

RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de INGENIERO MECATRÓNICO.
2. **TÍTULO:** CONTROL DEL MOVIMIENTO DE UN BRAZO ROBÓTICO EN UN PLANO MEDIANTE VISIÓN ARTIFICIAL.
3. **AUTORES:** David Felipe Meneses Garay y Miguel Ángel Ruiz Torres
4. **LUGAR:** Bogotá D.C
5. **FECHA:** 02/10/2018
6. **PALABRAS CLAVE:** Brazo, Contraste, Kinect, Manipulador, Segmentación, Visión artificial
7. **DESCRIPCIÓN DEL TRABAJO:** Este trabajo consta del desarrollo, simulación e implementación de un sistema capaz de captar y reconocer los movimientos de un brazo humano en un solo plano e imitarlos en un brazo robótico por medio de visión artificial a través de una cámara convencional y una cámara infrarroja de Kinect. Esto se logra desarrollando un código en el software Matlab® el cual se encarga de procesar las imágenes tomadas por la cámara discriminando el brazo del usuario y procesando dichas imágenes dando como resultado los ángulos que son enviados desde el programa hacia cada uno de los servomotores posicionados en las articulaciones del brazo robótico diseñado para emular el movimiento del brazo humano en el espacio o plano de trabajo.
8. **LÍNEAS DE INVESTIGACIÓN:** El campo temático al que va dirigido el proyecto es para el área de robótica del programa de ingeniería mecatrónica cuya sub-línea de facultad es la instrumentación y control de procesos, siendo parte del campo de tecnologías actuales y sociedad para la línea de investigación de la Universidad de San Buenaventura de Bogotá D.C.
9. **METODOLOGÍA:** Para llevar a cabo este proyecto se trabajará cada sistema de forma analítica e individual cumpliendo en primera instancia los objetivos específicos para luego realizar la comunicación entre los respectivos subsistemas, buscando así una integración sólida y una buena sinergia entre los componentes.
Cabe aclarar que la construcción de ningún sistema se efectuará hasta que el diseño de los mismos estén perfectamente establecidos tanto de forma individual para cada subsistema, como para su respectiva integración, siguiendo así los lineamientos VDI 2206 para el diseño de sistemas mecatrónicos.
10. **CONCLUSIONES:** De acuerdo con los experimentos realizados se concluye que todos los sistemas de detección pueden conseguir resultados satisfactorios en la segmentación del brazo humano y la emulación del movimiento del brazo robótico, sin embargo, debido a las características de cada uno de los sistemas se debe considerar las restricciones que estos presentan con respecto a parámetros variables como la iluminación y el color del fondo.

UNIVERSIDAD DE SAN BUENAVENTURA DE BOGOTA D.C.



PROYECTO DE GRADO

**CONTROL DEL MOVIMIENTO DE UN BRAZO ROBÓTICO EN UN PLANO
MEDIANTE VISIÓN ARTIFICIAL.**

Presentado por:

DAVID FELIPE MENESES GARAY

MIGUEL ANGEL RUIZ TORRES

ASESOR: JULIAN SEVERIANO RODRÍGUEZ A.

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA MECATRÓNICA

BOGOTA D.C.

2018

Resumen

Este trabajo consta del desarrollo, simulación e implementación de un sistema capaz de captar y reconocer los movimientos de un brazo humano en un solo plano e imitarlos en un brazo robótico por medio de visión artificial a través de una cámara convencional y una cámara infrarroja de Kinect. Esto se logra desarrollando un código en el software Matlab® el cual se encarga de procesar las imágenes tomadas por la cámara discriminando el brazo del usuario y procesando dichas imágenes dando como resultado los ángulos que son enviados desde el programa hacia cada uno de los servomotores posicionados en las articulaciones del brazo robótico diseñado para emular el movimiento del brazo humano en el espacio o plano de trabajo.

Tabla de contenido

1.Introducción	1
2.Planteamiento del problema	3
2.1. Justificación.....	3
2.2. Objetivos	4
2.2.1. Objetivo general	4
2.2.2. Objetivos específicos	4
3.Antecedentes	4
4.Desarrollo	11
4.1. Etapa 1. Sistemas de visión artificial.....	11
4.1.1. Detección del brazo humano	12
4.1.1.1. Sistema de detección 1 – Segmentación por contraste.....	12
4.1.1.2. Sistema de detección 2 – Con guante bicromático.....	38
4.1.1.3. Sistema de detección 3 – Con dispositivo Kinect Xbox 360	51
4.1.2. Cálculo de parámetros requeridos	57
4.1.2.1. Cálculo de la posición angular de cada segmento del brazo	57
4.1.2.2. Cálculo del porcentaje de apertura y cierre de la pinza.....	61
4.2. Etapa 2. Diseño y construcción del brazo robótico	64
4.2.1. Problema cinemático directo.....	64
4.2.2. Problema cinemático inverso – Método geométrico	67
4.2.3. Modelo diferencial	69
4.2.4. Modelo dinámico según Euler Lagrange	71
4.2.5. Diseño CAD y construcción del brazo robótico.....	80
4.3. Etapa 3. Diseño electrónico y comunicación entre visión artificial y brazo robótico	82
4.3.1. Configuración de los servomotores en el brazo robótico	82
4.3.2. Circuito de control del brazo robótico.....	84
4.3.3. Comunicación serial entre pc y microcontrolador	86
4.3.4. Clasificación de datos recibidos por el microcontrolador	87
4.3.5. Control de servomotores	88
5.Resultados	93
5.1. Resultados de sistemas de visión artificial	93
5.2. Resultados análisis dinámico	101
5.3. Resultados generales y validación del sistema.....	115
6.Conclusiones	118
7.Referencias	119

Lista de tablas

Tabla 1. Comparación ventajas y desventajas.....	9
Tabla 2. Costos.....	10
Tabla 3. Parámetros de Denawit-Hartenberg	66
Tabla 4. Medidas (cm)	80
Tabla 5. Peso eslabones.....	81
Tabla 6. Resultados obtenidos en las pruebas de distancia	98

Lista de figuras

Figura 1. Prototipo de brazo robótico.....	5
Figura 2. Robot recolector.....	6
Figura 3. Diagrama esquemático y cinemático del robot manipulador de 4 grados de libertad	7
Figura 4. Ensamble final del manipulador, la placa Arduino Mega 2560 y la placa electrónica	8
Figura 5. Imagen original del brazo humano	13
Figura 6. Imagen RGB.	14
Figura 7. Imagen R.....	14
Figura 8. Imagen G.	14
Figura 9. Imagen B.....	14
Figura 10. Histograma.....	16
Figura 11. Imagen original en escala de grises.	16
Figura 12. Imagen segmentada.....	16
Figura 13. Histograma de imagen original en escala de grises.	17
Figura 14. Histograma de imagen segmentada	17
Figura 15. Proceso de umbralización	18
Figura 16. Umbralización.....	22
Figura 17: Regiones indeseadas de pixeles	23
Figura 18. Imagen sin áreas indeseadas de pixeles	24
Figura 19. Segmentación por contraste (chaqueta 1)	25
Figura 20. Segmentación por contraste (chaqueta 2)	26
Figura 21. Segmentación por contraste (chaqueta 3)	26
Figura 22. Diagrama de flujo 1 (Almacenamiento de coordenadas en vector)	28
Figura 23. Datos almacenados por el vector 1	29
Figura 24. Imagen con puntos en las articulaciones.....	30
Figura 25. Diagrama de flujo de almacenamiento de datos	30
Figura 26. Diagrama de flujo para generar línea inferior de la pinza.....	31
Figura 27. Almacenamiento de coordenadas en vector 1 y 2	32
Figura 28. Error cuando la pinza apunta hacia arriba.....	33
Figura 29. Diagrama de flujo para ampliar el rango de acción de la pinza	34
Figura 30. Ampliación de rango de acción de la pinza	34
Figura 31. Adelgazamiento	35
Figura 32. Imagen esqueleto	36
Figura 33. Esqueleto con saturación de calores	37
Figura 34. Pinza abierta.....	37
Figura 35. Pinza cerrada.....	37
Figura 36. Guante bicromático día.....	38

Figura 37. Guante bicromático noche	38
Figura 38. Matrices RGB	39
Figura 39. Articulaciones resaltadas en la imagen	41
Figura 40. Brazo completo resaltado en la imagen	41
Figura 41. Histograma de articulaciones y brazo completo	41
Figura 42. Umbralización de articulaciones	43
Figura 43. Umbralización de brazo completo	43
Figura 44. Algoritmo de etiquetado	44
Figura 45. Lógica del algoritmo de etiquetado	45
Figura 46. Proceso de etiquetado de articulaciones	46
Figura 47. Creación del esqueleto	47
Figura 48. Detección del guante bicromático	47
Figura 49. Unión de etiquetas correctamente	49
Figura 50. Error al unir los labels	49
Figura 51. Diagrama de flujo para corrección de numeración de labels	50
Figura 52. Partes del Kinect	52
Figura 53. Imagen de profundidad resultante	53
Figura 54. Proyector infrarrojo	53
Figura 55. Segmentación por distancia, ejemplo 1	54
Figura 56. Segmentación por distancia, ejemplo 2	54
Figura 57. Segmentación por distancia	55
Figura 58. Detección de articulaciones	56
Figura 59. Creación del esqueleto	56
Figura 60. Ángulos con respecto a un eje fijo (eje horizontal)	57
Figura 61. Ángulos con respecto a eje móvil	57
Figura 62. Cálculo del parámetro q_1	58
Figura 63. Cálculo del parámetro d_1	59
Figura 64. Cálculo del parámetro q_2	60
Figura 65. Cálculo del parámetro q_3 con pinza cerrada	60
Figura 66. Ángulo medio	61
Figura 67. Medición de diferentes manos	62
Figura 68. Mano con mayor ángulo de apertura	62
Figura 69. Mano con menor ángulo de apertura	62
Figura 70. Cálculo del ángulo de apertura	63
Figura 71. Etiquetas convencionales	64
Figura 72. Etiquetas invertidas	64
Figura 73. Diagrama del brazo	65

Figura 74. Ejemplo de posición para el actuador final.....	68
Figura 75. Posición evaluada	77
Figura 76. Diseño CAD. a. Ensamble, b. Húmero, c. Radio, d. Articulaciones, e. Muñeca, f. Pinza.	81
Figura 77. Configuración de los servomotores en el brazo robótico.....	82
Figura 78. Ángulos teóricos	83
Figura 79. Conversión a ángulos reales	83
Figura 80. Circuito de control	85
Figura 81. Señal de control PWM.....	86
Figura 82. Módulo FT232.....	87
Figura 83. Clasificación de ángulos	88
Figura 84. Sincronización de señales con timer TCNT1.....	90
Figura 85. Pulso con par de interrupciones	91
Figura 86. Generación de varias salidas PWM	92
Figura 87. Pruebas fondo claro, manda oscura y variación de luz.....	93
Figura 88. Prueba fondo azul, sin manga e iluminación alta.....	94
Figura 89. Prueba de fondo con dos colores	94
Figura 90. Pruebas con ambiente totalmente controlado	95
Figura 91. Pruebas con fondo claro y variación de luz	95
Figura 92. Pruebas de fondo oscuro con variación de luz.....	96
Figura 93. Pruebas de fondo combinado con variación de luz.....	96
Figura 94. Pruebas de segmentación con fondo azul	97
Figura 95. Pruebas de alcance a 1,30 m	97
Figura 96. Pruebas de alcance a 1,10 m	97
Figura 97. Pruebas de alcance a 1 m	97
Figura 98. Pruebas de alcance a 80 cm	97
Figura 99. Pruebas de alcance a 60 cm	98
Figura 100. Pruebas con objetos	98
Figura 101. Desplazamiento en el eje x	102
Figura 102. Valores de torque requeridos en la articulación 1 para generar desplazamiento en el eje x	102
Figura 103. Imagen aumentada 1	102
Figura 104. Valores de torque requeridos en la articulación 2 para generar desplazamiento en el eje x.....	103
Figura 105. Imagen aumentada 2	103
Figura 106. Valores de torque requeridos en la articulación 3 para generar desplazamiento en el eje x	103
Figura 107. Desplazamiento en el eje y	104

Figura 108. Valores de torque requeridos en la articulación 1 para generar desplazamiento en el eje y	104
Figura 109. Imagen aumentada 3	104
Figura 110. Valores de torque requeridos en la articulación 2 para generar desplazamiento en el eje y	105
Figura 111. Imagen aumentada 4	105
Figura 112. Valores de torque requeridos en la articulación 3 para generar desplazamiento en el eje y	105
Figura 113. Desplazamiento en los ejes X & Y	106
Figura 114. Valores de torque requeridos en la articulación 1 para generar desplazamiento diagonal	107
Figura 115. Imagen aumentada 5	107
Figura 116. Valores de torque requeridos en la articulación 2 para generar desplazamiento en el eje	107
Figura 117. Imagen aumentada 6	107
Figura 118. Valores de torque requeridos en la articulación 3 para generar desplazamiento diagonal	108
Figura 119. Imagen aumentada 6	108
Figura 120. Barrido de ángulos	109
Figura 121. Valores de torque requeridos en la articulación 1 para generar desplazamiento circular.	110
Figura 122. Valores de torque requeridos en la articulación 2 para generar desplazamiento circular.	110
Figura 123. Valores de torque requeridos en la articulación 3 para generar desplazamiento circular.	111
Figura 124. Valores de torque requeridos en la articulación 1 para generar el desplazamiento circular con velocidad de 20 cm/s	112
Figura 125. Servomotor MG995	113
Figura 126. Prueba de validación sistema 1	116
Figura 127. Prueba de validación sistema 2	116-117
Figura 128. Prueba de validación sistema 3	117

1. Introducción

En la actualidad los avances tecnológicos han llevado al hombre a buscar la automatización de procesos y el uso de la robótica como instrumento para facilitar el desarrollo de una manera más eficiente en las diferentes actividades del diario vivir, sin embargo existen muchas actividades donde no es conveniente utilizar sistemas automatizados por lo cual se prefiere la ejecución manual de dicha actividad por parte de un operario aun cuando se pone en riesgo la integridad física de esta persona, esto debido a que la actividad realizada no es una tarea repetitiva sino que requiere de un análisis del operario para luego ejecutar movimientos o acciones de acuerdo a las decisiones tomadas por la persona, es decir, se requiere de un análisis inteligente de la situación y una ejecución de la tarea menos genérica y más personalizada de acuerdo a la situación.

Para automatizar una tarea de este tipo sin poner en riesgo al operario sería necesario implementar un sistema inteligente de alta tecnología y de elevado costo, que en la mayoría de casos la actividad no amerita el uso de sistemas tan sofisticados, es allí donde la idea de emular el movimiento humano de forma completa o una sola extremidad a través de robots se vuelve más viable, debido a que no se vulnera la integridad física de la persona pero sigue siendo él quien toma las decisiones y ejecuta las acciones.

Esta emulación puede hacerse aplicando diferentes metodologías, sin embargo, una de las opciones más viables por el bajo costo y pocos dispositivos de sensorizado es gracias al uso de una cámara, ya que a partir de una imagen digital del entorno es posible obtener bastante información del mismo, a esto se le conoce como visión artificial y es ampliamente usado en diversos sistemas automáticos donde se busca obtener diferentes características de un objeto sin la implementación de una numerosa cantidad de sensores, como Sánchez (2005) afirma. “la visión artificial pretende, en cierta medida, reproducir artificialmente el sentido

de la vista mediante el procesamiento e interpretación de imágenes, captadas con distintos tipos de sensores (fundamentalmente, cámaras), y utilizando para ello las prestaciones de los ordenadores”.

Teniendo en cuenta lo anterior el desarrollo del proyecto consiste en la implementación de un prototipo de un brazo robótico que posea 4 grados de libertad y que físicamente se asemeje al brazo del ser humano. La idea es capturar por medio de una cámara el movimiento en tiempo real que realice el brazo humano, para ello se busca que la cámara capte puntos bases del movimiento, es decir ciertos puntos del brazo humano que posteriormente se procesan para su discriminación.

Adicionalmente se hace la discriminación de las articulaciones mediante la implementación de algoritmos de visión artificial para así obtener la información requerida para emular satisfactoriamente el movimiento sobre el plano lateral al brazo humano, que es el plano de trabajo. Una vez procesado el movimiento, se envían señales independientes a cada actuador del brazo robótico con el fin de que cada articulación llegue a la posición deseada.

2. Planteamiento del problema

El proyecto busca evitar que una persona quede expuesta a algún riesgo debido a la manipulación de materiales o elementos peligrosos existentes en diversos procesos industriales o investigativos, por este motivo el proyecto pretende dar respuesta a la pregunta: ¿Cómo diseñar y construir un brazo robótico capaz de emular los movimientos de un brazo humano?

2.1. Justificación

Aunque se evidencia una disminución en los accidentes laborales en Colombia para el año 2017 con 643.501 frente al año 2016 con 701696, según análisis de la Federación de Aseguradores Colombianos (Fasecolda), las cifras siguen siendo altas, es por esto la necesidad de buscar alternativas que ayuden no solo a automatizar y agilizar procesos, sino también que brinden seguridad en los diferentes entornos de trabajo que se presenten.

La propuesta pretende crear ambientes seguros en lugares de trabajo donde se expone la vida e integridad de las personas, teniendo en cuenta que del total de accidentes laborales, el 53% de estos procesos dependen en algún aspecto de la ingeniería. El poder controlar el brazo a distancia permite disminuir los riesgos, pero disminuye la precisión del trabajo, es por esto por lo que se evaluaron 3 diferentes estrategias para detectar el brazo humano con el fin de obtener una correcta emulación en el brazo robótico.

La presente investigación y desarrollo tiene como fin emular el comportamiento de un brazo humano en un brazo artificial controlado por una persona, la captura de los movimientos del brazo humano se hace por medio de una cámara la cual discriminará algún tipo de referencia y hará el seguimiento de dicha referencia, este movimiento es traducido en el control hacia los actuadores del brazo artificial.

2.2. Objetivos

2.2.1. Objetivo general.

Diseñar y construir un sistema de imitación de movimiento entre un brazo humano y un brazo robótico en un solo plano por medio de visión artificial.

2.2.2. Objetivos específicos.

- Implementar un sistema de visión artificial que consiga la detección de los movimientos de un brazo humano en un solo plano.
- Diseñar un brazo robótico capaz de emular el movimiento real de un brazo humano en un plano.
- Construir un brazo robótico e implementar en este el sistema de visión artificial con el fin de poder imitar el movimiento real de un brazo humano en una máquina.
- Validar el funcionamiento del brazo robótico.

3. Antecedentes (ESTADO DEL ARTE)

En los últimos años se ha podido evidenciar que la robótica se ha abierto paso a través de diferentes campos de acción en una gran cantidad de procesos industriales, procesos de automatización, procesos ingenieriles, entre otros, puesto que esta ciencia tiene la capacidad de imitar de manera precisa y rápida cualquier tipo de tarea que realice el ser humano o alguna otra máquina, mejorando así los resultados obtenidos.

El uso de brazos o manipuladores robóticos se hace evidente generalmente en procesos repetitivos o cíclicos tales como líneas de ensamblado, de selección, de cortado, de maquinado, etc. Tareas las cuales pueden ser realizadas por cualquier ser humano, pero dando a su vez un valor agregado de rapidez y seguridad.

La idea de implementar a estos procesos lo que conocemos como visión artificial, da a los mismos una mayor capacidad de autonomía en el desarrollo de sus tareas para los cuales están diseñados.

Múltiples proyectos se han desarrollado en este sentido, algunos cercanos a la propuesta realiza en este proyecto se presentan a continuación:

Diseño de un Prototipo Electromecánico para la Emulación de los Movimientos de un Brazo humano. En este artículo (Gallo Sánchez, Guerrero Ramírez, Vásquez Salcedo, & Alonso Castro, 2016) explican el diseño y construcción de un prototipo de brazo robótico con 13 grados de libertad capaz de emular todos los movimientos realizados por el brazo humano por medio de una estructura con diferentes sensores que posee el operador del brazo,

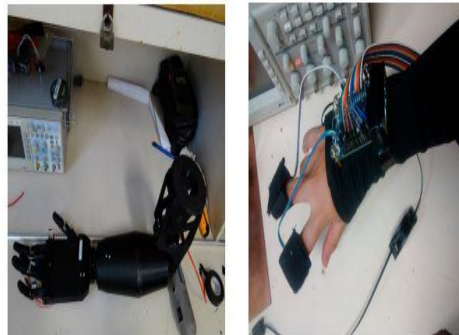


Figura 1. Prototipo de brazo robótico.

Como se puede ver en la figura 1. La mano del brazo robótico tiene 5 dedos, muñeca, antebrazo, bíceps y hombro con el fin de que el prototipo sea lo más semejante a un brazo humano. Las dimensiones del brazo robot corresponden a los de un brazo humano adulto y su respectivo control está dado por el uso de una tarjeta FPGA programada en VHDL con acelerómetros como sensores de posición, por último, el material de la estructura externa es PLA negro.

A review of key techniques of vision-based control for harvesting robot (Zhao, Gong, Huang, & Liu, 2016).

En este trabajo se presenta un sistema de cosecha agrícola, en el cual por medio de visión artificial se busca detectar frutas de interés para su recolección. El sistema se constituye de un brazo robótico o manipulador que está encargado de recoger, cosechar y almacenar dichos frutos. Esto se logra a través de una señal que envía el sistema de visión artificial al detectar la posición de la fruta al actuador del brazo robótico que se mueve hasta dicha posición y recoge la fruta.

El robot consta de un sistema monocular para la detección y un sistema binocular de estereovisión el cual está compuesto por dos cámaras separadas por una ligera distancia que apuntan a la misma escena con un ángulo entre ellas presentando una imagen tridimensional en conjunto. Ver figura 2

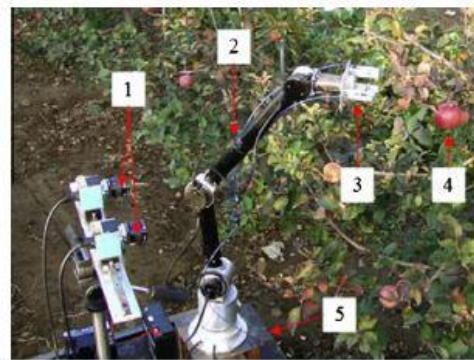


Figura 2. Robot recolector.

El proyecto utiliza una serie de algoritmos para procesar en tiempo real las imágenes detectadas con las coordenadas de posición de la fruta a coleccionar.

Touch-Less Java Based Hand Tracking System to Control an Artificial Arm(Grzegorz Baron, Piotr Czekalski, Daniel Malicki, 2014).

Este artículo presenta un nuevo método que permite la conducción de un Mindstorms NXT(Lego Group, 2018) brazo artificial con Microsoft Kinect (Microsoft, 2012), usando reconocimiento de gestos o movimientos y seguimiento de la mano. El artículo presenta aparte de la imitación en un brazo artificial, un algoritmo de interacción entre computador y humano usado en la implementación del movimiento esquelético y control de gestos en espacio 3D. Dicha solución esta implementada usando software libre y abierto basado en Java.

Sistema Teledirigido de un brazo robótico de 4 grados de libertad aplicando visión de máquina (Salazar, Darío, Sandoval, & Lucía, 2015).

Este articulo presenta los resultados del diseño y desarrollo de un sistema de control para el movimiento teledirigido o remoto de un brazo robótico de 4 grados de libertad a través de un Joystick o controlador manual manipulado por una persona aplicando visión artificial.

Para determinar la posición del brazo robótico se utiliza un sensor Kinect y una figura de referencia que se sitúa en la posición final del espacio de trabajo la cual puede ser reconocida por un sistema de visión artificial. Ver figura 3.

Se estima la distancia del manipulador con respecto a la referencia utilizando el mapa de profundidad de cámara infrarroja (IR).

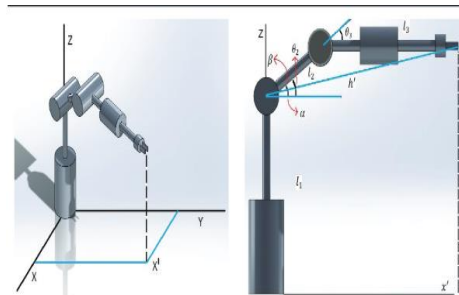


Figura 3. Diagrama esquemático y cinemático del robot manipulador de 4 grados de libertad.

El movimiento del brazo se efectúa aplicando un algoritmo de visión artificial, el cual procesa las imágenes para poder discriminar la posición del robot.

Generación De Trayectorias Para El Brazo Robótico (Armx)(Hermosilla & Castilla Pérez, 2016).

Este trabajo presenta la modelación y simulación cinemática directa e inversa de un robot articular de cuatros grados de libertad con pinzas como efector final para manipular objetos. El control cinemático se realiza por medio de seguimiento de trayectorias punto a punto, coordenadas y continuas para el extremo

Se emplea de plataforma el hardware libre Arduino y su entorno de desarrollo, además se diseñó e implemento una placa del tipo escudo con la electrónica necesaria para el control del manipulador, así como una interfaz gráfica para poder controlar dichas trayectorias por medio del software LabWindows y CVI 9.0 de National Instruments. Ver figura 4



Figura 4. Ensamble final del manipulador, la placa Arduino Mega 2560 y la placa electrónica.

De acuerdo con los proyectos presentados anteriormente y a demás proyectos de robótica y visión artificial de la actualidad es posible destacar una serie de ventajas y desventajas de nuestro proyecto en comparación con los demás. Cabe resaltar que nuestro proyecto es un sistema que podría ser aún más preciso y avanzado implementando mejoras como dispositivos de sensorizado en cada articulación del manipulador robótico que

permitirían la implementación de controles en lazo cerrado o también el uso un control inalámbrico entre el manipulador robótico y el PC, o la ampliación de la emulación de un plano de trabajo a dos planos. Estas mejoras podrían llevar nuestro sistema a campos donde se requiera sistemas de emulación más avanzados, sin embargo, para el propósito de este proyecto nuestro sistema es bastante competitivo frente a demás proyectos de este campo. En la tabla 1 se presentan las ventajas y desventajas de nuestro proyecto.

Tabla 1.Comparación ventajas y desventajas

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> • Facilidad para ser implementado en diversos campos y entornos empresariales sin necesidad de un excesivo espacio de trabajo o el uso de múltiples dispositivos para la instalación del sistema. • Sistema de visión monocular con bajo costo y bajo consumo energético en comparación a un sistema estereoscópico. • Generación de movimiento a partir del movimiento real de un brazo humano y no a través de mandos o controladores remotos. • Interfaz de comunicación por USB que permite la implementación de cualquier otro sistema de control para generar movimiento en el manipulador robótico. • El sistema presenta bajos costos siendo accesible y viable para aplicaciones donde no se justifique el uso de un sistema más avanzado y costoso. 	<ul style="list-style-type: none"> • Pocos grados de libertad en el manipulador robótico y emulación únicamente en un plano. • Sistema de visión monocular sin generación de imagen 3D para el seguimiento del brazo humano. • Dependencia de variables en el entorno que pueden afectar la imagen capturada por la cámara RGB. • Sistema de comunicación alámbrico que requiere de más cableado si se busca obtener una mayor distancia entre el manipulador robótico y el PC. • Falta de robustez para mantener un seguimiento preciso del brazo humano debido a una incorrecta detección en la cámara.

El desarrollo del proyecto presenta bajos costos en relación a trabajos de robótica y visión artificial que se encuentran comúnmente en el ámbito empresarial, siendo así una ventaja altamente competitiva en relaciones a sus similares. En la tabla 1 podemos destacar los costos aproximados de acuerdo a las herramientas utilizadas por los autores al momento de desarrollar el proyecto,

Tabla 2. Costos directos

MATERIAL	COSTO (COP)	UNIDAD
CÁMARA EXTERNA	30.000	1
CÁMARA INTERNA	0	N/A
SERVOS	25.000	5
PIEZAS ACRÍLICO	18.000	N/A
BASE EN HIERRO	27.000	1
ARANDELAS TORNILLOS TUERCAS VARILLAS ROSCADOS	23.000	N/A
CABLES PROTECTORES	19.000	N/A
MICROCONTROLADOR	11.000	1
PCB	14.000	1
INTERRUPTOR	1.500	1
MÓDULO DE COMUNICACIÓN SERIAL	21.000	1
FUENTES DE VOLTAJE	20.000	4
KINECT	230.000	1

Teniendo en cuenta los costos de la tabla 2 el total del proyecto es de \$549.500

4. Desarrollo

El diseño y realización del proyecto consta de 3 etapas, cada una de las cuales se especifican a continuación:

1. La primera etapa consiste en el diseño e implementación de 3 sistemas de visión artificial que tienen como objetivo capturar en tiempo real el movimiento real de un brazo humano por 3 diferentes métodos evaluando así sus ventajas y desventajas entre sí
2. La segunda etapa consiste en diseñar y construir un brazo robótico de 4 gdl y configuración tipo RRR similar al brazo de un ser humano, con el fin de controlar su movimiento a partir de la implementación de los sistemas de visión artificial.
3. La tercera etapa consiste en el diseño e implementación del circuito capaz de controlar y comunicar el sistema de visión artificial con el brazo robótico diseñado y construido.

4.1. Etapa 1 - Sistemas de visión artificial

En este capítulo se hace una explicación de los 3 sistemas de detección que hacen parte de la etapa de visión artificial para posteriormente hacer un estudio comparativo entre ellos, adicionalmente se hace el cálculo de parámetros necesarios para controlar el brazo robótico a partir de la detección realizada en cada sistema. Teniendo en cuenta lo anterior, este capítulo se divide en 2 partes que son:

- Detección del brazo humano
- Cálculo de parámetros requeridos

4.1.1. Detección del brazo humano.

4.1.1.1. *Sistema de detección 1 – Segmentación por contraste.*

El sistema de detección 1 busca detectar el brazo del usuario a partir del contraste que genere los colores del brazo con los colores del fondo, este contraste se logra con un fondo iluminado y un brazo con un color que esté entre una gama donde se encuentren el gris, negro y azul principalmente, debido a que estas tonalidades se encuentran comúnmente en la mayoría de las prendas empleadas en el sector industrial, y el objetivo de este sistema de detección es poder ser capaz de detectar el brazo humano a pesar de la ropa utilizada. Por otra parte, es común encontrar fondos con colores claros debido a que los interiores de los recintos generalmente manejan tonos claros, sin embargo, no siempre se dará el caso en el que el fondo este iluminado y la ropa utilizada por el usuario tenga tonalidades oscuras, por lo que es importante que este sistema opere en un ambiente controlado para detectar correctamente. El sistema de detección 1 logra su objetivo a partir de una serie de procesos que son:

- Adquisición de la imagen en tiempo real
- Segmentación por contraste:
 - Detección de colores RGB
 - Umbralización
 - Eliminación de áreas indeseadas en la imagen
- Detección de articulaciones
- Apertura y cierre de la pinza
- Ampliación de rango de acción de la pinza
- Creación de esqueleto

4.1.1.1.1. Adquisición de la imagen en tiempo real.

La adquisición de la imagen es quizás la parte más importante en el campo de visión artificial puesto que la captura de una imagen con buena resolución y calidad puede hacer del preprocesamiento un paso más simple. Esta se logra a través de un sensor o cámara capaz de digitalizar la imagen, una imagen digital consta de una matriz de $N \times M$ números en la cual cada celda de la matriz es un pixel. En este caso la adquisición de la imagen se hace a través de cualquier cámara capaz de conectarse a un computador y que sea reconocida por el software Matlab®.

