UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA
INGENIERÍA ELECTRÓNICA

Trabajo de Titulación previo a la obtención del título de INGENIERA ELECTRÓNICA

TEMA
DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL A TRAVÉS DE UNA TARJETA DE DESARROLLO PARA CONTROLAR EL BRAZO ROBÓTICO MITSUBISHI

AUTORA
ANDREA LILIANA ASNALEMA CONDO

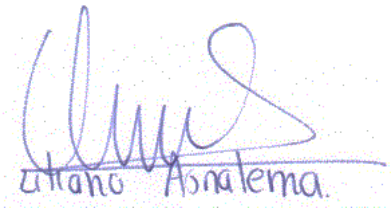
DIRECTORA
CELI SÁNCHEZ CARMEN JOHANNA

Quito, agosto 2017

CESIÓN DE DERECHOS DE AUTOR

Yo Andrea Liliana Asnalema Condo, con documento de identificación N° 1724461205, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy la autora del trabajo de grado/titulación intitulado: " Desarrollo de un sistema de visión artificial a través de una tarjeta de desarrollo para controlar