Conociendo el proceso de adquisición de imágenes, para su procesamiento se requiere que esta sea en tiempo real con el fin de poder seguir la trayectoria del brazo humano, esto se logra por medio de la toma consecutiva de fotogramas a una velocidad que no puede ser percibida por el ojo humano. Una vez se comienzan a tomar las capturas en tiempo real estas se almacenan en una variable temporal para realizar el procesamiento de cada una de estas de manera individual. En la figura 5 se muestra una captura del brazo humano durante el proceso de adquisición de imágenes en tiempo real.



Figura 5. Imagen original del brazo humano.

4.1.1.1.2. Segmentación por contraste.

4.1.1.1.2.1. Detección de colores RGB.

Una imagen digital a color está conformada por 3 matrices cuyos elementos representan una intensidad de colores en el espacio RGB, la unión de dicha intensidad de colores de cada una de estas matrices es lo que da como resultado toda la gama de colores que pueden ser identificados. En la figura 6 se puede apreciar un ejemplo con varios colores y en las figuras 7, 8 y 9 su descomposición en el espacio RGB.



Figura 6. Imagen RGB.



Figura 7. Imagen en R.



Figura 8. Imagen en G.



Figura 9. Imagen en B.

A partir del espacio RGB se busca implementar un algoritmo capaz de segmentar el color que diferencie el brazo humano del resto de la imagen. Para ello se busca una combinación en la cual las matrices R, G, y B resalten los tonos grises, negros o azules del brazo teniendo en cuenta que el fondo de la imagen debe tener un nivel de luz que genere un contraste adecuado con el brazo y no permita una mimetización del brazo con el entorno. La combinación de matrices implementada se muestra en la ecuación 1.

$$\text{Imagen segmentada} = Kr(R - \text{gris}) + Kg(G - \text{gris}) + Kb(B - \text{gris}) \quad (1)$$

Donde:

gris = Imagen original en escala de grises

(R, G, B) = matrices que componen la imagen original

(K_r, K_g, K_b) = coeficientes para controlar la prioridad de cada matriz R, G y B en la imagen resultante.

De acuerdo con el entorno en el que se detecta el brazo humano se pueden variar los coeficientes K_r , K_g , y K_b para resaltar unas tonalidades más que otras, sin embargo, el valor de los coeficientes es definido por los autores como $K_r = 0.4$, $K_g = 0.6$ y $K_b = 2$ a partir de diversas pruebas realizadas debido a que esta combinación permite segmentar gran cantidad de tonalidades dentro de la gama del gris, negro y azul en diversos entornos.

Al aplicar la ecuación 1 las matrices R, G y B de la imagen original sufren una transformación en la cual la mayoría de píxeles en cada matriz son iguales a cero, es decir son píxeles de color negro, y únicamente una minoría de píxeles de cada matriz se aprecian en la imagen resultante. Si el entorno y las tonalidades en la ropa del usuario son los adecuados, esta minoría de píxeles resaltados debe corresponder al brazo humano.

En la figura 10 se muestra el histograma de las matrices R, G y B de la imagen original y su posterior transformación para obtener la imagen segmentada.

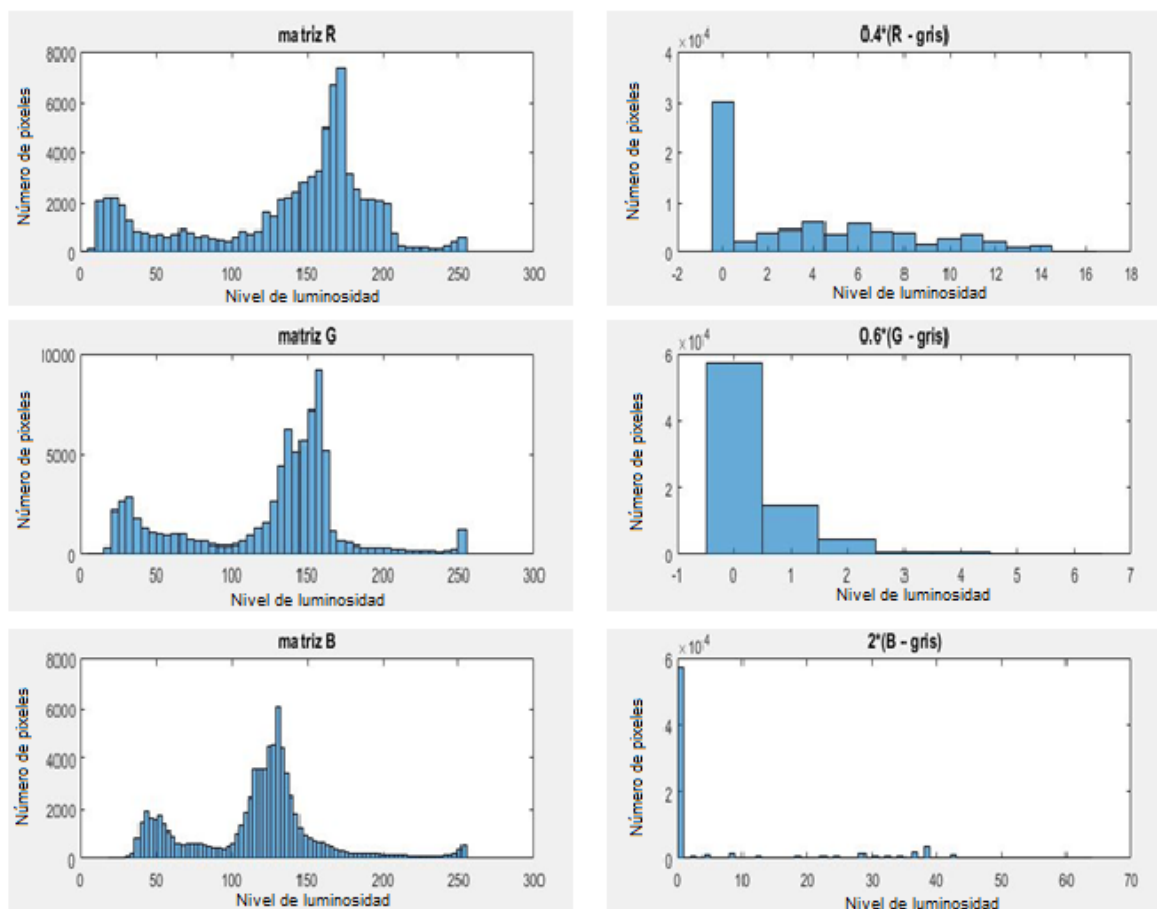


Figura 10. Histogramas.

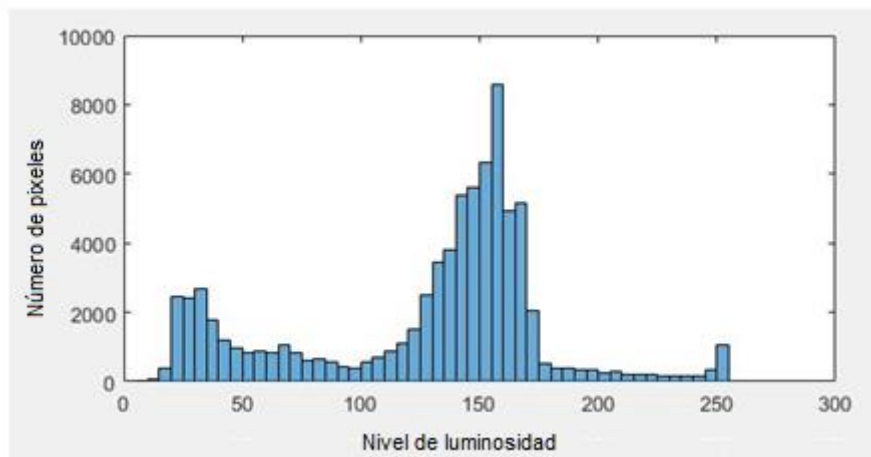
El resultado de la ecuación 1 es la imagen del brazo humano segmentado en escala de grises (figura 12) a partir de las matrices R, G, B, y de la imagen original en escala de grises (figura 11).



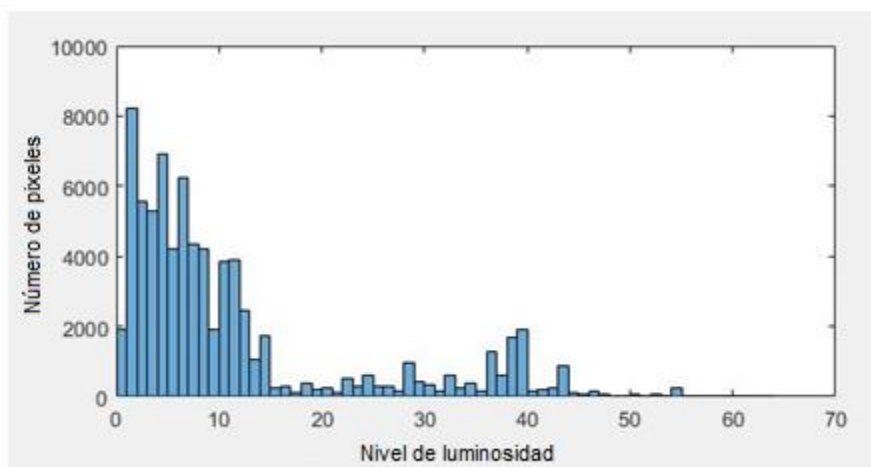
Figura 11. Imagen original en escala de grises.

Figura 12. Imagen segmentada.

En la figura 13 se puede apreciar el histograma de la imagen original en escala de grises y en la figura 14 el histograma de la imagen segmentada.



*Figura 13.*Histograma de imagen original en escala de grises.



*Figura 14.*Histograma de imagen segmentada.

En la figura 14 se puede apreciar también el aumento en píxeles oscuros entre el rango lumínico de 0 a 20 aproximadamente correspondientes al fondo, y una pequeña minoría de píxeles entre el rango lumínico de 30 a 50 aproximadamente correspondientes a la figura del brazo segmentado y demás objetos que hayan sido resaltados en la imagen al implementar la ecuación 1. Se puede apreciar a su vez la reducción en el rango de iluminación del histograma de la imagen segmentada con respecto al rango de iluminación de la imagen original en escala de grises evidenciando la eliminación de tonalidades claras de la imagen.

4.1.1.1.2.2. Umbralización.

La umbralización busca segmentar de manera clara el brazo humano del resto de la imagen, esto se puede lograr a través de convertir una imagen a escala de grises en una imagen binaria que sea capaz de resaltar la parte de interés. En otras palabras, colocar la imagen a blanco y negro donde se puede diferenciar el brazo humano del fondo de la imagen, este proceso de umbralización está definido por la ecuación 2.

$$\begin{aligned} g(x,y) &= 1 \quad \text{para } f(x,y) \geq T \\ g(x,y) &= 0 \quad \text{para } f(x,y) < T \end{aligned} \quad (2)$$

Donde:

T = nivel de umbral entre los pixeles a resaltar y los pixeles de fondo.

$g(x, y)$ = valor de intensidad lumínica en la posición (x, y) de la imagen umbralizada

$f(x, y)$ = valor de intensidad lumínica en la posición (x, y) de la imagen a escala de grises

En la figura 15 se muestra el proceso de umbralización definido en la ecuación 2 a partir del histograma de la imagen segmentada obtenida anteriormente (figura 14).

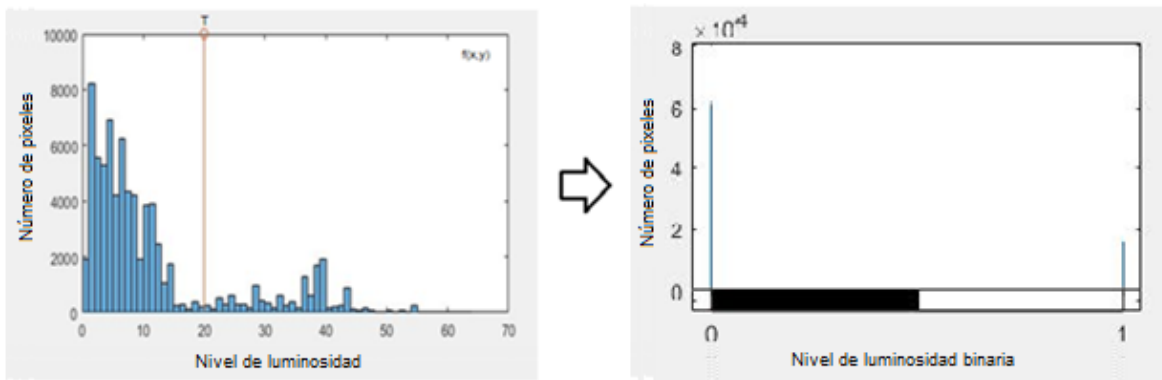


Figura 15. Proceso de umbralización.

- Umbral Óptimo: El umbral óptimo se puede definir como aquel rango de trabajo ideal para que el sistema pueda segmentar el objeto de interés, para lograr esto

Cattaneo, Larcher, Ruggeri, Herrera, & Biasoni (2011) exponen algunos métodos existentes, los cuales son:

- **Métodos basados en la forma del histograma:** En esta clasificación se tienen en cuenta las propiedades del histograma tales como picos, valles o curvaturas.
- **Métodos basados en la clusterización:** Se busca modelar el histograma como una superposición de funciones gaussianas. Estos algoritmos reducen el análisis de datos en dos regiones (background y foreground). Las dos regiones corresponden a los dos lóbulos del histograma.
- **Métodos basados en la entropía:** Por medio de los niveles de gris en una imagen se aplica la entropía. La máxima entropía es interpretada como la máxima información transferida y es el umbral óptimo a elegir.
- **Métodos basados en los atributos de la imagen:** Son técnicas que seleccionan un valor de umbral T basado en atributos que buscan una medida de similitud entre la imagen original y la imagen binarizada. Estos atributos pueden ser: bordes, formas, momentos de niveles de gris, conectividad, textura o estabilidad de los objetos
- **Métodos basados en información espacial:** A diferencia de los métodos anteriores que utilizan el valor de gris de cada píxel, estos los algoritmos dependen de la información espacial de los píxeles, por ejemplo, las probabilidades de su contexto, funciones correlación, probabilidades de coocurrencia, modelos locales dependientes de píxeles, entropía en bidimensional, etc.

- **Métodos basados en características locales:** Adaptan el umbral en cada píxel en función de las características locales de la imagen tales como rango, varianza, parámetros de superficie. El umbral $T(i, j)$ es en función de sus coordenadas, los píxeles del objeto y fondo son indicados por la variable $B(i, j)$.

De los métodos mencionados anteriormente para la umbralización del brazo, se proponen los métodos basados en la clusterización, entre los cuales se encuentra el método propuesto por Otsu, (1979), el cual está basado en el histograma de la imagen capturada (Figura 11 y 12). Dicho método utiliza la distribución de probabilidad, la media y la varianza de las intensidades para definir el umbral óptimo, a través de normalizar el histograma como una función de densidad de probabilidad de tipo discreta **pi**.

Se considera una imagen como una función bidimensional de intensidad del nivel de gris que contienen N píxeles cuyos niveles de gris se encuentran entre 1 y L , El número de píxeles con nivel de gris i se denota como n_i y el total de número de píxeles está dado por $N=n_1+n_2+\dots+n_L$, teniendo en cuenta lo anterior la probabilidad de ocurrencia del nivel de gris i en la imagen está dada por la ecuación 3.

$$p_i = \frac{n_i}{N} \quad ; \quad p_i \geq 0 \quad (3)$$

Donde:

p_i = la probabilidad de ocurrencia del nivel de gris i .

Ya normalizado el histograma se divide la imagen en dos clases por un nivel de umbral K tal que C_0 represente un conjunto de píxeles con niveles entre $[1, 2, 3, \dots, K]$ y C_1 otro grupo de píxeles con niveles $[K+1, K+2, \dots, L]$, por lo cual el método de Otsu selecciona

el valor del umbral maximizando la varianza entre clases σ_B^2 la cual se encuentra definida por la ecuación 4.

$$\sigma_B^2 = w_0 * (\mu_1 - \mu_T)^2 + w_1(\mu_0 - \mu_T)^2 \quad (4)$$

Donde:

$$w_0 = \sum_{i=1}^k p_i = w(k) \quad (5)$$

$$w_1 = \sum_{i=k+1}^L p_i = 1 - w(k) \quad (6)$$

$w_0 = \text{probabilidad de ocurrencia de } C_0$

$w_1 = \text{probabilidad de ocurrencia de } C_1$

Lo niveles medios de cada clase están dados por

$$\mu_0 = \sum_{i=1}^k \frac{i * p_i}{w_0} \quad (7)$$

$$\mu_1 = \sum_{i=k+1}^L \frac{i * p_i}{w_1} \quad (8)$$

Donde:

$\mu_0 = \text{es el nivel medio de } C_0$

$\mu_1 = \text{es el nivel medio de } C_1$

El nivel medio total K de la imagen original está dado por la ecuación 9.

$$\mu_T = \sum_{i=1}^L i * p_i \quad (9)$$

Por otro lado si se busca umbralizar un número de M niveles, el método Otsu se generaliza obteniendo el conjunto multinivel que maximiza la varianza entre las M clases $[K_1, K_2, K_3, \dots, K_M]$, de tal forma que se satisface la ecuación 10.

$$\{K_1, K_2, K_3 \dots K_M\} = MAX\{\sigma_B^2(k_1, k_2, k_3 \dots k_M)\} \quad (10)$$

Para $1 \leq k_1 \leq k_2 \leq k_3 \leq k_M \leq L$

Donde:

$$\sigma_B^2 = \sum_{k=1}^{M+1} (\mu_k - \mu_T)^2 * w_k \quad (11)$$

$$w_k = \sum_{i \in C_k} P_i \quad (12)$$

$$u_k = \sum_{i \in C_k} i * P_i \quad (13)$$

El resultado del proceso de umbralización es una imagen cuyos pixeles poseen una información binaria en la cual el valor de 1 corresponde a un pixel blanco y el valor de 0 corresponde a un pixel negro obteniendo así una imagen en blanco y negro donde se hacen visibles los objetos que superan el umbral obtenido como se muestra en la figura 16.

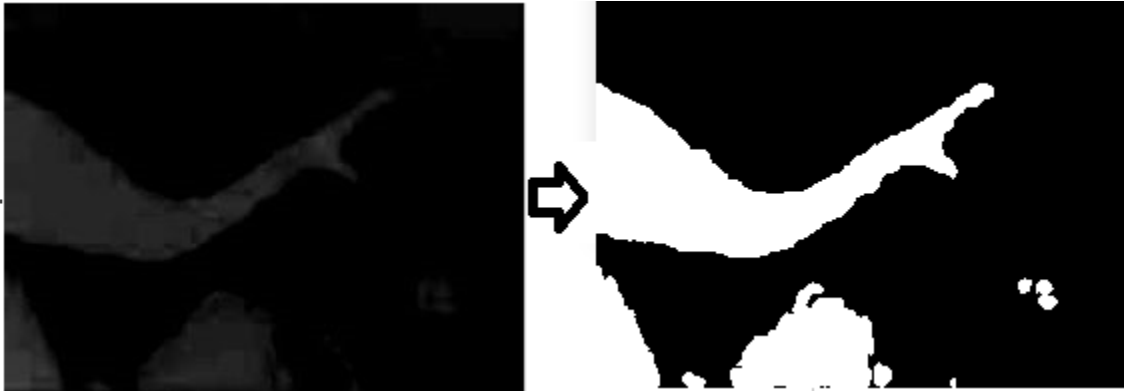


Figura 16. Umbralización.

4.1.1.1.2.3. Eliminación de áreas indeseadas en la imagen.

Obtener una imagen bien umbralizada y de calidad del brazo humano requiere otros procesos que perfeccionen y entreguen una imagen umbralizada más precisa y homogénea aparte de elegir un nivel de umbral optimo, sin embargo para una detección en tiempo real estos procesos consumen bastante capacidad de procesamiento y a pesar de la mejoría que ofrecen a la imagen umbralizada, la mayoría de procesos no son esenciales para la detección debido a que únicamente con la imagen umbralizada ya se obtiene una máscara aceptable del brazo. No obstante, si es necesario realizar un proceso de eliminación de áreas indeseadas en la imagen correspondientes a ruido o impurezas debido a objetos en el fondo que también han sido resaltados por su alto contraste con el fondo como se muestra en la figura 17.

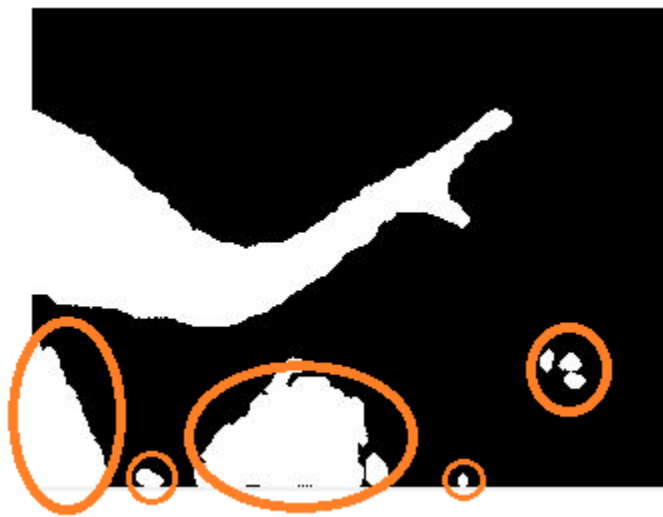


Figura 17. Regiones indeseadas de pixeles.

Estas regiones indeseadas de pixeles generalmente son una minoría en comparación con el área de pixeles del brazo humano, por lo que se pueden discriminar según su tamaño. Para eliminar estas regiones indeseadas con poca cantidad de pixeles se utiliza la ecuación 14.

$$\text{si } g(x, y) < R \text{ entonces } g(x, y) = 0 \quad (14)$$

Donde R es el valor de las áreas mínimas, es decir, cualquier área segmentada menor a dicho número es suprimida.

En la figura 18 se aprecia el resultado de la eliminación de áreas indeseadas en una imagen umbralizada.

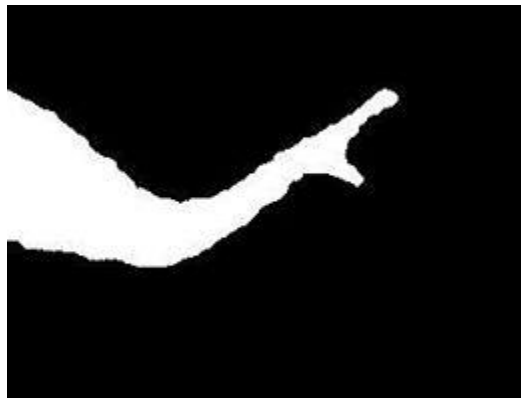


Figura 18. Imagen sin áreas indeseadas de pixeles.

A pesar de que el sistema de detección 1 busca segmentar únicamente el brazo humano, no se limita únicamente a ello, ya que el sistema es capaz de segmentar siempre y cuando exista un contraste entre el fondo y el objeto a segmentar y adicionalmente el objeto posea colores que se encuentren dentro de la gama del negro, gris y azul. En las figuras 19 a 21 se puede apreciar el comportamiento del sistema de detección 1 frente a la figura del cuerpo humano.

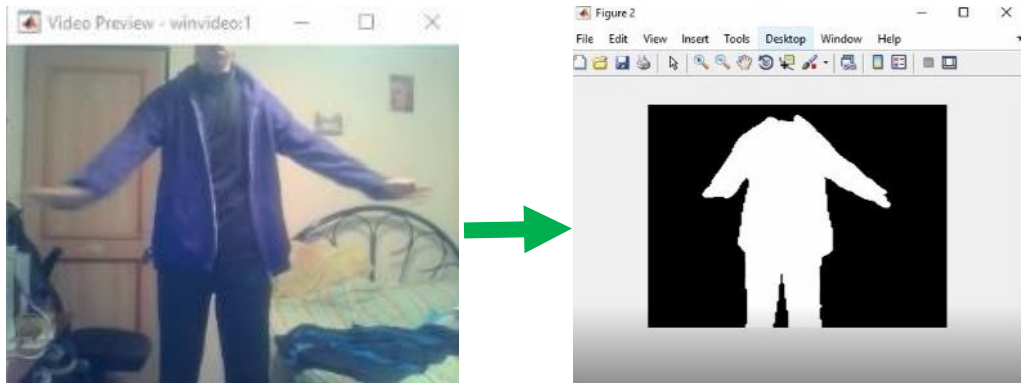


Figura 19. Segmentación por contraste (chaqueta 1).

Se debe considerar que a pesar de tener un fondo con bastante carga visual como se muestra en la figura 19 el sistema de detección demuestra resultados satisfactorios siempre y cuando alguna parte del cuerpo no se superponga con otro objeto que también sea detectado por el sistema de visión artificial ya que deformará la figura del cuerpo humano. Esto ocurre ya que varios objetos en el fondo también han sido detectados, sin embargo, al ser regiones con poca cantidad de píxeles se han eliminado de la imagen permitiendo una detección limpia del cuerpo humano, no obstante si algún objeto detectado del fondo se superpone con alguna extremidad del cuerpo humano, el sistema de visión artificial no podrá eliminar la región indeseada de píxeles ya que estará mimetizada con la extremidad del cuerpo afectando así la detección y deformando la figura.

La figura 20 muestra el comportamiento del sistema de detección 1 utilizando otra ropa con colores dentro de la gama del negro, gris y azul, a razón de corroborar el funcionamiento del sistema de detección frente al contraste que genera el fondo.

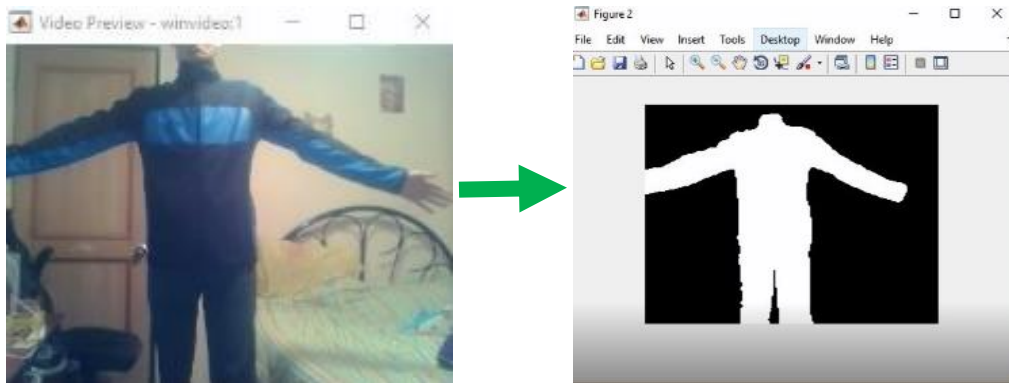


Figura 20. Segmentación por contraste (chaqueta 2).

Recordemos que la ropa con colores fuera de la gama del negro, gris y azul no son segmentados correctamente debido al bajo contraste con fondos iluminados y a la falta de efectividad en la combinación de matrices R, G, y B implementada, para segmentar en otras condiciones.

La figura 21 corresponde a la segmentación que realiza el sistema de detección 1 en la oscuridad o con poca luz del ambiente.

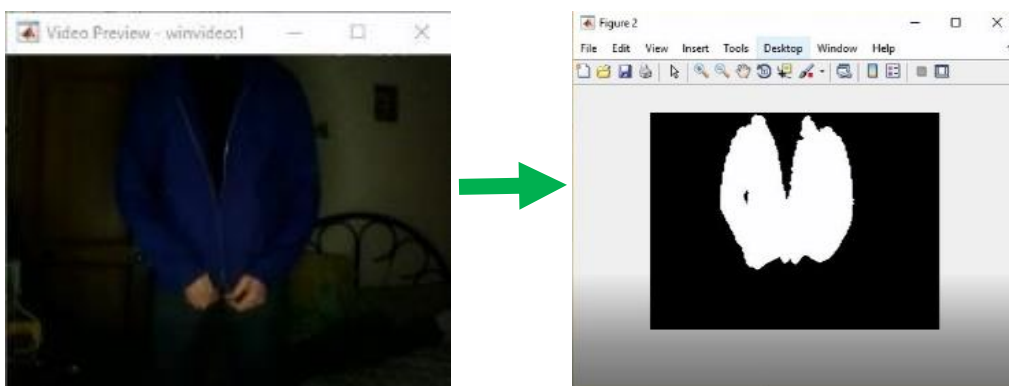


Figura 21. Segmentación por contraste (chaqueta 3).

Se pudo apreciar que en la oscuridad los colores oscuros de la ropa se mimetizan con el entorno afectando la segmentación.

4.1.1.1.3. *Detección de articulaciones del brazo.*

Para detectar las articulaciones del brazo humano se diseña e implementa un algoritmo el cual mediante un ciclo iterativo recorre todos los pixeles de la imagen umbralizada columna a columna; En cada columna de pixeles el algoritmo guarda la posición del primer pixel blanco que encuentre que corresponde al borde superior del brazo y también guarda la posición del último pixel blanco que corresponde al borde inferior del brazo (ver figura 23 y 24) con estas dos posiciones se calcula el promedio y se guarda ese valor medio del brazo en un arreglo unidimensional que se va llenando a medida que se va realizando este proceso con cada columna de pixeles de la imagen. Con la ayuda de una bandera **k** se determina si existe otra región blanca debajo de la región que corresponde al brazo, esta bandera servirá posteriormente para detectar la posición de la línea inferior de la pinza. La lógica del proceso se presenta en el diagrama de flujo 1 que se encuentra en la figura 22.

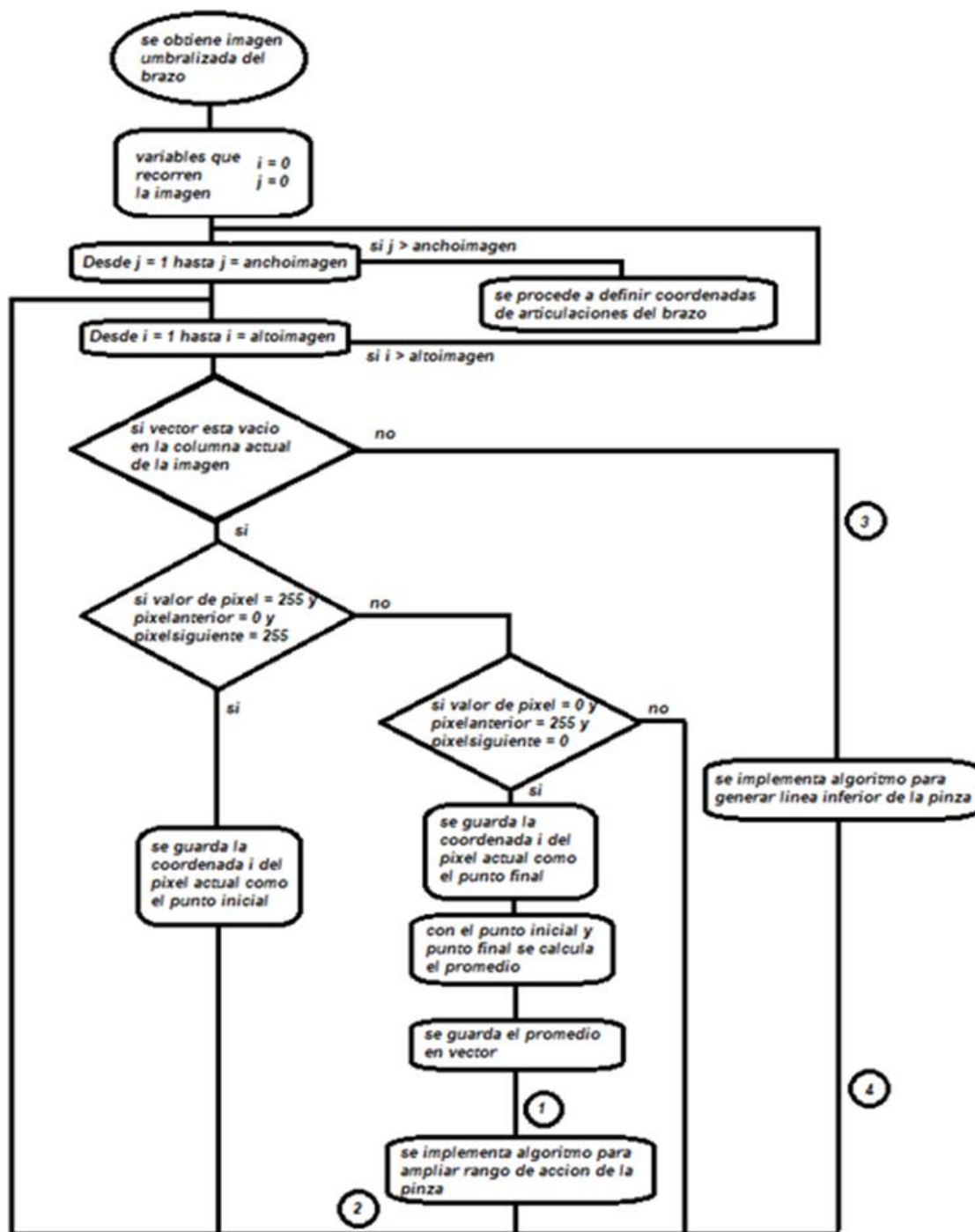


Figura 22. Diagrama de flujo 1 (Almacenamiento de coordenadas en vector).

Este proceso se realiza al finalizar la segmentación de la imagen y su objetivo es llenar el arreglo unidimensional “vector” que contiene las coordenadas correspondientes a las filas

que se encuentran como valor medio entre el contorno superior y el contorno inferior del brazo a lo largo de todas las columnas que abarcan la región blanca de píxeles en la imagen.

Los algoritmos para generar la línea inferior de la pinza y para ampliar el rango de acción de la pinza se explican más adelante y se detallan en los diagramas de flujo de las figuras 26 y 29 respectivamente.

Para validar el funcionamiento de este algoritmo se prueba el almacenamiento de datos del vector inicialmente en una imagen construida que simula la figura del brazo humano sin contemplar la pinza (ver figura 23). La línea central de píxeles negros corresponde a los datos almacenados por el vector evidenciando un correcto funcionamiento.

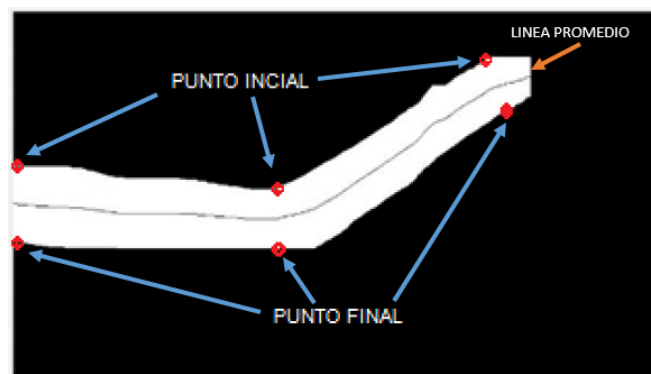


Figura 23. Datos almacenados por el vector 1.

Una vez se finaliza la obtención de datos del brazo, mediante un proceso experimental se determina que la ubicación del codo se encuentra aproximadamente a una distancia del 45 por ciento de la longitud del vector, del mismo modo se encontró experimentalmente que la posición de la muñeca equivale a un aproximado de 74 por ciento (Ver figura 24).

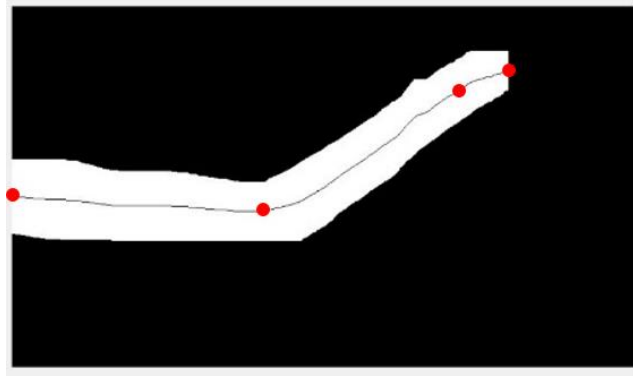


Figura 24. Imagen con puntos en las articulaciones.

4.1.1.1.4. Apertura y cierre de la pinza.

Para detectar la apertura y cierre de la pinza se implementa el uso de un segundo vector. Para entender esto hay que tener en cuenta que cuando la pinza se encuentra abierta hay dos regiones blancas en varias columnas de píxeles. Una región corresponde a los dedos de la mano y la otra al dedo pulgar, para esta situación un segundo vector almacena las posiciones de los píxeles promedio que se encuentran en el segmento de la pinza inferior o el dedo pulgar de la mano, esto se explica mejor en el diagrama de flujo de la figura 25.

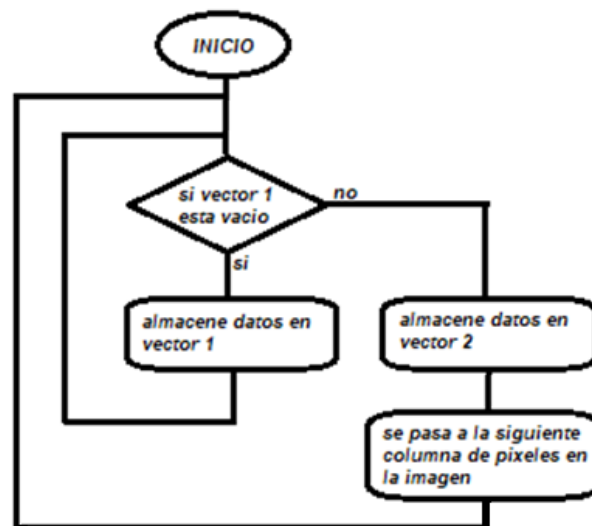


Figura 25: Diagrama de flujo de almacenamiento de datos.

Al igual que con las articulaciones el objetivo de este proceso es obtener unas coordenadas determinadas de los vectores 1 y 2 ya que la pinza requiere de 3 coordenadas determinadas para poder ser detectada, la primera coordenada corresponde a la longitud final de los dedos, la segunda coordenada a la longitud final del dedo pulgar y la tercera coordenada a la muñeca.

El uso del vector 2 permite obtener la coordenada faltante para detectar la pinza, ya que la posición de la muñeca y los dedos ya han sido obtenidas a partir del uso del vector 1. La lógica para almacenar los datos en el vector 2 se presenta en el diagrama de flujo de la figura 26.

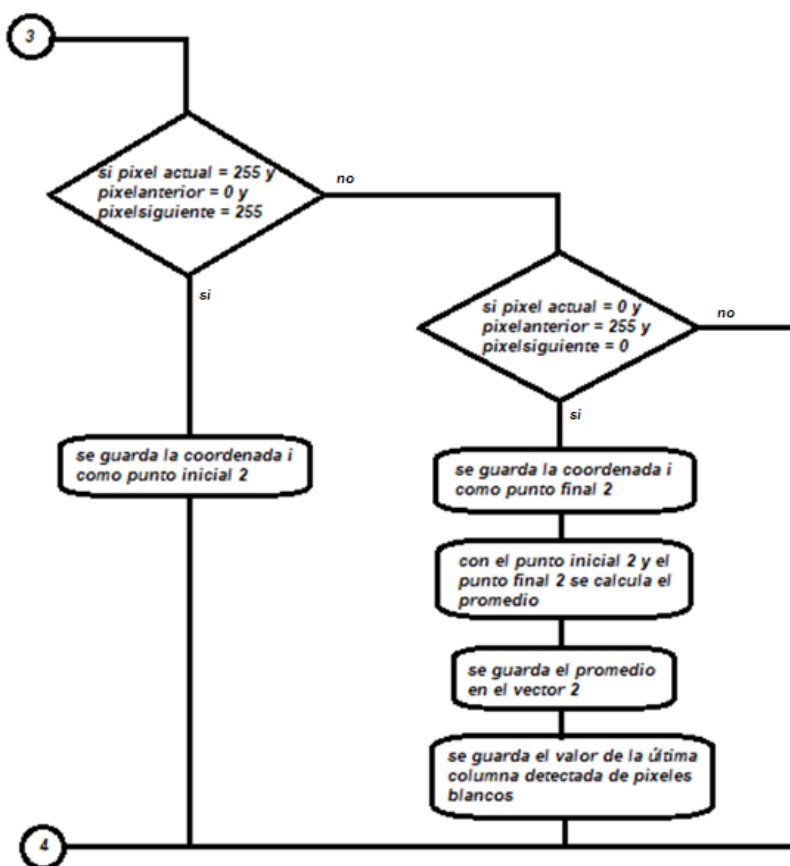


Figura 26. Diagrama de flujo para generar línea inferior de la pinza.

Cabe resaltar que la figura 26 hace parte de la figura 22 y tiene como objetivo almacenar en el arreglo unidimensional “vector 2” la coordenada “i” del valor medio de cualquier región blanca que se encuentre debajo de otra región blanca ya detectada y almacenada por el arreglo “vector 1”. El proceso de almacenar datos en el vector 2 tiene como objetivo encontrar las coordenadas del segmento inferior de la pinza o en otras palabras del dedo pulgar en la mano ya que es la única región de píxeles que se encuentra debajo de otra región de píxeles correspondiente a los dedos de la mano. En la figura 27 se muestra gráficamente el proceso de almacenamiento de datos del vector 1 y 2, donde $k = 0$ representa que no se ha encontrado una región de píxeles blancos en el recorrido vertical y $k = 1$ representa que ya se han almacenado los datos del vector 1 y los próximos datos serán almacenados en el vector 2.

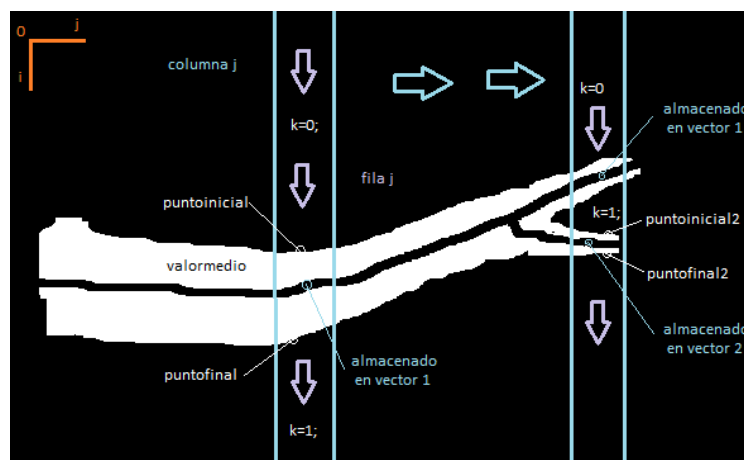


Figura 27. Almacenamiento de coordenadas en vector 1 y 2.

4.1.1.1.5. Ampliación del rango de acción de la pinza.

Al realizar el almacenamiento de datos del vector 2 se presenta un problema en caso tal de que el dedo pulgar se encuentre en una coordenada “j” mayor que los dedos que corresponden a la pinza superior, en otras palabras, se presenta un problema cuando la pinza

está apuntando hacia arriba debido a que llegará un punto en el proceso de almacenamiento en el que el sistema de visión artificial empezará a guardar las coordenadas del valor medio del dedo pulgar en el vector 1 y no en el vector 2 como debería hacerlo, en consecuencia los datos tanto del vector 1 como del vector 2 serán erróneos (ver figura 28).

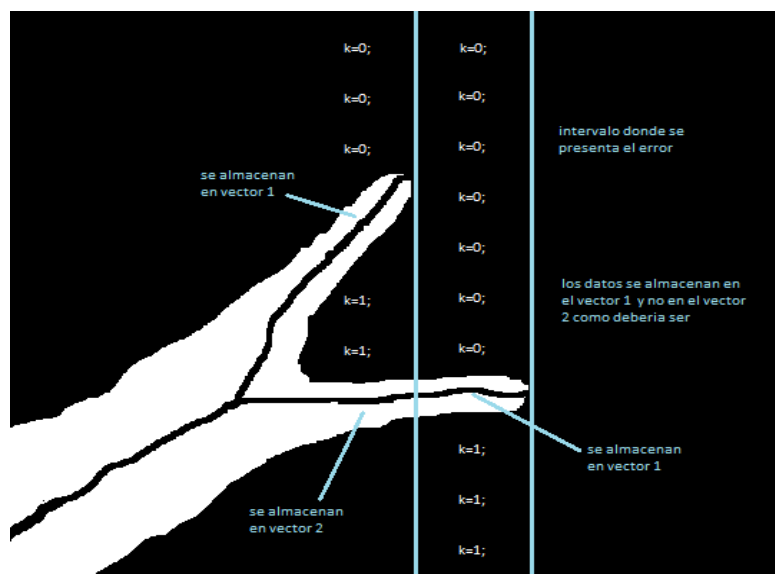


Figura 28. Error cuando la pinza apunta hacia arriba.

Para evitar esta situación se implementa el algoritmo de ampliación del rango de acción de la pinza el cual tiene como objetivo almacenar los datos en el vector 2 que se encuentran en el intervalo donde se presenta el error, para ello se evalúa en cada columna de la imagen si el vector1 tiene un cambio abrupto de coordenada “i” de una columna “j-1” a una columna “j” es decir, si presenta una discontinuidad de una columna de pixeles a otra. Si este es el caso los datos siguientes a la discontinuidad corresponden a las coordenadas del dedo pulgar y por lo tanto se deben almacenar estos datos en el vector2 y no en el vector1. El algoritmo para realizar este proceso se presenta en el diagrama de flujo mostrado en la figura 29. Donde la variable **flag** permite evidenciar la discontinuidad siendo flag = 1 cuando ya se ha presentado la discontinuidad y flag = 0 cuando aún no se ha presentado la discontinuidad.

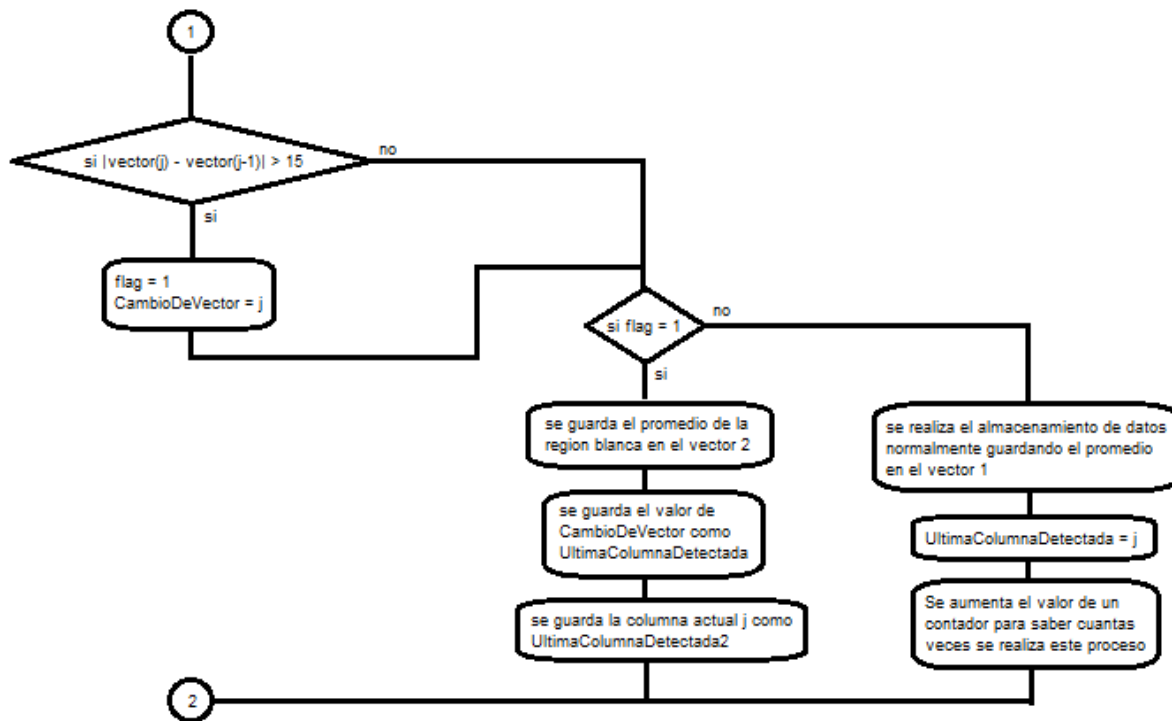


Figura 29. Diagrama de flujo para ampliar el rango de acción de la pinza.

Este algoritmo permite detectar la pinza aun cuando exista una discontinuidad en el almacenamiento de datos, permitiendo así una ampliación del rango de acción de la pinza (ver figura 30).

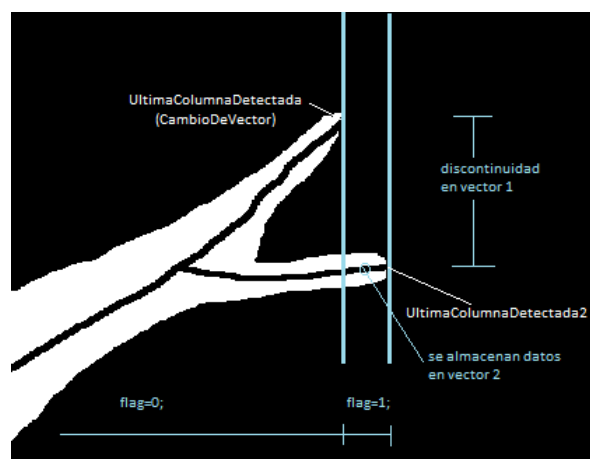


Figura 30. Ampliación de rango de acción de la pinza.

Los valores de ÚltimaColumnaDetectada y ÚltimaColumnaDetectada2 son necesarios para conocer la distancia en el eje “j” que hay entre los dedos de la pinza superior y el dedo pulgar de la pinza inferior, de la siguiente forma:

$$R_{pinza} = \text{ÚltimaColumnaDetectada2} - \text{ÚltimaColumnaDetectada} \quad (15)$$

La distancia Rpinza servirá para graficar las líneas que corresponden al esqueleto posteriormente.

4.1.1.1.6. Creación del esqueleto.

La creación del esqueleto es una parte fundamental del sistema de visión artificial ya que permite evidenciar visualmente si la distancia existente entre segmento y segmento del brazo humano está correctamente detectada, en otras palabras, se verifica si la proporción del esqueleto coincide con la proporción del brazo humano. Inicialmente se busca obtener el esqueleto con algoritmos de adelgazamiento, cuya función es reducir el número de píxeles de la figura como lo hace el algoritmo de Zhang & Suen, (1984) obteniendo una figura esqueletizada del objeto detectado, pero con bastantes irregularidades que no permiten una creación clara del esqueleto (ver figura 31).



Figura 31. Adelgazamiento. Imagen tomada de www.incress.com.

Por este motivo se prefiere crear el esqueleto a partir de la unión entre las articulaciones detectadas en la sección anterior dibujando líneas rectas sobre la imagen obteniendo cada segmento del brazo en forma de esqueleto completamente limpio y sin raíces o algún otro tipo de ruido en la esquelitización como se aprecia en la figura 32.

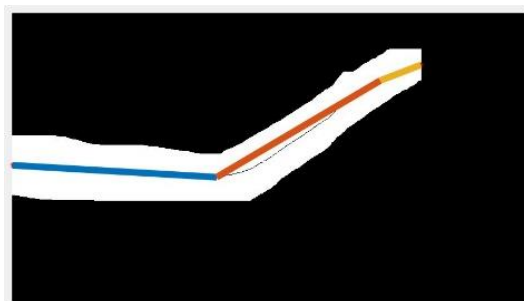


Figura 32. Imagen esqueleto.

La creación del esqueleto es un proceso que depende del correcto funcionamiento de los procesos previos, si alguno de estos procesos presenta alguna falencia debido a factores externos como lo son algunos fondos con mucha carga visual, el sistema calculará de forma errónea los puntos de las articulaciones y por consiguiente el esqueleto. Para solucionar este problema se realiza una saturación de valores para cada articulación del brazo, permitiendo que la detección no sufra alteraciones dentro del intervalo de saturación, es decir, si se desvía un punto de una articulación del brazo más de cierto rango, la coordenada de la articulación se mantiene en el valor anterior y no puede moverse más de esa posición, evitando variaciones indeseadas de las articulaciones en la imagen. En la figura 33 se aprecia el resultado de la saturación a pesar de una segmentación indeseada de pequeñas áreas que alteran la figura del brazo humano y la posición de las articulaciones.

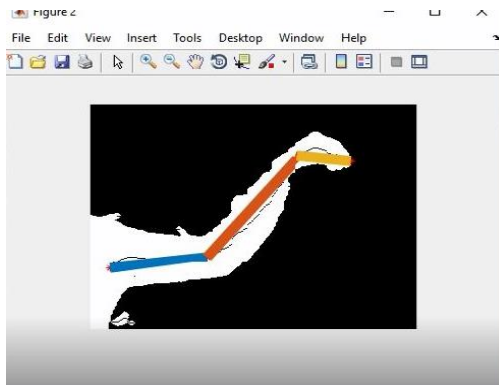


Figura 33. Esqueleto con saturación de valores

Para desarrollar el esqueleto correspondiente a la apertura y cierre de la mano humana se implementa una segunda línea que une la articulación de la muñeca con la coordenada del dedo pulgar, esto ocurre una vez la pinza está abierta (ver figura 34), si la pinza se encuentra cerrada la línea que une estas dos articulaciones desaparece (ver figura 35).



Figura 34. Pinza abierta.



Figura 35. Pinza cerrada.

Para la creación de la segunda línea articulada correspondiente a la pinza se evalúa el último dato almacenado en el vector 2 correspondiente a la posición final del dedo pulgar. Esta posición final es posible conocerla gracias al parámetro “Rpinza” (ecuación 15) que contiene la distancia en el eje X entre el valor final del segmento superior de la pinza y el segmento inferior, este parámetro permite conocer la coordenada en el eje X del dedo pulgar a partir de la coordenada en el eje X de los dedos de la mano permitiendo así la

implementación de la línea dibujada sobre el dedo pulgar siempre y cuando la pinza esté abierta.

4.1.1.2. Sistema de detección 2 – Con guante bicromático.

Para este sistema de detección se utiliza un guante de color rojo con segmentos en color amarillo que representan las articulaciones, este guante bicromático se utiliza con el fin de obtener la imagen umbralizada del brazo humano y la ubicación de las articulaciones a partir de la segmentación de los colores rojo y amarillo, para generar alto contraste con las tonalidades que se encuentran comúnmente en el entorno. En las figuras 36 y 37 se aprecia la apariencia del guante bicromático con distintas iluminaciones del ambiente.



Figura 36. Guante bicromático día.



Figura 37. Guante bicromático noche.

Para que este sistema de detección logre su objetivo se siguen las siguientes etapas:

- Segmentación por color
 - Detección de colores RGB
 - Umbralización
- Detección de articulaciones
- Creación de esqueleto
- Ampliación del rango de movimiento permisible

4.1.1.2.1. Segmentación por color.

4.1.1.2.1.1. Detección de colores RGB.

La imagen capturada en tiempo real por la cámara es una agrupación de 3 matrices superpuestas cuyo resultado es una imagen donde se puede apreciar a simple vista cualquier combinación de colores, las matrices que componen esta imagen son conocidas como R, G, B o matrices ROJA ,VERDE y AZUL (figura 38), y cada matriz contiene la información respectiva de la intensidad de su color en cada pixel de la imagen, siendo cada pixel un registro de hasta 8 bits de almacenamiento que es el equivalente a un intervalo de 0 a 255 en formato decimal.

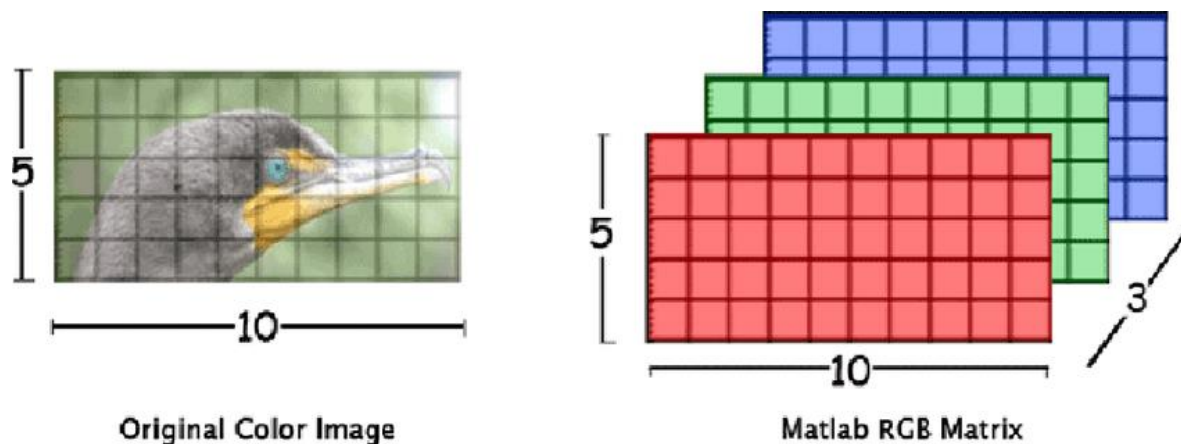


Figura 38. Matrices RGB. Tomado de (Singh, 2015)

Para el desarrollo de este sistema de visión artificial se busca obtener la información de la matriz R y la matriz G de la imagen capturada por la cámara ya que el guante bicromático está compuesto por los colores rojo y amarillo, cabe resaltar que al trabajar con imágenes el color amarillo opera como un tono aditivo secundario lo que significa que puede ser conformado por dos o más colores primarios del sistema aditivo de síntesis de color o en otras palabras puede obtenerse del espacio de color RGB.

Para el caso específico del amarillo, este es conformado por la matriz R y la matriz G, ya que tanto para detectar el color rojo o el color amarillo del guante se necesitan una de estas dos matrices o ambas.

El objetivo es obtener dos imágenes en escala de grises a partir de la matriz R y la matriz G, estas imágenes tienen las siguientes características:

- La imagen número uno debe resaltar en intensidad los segmentos del guante bicromático en color amarillo siendo los pixeles amarillos en la imagen los que posean un valor mayor entre 0 y 255 que los demás pixeles de la imagen que correspondan a otros colores. Esta imagen servirá para conocer la posición de las articulaciones del brazo en cada captura de imagen que haga el video. Para obtener esta imagen se combinan las matrices R y G dándole mayor peso a la información de la matriz G de tal forma que el color amarillo sea el color con mayor intensidad (figura 39). En la ecuación 16 se muestra la combinación de matrices implementada.

$$articulaciones = 2 * (gris - B - 0.2 * R) \quad (16)$$

Donde:

gris = imagen original en escala de grises

- La imagen numero dos debe resaltar la intensidad de los colores rojo y amarillo es decir resaltar todo el guante bicromático en relación con la intensidad del fondo de la imagen. Esta imagen servirá posteriormente para conocer la imagen umbralizada del brazo completo. Para obtener esta imagen se combina el espacio de color R, G, B dándole mayor prioridad a la información de la matriz R y G en comparación con los pixeles con otras tonalidades (figura 40). En la ecuación 17 se muestra la combinación de matrices implementada.

$$\text{Brazo completo} = R - 2 * B + \text{articulaciones} \quad (17)$$

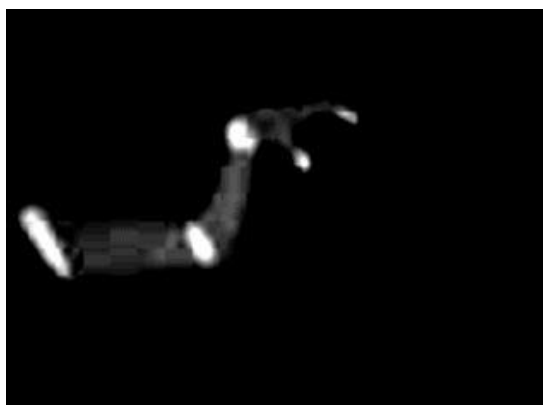


Figura 39. Articulaciones resaltadas en la imagen.



Figura 40. Brazo completo resaltado en la imagen.

Las constantes implementadas en las ecuaciones 16 y 17 se definieron realizando diversas pruebas para obtener la mejor segmentación de los colores rojo y amarillo. Por otra parte, el uso de la matriz B permite eliminar gran parte del fondo obteniendo así una mayoría de píxeles de color negro en las imágenes resultantes y una minoría de píxeles con tonalidades claras que corresponden a las articulaciones y al guante bicromático completo debidamente segmentados. Esto se puede apreciar en los histogramas de la figura 41 donde se evidencia el aumento de píxeles negros y la reducción de píxeles claros en la imagen de las articulaciones y la imagen del brazo completo.

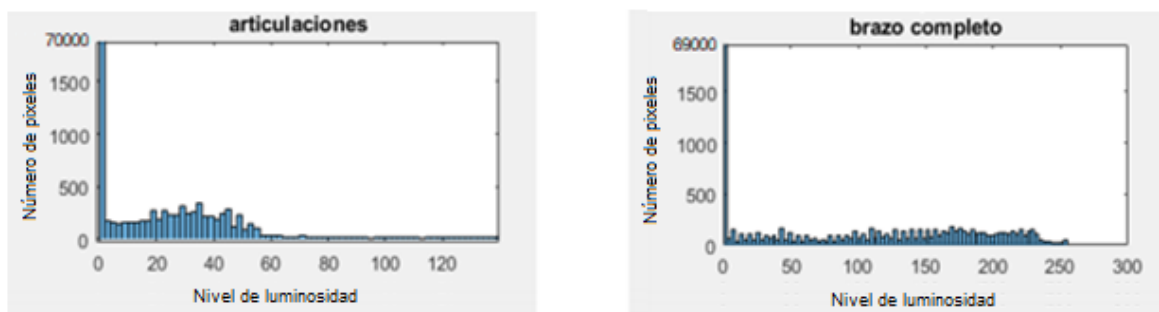


Figura 41. Histograma de articulaciones y brazo completo.

4.1.1.2.1.2. Umbralización.

Para obtener una imagen umbralizada a partir de las imágenes en escala de grises obtenidas anteriormente (figura 39 y 40) se realiza un proceso de umbralización que consiste en transformar la información de 8 bits de cada pixel de la imagen a escala de grises, en una información de un único bit por pixel, pasando de una imagen con una variación lumínica entre 0 y 255 a una entre 0 y 1, en otras palabras se convierte la imagen de entrada en una imagen a blanco y negro cuyo factor que determina cuales pixeles de la imagen se convierten en blancos y cuales en negros se le conoce como umbral.

El método más común y utilizado para determinar el umbral es el método de Otsu explicado anteriormente en el sistema de detección 1(ver página 20). Con ayuda de esta metodología se obtiene el umbral más adecuado para la umbralización o binarización de la imagen, de acuerdo a los factores variables del entorno que puedan afectar la imagen.

La umbralización permite segmentar los colores rojo y amarillo resaltados en las figuras 39 y 40, Lo que significa que nuevamente se obtienen dos imágenes, la primera imagen segmenta las articulaciones, y la segunda imagen segmenta el brazo completo.

La segmentación de las articulaciones (figura 42) se hace con el fin de encontrar la posición en coordenadas (x, y) de las regiones de pixeles blancos en la imagen para así conocer la localización de las articulaciones, este proceso se detalla en la sección de detección de articulaciones.

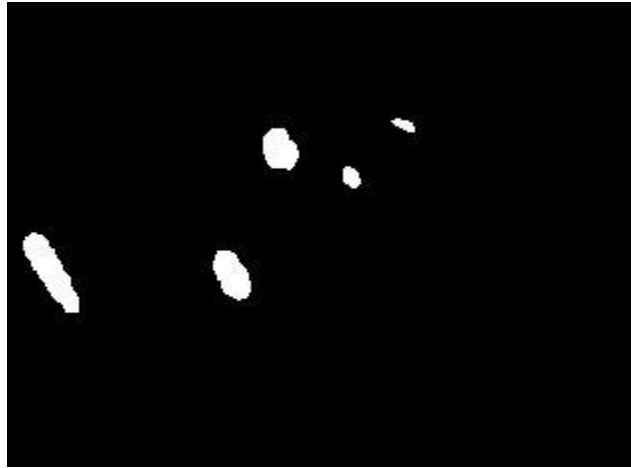


Figura 42. Umbralización de articulaciones.

La segmentación del brazo completo (figura 43) se realiza con el fin de servir como un indicador para el usuario del sistema, de tal forma que permita evidenciar si la detección del brazo es correcta, sin embargo, esta segmentación no es tan relevante para los procesos posteriores.

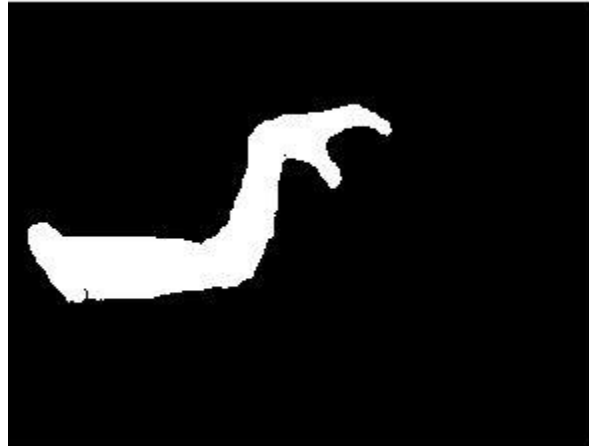


Figura 43. Umbralización de brazo completo.

De acuerdo a factores variables que presenta la imagen RGB original es posible que el sistema de visión artificial segmente regiones de la imagen que sean indeseadas para una correcta detección. Estas regiones indeseadas pueden ser consecuencia de objetos en el fondo de la imagen, efectos de iluminación indeseados, o cambios en la saturación o contraste de

la imagen RGB que resaltan regiones de la imagen, afectando la umbralización y los procesos posteriores.

Para evitar una mala detección se debe asegurar que la imagen esté libre de impurezas y para ello se eliminan estas áreas con muy baja cantidad de píxeles blancos de la misma forma que con el sistema de detección 1 (ver página 23).

4.1.1.2.2. *Detección de articulaciones.*

La detección de articulaciones se hace a partir de la imagen umbralizada de las articulaciones (figura 42) donde es posible conocer la región de píxeles a la que corresponde cada articulación por medio de un algoritmo de etiquetado (M, 2009). El proceso consiste en etiquetar las agrupaciones de píxeles blancos de la imagen con números, para ello es necesario tener como matriz de entrada del algoritmo la imagen binarizada y a la salida se obtendrá la imagen etiquetada (figura 44).

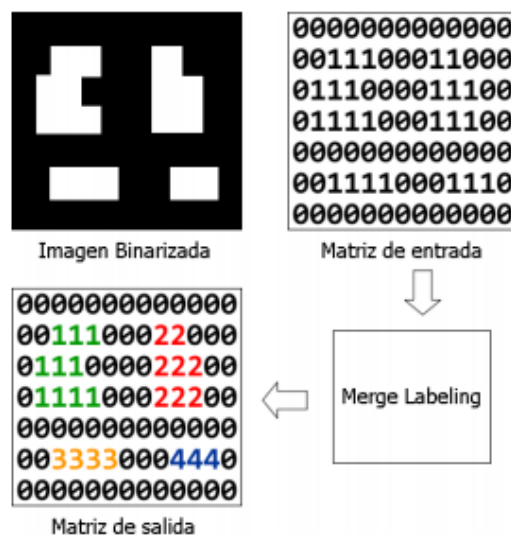


Figura 44. Algoritmo de etiquetado. Tomado de (M, 2009)

La lógica del algoritmo de etiquetado se presenta en el diagrama de flujo de la figura 45.

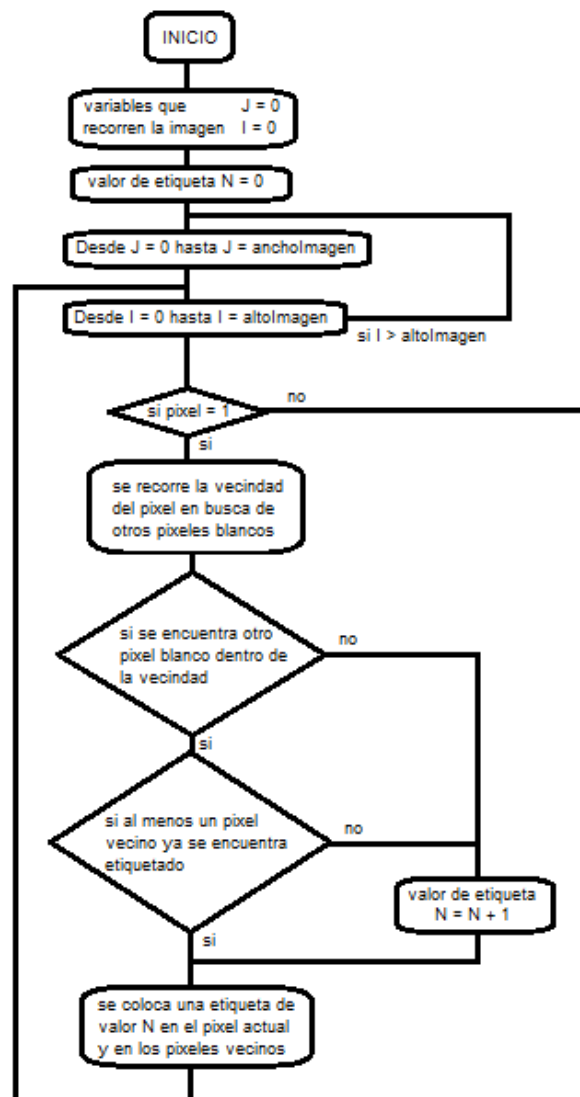


Figura 45. Lógica del algoritmo de etiquetado.

Este proceso de etiquetado tiene como fin relacionar una determinada etiqueta con una articulación, y asigna una numeración a las regiones blancas de izquierda a derecha de la imagen, es decir que la primera región de izquierda a derecha será considerada como el hombro, la segunda como el codo, la tercera como la muñeca, y la cuarta y quinta como la pinza. (Ver figura 46).

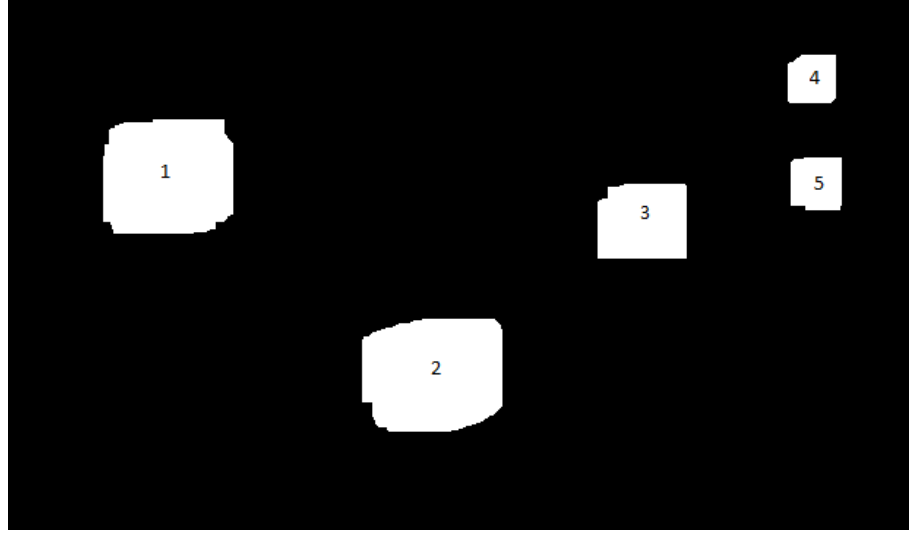


Figura 46. Proceso de etiquetado de articulaciones.

Para realizar una correcta detección de las articulaciones no es suficiente con un proceso de etiquetado de regiones blancas, sino que adicionalmente es necesario detectar la posición exacta de las articulaciones, para ello se calcula el centroide de cada región obteniendo una coordenada (x, y) correspondientes al ancho y alto de la imagen respectivamente. Teniendo en cuenta que las regiones blancas son figuras amorfas, y una imagen es una representación discreta de la realidad, el cálculo de los centroides está dado por las ecuaciones 18 y 19.

$$\bar{x} = \frac{\sum \tilde{x}_i * A_i}{\sum A_i} \quad (18)$$

$$\bar{y} = \frac{\sum \tilde{y}_i * A_i}{\sum A_i} \quad (19)$$

Donde:

A_i = área de cada pixel de la región blanca

(\tilde{x}, \tilde{y}) = coordenadas del centroide de cada pixel de la región blanca

(\bar{x}, \bar{y}) = coordenadas del centroide de la región completa

4.1.1.2.3. Creación de esqueleto.

La creación del esqueleto se realiza a partir de la imagen umbralizada, etiquetada, y con los centroides calculados. El proceso consiste en dibujar líneas que unan los centroides de cada región blanca de izquierda a derecha de la misma forma que se realizó el proceso de etiquetado, el resultado serán 4 líneas, la primera línea será la unión de la etiqueta 1 y 2, es decir el hombro y el codo, la segunda línea será la unión de la etiqueta 2 y 3 es decir el codo y la muñeca, y las dos últimas líneas serán las uniones de las etiquetas 3 a 4 y 3 a 5 correspondientes a la pinza (Figura 47).

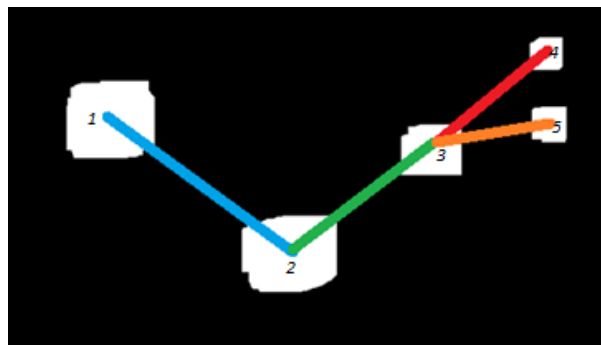


Figura 47. Creación de esqueleto.

Una vez obtenido el esqueleto a partir de la imagen umbralizada de las articulaciones, se superpone el esqueleto con la imagen umbralizada del guante bicromático completo (figura 48)

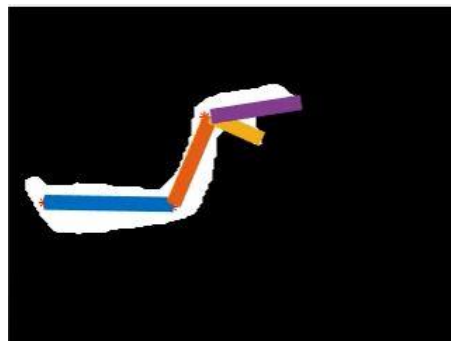


Figura 48. Detección del guante bicromático.

Cabe aclarar que, si la pinza se encuentra cerrada, la imagen umbralizada de las articulaciones solo tendrá 4 regiones de píxeles blancos en vez de 5, si este es el caso solo se dibuja 3 líneas para crear el esqueleto y se omite la línea que une la etiqueta 3 con la etiqueta 5 es decir se omite una línea correspondiente a un segmento de la pinza como se muestra en la ecuación 20.

$$\begin{aligned} y_4 &= 0 & \text{para } N &= 4 \\ y_4 &= m_4 x_4 & \text{para } N &= 5 \end{aligned} \quad (20)$$

Donde:

y_4 = recta correspondiente a la línea inferior de la pinza con respecto al sistema de referencia de la muñeca.

m_4 = pendiente de la recta de la línea inferior de la pinza.

N = número de etiquetas en la imagen.

4.1.1.2.4. Ampliación del rango de movimiento permisible.

Una vez se ha creado el esqueleto del brazo, el sistema de visión artificial está listo para calcular los parámetros angulares necesarios para controlar el movimiento del brazo robótico, sin embargo, para este sistema de visión artificial se realiza una pequeña ampliación del rango de movimiento permisible ya que este proceso de detección presenta un problema, en el cual el rango de movilidad del brazo humano frente a la cámara está limitado a la incapacidad del proceso de etiquetado para cambiar de numeración, un ejemplo claro de esto es cuando el brazo humano se encuentra retraído completamente (figura 49).

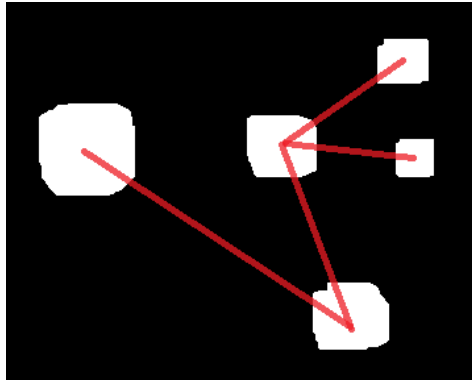


Figura 49. Unión de etiquetas correctamente.

En este caso el label o etiqueta de la muñeca se encuentra más a la izquierda que la etiqueta del codo, por lo que el sistema automáticamente numera la etiqueta de la muñeca con el número 2, en consecuencia el sistema de visión artificial no realiza una correcta detección ya que ahora la muñeca se reconoce como el codo y viceversa, esto quiere decir que la movilidad del brazo humano en la detección está limitado a que la coordenada X de la muñeca correspondiente al ancho de la imagen sea mayor a la del codo. Ver figura 50.

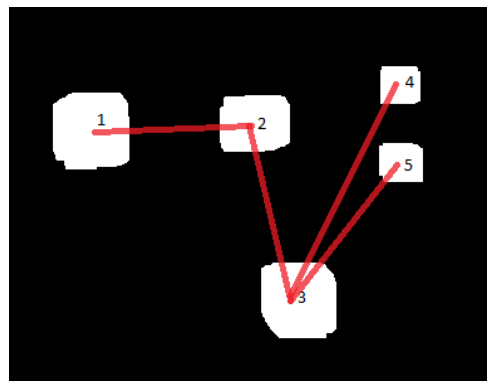


Figura 50. Error al unir los labels.

Esto ocurre ya que las etiquetas o labels están definidos para ser una articulación específica siendo la etiqueta 2 la articulación del codo y la etiqueta 3 la muñeca generando este problema cuando el brazo se encuentra completamente retraído.

Para solucionar este problema se invierten las etiquetas 2 y 3 cuando se trate de superar el rango de movimiento permisible, la metodología implementada se presenta en el diagrama de flujo de la figura 51.

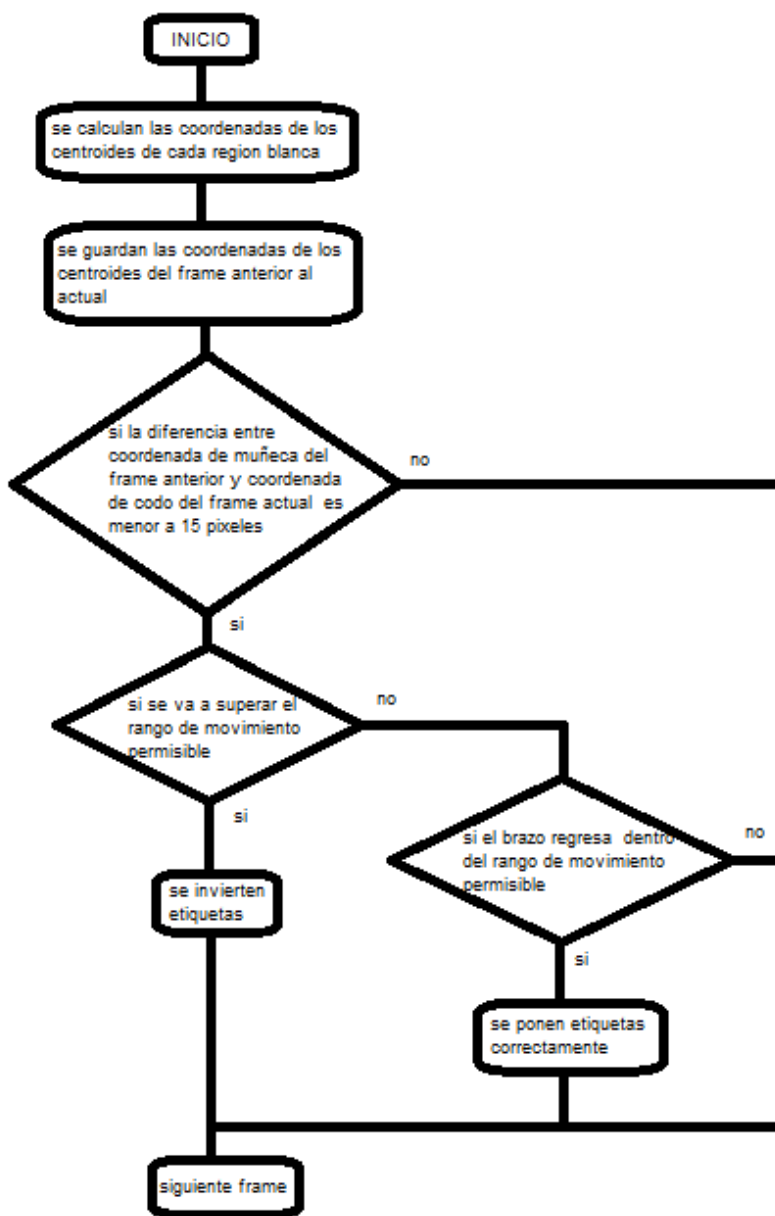


Figura 51. Diagrama de flujo para corrección de la numeración de los labels.

La corrección de las etiquetas o labels para ampliar el rango de movimiento permisible del brazo durante la detección se basa principalmente en evaluar si las coordenadas del centroide del codo y las coordenadas del centroide de la muñeca son

cercanas al pasar del frame anterior al actual (diferencia menor a 15 pixeles), si esto ocurre quiere decir que se presentó un cambio de etiquetado de un frame a otro ya que ahora la región blanca de la muñeca se encuentra situada en una posición más cercana al origen de la imagen que la región blanca del codo, en consecuencia el sistema de visión artificial renombrara las etiquetas estableciendo que la antigua coordenada de la muñeca sea ahora la nueva coordenada del codo y viceversa, por esta razón se debe hacer la debida corrección realizando la inversión de etiquetas.

Al solucionar este problema se aumenta el rango de movilidad en la detección del brazo humano frente a la cámara debido a que ahora el brazo puede retraerse mucho más y el sistema seguirá detectándolo correctamente.

4.1.1.3. Sistema de detección 3 – Con dispositivo Kinect Xbox 360.

El sistema de detección 3 busca detectar el brazo humano de una forma más efectiva que con los sistemas anteriores. Consiste en el uso del sistema de sensorizado Kinect diseñado para la consola de videojuegos Xbox® 360 pero adaptado al uso en PC por medio de un adaptador tipo USB que permite la conexión entre este periférico y el PC. El uso del sensor Kinect permite la detección de cualquier brazo humano sin necesidad de un guante bicromático como el sistema de detección 2 o un ambiente con una luz determinada para detectar el brazo como ocurre con el sistema de detección 1.

Para que el sistema de detección 3 logre su objetivo se siguen las siguientes etapas:

- Adquisición de imagen
- Segmentación por distancia
- Detección de articulaciones

- Creación de esqueleto

4.1.1.3.1. *Adquisición de imagen.*

El dispositivo Kinect(Microsoft, 2012) utiliza dos tipos de cámaras diferentes (figura 52), la primera es una cámara RGB estándar y la segunda es una cámara IR o cámara de profundidad que obtiene una imagen en escala de grises a partir de la distancia a la que se encuentren los objetos frente al dispositivo Kinect. Este dispositivo de sensorizado permite obtener imágenes de una cámara o de otra incluyendo también la posibilidad de obtener una imagen combinada producto de la convolución de las imágenes de ambas cámaras, sin embargo, para la adquisición de una imagen que permita detectar el brazo de una persona no hace falta el uso de la cámara RGB integrada en el Kinect ya que únicamente con la cámara de profundidad es posible obtener una detección adecuada.



Figura 52. Partes del Kinect. Tomado de (Microsoft, 2012)

El proceso de adquisición de imagen consiste en lo siguiente; Un proyector infrarrojo del sensor Kinect dispara un patrón de infrarrojos hacia el área de proyección y la cámara de profundidad infrarroja analiza el tamaño de los puntos de los laser en cada región de la malla disparada por el proyector, de esta forma la cámara de profundidad puede saber si un objeto

está cerca o lejos del Kinect y esa profundidad se trasmite en términos de una imagen en escala de grises donde lo que está alejado se muestra como blanco, y lo que está cerca se visualiza como negro, en las figuras 53 y 54 se muestran ejemplos de su funcionamiento.



Figura 53. Imagen de profundidad resultante. Tomado de (Mind treat studios, 2012)

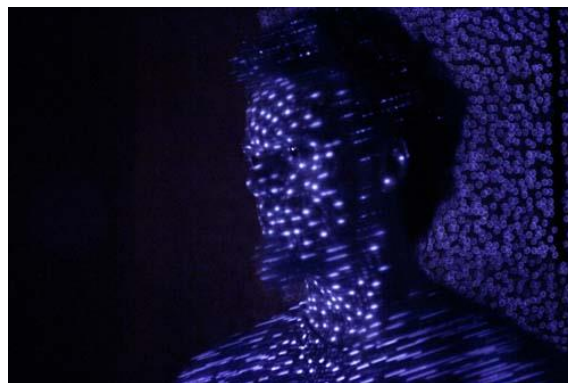


Figura 54. Proyector infrarrojo. Tomado de (Garaizar Sagarminaga, 2013)

4.1.1.3.2. Segmentación por distancia.

Gracias a la cámara de profundidad se puede obtener una imagen en la cual a partir de una gama de grises (ver figura 55) se puede realizar una segmentación por distancia cuyo objetivo es segmentar únicamente los objetos que estén en un rango de distancia definido por dos umbrales que delimiten el rango de tonos de grises que serán convertidos en regiones de píxeles blancos en el proceso de umbralización estableciendo así que

cualquier tono de gris que este fuera de los umbrales previamente definidos serán píxeles de color negro en la imagen umbralizada (figura 55 y figura 56)

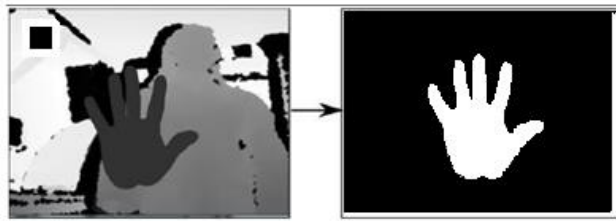


Figura 55. Segmentación por distancia ejemplo1 tomado de (Ebert, Hatch, Thali, & Ross, 2013)



Figura 56. Segmentación por distancia, ejemplo 2.

Para seleccionar los umbrales adecuados se busca una relación umbral-distancia en la cual se establezca el rango de distancia deseado en el que el sistema de visión artificial detecta el brazo humano (figura 57). El rango de distancia para segmentar el brazo humano es definido de 0.8 metros a 1.3 metros por los autores, este rango puede cambiar de acuerdo con la conveniencia del usuario teniendo en cuenta los límites de operación del Kinect establecidos entre 0.8 m y 2.5 m para una detección adecuada.

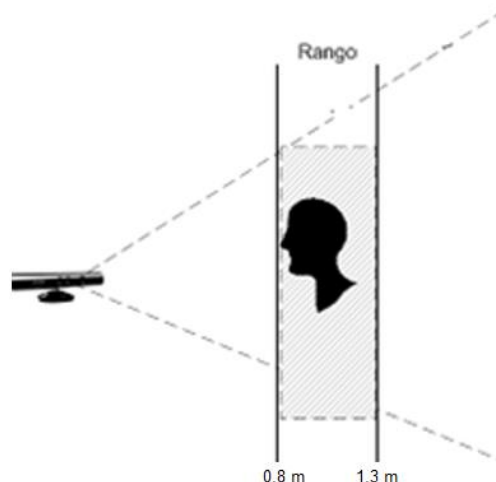


Figura 57. Segmentación por distancia.(Serrano, Bueno, Salido, & Déniz, 2011)

Cualquier objeto que se encuentre en el rango de distancia previamente definido será detectado, por lo que es importante asegurarse que únicamente se encuentre el brazo de la persona en ese rango de distancia para que la segmentación sea adecuada.

4.1.1.3.3. Detección de articulaciones.

Con la imagen umbralizada del brazo detectado se aplica la misma metodología realizada en el sistema de detección 1 para encontrar las coordenadas (x,y) de cada articulación(ver página 27), donde a partir de un análisis pixel por pixel de la imagen se obtiene un arreglo unidimensional en el cual los datos almacenados son las coordenadas en el eje (y) de los pixeles promedio del brazo y las posiciones de los datos en el arreglo corresponden a la posición en el eje (x) de cada uno de esos pixeles. Este proceso se hace con el fin de definir las coordenadas de las articulaciones a partir de la longitud total de arreglo, por otra parte, el proceso para detectar la apertura y cierre de la pinza también sigue la misma metodología que el sistema de detección 1 (ver página 30), donde se implementa un segundo arreglo para almacenar las coordenadas del segmento inferior de la pinza. El resultado final

de este proceso son las coordenadas de 4 puntos en la imagen si la mano del usuario se encuentra con la pinza cerrada o 5 puntos en la imagen si la mano del usuario se encuentra con la pinza abierta (figura 58).

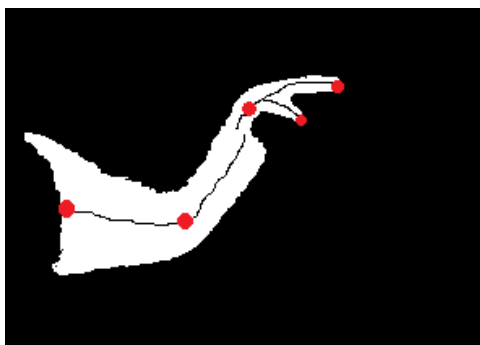


Figura 58. Detección de articulaciones.

4.1.1.3.4. Creación de esqueleto.

El proceso para la creación del esqueleto consiste en dibujar rectas que unan las articulaciones previamente calculadas, la metodología aplicada es la misma utilizada en el sistema de detección 1 (ver página 35) donde el resultado es la unión de 3 líneas correspondientes al humero, antebrazo y mano, sin embargo, si la pinza está abierta se dibuja una cuarta línea para representar el segmento inferior de la pinza (figura 59).

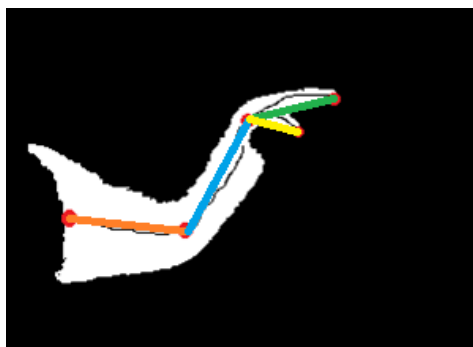


Figura 59. Creación del esqueleto.

Ya que la metodología para obtener el esqueleto del brazo es la misma aplicada en el sistema de detección 1 se realiza también una ampliación del rango de acción de la pinza (ver página 32) para mejorar la detección de la pinza y así permitir una mayor movilidad del brazo

por parte del usuario del sistema sin afectar la detección de articulaciones y por consiguiente la creación del esqueleto.

4.1.2. Cálculo de parámetros requeridos.

4.1.2.1. Cálculo de la posición angular de cada segmento del brazo.

La posición angular de cada segmento del brazo es el objetivo por el cual se implementa el sistema de visión artificial, debido a que las posiciones angulares son los datos requeridos para controlar el brazo robótico, generando así una emulación exitosa del movimiento.

La cantidad de parámetros angulares que se deben calcular está definida por la cantidad de articulaciones del brazo, esto da como resultado un cálculo de 3 parámetros angulares q_1 , q_2 , y q_3 correspondientes a los ángulos del hombro, codo, y muñeca respectivamente.

Para realizar un cálculo adecuado de estos parámetros es importante tener en cuenta que los ángulos del antebrazo y muñeca (q_2 y q_3) no se calculan con respecto a un eje fijo (figura 60), sino que deben ser calculados con respecto a un eje móvil (figura 61), para el caso del ángulo q_1 correspondiente al humero el cálculo si se realiza con respecto a un eje fijo debido a que este parámetro varía con respecto a la base estática en la que se apoya el hombro del brazo robótico.

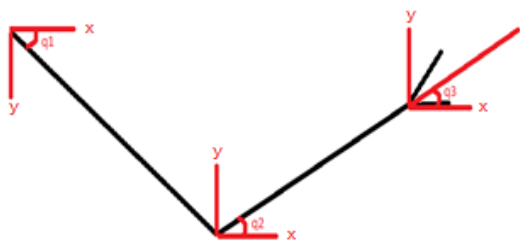


Figura 60. Ángulos con respecto a un eje fijo (eje horizontal).

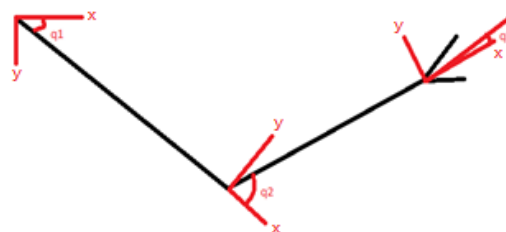


Figura 61. Ángulos con respecto a eje móvil.

Los ángulos q_2 y q_3 se calculan con respecto a un eje móvil, es decir que el eje coordenado de estas articulaciones varía en función del movimiento del eslabón anterior a la articulación. Esto se debe a que el ángulo de un eslabón no puede variar si se presenta movimiento en el eslabón anterior.

Para realizar el cálculo del parámetro angular q_1 se utilizan las coordenadas correspondientes a la posición del hombro y del codo obteniendo así una relación trigonométrica que permite calcular este parámetro (ver figura 62).

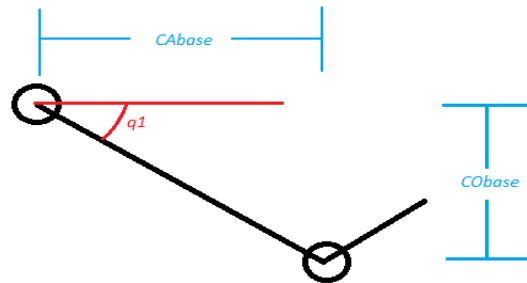


Figura 62. Cálculo del parámetro q_1 .

El ángulo q_1 se define entonces como:

$$q_1 = \arctan\left(\frac{CObase}{CAbase}\right) \quad (21)$$

Donde:

$$CObase = y_2 - y_1$$

$$CAbase = x_2 - x_1$$

(x_1, y_1) = coordenadas de la articulación del hombro.

(x_2, y_2) = coordenadas de la articulación del codo.

Para obtener el parámetro angular q_2 primero es necesario conocer el ángulo del antebrazo con respecto a la horizontal (d_1) para ello se utilizan las coordenadas

correspondientes al codo y a la muñeca encontrando así una relación trigonométrica que permite obtener este parámetro (figura 63).

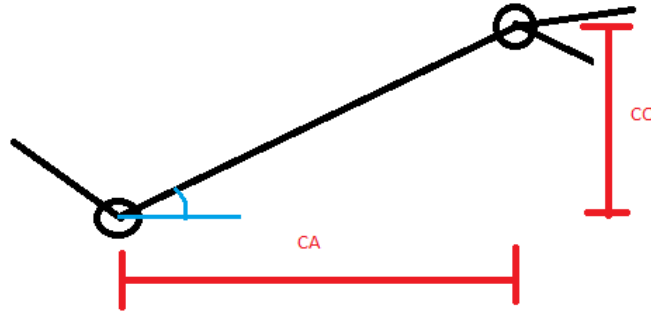


Figura 63. Cálculo del parámetro d1.

El parámetro d1 se define entonces como:

$$d1 = \arctan\left(\frac{CO}{CA}\right) \quad (22)$$

Donde:

$$CO = y2 - y3$$

$$CA = x3 - x2$$

(x2 , y2) = coordenadas de la articulación del codo.

(x3 , y3) = coordenadas de la articulación de la muñeca.

A partir del parámetro d1 es posible obtener el ángulo q2 en su respectivo eje coordenado (figura 64).

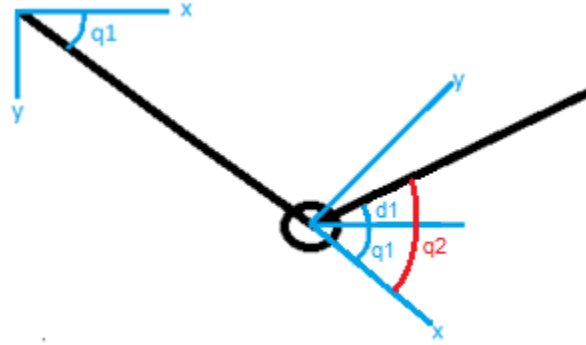


Figura 64. Cálculo del parámetro q_2 .

El ángulo q_2 se define entonces como:

$$q_2 = q_1 + d_1 \quad (23)$$

Cuando se calcula el parámetro q_3 es posible encontrar dos situaciones, cuando la pinza se encuentra abierta (figura 66) y cuando se encuentra cerrada (figura 65).

Si la pinza se encuentra cerrada el parámetro q_3 se calcula usando la ecuación 24.

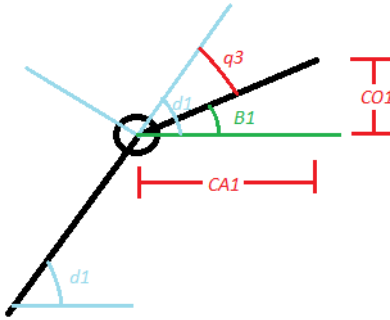


Figura 65. Cálculo del parámetro q_3 con pinza cerrada.

$$q_3 = (d_1 - B_1) \quad (24)$$

$$B_1 = \arctan\left(\frac{CO_1}{CA_1}\right) \quad (25)$$

Donde:

$$CO_1 = y_3 - y_4$$

$$CA1 = x4 - x3$$

($x3$, $y3$) = coordenadas de la articulación de la muñeca

($x4$, $y4$) = coordenadas de la posición final de los dedos

Si la pinza se encuentra abierta el parámetro $q3$ se calcula usando la ecuación 26.

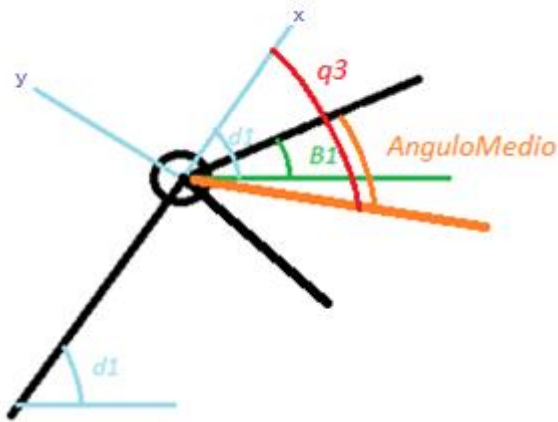


Figura 66. Ángulo medio.

$$q3 = (d1 - B1) + \text{AnguloMedio} \quad (26)$$

Donde:

$$\text{AnguloMedio} = \frac{\text{AnguloDeApertura}}{2} \quad (27)$$

El cálculo del AnguloDeApertura se presenta en la ecuación 28.

4.1.2.2. Cálculo del porcentaje de apertura y cierre de la pinza.

Para conocer el nivel de apertura de la pinza, se realiza un cálculo de porcentaje dependiendo el ángulo que exista entre el segmento superior y el segmento inferior de la pinza. Para realizar esta conversión de ángulo a porcentaje se debe conocer el ángulo que corresponde al 100 por ciento de apertura, es decir el valor equivalente al máximo de apertura

de la pinza de una persona común. Para conocer este valor se realiza una medición de diferentes ángulos de apertura máxima de diferentes personas encontrando así el promedio de apertura máxima de la muestra seleccionada.



Figura 67. Medición de diferentes manos.

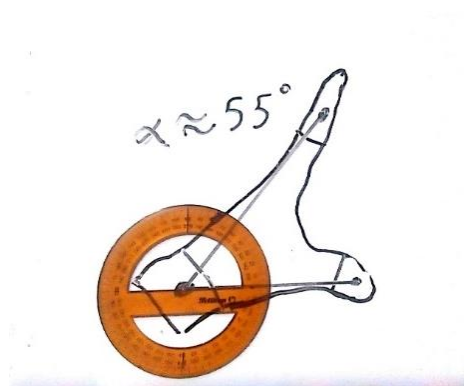


Figura 68. Mano con mayor ángulo de apertura.

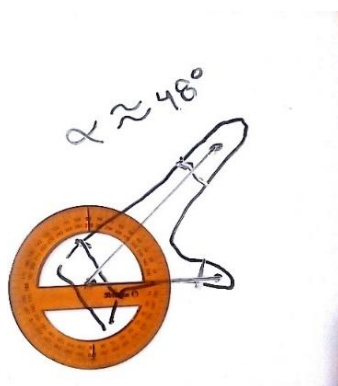


Figura 69. Mano con menor ángulo de apertura.

De 5 diferentes manos que se midieron, el mayor ángulo de apertura corresponde a una mano de un hombre joven de 180 cm de estatura con un ángulo de apertura máximo de 55 grados y el menor ángulo de apertura corresponde a la mano de una mujer de 40 años y una estatura de 162 cm con un ángulo de apertura máximo de 48 grados (figuras 67 a 69).

Finalmente, el promedio de datos da que el ángulo máximo de apertura es de 50 grados. De acuerdo con esta medición el ángulo de 50 grados es el equivalente al 100 por

ciento de apertura en el brazo robótico, sin embargo, aún es necesario calcular en tiempo real el ángulo de apertura de la mano detectada para su posterior conversión a porcentaje.

A partir de relaciones trigonométricas el ángulo de apertura se define como:

$$\text{AnguloDeApertura} = B_1 + B_2 \quad (28)$$

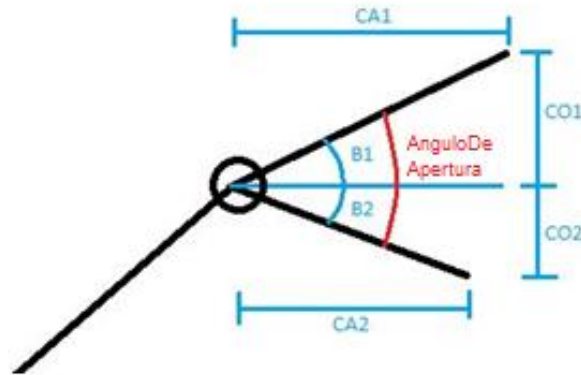


Figura 70. Cálculo del ángulo de apertura.

Donde:

$$B_2 = \arctan\left(\frac{CO2}{CA2}\right) \quad (29)$$

$$CO2 = y5 - y3$$

$$CA2 = x5 - x3$$

$(x3, y3)$ = coordenadas de la articulación de la muñeca.

$(x5, y5)$ = coordenadas de la posición final del dedo pulgar.

El ángulo B1 fue calculado anteriormente y está definido por la ecuación 25.

A partir del cálculo del ángulo de apertura se realiza la debida conversión a porcentaje teniendo en cuenta que el 100 por ciento de la apertura equivale a 50 grados. Todo este cálculo se realiza en tiempo real para cada fotograma.

Para el caso específico del porcentaje de apertura y cierre de la pinza del sistema de detección 2 (con guante bicromático), es fundamental calcular el porcentaje de apertura y cierre como

un valor absoluto tal y como se presenta en la ecuación 30 debido a que el cálculo puede dar negativo si se invierten las etiquetas 4 y 5 debido a movimientos en la muñeca (figura 71 y 72).

$$\text{Porcentaje de apertura} = \frac{|\text{Angulo de apertura}|}{50} * 100 \quad (30)$$

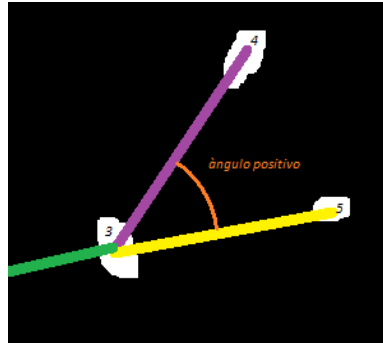


Figura 71. Etiquetas convencionales.

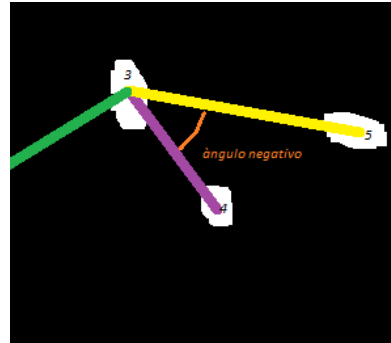


Figura 72. Etiquetas invertidas.

4.2. Etapa 2 – Diseño y construcción de brazo robótico

Para la emulación del movimiento de un brazo humano en un plano se diseña un robot manipulador de configuración tipo RRR y 4 grados de libertad permitiendo la realización de gran cantidad de movimientos en un único plano. El brazo robot es una cadena cinemática de 3 eslabones correspondientes al humero, antebrazo y muñeca, siendo el hombro la articulación que está sujeta a una base que sirve de apoyo para que el manipulador opere sin problemas.

4.2.1. Problema cinemático directo.

El problema cinemático directo tiene como objetivo encontrar las coordenadas X, Y del efector final a partir de los parámetros angulares q_1 , q_2 , y q_3 . La figura 73 muestra el diagrama del manipulador con los parámetros requeridos en la cinemática directa.

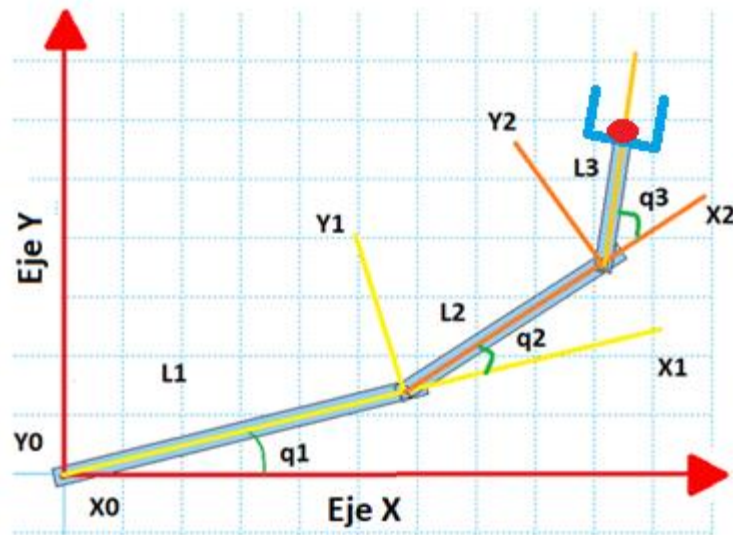


Figura 73. Diagrama del brazo.

Las ecuaciones 31 y 32 presentan la solución geométrica para conocer la posición de las coordenadas X y Y del efector final.

$$X = L1 \cos(\theta1) + L2 \cos(\theta1 + \theta2) + L3 \cos(\theta1 + \theta2 + \theta3) \quad (31)$$

$$Y = L1 \sin(\theta1) + L2 \sin(\theta1 + \theta2) + L3 \sin(\theta1 + \theta2 + \theta3) \quad (32)$$

Donde:

$$\theta1 = q1$$

$$\theta2 = q2$$

$$\theta3 = q3$$

Las ecuaciones 31 y 32 no proporcionan la información requerida para una solución completa de la cinemática directa del manipulador, por esta razón se emplea la convención propuesta por Hartenberg & Denavit, (1955) o parámetros DH, los cuales se presentan en la tabla 1 Y mediante la cual se obtienen matrices de transformación homogénea para conseguir la información de rotación y translación del manipulador.

Tabla 3. Parámetros Denavit-Hartenberg.

Articulación	Θ_i	d_i	a_i	A_i
1	Θ_1	0	L1	0
2	Θ_2	0	L2	0
4	Θ_3	0	L3	0

Las ecuaciones 33 a 35 representan las matrices de transformación homogénea de cada eslabón.

$$A_1^0 = \begin{bmatrix} \cos\theta_1 & -\text{sen}\theta_1 & 0 & l_1\cos\theta_1 \\ \text{sen}\theta_1 & \cos\theta_1 & 0 & l_1\text{sen}\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (33)$$

$$A_2^1 = \begin{bmatrix} \cos\theta_2 & -\text{sen}\theta_2 & 0 & l_2\cos\theta_2 \\ \text{sen}\theta_2 & \cos\theta_2 & 0 & l_2\text{sen}\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (34)$$

$$A_3^2 = \begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & l_3\cos\theta_3 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & l_3\text{sen}\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (35)$$

La ecuación 36 es el producto de las ecuaciones 33 a 35 y muestra la matriz de rotación y el vector de posición del efector final con respecto al sistema de referencia base.

$$A_3^0 = \begin{bmatrix} \cos(a) & -\text{sen}(a) & 0 & b + c + d \\ \text{sen}(a) & \cos(a) & 0 & e + f + g \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (36)$$

Donde:

$$a = \theta_1 + \theta_2 + \theta_3 \quad (37)$$

$$b = l_1 * \cos\theta_1 \quad (38)$$

$$c = l_2 * \cos(\theta_1 + \theta_2) \quad (39)$$

$$d = l_3 * \cos(\theta_1 + \theta_2 + \theta_3) \quad (40)$$

$$e = l_1 * \sin\theta_1 \quad (41)$$

$$f = l_2 * \sin(\theta_1 + \theta_2) \quad (42)$$

$$g = l_3 * \sin(\theta_1 + \theta_2 + \theta_3) \quad (43)$$

4.2.2. Problema cinemático inverso – Método geométrico.

El PCI o problema cinemático inverso tiene como objetivo conocer los parámetros angulares q_1 , q_2 y q_3 a partir de unas coordenadas X , Y definidas en el efector final, esto con el fin de poder simular los diferentes movimientos realizados por el brazo humano hallando así los resultados de un análisis dinámico posterior donde se encuentran cada uno de los torques requeridos en cada una de las articulaciones. Sin embargo, encontrar una configuración de ángulos que satisfagan la posición X , Y deseada no será posible debido a que toda cadena cinemática de más de dos eslabones tiene más de dos posibles configuraciones de ángulos para llegar a la posición deseada, en consecuencia, todo intento de resolver el sistema de ecuaciones que genera este problema cinemático será matemáticamente indeterminado. Para resolver esta indeterminación es necesario elegir una de las posibles configuraciones de ángulos. La opción seleccionada es dejar estático el último eslabón de la cadena cinemática, es decir la muñeca, el procedimiento a seguir es definir las coordenadas de la muñeca a partir de las coordenadas del efector final siendo X y Y las coordenadas del efector final y X_m y Y_m las coordenadas de la muñeca (figura 74).

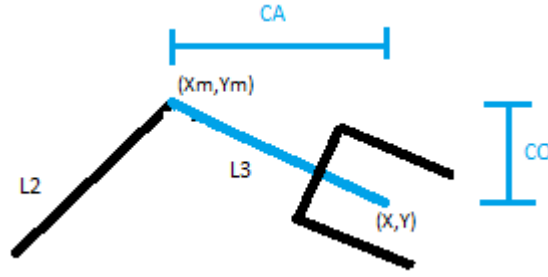


Figura 74. Ejemplo de posición para el actuador final.

A continuación, se elige una de las muchas posibilidades que existen para llegar a la coordenada definida en el efector final definiendo una posición en el eslabón **L3**.

$$CA = \sqrt{L3^2 + CO^2}$$

$$Ym = Y + CO$$

$$Xm = X - CA$$

Donde:

Y = Coordenadas en el eje Y del efector final

X = Coordenadas en el eje X del efector final

CO = Distancia vertical entre la coordenada Y y Ym

Al reducir el problema de esta forma es posible obtener las ecuaciones q_1 , q_2 , y q_3 a partir relaciones trigonométricas obteniendo como resultado las ecuaciones 44 a la 46.

$$q_2 = \cos^{-1}\left(\frac{Xm^2 + Ym^2 - l_1^2 - l_2^2}{2 * l_1 * l_2}\right) \quad (44)$$

$$q_1 = \tan^{-1}\left(\frac{Ym}{Xm}\right) - \tan^{-1}\left(\frac{l_2 * \sin(q_2)}{l_1 + l_2 * \cos(q_2)}\right) \quad (45)$$

$$q_3 = \cos^{-1}\left(\frac{X - Xm}{l_3}\right) - q_1 - q_2 \quad (46)$$

4.2.3. Modelo diferencial.

A partir de las posiciones X, Y del efector final definidas en las ecuaciones 18 y 19 es posible conocer las velocidades y aceleraciones angulares de cada articulación derivando la posición con ayuda de la matriz jacobiana (ecuación 47), donde los parámetros de la **a** hasta la **g** están definidos en las ecuaciones 37 a 43.

Jacobiano Analítico:

$$J = \begin{bmatrix} \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \frac{\partial X}{\partial q_3} \\ \frac{\partial Y}{\partial q_1} & \frac{\partial Y}{\partial q_2} & \frac{\partial Y}{\partial q_3} \\ \frac{\partial Z}{\partial q_1} & \frac{\partial Z}{\partial q_2} & \frac{\partial Z}{\partial q_3} \end{bmatrix} = \begin{bmatrix} -e - f - g & -f - g & -g \\ b + c + d & c + d & d \\ 0 & 0 & 0 \end{bmatrix} \quad (47)$$

4.2.3.1. Velocidades angulares.

Se define el vector de velocidades lineales $\dot{A}(\dot{x}, \dot{y}, \dot{z})$ del efector final para conocer las velocidades angulares $\dot{q}(q_1, q_2, q_3)$ de cada articulación como se presenta en la ecuación 39.

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} * J^{-1} \quad (48)$$

Al implementar la ecuación 48 se presenta una indeterminación debido a que el jacobiano presenta una fila de ceros, esto se debe a que físicamente el robot puede tener diferentes configuraciones de velocidades angulares para ejecutar una trayectoria en el efector final dependiendo del tipo de movimiento que realice el robot para llegar a las

coordenadas deseadas. Debido a esto es necesario alterar el jacobiano ubicando un uno (1) en la fila de ceros como se muestra en la ecuación 49.

$$J = \begin{bmatrix} -e - f - g & -f - g & -g \\ b + c + d & c + d & d \\ 0 & 0 & 1 \end{bmatrix} \quad (49)$$

Esto permite definir la velocidad angular q3 es decir, una velocidad angular para la muñeca con el fin de encontrar las velocidades angulares q2 y q1 y así encontrar una de las muchas posibles combinaciones de velocidades angulares en los eslabones para ejecutar el movimiento en el efector final a las velocidades lineales en X y Y que se desean.

4.2.3.2. Aceleraciones angulares.

Si se toma los vectores de posición X y Y obtenidos en la cinemática directa y se deriva dos veces se obtienen los vectores \ddot{X} y \ddot{Y} . Al agrupar términos de forma matricial se obtiene la expresión definida por la ecuación 50.

$$\begin{aligned} \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} &= \begin{bmatrix} -e - f - g & -f - g & -g \\ b + c + d & c + d & d \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} \\ &+ \begin{bmatrix} -b - c - d & -c - d & -d \\ -e - f - g & -f - g & -g \\ 0 & 0 & 0 \end{bmatrix} * \left(\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \right)^2 \\ &+ \begin{bmatrix} -2 * (c - d) & -2 * d & -2 * d \\ -2 * f - g & -2 * g & -2 * g \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \dot{q}_2 \\ \dot{q}_1 \dot{q}_3 \\ \dot{q}_2 \dot{q}_3 \end{bmatrix} \end{aligned} \quad (50)$$

Los vectores \ddot{X} , \ddot{Y} y \ddot{Z} , presentados en la ecuación 41 poseen la estructura definida por la ecuación 51.

$$\ddot{A} = J * \ddot{q} + \frac{dJ}{dq} * \dot{q}^2 + D * \dot{q}_A * \dot{q}_B \quad (51)$$

Al despejar el vector de aceleraciones angulares \ddot{q} se obtiene la ecuación 52.

$$\ddot{q} = \ddot{A} * J^{-1} - \frac{dJ}{dq} * \dot{q}^2 * J^{-1} - D * \dot{q}_A * \dot{q}_B * J^{-1} \quad (52)$$

La ecuación 52 depende de las velocidades angulares \dot{q} ya obtenidas previamente y del vector de aceleraciones lineales del efector final \ddot{A} que se obtiene a partir de la derivada del vector de velocidades lineales \dot{A} . En tal caso de que el vector de velocidades lineales \dot{A} sea definido con valores constantes el vector de aceleraciones lineales \ddot{A} será cero.

4.2.4. Modelo dinámico según Euler Lagrange.

A partir de los resultados cinemáticos del modelo diferencial se busca obtener los torques requeridos para cada articulación desarrollando la dinámica inversa según el método de Euler Lagrange basados en (Cortés Reyes, 2011, p.255-257) y (Ulloa Calzonzin, 2017, p. 1 y 2).

Modelo General según Euler Lagrange:

$$\frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = T \quad (53)$$

Donde:

L = Lagrangiano (diferencia entre la energía cinética y potencial)

T = Torque requerido en las articulaciones

Modelo Matricial de un robot manipulador:

$$M(q)\ddot{q} + c(q, \dot{q})\dot{q} + g(q) = T \quad (54)$$

Donde:

$M(q)$ = Matriz de inercia del robot manipulador.

$c(q, \ddot{q})$ = Matriz de Coriolis y fuerzas centrífugas.

$g(q)$ = Matriz de pesos y fuerzas debido a la gravedad.

La matriz de inercia del robot manipulador se obtiene del modelo general de Euler Lagrange a partir de la agrupación de términos y de los modelos de energías en términos matriciales dando como resultado la ecuación 55.

$$M(q) = \sum_{i=1}^n m_i * J_{vi}^T * J_{vi} + J_{wi}^T * J_{wi} * {}^0R_i^T * {}^0R_i * I_i \quad (55)$$

Donde:

i=eslabón

m=masa del eslabón

J_{vi} =Jacobiano velocidad lineal del centro de masa del eslabón

J_{wi} =Jacobiano velocidad angular del eslabón

R_i =matriz de rotación con respecto al sistema base

I =tensor de inercia del eslabón en su propio sistema de referencia

Para calcular el jacobiano J_{vi} de cada eslabón se relaciona las velocidades angulares con las velocidades lineales del centro de masa del eslabón realizando el cálculo con respecto al sistema de referencia base (ecuación 56 y 57)

$$V_{ic} = J_v^i * \dot{q} \quad (56)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J_v^i * \begin{bmatrix} \dot{q}1 \\ \dot{q}2 \\ \dot{q}3 \end{bmatrix} \quad (57)$$

El jacobiano J_w^i relaciona las velocidades angulares de cada eslabón con las velocidades angulares de los ejes X, Y y Z del efector final con respecto al sistema de referencia base (ecuación 58 y 59).

$$W_i = J_w^i * \dot{q} \quad (58)$$

$$\begin{bmatrix} W_X \\ W_Y \\ W_Z \end{bmatrix} = J_w^i * \begin{bmatrix} \dot{q}1 \\ \dot{q}2 \\ \dot{q}3 \end{bmatrix} \quad (59)$$

Los jacobianos de velocidad lineal J_v^i y velocidad angular J_w^i de cada eslabón con respecto al sistema de referencia base están definidos por las ecuaciones 60 a 65.

$$J_v^1 = \begin{bmatrix} -\frac{e}{2} & 0 & 0 \\ \frac{b}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (60)$$

$$J_w^1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (61)$$

$$J_v^2 = \begin{bmatrix} -e - \frac{f}{2} & -\frac{f}{2} & 0 \\ b + \frac{c}{2} & \frac{c}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (62)$$

$$J_w^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad (63)$$

$$J_v^3 = \begin{bmatrix} -e - f - \frac{g}{2} & f - \frac{g}{2} & -\frac{g}{2} \\ b + c + \frac{d}{2} & c + \frac{d}{2} & \frac{d}{2} \\ 0 & 0 & 0 \end{bmatrix} \quad (64)$$

$$J_w^3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (65)$$

Las matrices de rotación de cada eslabón con respecto al sistema de referencia base están definidas por las ecuaciones 66 a 68.

$$R_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (66)$$

$$R_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & \mathbf{0} \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \quad (67)$$

$$R_3 = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & \mathbf{0} \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \quad (68)$$

Los tensores de inercia de cada eslabón fueron considerados como inercias para una varilla rígida; Siendo este un sólido simétrico los tensores de inercia están definidos por las ecuaciones 69 a 71.

$$I_1 = \begin{bmatrix} 0 & 0 & \mathbf{0} \\ 0 & \frac{m_1 * l_1^2}{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{m_1 * l_1^2}{12} \end{bmatrix} \quad (69)$$

$$I_2 = \begin{bmatrix} 0 & 0 & \mathbf{0} \\ 0 & \frac{m_2 * l_2^2}{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{m_2 * l_2^2}{12} \end{bmatrix} \quad (70)$$

$$I_3 = \begin{bmatrix} 0 & 0 & \mathbf{0} \\ 0 & \frac{m_3 * l_3^2}{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{m_3 * l_3^2}{12} \end{bmatrix} \quad (71)$$

La matriz de inercia del robot manipulador se define entonces como la sumatoria de las 3 matrices de inercia correspondiente a cada eslabón (ecuación 72 y 73).

$$M(q) = M(q)1 + M(q)2 + M(q)3 \quad (72)$$

$$M(q) = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (73)$$

La matriz de Coriolis y fuerzas centrífugas (ecuación 74) se calcula con ayuda de los símbolos de Christoffel (ecuación 75 y 76)

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (74)$$

Donde los Símbolos de Christoffel son calculados mediante la ecuación 75 y 76:

$$C_{ij} = \sum_{k=1}^n C_{ijk} * \dot{q}_k \quad (75)$$

$$C_{ijk} = \frac{1}{2} * \left(\frac{\partial h_{ij}}{\partial q_k} + \frac{\partial h_{jk}}{\partial q_j} - \frac{\partial h_{ki}}{\partial q_i} \right) \quad (76)$$

Si remplazamos C_{ijk} en la ecuación de C_{ij} obtenemos la ecuación 77.

$$C_{ij} = \sum_{k=1}^n \frac{1}{2} * \left(\frac{\partial h_{ij}}{\partial q_k} + \frac{\partial h_{jk}}{\partial q_j} - \frac{\partial h_{ki}}{\partial q_i} \right) * \dot{q}_k \quad (77)$$

Donde:

k = número de eslabón

A partir de la derivada parcial de los términos de la matriz de inercia $M(q)$ es posible obtener la matriz de Coriolis y fuerzas centrífugas $C(q, \dot{q})$ donde cada termino C_{ij} corresponde a la sumatoria del efecto Coriolis de cada eslabón (ecuación 78).

$$C_{ij} = C_{ij1} + C_{ij2} + C_{ij3} \quad (78)$$

Al ser la matriz de Coriolis 3x3 se debe calcular un total de 27 términos debido a que la matriz tiene 9 términos C_{ij} pero cada término C_{ij} lo componen 3 términos más.

Por último, se calcula el vector de pesos debido a las fuerzas de gravedad, cuya expresión matricial es de tamaño 3x1 y está dada por la ecuación 79.

$$g(q) = - \sum_{i=1}^n J_{vi}^T * m_i * g_0 \quad (79)$$

Donde:

g_0 = vector de aceleración gravitacional

m_i = masa de cada eslabón

J_{vi}^T = jacobiano traspuesto de velocidad lineal del centro de masa de cada eslabón

$$g_0 = [0 \quad -9.81 \quad 0] \quad (80)$$

El vector $g(q)$ es entonces la sumatoria negativa de los pesos de cada eslabón (ecuación 81).

$$g(q) = -g(q)1 - g(q)2 - g(q)3 \quad (81)$$

Con estas matrices ya calculadas es posible obtener los valores de torque requeridos en las articulaciones y aún más obtener la potencia de cada una:

$$\text{Potencia máxima} = \text{Torque máximo} * \text{Velocidad angular máxima} \quad (82)$$

Para obtener los resultados dinámicos a partir de estas formulaciones planteadas, se calculan cada una de las matrices según la posición angular de cada segmento del brazo robot es decir que para un movimiento determinado que realice el brazo robot se debe obtener múltiples resultados que corresponden a cada una de las posiciones en las que estuvo el brazo robótico en el momento de ejecutar su trayectoria, es por eso que este proceso de evaluación constante de matrices se realiza de forma computacional.

Para ejemplificar el procedimiento lo primero que se realiza es definir una trayectoria en el efector final con el fin de calcular la cinemática inversa del brazo robot y así obtener los ángulos q_1 , q_2 , y q_3 que van a ir variando a medida que el brazo robot se vaya desplazando, sin embargo, para el caso del ejemplo se realizara el procedimiento a partir de una posición estática del brazo robot, para ello se definen arbitrariamente los ángulos q_1 , q_2 y q_3 de la siguiente forma:

$$q1 = -38 \text{ grados}$$

$$q2 = 110 \text{ grados}$$

$$q3 = -50 \text{ grados}$$

Estos ángulos corresponderían a la posición del brazo robot que se ilustra en la figura 75.

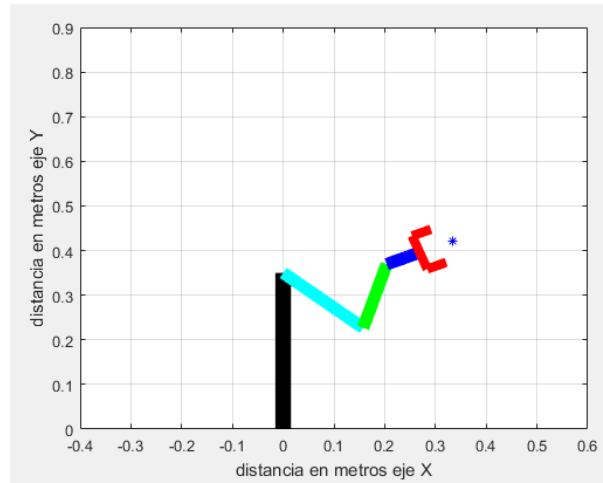


Figura 75. Posición evaluada.

A partir de estos ángulos se busca obtener las velocidades angulares de cada articulación, para ello se evalúan los ángulos $q1$, $q2$, y $q3$ en la ecuación 48 y adicionalmente se define la velocidad lineal del efector final teniendo en cuenta la aplicación para la que se requiera el brazo robot, para el caso del ejemplo se define una velocidad lineal constante tanto en el eje X como en el eje Y de 10 cm/s que es igual a 0.1 m/s. la ecuación 48 quedaría entonces como:

$$\begin{bmatrix} \dot{q1} \\ \dot{q2} \\ \dot{q3} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0 \end{bmatrix} * \begin{bmatrix} -0.0720 & -0.1951 & -0.0524 \\ 0.3338 & 0.1762 & 0.1298 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

$$\begin{bmatrix} \dot{q1} \\ \dot{q2} \\ \dot{q3} \end{bmatrix} = \begin{bmatrix} 0.7080 \\ -0.7737 \\ 0 \end{bmatrix}$$

Una vez obtenidas las velocidades angulares se desean obtener las aceleraciones angulares reemplazando la ecuación 52 para ello es necesario definir la aceleración lineal del

efector final que se obtiene a partir de la derivada de la velocidad lineal, sin embargo, como definimos una velocidad lineal constante su derivada es cero, la ecuación 52 quedaría entonces como:

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} * J^{-1} - \begin{bmatrix} -0.3338 & -0.1762 & -0.1298 \\ -0.0720 & -0.1951 & -0.0524 \\ 0 & 0 & 0 \end{bmatrix} * \dot{q}^2 * J^{-1} \\ - \begin{bmatrix} -0.3523 & -0.2596 & -0.2596 \\ -0.3902 & -0.1049 & -0.1049 \\ 0 & 0 & 0 \end{bmatrix} * \dot{q}_A * \dot{q}_B * J^{-1}$$

Donde los parámetros faltantes se obtienen de la siguiente forma:

$$\begin{bmatrix} \dot{q}_1 \dot{q}_2 \\ \dot{q}_1 \dot{q}_3 \\ \dot{q}_2 \dot{q}_3 \end{bmatrix} = \begin{bmatrix} -0.5478 \\ 0 \\ 0 \end{bmatrix}$$

$$\left(\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \right)^2 = \begin{bmatrix} 0.5012 \\ 0.5986 \\ 0 \end{bmatrix}$$

$$J^{-1} = \begin{bmatrix} -0.0720 & -0.1951 & -0.0524 \\ 0.3338 & 0.1762 & 0.1298 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \\ = \begin{bmatrix} 3.3593 & 3.7205 & -0.3068 \\ -6.3647 & -1.3725 & -0.1556 \\ 0 & 0 & 1 \end{bmatrix}$$

La aceleración angular de cada segmento del brazo sería entonces:

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} 0.0414 \\ -0.4241 \\ 0 \end{bmatrix}$$

Una vez obtenidas las velocidades y aceleraciones angulares se procede a evaluar la ecuación 54 correspondiente al modelo matricial de un robot manipulador según Euler Lagrange, para ello se debe obtener la matriz de inercia, la matriz de Coriolis y la matriz de pesos que se obtienen evaluando las ecuaciones 55 a 81 a partir de los valores ya obtenidos de posición angular, velocidad angular y aceleración angular.

$$\begin{bmatrix} 0.0120 & 0.0042 & 0.0020 \\ 0.0038 & 0.0050 & 0.0013 \\ 0.0019 & 0.0013 & 0.0007 \end{bmatrix} \ddot{q} + c(q, \ddot{q}) \ddot{q} + \begin{bmatrix} 0.5184 \\ 0.1319 \\ 0.0637 \end{bmatrix} = T$$

Para el caso específico de la matriz de Coriolis es importante aclarar que el cálculo de esta matriz depende de los valores de la matriz de inercia de la posición anterior a la actual debido a que se involucra la derivada de los parámetros de la matriz de inercia con respecto a los ángulos q_1 , q_2 y q_3 tal y como se muestra en la ecuación 77. Para realizar el cálculo de estas derivadas de forma computacional se obtiene la pendiente de la recta que existe entre el dato actual (parámetro de la matriz de inercia con respecto a la posición angular) y el dato anterior.

Para el caso del ejemplo supondremos una posición anterior a la actual para obtener una matriz de inercia anterior y a partir de estos datos poder calcular las derivadas requeridas para obtener la matriz de Coriolis.

$$\text{Posición anterior} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} -37 \\ 115 \\ -52 \end{bmatrix}$$

$$\text{Matriz de inercia de la posición anterior} = \begin{bmatrix} 0.0111 & 0.0037 & 0.0019 \\ 0.0033 & 0.0049 & 0.0013 \\ 0.0018 & 0.0013 & 0.0007 \end{bmatrix}$$

Las derivadas se calculan computacionalmente como se muestra en la ecuación 83.

$$\frac{\partial h_{ij}}{\partial q_k} = \frac{h_{ij} - h_{ij}^{\text{anterior}}}{q_k - q_k^{\text{anterior}}} \quad (83)$$

Una vez obtenidas las derivadas, se rempazan los valores en la ecuación 77 para obtener los parámetros de la matriz de Coriolis y fuerzas centrífugas.

$$c(q, \ddot{q}) = \begin{bmatrix} -0.2418 & -0.0300 & 0.0639 \\ -0.2268 & -0.0285 & 0.0613 \\ -0.1349 & -0.0769 & 0.0129 \end{bmatrix} * 10^{-3}$$

Finalmente, los torques de cada articulación en kg-cm para la posición actual del brazo robot son los siguientes:

$$T = \begin{bmatrix} 5.2699 \\ 1.3229 \\ 0.6437 \end{bmatrix}$$

Cabe aclarar que este ejemplo se desarrolló para una posición determinada, sin embargo para un análisis dinámico del brazo robot estos cálculos se realizan múltiples veces de acuerdo a la trayectoria definida para el análisis, obteniendo así un barrido de valores de torques en función de la posición del brazo robot, es así como se pueden conocer los torques máximos en una trayectoria determinada.

4.2.5. Diseño CAD y construcción del brazo robótico.

El diseño de cada una de las partes comprendidas por el brazo robótico corresponde a aproximadamente el 70% del total del brazo humano real, esto con el fin de que el movimiento sea notorio en cuanto al seguimiento, las medidas del brazo robótico se presentan en la tabla 2.

Tabla 4. Medidas.

Medidas generales (cm)		Distancia entre ejes (cm)	
Sección	Medida	Sección	Medida
Hombro	0	Hombro	2.5
Codo	16	Codo	20
Muñeca	29	Muñeca	32.5
Pinza	38	Pinza	40

A su vez es muy necesario tener en cuenta el material en que se realizan dichos prototipos, debido a que algunos materiales idealmente resistentes son demasiado pesados lo que conlleva a requerir más torque para su funcionamiento, es por este motivo que se decide usar el acrílico como material principal debido a su bajo peso, bajo costo, buena resistencia

y buena resiliencia. Por otra parte, se emplea acero para los tornillos y uniones entre láminas de acrílico.

Los eslabones se diseñaron con una ligera curvatura cóncava con el fin de reducir el peso de cada uno de estos. Tal como se muestra en la figura 76, estos eslabones fueron diseñados en el software Inventor y cortados en maquina laser CNC.

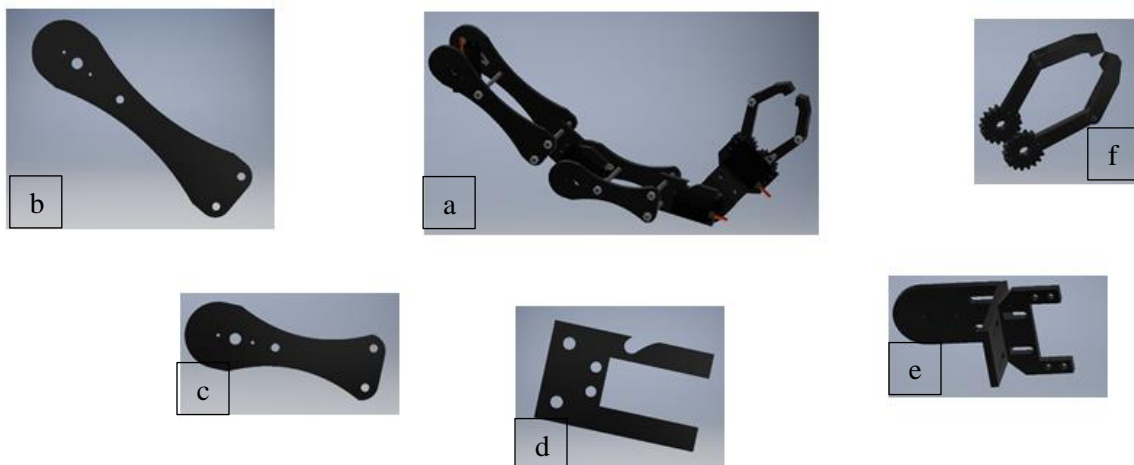


Figura 76. Diseño CAD. a. Ensamble, b. Húmero, c. Radio, d. Articulación, e. Muñeca, f. Pinza.

La tabla 3 presenta los pesos obtenidos de cada eslabón luego de su diseño, construcción y posterior ensamble individual de cada uno.

Tabla 5. Pesos eslabones

Eslabón	Peso (gramos)
Húmero	114
Antebrazo	108
Muñeca	123

Estos datos de pesos obtenidos a partir de pruebas realizadas permiten posteriormente obtener los resultados del análisis dinámico.

4.3. Etapa 3 - Diseño electrónico y comunicación entre visión artificial y brazo robótico

4.3.1. Configuración de los servomotores en el brazo robótico.

El resultado de los sistemas de visión artificial son los ángulos q_1 , q_2 , q_3 y el porcentaje de apertura de la pinza, pero estos ángulos a excepción del porcentaje de apertura deben sufrir una pequeña conversión ya que son ángulos teóricos que están medidos con respecto a un eje de referencia móvil como se explicó anteriormente en los sistemas de visión artificial. Sin embargo, estos ángulos a pesar de que deben variar con respecto a un sistema de referencia móvil también deben adaptarse a la posición en la que se encuentre el servomotor en el brazo robot, esto debido a que los servomotores tienen un rango de movimiento de 180 grados y están ubicados en el brazo robótico de tal forma que éste sea capaz de ejecutar la mayor cantidad de movimientos posibles.

Para realizar esta conversión entre lo que denominamos ángulos teóricos y ángulos reales que serán enviados a los servomotores, primero es necesario definir la posición de cada servomotor dentro del brazo robot, la configuración de los servomotores diseñada para el brazo robot se presenta en la figura 77.

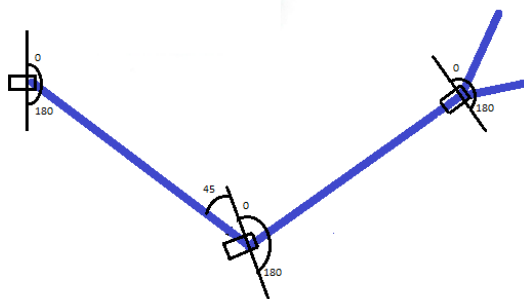


Figura 77. Configuración de los servomotores en el brazo robótico.

Esta configuración permite que el brazo robótico pueda emular gran cantidad de movimientos que haría un brazo humano real. Una vez definidas las posiciones de los servomotores en el brazo robot se realiza la conversión de ángulo teórico a ángulo real, para ello es importante recordar que los ángulos teóricos se calculan de acuerdo con relaciones trigonométricas entre las coordenadas de las articulaciones del brazo detectado, estos ángulos están definidos con respecto al eslabón anterior tal y como se explicó en la sección de cálculo de parámetros angulares. Los ángulos teóricos se presentan en la figura 78.

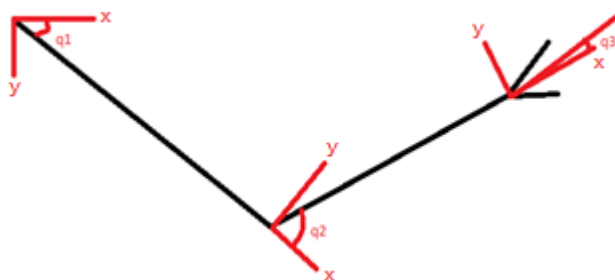


Figura 78. Ángulos teóricos.

Los ángulos reales que deben ser enviados a los servomotores se calculan a partir de los ángulos teóricos q_1 , q_2 y q_3 . En la figura 79 se presentan, en color rojo, los ángulos reales.

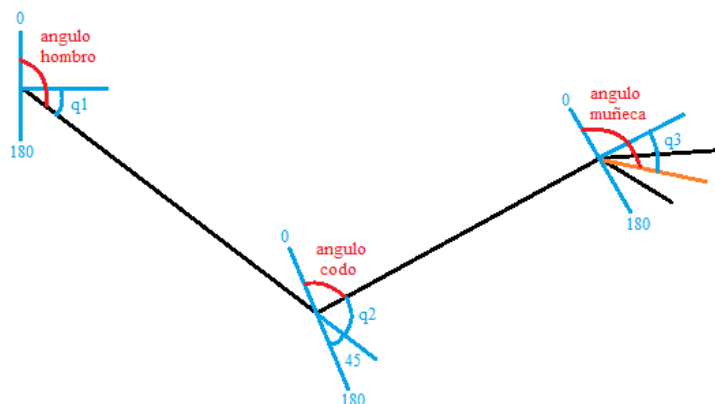


Figura 79. Conversión a ángulos reales.

La conversión de ángulos teóricos a reales se presenta a continuación:

- Angulo hombro = $90 - q_1$
- Angulo codo = $180 - 45 - q_2$
- Angulo muñeca = $90 - q_3$

En el caso del servomotor que controla la apertura y cierre de la pinza el ángulo calculado a partir de la detección del brazo humano no es el mismo en la pinza del brazo robot debido a la arquitectura de la pinza mecánica. Experimentalmente se observa que el 0% de apertura en la pinza del brazo robot equivale a 14 grados y el 100% de apertura equivale a 50 grados, de esta forma se realiza una conversión para este servomotor definida en la ecuación 84.

$$\text{Angulo Real de pinza} = 14 + (\text{PORCENTAJEAPERTURA} * 36)/100 \quad (84)$$

Donde:

PORCENTAJEAPERTURA = valor obtenido en la sección Cálculo del porcentaje de apertura y cierre de la pinza.

4.3.2. Circuito de control del brazo robótico.

Las etapas descritas anteriormente generan los datos correspondientes a los ángulos reales de cada articulación del brazo robótico, estos datos son enviados desde el PC hacia un microcontrolador que recibe y clasifica cada dato y además genera la señal de control de cada servomotor de acuerdo con el dato recibido. El esquema del circuito se presenta en la figura

80.

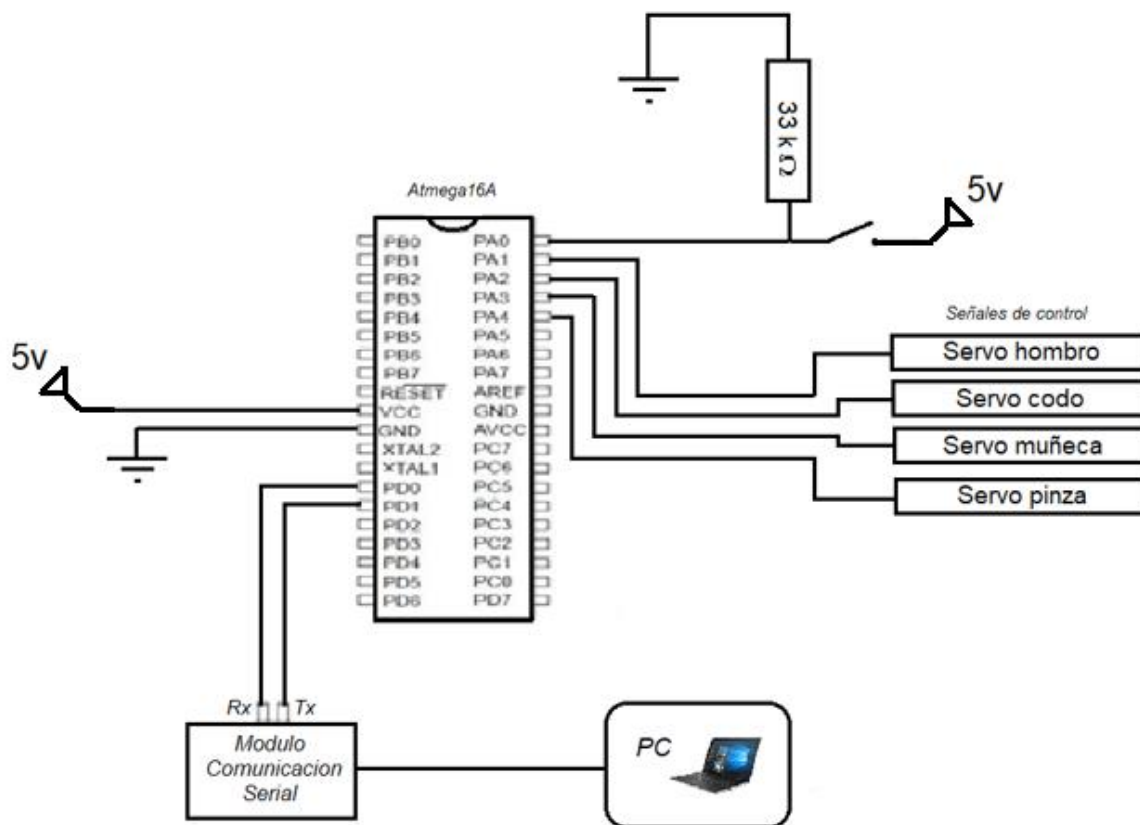


Figura 80. Circuito de control.

El microcontrolador empleado es el atmega16A, que tiene una arquitectura de 8 bits, 32 pines I/O disponibles, memoria de programa flash de 16 kB, SRAM de 1 kB, EEPROM de datos de 512 Bytes, ADC de 10 bits y 8 canales, 3 timers/comparadores/contadores, 4 canales PWM, USART, SPI, TWI y un comparador analógico.

Como medida de seguridad se tiene un interruptor que corta las comunicaciones entre el PC y el microcontrolador dejando el último dato enviado activo en el brazo robot, esto con el fin de realizar un apagado del brazo robótico sin des-energizar los servomotores evitando así una caída violenta del brazo robot.

El microcontrolador es operado a 5 voltios, mientras que cada servomotor es operado a 6 voltios y a un periodo de 20 ms en la señal de control.

El microcontrolador atmega16A tiene una arquitectura de 8 bits sin embargo el timer 1 del microcontrolador posee 16 bits de resolución. Gracias a este timer se puede obtener una variación del ciclo útil de 0 a 65536 unidades, a pesar de tener este rango de valores tan amplio se define el valor máximo del timer como 20000 ya que se puede obtener suficiente precisión en este intervalo de valores, obteniendo también cierta relación con el periodo de 20 ms de la señal PWM que se debe ingresar a los servomotores (figura 81).

Para controlar los ángulos de un servomotor de 0 a 180 grados se debe variar el pulso de la señal de control entre 0.5 y 2.5 ms. El rango de 0 a 20000 unidades de ciclo útil permite implementar un máximo de 2000 unidades para controlar los 180 grados en el intervalo de 2 ms que exige el servomotor en la señal de control, estas 2000 unidades son suficientes para tener una buena resolución de control obteniendo así un funcionamiento del brazo robot con mayor precisión.

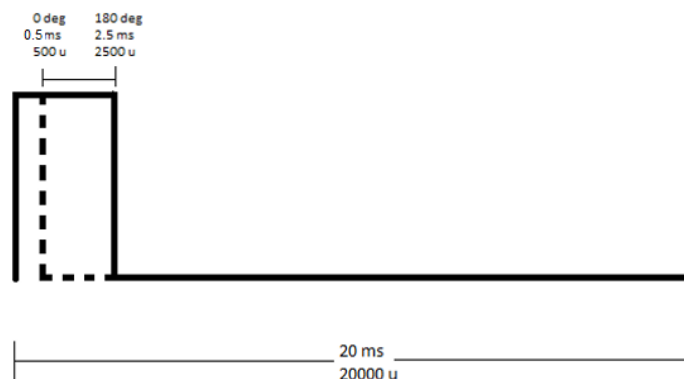


Figura 81. Señal de control PWM.

4.3.3. Comunicación serial entre pc y microcontrolador.

Para realizar una comunicación de datos es necesario tener un dispositivo emisor y un dispositivo receptor, en este caso el software Matlab® es el emisor de los datos ya que es allí donde se implementa el sistema de visión artificial, siendo el microcontrolador el receptor

de los datos. Debido a que la comunicación desde el PC se hace a través de un puerto USB se requiere el uso del módulo de comunicación serial FT232 (ver figura 82).



Figura 82. Módulo FT232.

La comunicación serial se realiza de tipo asíncrona utilizando una velocidad de transmisión de 5000 baudios lo que actualmente se puede considerar una velocidad de transmisión lenta, sin embargo, es suficiente para el control del brazo robótico. La comunicación está configurada para tener un frame de información con un bit de inicio, un bit de parada, y 8 bits de datos entregando así la información desde un puerto COM del PC hacia el pin Rx del módulo UART que posee el microcontrolador. Exceptuando la configuración de la comunicación, el resto es posible gracias al buffer incorporado en el módulo FT232 que permite el acoplamiento en términos de impedancias entre el PC y el microcontrolador.

4.3.4. Clasificación de datos recibidos por el microcontrolador.

Una vez recibidos los datos por el microcontrolador el circuito de control debe ser capaz de enviar a cada servomotor su correspondiente ángulo, para ello se envía desde Matlab® el ángulo precedido de un dato de sincronización que le permite al microcontrolador saber hacia que servomotor debe enviar el ángulo (figura 83). Los datos de sincronización seleccionados son 201, 202, 203, y 204. Si el valor de sincronización es 201 el microcontrolador sabe que el dato siguiente a este valor es un ángulo para el servomotor 1

correspondiente al hombro, si el dato de sincronización es 202 el microcontrolador sabe que el siguiente dato será un ángulo para el servomotor 2, correspondiente al codo, el valor 203 el corresponde a la muñeca y el valor 204 el corresponde a la pinza. La razón por la cual se seleccionan estos valores de sincronización es debido a que no corresponden a ningún valor entre 0 y 180 y tampoco son mayores a 255 que es el valor máximo para un dato de 8 bits de resolución. Esto evita confusiones entre el dato de sincronización y los ángulos recibidos por el microcontrolador, sin embargo, estos datos de sincronización pueden tomar cualquier valor entre 180 y 255.

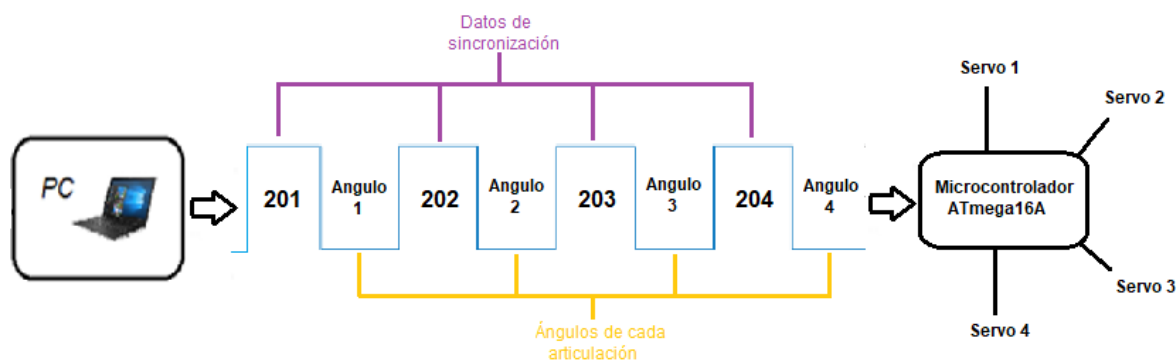


Figura 83. Clasificación de ángulos.

4.3.5. Control de servomotores.

Para controlar los servomotores se deben generar 4 señales PWM cuyo ciclo útil será proporcional a los valores de los cuatro ángulos. Generalmente se utilizan los módulos PWM del microcontrolador en los cuales se configura el timer de cada PWM y se define el ciclo útil de cada uno. El microcontrolador se encarga de sincronizar el ancho de pulso con un valor del timer automáticamente, sin embargo, utilizar los módulos PWM del microcontrolador no es suficiente para controlar el brazo robótico debido a que el

microcontrolador posee 3 módulos PWM, y son necesarias 4 señales de control PWM independientes.

La falta de señales de control PWM es un problema usual en el campo de la robótica ya que por lo general se necesita gran cantidad de salidas PWM para controlar un robot manipulador. La alternativa en estos casos es generar manualmente las señales de control requeridas, evitando el uso de los módulos PWM predeterminados en el microcontrolador. Estas señales de control pueden generarse a partir de diferentes metodologías.

El control de una señal PWM puede realizarse utilizando un timer del microcontrolador con suficiente resolución para poder modificar el ancho de pulso con precisión. Generalmente el procedimiento para generar la señal de control a partir de un timer del microcontrolador consiste en poner en HIGH la cantidad de salidas digitales que se requieran al mismo tiempo, y mientras permanecen en HIGH se evalúa constantemente para cada una de las salidas si el valor del timer ha llegado al valor requerido y definido previamente de acuerdo al ciclo útil que se necesite en cada señal. Una vez haya llegado el timer al valor requerido para generar un ciclo útil en una de las salidas se pone en LOW la señal de la respectiva salida. En otras palabras, se sincroniza el flanco de bajada del pulso de la señal de salida con un valor del registro del timer, una vez el registro del timer llega al valor de sincronización, el estado de la salida pasa de HIGH a LOW. Este proceso se realiza con la cantidad de salidas que se requieran (Figura 84).

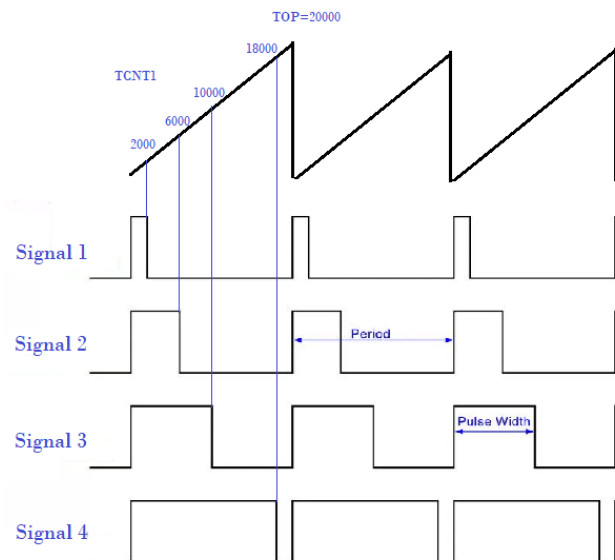


Figura 84. Sincronización de señales con timer TCNT1.

Sin embargo, sincronizar valores del timer con señales de control requiere estar evaluando constantemente el valor del timer para poner en LOW la señal que ha llegado a su valor de sincronización. Si no se evalúa constantemente el registro del timer el sistema podría poner en LOW la señal un poco después del valor de sincronización, haciendo que el ciclo útil de la señal cambie y por consiguiente el ángulo del servomotor.

Este proceso pone como prioridad en el microcontrolador la generación de señales PWM ya que debe haber una evaluación constante del timer, y pone como tarea secundaria la recepción de los datos enviados por Matlab® debido a que se reciben en una interrupción del sistema por parte del módulo UART del microcontrolador, haciendo que la recepción de datos termine siendo una tarea lenta y de poca prioridad.

Para un funcionamiento adecuado del circuito de control, la recepción de datos debe tener prioridad incluso mayor que la generación de señales PWM ya que generar una señal PWM requiere de dos acciones cada 20 milisegundos que equivale al periodo de la señal, estas dos acciones son poner en HIGH la señal por un tiempo y luego poner en LOW la señal

por otro periodo de tiempo, mientras que la recepción de datos debe hacerse a una velocidad superior que la del PWM si no se quiere perder información en el proceso.

Teniendo claro estas consideraciones se opta por una metodología que permite mantener la recepción de datos como prioridad en el microcontrolador teniendo la generación de señales de control como tarea secundaria.

La idea consiste en generar el pulso de las señales PWM por medio de dos interrupciones del timer, la primera interrupción del timer permite poner el pulso de la señal PWM en HIGH durante un tiempo t_h y la segunda interrupción pone en LOW la señal por otro periodo de tiempo t_l , es decir que el ancho de pulso requerido corresponde al valor máximo del timer (Registro ICR1), según se ilustra en la figura 85.

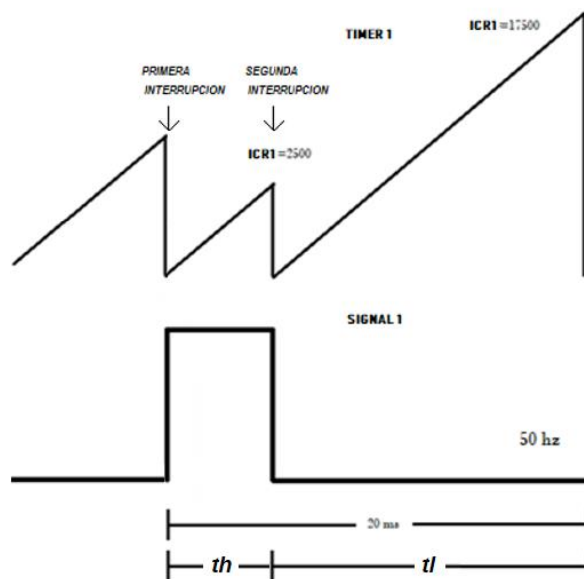


Figura 85. Pulso con par de interrupciones.

Utilizando el par de interrupciones se modifica el ancho de pulso de una señal PWM, sin embargo, para controlar el brazo robótico es necesario generar 4 señales PWM independientes. Para conseguir esto se aprovecha el periodo de tiempo desperdiciado en el que una señal se encuentra en LOW, ya que el máximo de tiempo posible en HIGH para una

señal PWM que busca controlar un servo es de 2500 unidades es decir 2.5 ms lo que equivale a 180 grados, esto significa que tenemos 17.5 ms de tiempo (dado que el periodo de la señal es de 20 ms) que podemos usar para controlar otras salidas antes de que el periodo de la señal se repita nuevamente. Esta metodología permite controlar máximo 8 servomotores por timer ya que si se divide el periodo de la señal de 20 ms en 8 servomotores se obtiene un tiempo de 2.5 ms para cada servomotor, suficiente para controlar todos los ángulos de cada servomotor.

Siguiendo esta idea y teniendo en cuenta que solo se necesitan 4 salidas de PWM se divide el periodo de 20 ms en 4 servomotores, es decir un tiempo de 5 ms para controlar cada servomotor. La figura 86 Muestra un ejemplo con tiempos diferentes para cada señal.

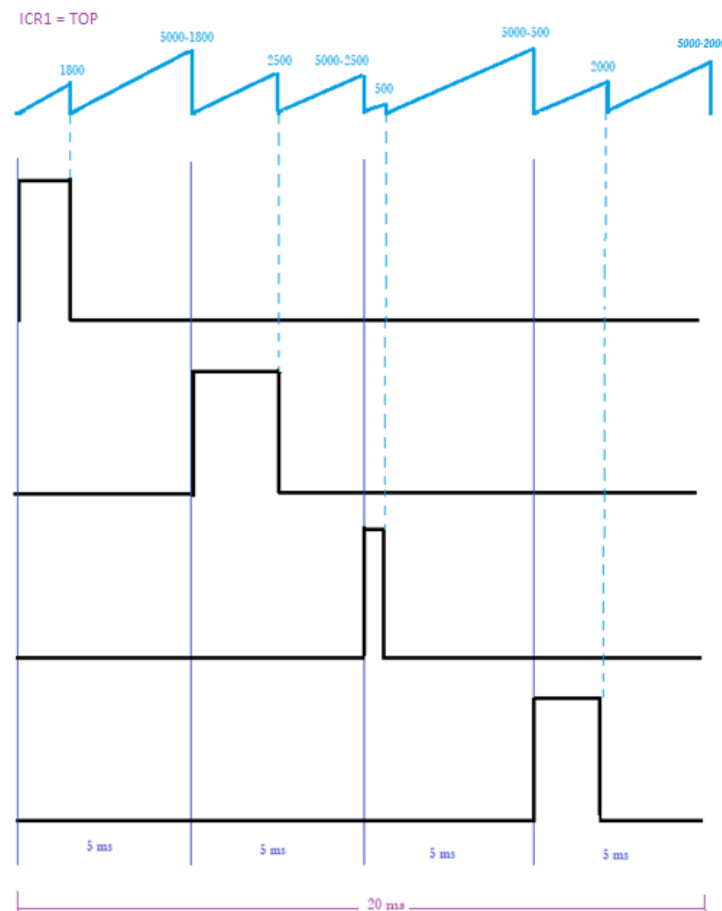


Figura 86. Generación de varias salidas PWM.

Los valores máximos del timer se obtienen realizando una conversión de los valores de los cuatro ángulos a sus respectivas equivalencias en ancho de pulso teniendo en cuenta que 0 grados equivalen a 500 unidades del timer, y 180 grados equivalen a 2500 unidades del timer.

5. Resultados

5.1. Resultados de sistemas de visión artificial

Para corroborar el funcionamiento y comportamiento de cada uno de los sistemas de visión artificial y de movilidad de brazo robótico se realizaron pruebas de campo a fin de conocer cuáles son las ventajas de desventajas de cada uno de estos sistemas.

5.1.1. Sistema de detección 1 – segmentación por contraste.

Para el primer sistema se realizaron un total de 20 pruebas en las cuales se buscó verificar el comportamiento de este frente a diferentes entornos propuestos. Las pruebas se realizaron modificando 3 variables, el fondo, el color de manga a segmentar y la iluminación del ambiente. Algunas de estas pruebas se documentan en las figuras 87 a 90

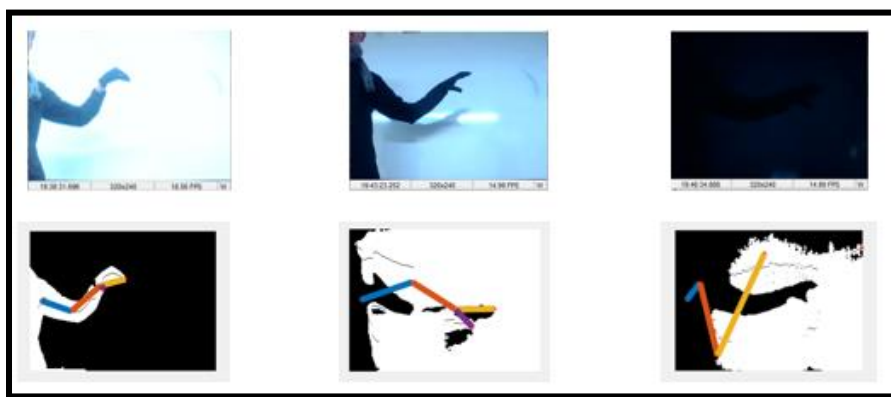


Figura 87. Prueba fondo claro, manga oscura y variación de luz.

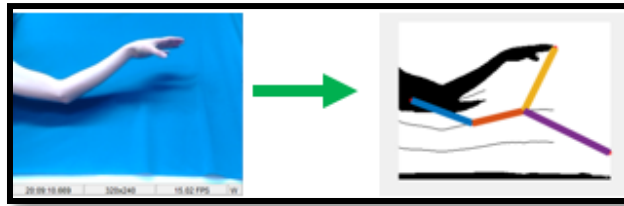


Figura 88. Prueba fondo azul, sin manga e iluminación alta.

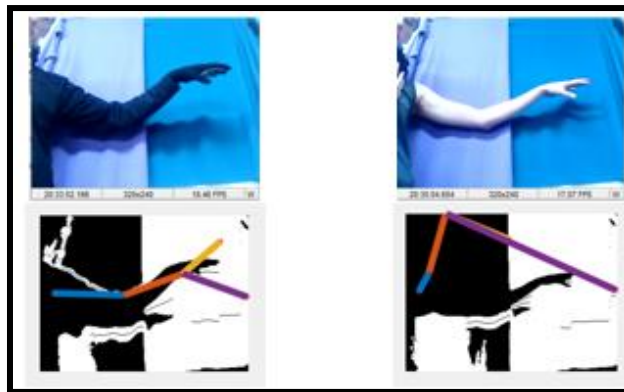


Figura 89. Prueba de fondo con dos colores.

Del total de las pruebas realizadas se presentaron anomalías por segmentación en el 20 por ciento de las pruebas, en el 70 por ciento no tuvo funcionalidad y solo el 10 por ciento fueron exitosas, tal como se aprecia en las figuras 87 a 89. Esto se debe a que el sistema requiere para su funcionamiento un ambiente totalmente controlado, donde la segmentación pueda realizarse de manera satisfactoria. El efecto inverso de la figura 88 se debe a que el sistema identifica como objetivo de interés a segmentar el fondo y no el brazo, lo cual afecta totalmente la detección y el movimiento del brazo robótico.

En la figura 90 se evidencia la segmentación frente a un ambiente controlado, lo cual ayuda a potenciar la detección del sistema.

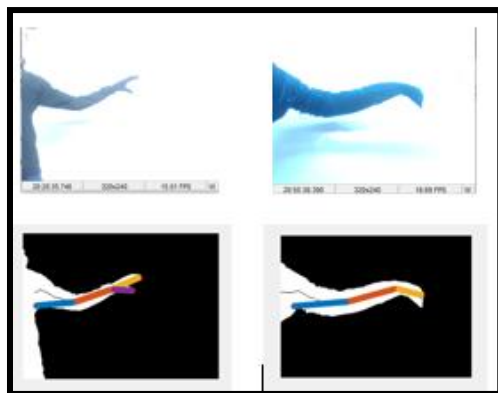


Figura 90. Pruebas con ambiente totalmente controlado.

5.1.2. Sistema de detección 2 – con guante bicromático.

Para el sistema 2 se realizaron 12 pruebas de las cuales el 80 por ciento tuvo una segmentación adecuada y por tanto un seguimiento de los movimientos exitoso, el porcentaje restante presento fallas debido a la alta iluminación sobre un fondo claro. La diferencia de este sistema con el sistema 1 es el reconocimiento del brazo, pues este lo hace a través de las articulaciones como se mencionó anteriormente. Las figuras 91 a 94 muestran algunos de los resultados obtenidos de las pruebas aplicadas al sistema.

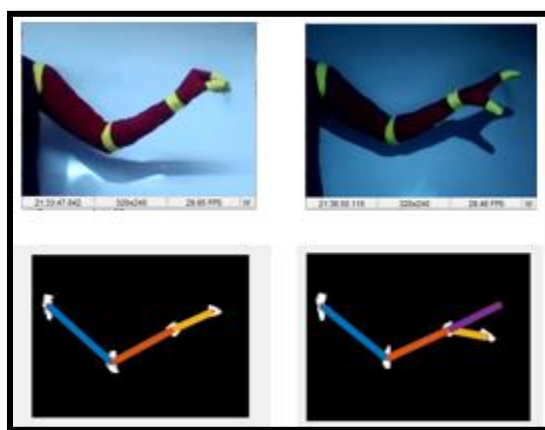


Figura 91. Pruebas con fondo claro y variación de luz.



Figura 92. Prueba de fondo oscuro con variación de luz.

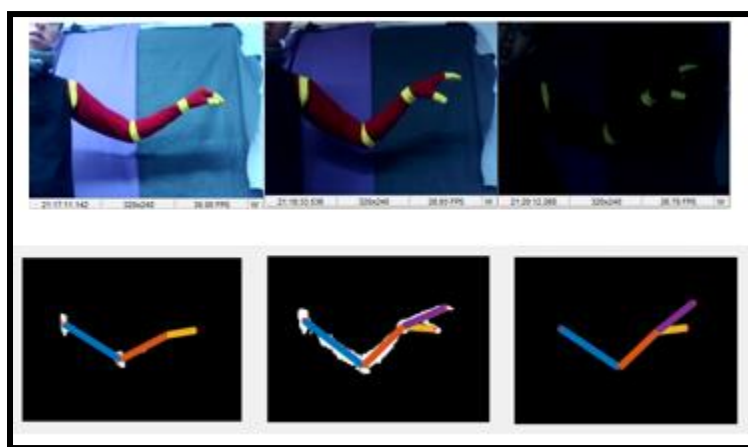


Figura 93. Prueba de fondo combinado con variación de luz.

Se aprecia en las imágenes que la mayoría de los resultados fueron positivos, esto se debe a que la prioridad del sistema es segmentar al amarillo de las articulaciones sin importar fondo o iluminación. La única desventaja evidenciada se presenta cuando se emplea una iluminación alta sobre un fondo claro, donde se pueden llegar a perder los colores que destacan las articulaciones. La figura 94 muestra otros resultados obtenidos en pruebas donde el sistema 1 no funcionaba e invertía la segmentación.

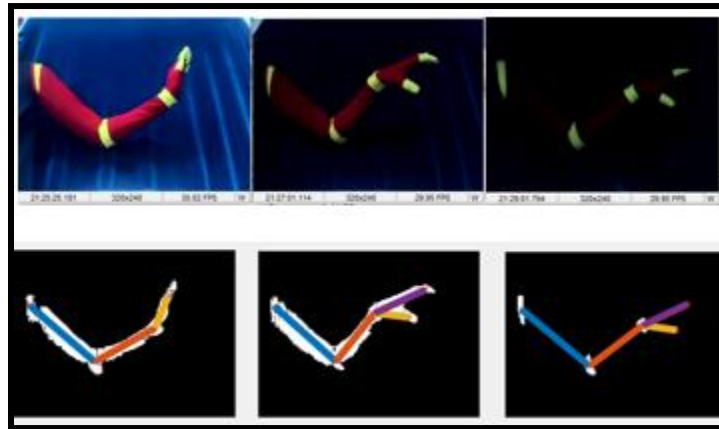


Figura 94 .Prueba de segmentación con fondo azul.

5.1.3. Sistema de detección 3 – con dispositivo Kinect Xbox 360.

Para el sistema de detección 3 se realizaron 9 pruebas que consisten en modificar la distancia entre la persona y el dispositivo Kinect con el fin de poder ubicar exactamente el rango de trabajo que puede tener el sistema. Estas pruebas se pueden apreciar en las imágenes 95 a 99.

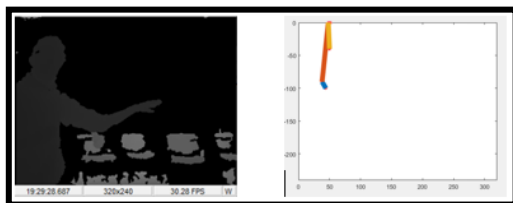


Figura 95. Prueba de detección a 1,30 m.

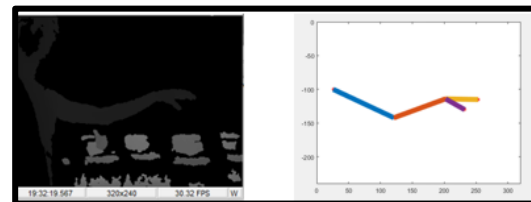


Figura 97. Prueba de detección a 1 m.

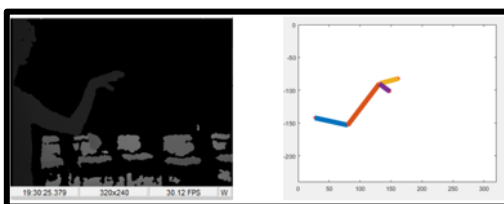


Figura 96. Prueba de detección a 1,10 m.

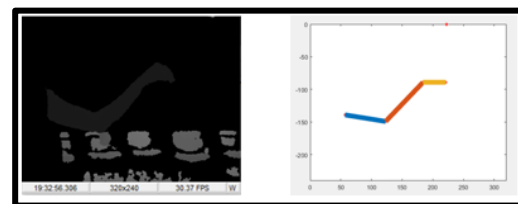


Figura 98. Prueba de detección a 80 cm.

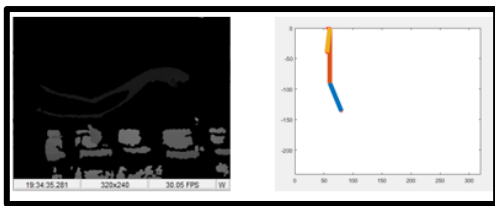


Figura 99. Prueba de detección a 60 cm.

Tabla 6. Resultados obtenidos en las pruebas de distancia.

Distancia (cm)	Resultado
130	No detecta
110	Detección defectuosa
1	Buena detección pero intermitente
80	Buena detección
60	No detecta

Tal y como se evidencia en las imágenes y se resume en la tabla 3, el rango de detección para que no se presente falla en el sistema se encuentre entre 80 cm a 1 metro.

Un segundo experimento se realiza en este sistema de detección con el fin de determinar si la detección es afectada por objetos que se encuentren en el mismo plano de detección del brazo. Los resultados de este experimento se evidencian en la figura 100.

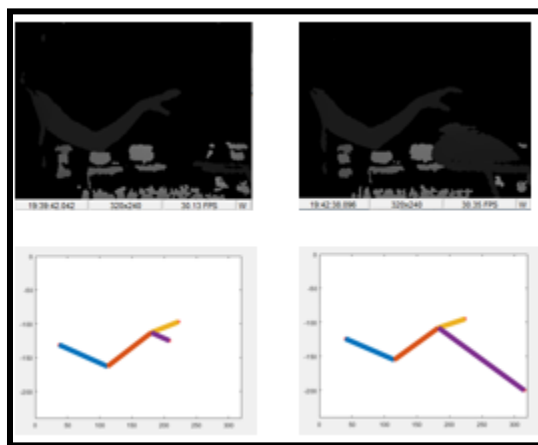


Figura 100. Prueba con objetos.

La deformación del esqueleto en la detección se debe a que el sistema es capaz de descartar los objetos y figuras menores a un área determinada, si estas figuras sobrepasan esa área el sistema toma como parte de la segmentación dichos objetos.

De las pruebas realizadas al sistema 3 tan solo el 30 por ciento fueron fallidos, el 50 por ciento funcionaron satisfactoriamente y el porcentaje restante detectaba, pero presentaba intermitencias.

A manera de resumen se presentan las observaciones obtenidas en las pruebas realizadas a los sistemas de detección 1 2 y 3 donde se evidencia lo siguiente:

- El sistema de detección 1 se presenta en las pruebas como un sistema poco robusto puesto que en la mayoría de las pruebas los resultados no fueron satisfactorios, sin embargo, los resultados pueden ser satisfactorios si el sistema se implementa en un ambiente completamente controlado.
- Para el sistema de detección 1 es indispensable evitar las sombras puesto que esto altera la detección del brazo y por ende la precisión del movimiento del robot.
- Cualquier fondo de tono azul invertirá el sistema de visión artificial, es decir, el sistema de visión segmentará como objeto a resaltar el fondo y no el brazo, dando una imagen umbralizada contraria a la que se desea.
- Para que la detección sea perfecta se recomiendan colores negros y azules, puesto que harán que el sistema segmente de buena manera.
- Los fondos con colores combinados pueden afectar la detección, puesto que esto modificará el histograma de la imagen y por ende su segmentación y umbralización.
- La iluminación media y baja para el sistema de visión artificial 1 puede resultar en un mal contraste y una segmentación nula, es por esto que al aplicar este sistema se debe tener buena iluminación, Por el contrario, el sistema dos funciona mejor con iluminaciones bajas, puesto que se logra resaltar más el color amarillo que es el color principal del sistema.

- Para el sistema de visión artificial 2 se observa una buena robustez en cuanto a detección en la mayoría de las pruebas, esto pues solo tiene que detectar el color amarillo y mientras el fondo no sea de este mismo color las articulaciones se segmentarán de manera satisfactoria.
- El sistema de visión 3 puede presentar fallas intermitentes cuando se Trabaja fuera del rango establecido, haciendo que la detección sea defectuosa y por ende un mal seguimiento.
- Para el caso del sistema de visión artificial 3 es necesario no tener objetos muy grandes en el rango de detección para que no dañe la segmentación del brazo.
- Para el sistema de detección 3 se debe tener en cuenta que la posición del brazo y su movimiento debe estar paralelo al dispositivo Kinect para obtener una mejor segmentación y movilidad del brazo humano.
- El movimiento realizado por el brazo robótico va a ser preciso siempre y cuando la segmentación se encuentre bien realizada.

5.2. Resultados análisis dinámico

Consideraciones:

- Las pruebas de desplazamiento se hicieron con una velocidad lineal del efector final de 10 cm/s en su respectivo eje de desplazamiento de acuerdo con la prueba realizada.
- Los desplazamientos descritos se hicieron a lo largo del volumen de trabajo permitido por el brazo.
- Los ángulos presentados en las gráficas son los ángulos de rotación de las articulaciones con respecto a los planos de referencia especificados en el algoritmo de Denavit Hartenberg.
- El análisis dinámico se hizo sin carga en el efector final en las 3 pruebas de desplazamiento.
- Se definieron positivos los torques generados por el efecto de la gravedad sobre los eslabones, por lo que todo torque que se oponga a la gravedad será negativo en el análisis.
- Se utilizó la cinemática inversa para simular cada una de las pruebas realizadas en el software Matlab®.

5.2.1. Prueba 1 - Movimiento en el eje x.

El desplazamiento se realiza de izquierda a derecha y se representa en la figura 100.

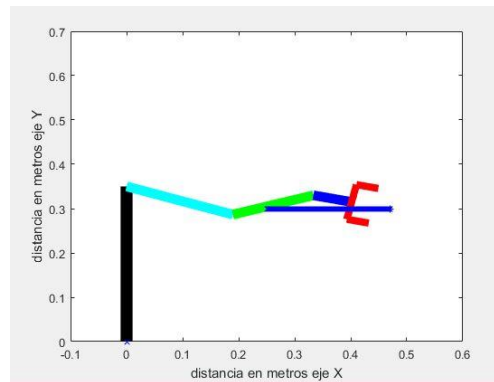


Figura 101. Desplazamiento en el eje x.

Eslabón 1.

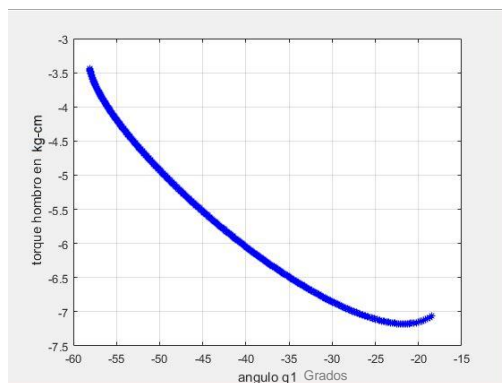


Figura 102. Valores de torque requeridos en la articulación 1 para generar desplazamiento en el eje x.

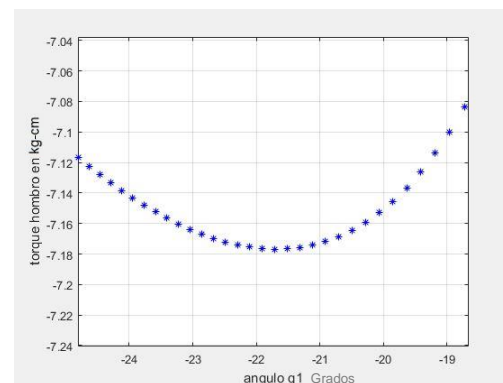


Figura 103. Imagen aumentada 1.

Torque máximo = -7.18 kg-cm

Eslabón 2.

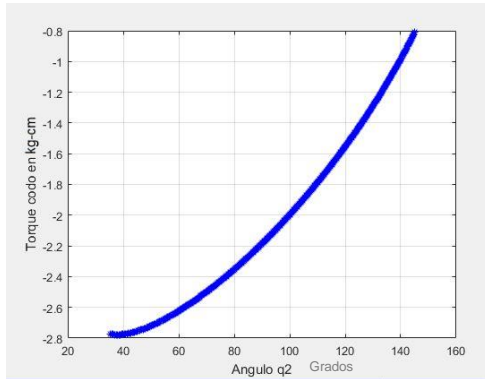


Figura 104. Valores de torque requeridos en la articulación 2 para generar desplazamiento en el eje x.

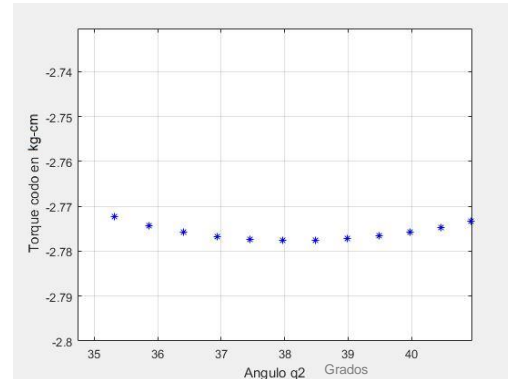


Figura 105. Imagen aumentada 2.

Torque máximo=-2.78 kg-cm.

Eslabón 3.

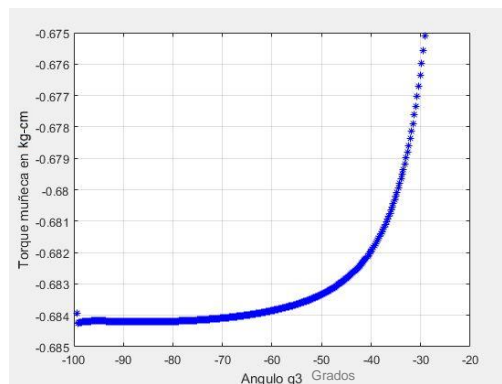


Figura 106. Valores de torque requeridos en la articulación 3 para generar desplazamiento en el eje x.

Torque máximo=-0.684 kg-cm.

En la prueba 1 se puede observar como a medida que se genera el desplazamiento de izquierda a derecha del efector final, el centro de masa del robot se aleja de su base y por lo tanto el torque requerido en cada articulación aumenta. Se puede apreciar también como

el hombro es la articulación que presenta mayor torque debido a los pesos del eslabón 2 y 3 que tiene que soportar, y de la misma forma aumenta su valor de torque a medida que el efector final se aleja de la base.

5.2.2. Prueba 2 - Movimiento en el eje y.

El desplazamiento se hizo de forma ascendente como se muestra en la figura 106 debido a que el movimiento se opone a la gravedad y los valores de torque requeridos aumentan.

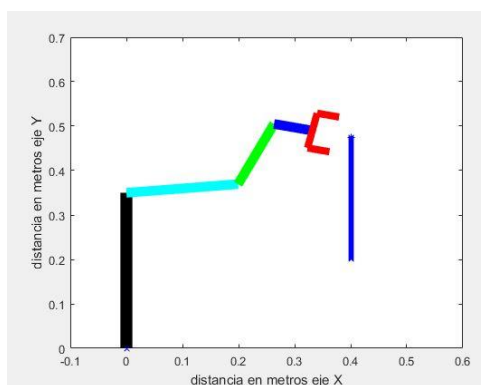


Figura 107. Desplazamiento eje y.

Eslabón 1.

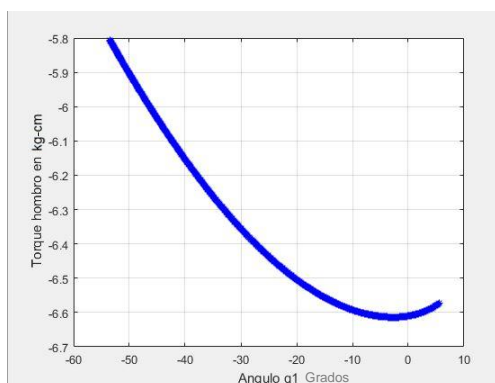


Figura 108. Valores de torque requeridos en la articulación 1 para generar desplazamiento en el eje y.

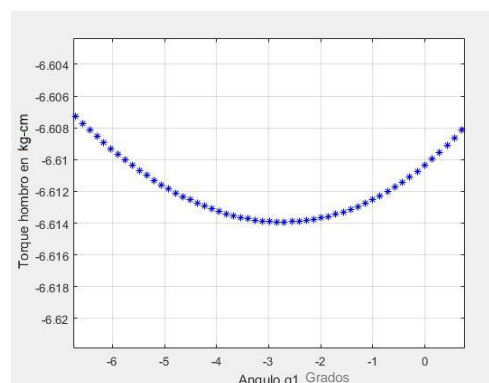
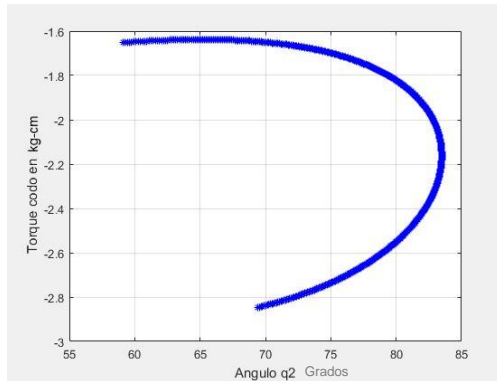


Figura 109. Imagen aumentada 3.

Torque máximo=-6.614 kg-cm.

Eslabón 2.



*Figura 110.*Valores de torque requeridos en la articulación 2 para generar desplazamiento en el eje y.

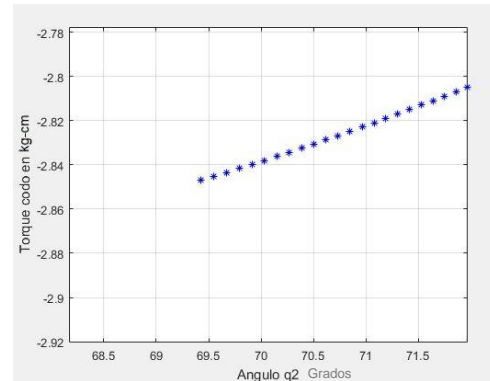
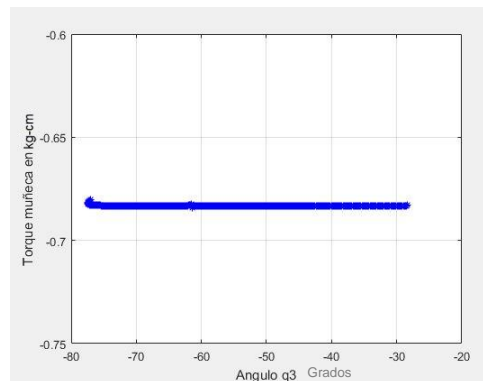


Figura 111. Imagen aumentada 4.

Torque máximo=-2.85 kg-cm.

Eslabón 3.



*Figura 112.*Valores de torque requeridos en la articulación 3 para generar desplazamiento en el eje y.

Torque máximo=-0.685 kg-cm.

En la prueba 2 se puede observar como el eslabón 1 aumenta el torque requerido a medida que asciende el efector final y el torque del eslabón 2 se va reduciendo progresivamente. Esto se debe a que inicialmente la articulación del eslabón 2 debe generar un movimiento angular en sentido contrario a la gravedad para el inicio del desplazamiento y a medida que va finalizando el desplazamiento, el sentido de giro de este eslabón se vuelve a favor de la gravedad, reduciendo el torque requerido en esa articulación, mientras tanto la articulación del eslabón 1 debe generar un movimiento angular en sentido contrario a la gravedad y a medida que asciende el efector final el centro de masa del brazo robot se va alejando de la base, aumentando el torque requerido. Por otra parte, no se presentan cambios significativos en el torque del eslabón 3 ya que este eslabón se encuentra estático en una posición definida a partir de la cinemática inversa (ver pág. 68).

5.2.3. Prueba3 -Movimiento en ambos ejes.

Al igual que en la prueba 2 el movimiento en el eje Y se hizo de forma ascendente y en el eje x de izquierda a derecha como se muestra en la figura 112.

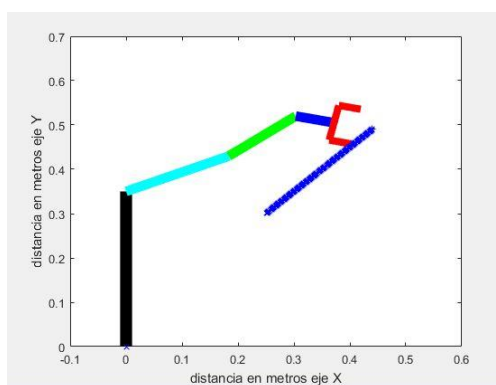
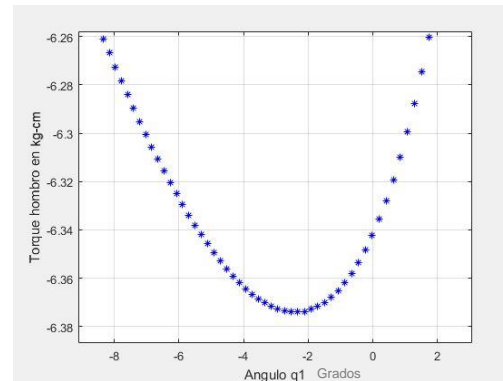
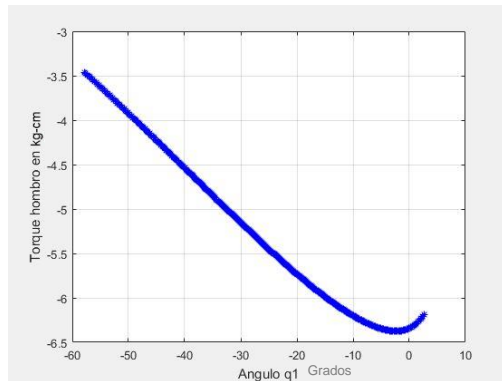


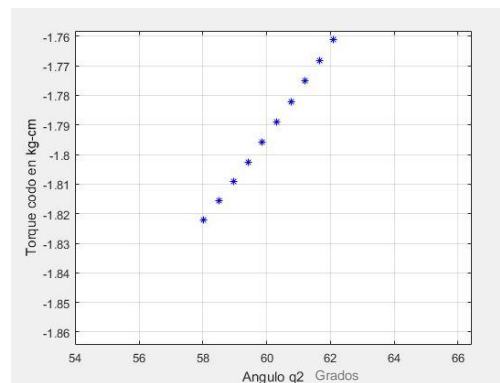
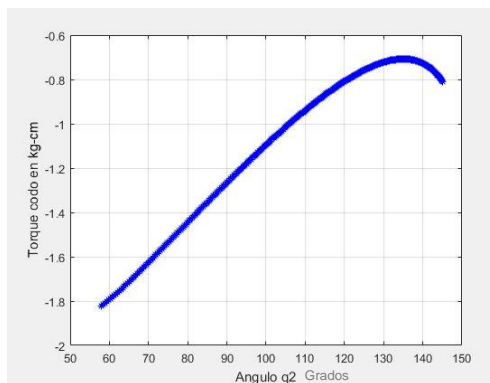
Figura 113. Desplazamiento en los ejes X & Y.

Eslabón 1.



Torque máximo=-6.37 kg-cm.

Eslabón 2.



Torque máximo=-1.822 kg-cm

Eslabón 3.

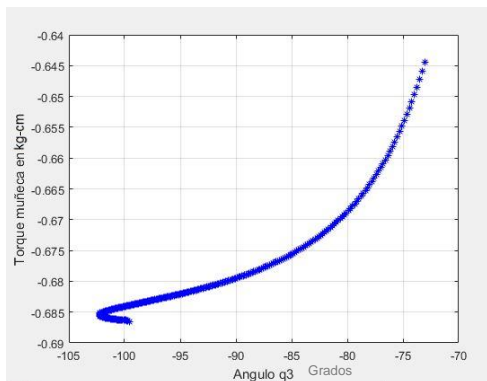


Figura 118. Valores de torque requeridos en la articulación 3 para generar desplazamiento diagonal.

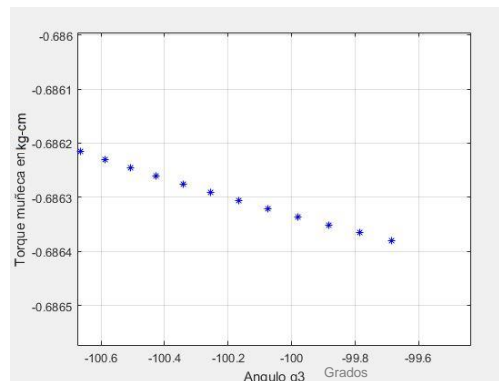


Figura 119. Imagen aumentada.

Torque máximo=-0.6864 kg-cm.

En la prueba 3 se puede observar un comportamiento similar al presentado en la prueba 1 donde el torque del eslabón 1 y 2 aumentan a medida que el efector final se aleja de la base del robot, el eslabón 1 aumenta su torque a medida que el ángulo q_1 , que se encuentra con respecto a la horizontal, se va reduciendo, es decir aumenta el torque a medida que el brazo asciende y el eslabón 2 aumenta el torque a medida que el ángulo q_2 , que corresponde al ángulo del eslabón 2 con respecto al eslabón anterior, se va reduciendo. Para el caso del eslabón 3 se presenta un ligero aumento de torque al inicio del desplazamiento debido a que la velocidad que genera el eslabón 2 sobre el eslabón 3 ocasiona un mayor torque a la articulación de la muñeca si el ángulo q_3 es mayor, y debido a que el ángulo q_3 se reduce a medida que se realiza el desplazamiento del efector final, el torque en esa articulación también se va reduciendo.

Además de estas pruebas se hicieron algunas otras donde la más relevante fue la prueba de barrido de ángulos mostrada en la figura 119, la cual permite conocer los torques de cada articulación haciendo dicho barrido de ángulos simultáneo a cada eslabón. Para esta prueba el barrido se hizo desde los 0 grados hasta los 90 grados.

5.2.4. Prueba de barrido de ángulos.

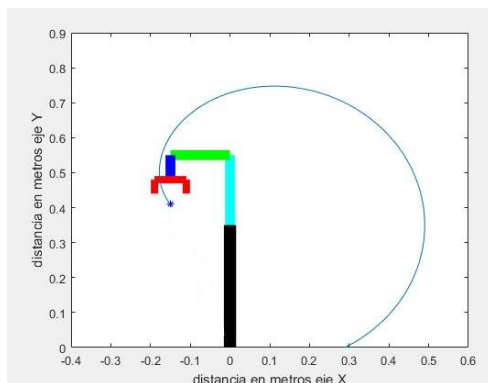


Figura 120. Barrido de ángulos.

En esta prueba se le ingresa al sistema una variación constante de ángulos q_1 , q_2 , y q_3 lo que genera un movimiento circular en cada eslabón de forma individual, pero de forma conjunta genera una trayectoria en caracol, tal y como se muestra en la figura 120. Esta prueba se hizo con el fin de conocer los torques de cada articulación en un amplio rango de ángulos, lo que hace que la prueba contemple la mayoría de posibles movimientos que genere el brazo robótico en una aplicación real y así es posible conocer la mayoría de torques a los que van a tener que enfrentarse los servomotores en cualquier tipo de desplazamiento.

Para esta prueba los valores de velocidad lineal y aceleración lineal del efector final no pueden ser constantes, por lo que se varían en función de la trayectoria descrita para así poder calcular los valores de velocidades y aceleraciones angulares correctamente.

Los resultados obtenidos son:

Eslabón 1.

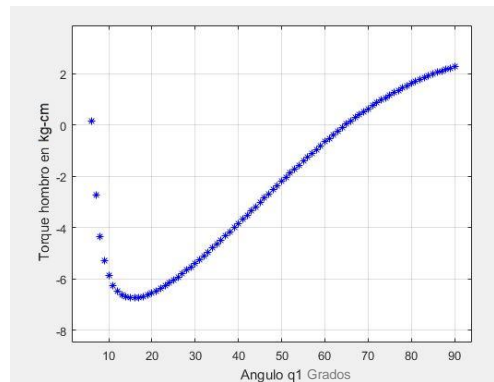


Figura 121. Valores de torque requeridos en la articulación 1 para generar desplazamiento circular.

Torque máximo=-6.8 kg-cm

Eslabón 2.

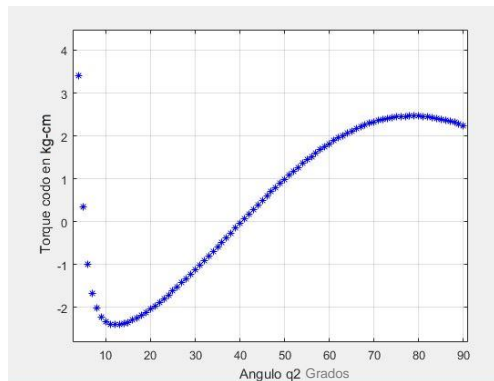


Figura 122. Valores de torque requeridos en la articulación 2 para generar desplazamiento circular.

Torque máximo= 2.51 kg-cm

Eslabón 3.

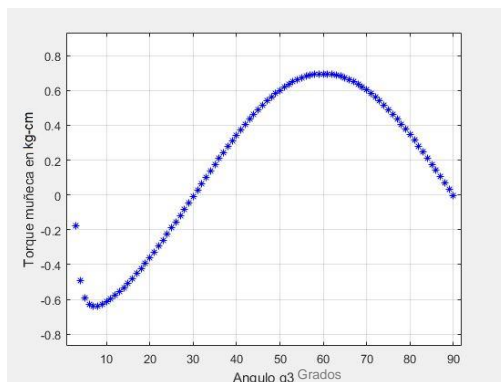


Figura 123. Valores de torque requeridos en la articulación 3 para generar desplazamiento circular.

Torque máximo= 0.70 kg-cm.

En esta prueba se evidencia como los torques requeridos en las articulaciones tienen un cambio de signo debido a que al inicio del desplazamiento el brazo robótico genera un movimiento en sentido contrario al efecto de la gravedad, sin embargo a medida que se finaliza el desplazamiento, el brazo robótico genera el movimiento a favor del efecto de la gravedad, esto quiere decir que los torques en las articulaciones deben cambiar de sentido dependiendo de si el efecto de la gravedad está a favor o en contra del movimiento.

Cabe resaltar que muchas de las pruebas se hicieron bajo el supuesto de una velocidad de trabajo normal definida en 10 cm/s, si se aumenta la velocidad, aumentarían los valores de torque. Sin embargo, los torques correspondientes a la muñeca y el codo no se verán tan afectados como la articulación del hombro debido a que es la articulación que mayor carga tiene que soportar, aun así, es importante analizar cómo reacciona esta articulación al duplicar la velocidad a 20 cm/s.

En la siguiente imagen se muestra un barrido de ángulos aplicado al eslabón 1 con una velocidad lineal de 20 cm/s en el efector final y un movimiento circular ascendente del eslabón 1.

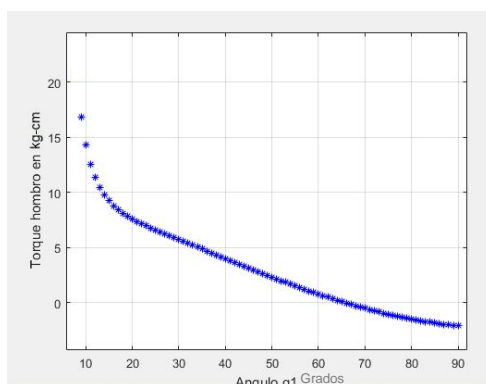


Figura 124. Valores de torque requeridos en la articulación 1 para generar el desplazamiento circular con velocidad de 20 cm/s.

Bajo estas condiciones se observa que al inicio del movimiento del eslabón 1, la inercia del robot genera un aumento significativo en el torque requerido para llegar a esta velocidad, este caso presenta el torque más alto registrado en todas pruebas realizadas con un valor de 17.1 kg-cm.

En conclusión, los torques máximos necesarios para ejecutar la mayoría de los movimientos según las pruebas son:

- *Servomotor hombro* = 7.18 a 17 kg-cm
- *Servomotor codo* = 2.85 a 5 kg-cm
- *Servomotor muñeca* = 0.7 a 1.5 kg-cm

Cabe resaltar que el rango de torques presentados anteriormente corresponde a intervalos entre los máximos torques registrados bajo las pruebas normales mostradas al comienzo de esta sección, y los torques máximos obtenidos en la prueba de mayor exigencia que formulamos anteriormente.

Al analizar estos resultados y comparándolos con valores de torques estandarizados en servomotores comerciales se seleccionó la referencia MG995, cuyas especificaciones técnicas se describen a continuación:

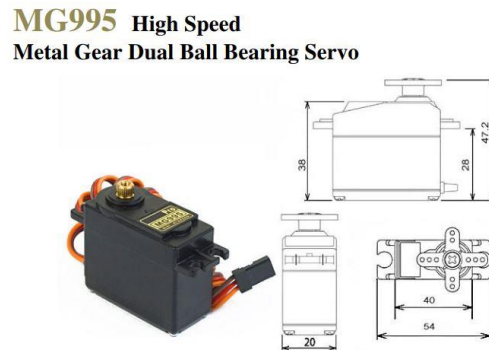


Figura 125. Servomotor MG995.

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Dead band width: 5 μ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

Se decide utilizar el mismo servomotor para todas y cada una de las articulaciones contemplando también el servomotor requerido para controlar la apertura y cierre del efector final.

Este servomotor MG995 (ver figura 125) tiene un torque de 10 kg-cm al ser operado con 6 voltios, este valor de torque es superior al requerido por la articulación del codo y la muñeca, pero no logra satisfacer el rango de torques requeridos de la articulación del hombro, es por eso que se implementan dos servomotores en la articulación del hombro, obteniendo un valor de torque de 20 kg-cm al sumar los dos torques individuales de cada uno.

5.2.5. Potencia en articulaciones.

De acuerdo con las pruebas realizadas se obtiene los valores de potencia máxima para cada una de las articulaciones a partir del producto del torque máximo y la velocidad angular correspondiente a cada articulación para generar el respectivo torque máximo.

Teniendo claro esto, los valores de potencia máxima para las pruebas realizadas son los siguientes:

<i>ServoHombro</i>	<i>ServoCodo</i>	<i>ServoMuñeca</i>
$P = 17 \text{ kg-cm} * 4.4 \text{ rad/s}$	$P = 5 \text{ kg-cm} * 1.3 \text{ rad/s}$	$P = 1.5 \text{ kg-cm} * 0.5 \text{ rad/s}$
$P = 1.66 \text{ N-m} * 4.4 \text{ rad/s}$	$P = 0.49 \text{ N-m} * 1.3 \text{ rad/s}$	$P = 0.147 \text{ N-m} * 0.5 \text{ rad/s}$
$P = 7.3 \text{ W}$	$P = 0.63 \text{ W}$	$P = 73.5 \text{ mW}$

El servomotor MG995 puede suministrar una velocidad angular de hasta 60 grados por cada 0.16 s es decir una velocidad angular de 375 grados/s o lo que es igual a 6.544 rad/s. Si se calcula la potencia máxima bajo el supuesto de que el servomotor está operando con la velocidad angular máxima que puede suministrar, los valores de potencia en las articulaciones son:

<i>ServoHombro</i>	<i>ServoCodo</i>	<i>ServoMuñeca</i>
$P = 17 \text{ kg-cm} * 6.544 \text{ rad/s}$	$P = 5 \text{ kg-cm} * 6.544 \text{ rad/s}$	$P = 1.5 \text{ kg-cm} * 6.544 \text{ rad/s}$
$P = 1.66 \text{ N-m} * 6.544 \text{ rad/s}$	$P = 0.49 \text{ N-m} * 6.544 \text{ rad/s}$	$P = 0.147 \text{ N-m} * 6.544 \text{ rad/s}$
$P = 10.86 \text{ W}$	$P = 3.2 \text{ W}$	$P = 0.9619 \text{ W}$

De acuerdo con las especificaciones técnicas del servomotor MG995 la potencia máxima que puede suministrar es la siguiente:

$$P = 10 \text{ kg-cm} * 6.544 \text{ rad/s}$$

$$P = 0.981 \text{ N-m} * 6.544 \text{ rad/s}$$

$$P = 6.41 \text{ W}$$

Cada servomotor implementado en las respectivas articulaciones del brazo robot tiene una potencia calculada anteriormente, sin embargo, para el caso específico del hombro donde se requiere el uso de dos servomotores la potencia máxima que se puede suministrar es la siguiente:

$$P = 20 \text{ kg-cm} * 6.544 \text{ rad/s}$$

$$P = 1.962 \text{ N-m} * 6.544 \text{ rad/s}$$

$$P = 12.83 \text{ W}$$

5.3. Resultados generales y validación del sistema

Una vez desarrolladas las diferentes etapas del proyecto es necesario validar el funcionamiento en conjunto tanto de los sistemas de visión artificial como del movimiento físico realizado por el manipulador robótico, es por esta razón que se realiza una serie de pruebas cuyo fin es comparar el movimiento real efectuado por brazo humano con el movimiento emulado por el brazo el brazo robótico.

Las pruebas consisten en grabar simultáneamente el movimiento realizado por el brazo humano del operador junto con el movimiento realizado por el brazo robótico, a fin de evidenciar una correcta emulación por parte del brazo robótico. Estas pruebas se evidencian en las imágenes 126 a 128, en las cuales se puede apreciar algunos de los fotogramas del video en tiempo real del movimiento realizado por la persona y el manipulador robótico. Se divide la validación en 3 grupos, el primero refiere al funcionamiento del sistema de visión artificial 1 (ver figura 126), el segundo corresponde al sistema de visión artificial 2 (ver figura 127) en los cuales se utiliza una cámara rgb y en el grupo 3 se valida el funcionamiento del sistema de visión 3 usando una cámara de profundidad infrarroja,

Sistema de visión artificial 1



Figura 126. Prueba de validación sistema 1.

Sistema de visión artificial 2

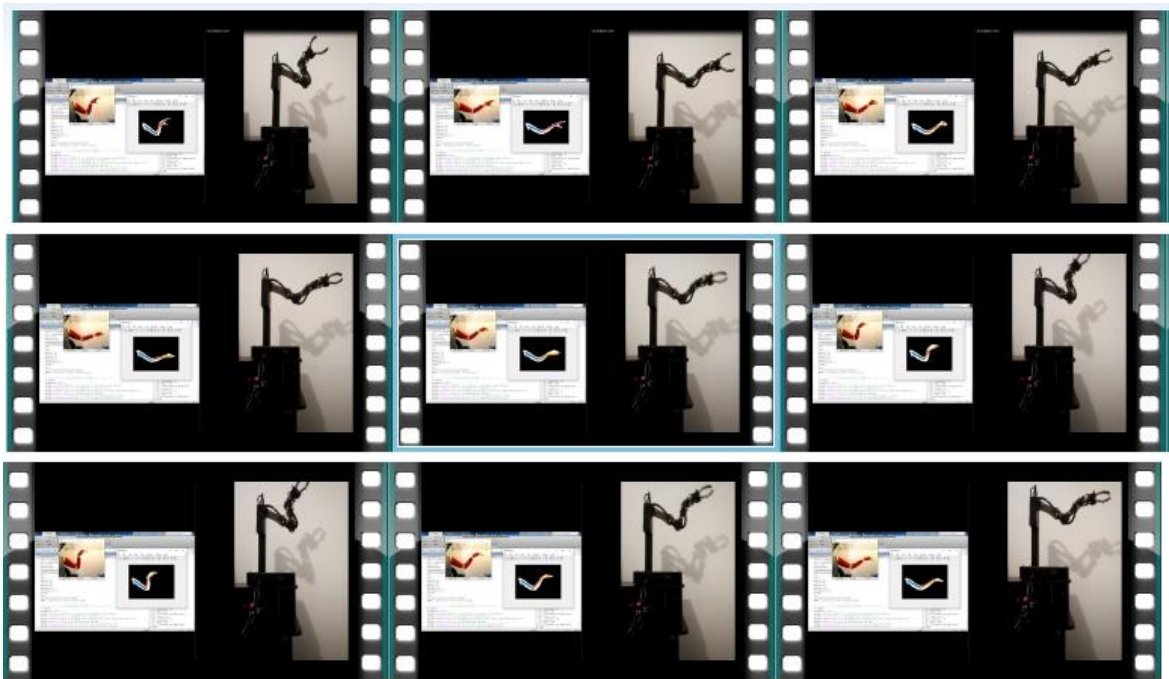


Figura 127. Prueba validación sistema 2.

Sistema de visión artificial 3.

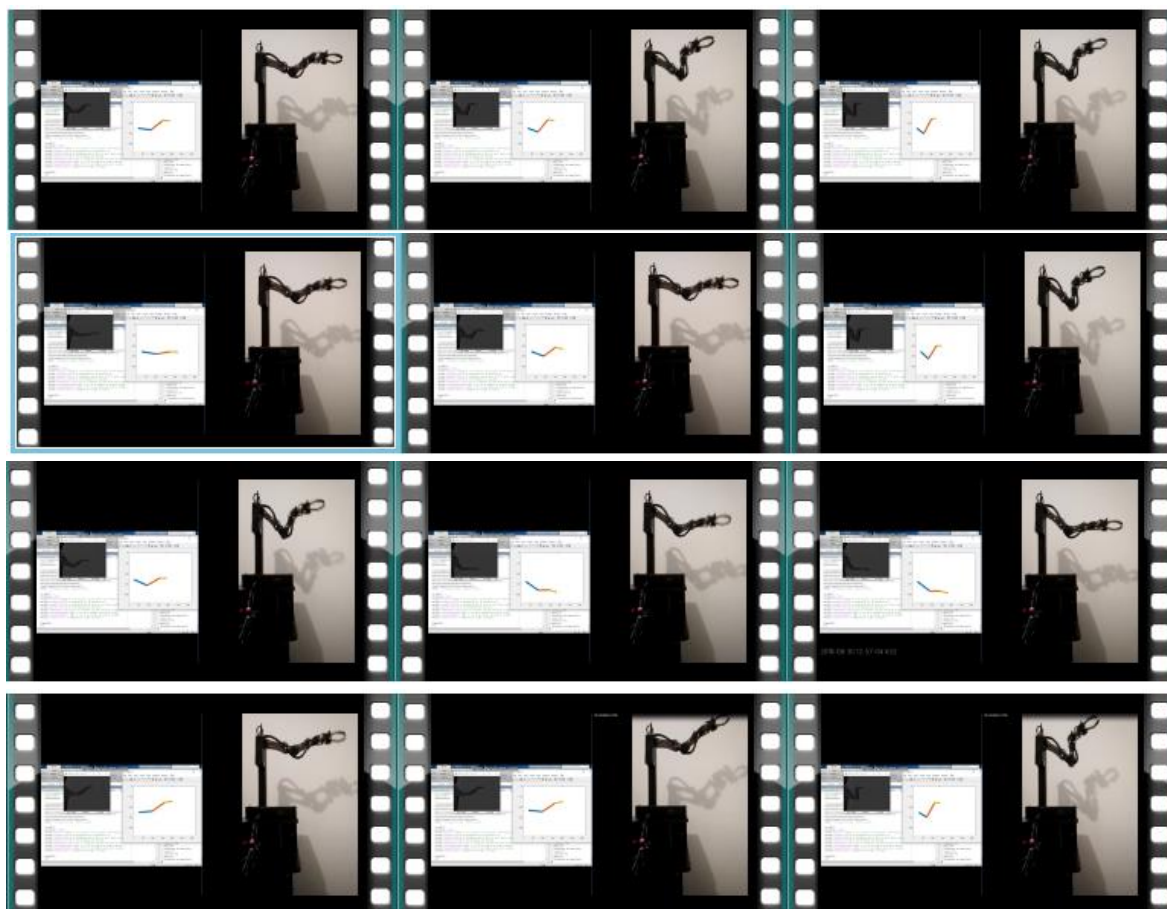


Figura 128. Prueba validación sistema 3.

A partir de estas pruebas de validación se observa que para los 3 sistemas hay un correcto funcionamiento tanto en la detección del brazo humano como en el seguimiento con el brazo robótico.

Cabe aclarar que aunque el brazo robótico emula el movimiento del brazo humano, no es una herramienta de precisión, por lo cual en futuras mejoras se pretende controlar el ángulo real del brazo robótico para así poder comparar este ángulo con el ángulo enviado por el controlador determinando así el grado de error que este presenta, con este error se puede diseñar un control en lazo cerrado asegurando no solo la emulación del movimiento sino la precisión del brazo robótico con respecto al brazo humano.

6. Conclusiones

De acuerdo con los experimentos realizados se concluye que todos los sistemas de detección pueden conseguir resultados satisfactorios en la segmentación del brazo humano y la emulación del movimiento del brazo robótico, sin embargo, debido a las características de cada uno de los sistemas se debe considerar las restricciones que estos presentan con respecto a parámetros variables como la iluminación y el color del fondo.

Con un brazo robótico de 4 grados de libertad es posible emular la mayoría de los movimientos que realiza el brazo humano en un plano, sin embargo, el movimiento del brazo robótico está limitado a las restricciones angulares de cada servo motor.

A partir de materiales de bajo costo como el acrílico es posible construir un robot manipulador de 4 grados de libertad capaz de emular el movimiento de un brazo humano en un plano.

Además de los parámetros variables como iluminación y color de fondo, la segmentación del brazo humano depende de la configuración de la cámara que captura la imagen y de la efectividad del computador para procesar las imágenes en tiempo real.

Con el uso del microcontrolador Atmega16a es posible tener un adecuado funcionamiento en el brazo robótico y en diferentes áreas de la robótica debido a su bajo costo y alto rendimiento.

La metodología implementada en este proyecto puede ser mejorada o adaptada en otros proyectos en los cuales se amplíe el rango de posibles movimientos que ejecutó un brazo humano u otra extremidad a través de una detección no solo en un plano sino en dos planos cubriendo el movimiento total del brazo humano.

7. Referencias.

- Cattaneo, C. A., Larcher, L. I., Ruggeri, A. I., Herrera, A. C., & Biasoni, E. M. (2011). Métodos De Umbralización De Imágenes Digitales Basados En Entropía De Shannon Y Otros, *XXX*, 2785–2805.
- Cortés Reyes, F. (2011). *Robótica: Control de Robots Manipuladores*. (Alfaomega, Ed.) (Primera). México D.F.
- Ebert, L. C., Hatch, G., Thali, M. J., & Ross, S. (2013). Invisible touch-control of a DICOM viewer with finger gestures using the kinect depth camera. *Journal of Forensic Radiology and Imaging*, 1(1), 14. <https://doi.org/10.1016/j.jofri.2012.11.006>
- Fasecolda. (2018). Reporte por clase de riesgo y actividad económica. Retrieved from <https://sistemas.fasecolda.com/rldatos/Reportes/xClaseGrupoActividad.aspx>
- Gallo Sánchez, L. F., Guerrero Ramírez, M. A., Vásquez Salcedo, J. D., & Alonso Castro, M. Á. (2016). Diseño de un prototipo electromecánico para la emulación de los movimientos de un brazo humano. *Inge Cuc*, 12(2), 17–25. <https://doi.org/10.17981/ingecuc.12.2.2016.02>
- Garaizar Sagarminaga, P. (2013). ¿Cómo consigue vernos nuestra videoconsola? Retrieved from <http://enciende.cosce.org/boletin/index.asp?item=75>
- Grzegorz Baron, Piotr Czekalski, Daniel Malicki, K. T. (2014). Touch-Less Java Based Hand Tracking System to Control an Artificial Arm, *214*(2), 1–10. <https://doi.org/10.4028/www.scientific.net/SSP.214.1>
- Hartenberg, R. S., & Denavit, J. (1955). A kinematic notation for lower-pair mechanisms based on metrics. *Transactions of the ASME. Journal of Applied Mechanics*. <https://doi.org/citeulike-article-id:7153318>
- Hermosilla, D. M., & Castilla Pérez, A. (2016). Generación de trayectorias para el brazo

- robótico (ArmX): EBSCOhost. *RIELAC*, XXXVII, 58–71. Retrieved from <http://web.a.ebscohost.com/ehost/detail/detail?vid=5&sid=28cfe709-889a-49e4-a444-618cac007c94%40sessionmgr4007&hid=4109&bdata=JnNpdGU9ZW9vc3QtbGl2ZQ%3D%3D#AN=119317869&db=zbh>
- Lego Group. (2018). Lego® Mindstorms® ntx 2.0. Retrieved July 14, 2018, from <https://shop.lego.com/en-US/LEGO-MINDSTORMS-NXT-2-0-8547>
- M, G. L. (2009). Segmentación de imágenes basado en etiquetado de píxeles.
- Microsoft. (2012). Kinect Sensor. Retrieved from <https://msdn.microsoft.com/en-us/library/hh438998.aspx>
- Mind treat studios. (2012). Real life occlusion of rendered content [Kinect]. Retrieved October 15, 2017, from <https://www.mindtreatstudios.com/kinect-real-life-occlusion-rendered-content/>
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
- Salazar, B., Darío, G., Sandoval, R., & Lucía, O. (2015). Remote control system of a robotic arm 4 degrees of freedom using machine vision. *Revista EIA*, 12, 121–219. <https://doi.org/http://dx.doi.org/10.14508/reia.2015.12.24.121-129>.
- Sánchez Calle, Á. (2005). *Aplicaciones de la visión artificial y la biometría informática*. (Dykinson, Ed.). Madrid , España. <https://doi.org/10.17981/ingecuc.12.2.2016.02>
- Serrano, I., Bueno, G., Salido, J., & Déniz, O. (2011). ASM Facial Combinado Image Visible y de Profundidad ASM Facial Combinando Imagen Visible y de Profundidad, (May), 8.
- Singh, B. P. (2015). Imaging Applications of Charge Coupled Devices (CCDs) for

Cherenkov Telescope Summer Training Project Report, (October), 50.

Ulloa Calzonzin, M. G. del C. (2017). *MECÁNICA LAGRANGIANA y HAMILTONIANA*.

Guanajuato.

Zhang, T. Y., & Suen, C. Y. (1984). A Fast Parallel Algorithm for Thinning Digital Patterns, 27(3), 236–239.

Zhao, Y., Gong, L., Huang, Y., & Liu, C. (2016). A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127, 311–323. <https://doi.org/10.1016/j.compag.2016.06.022>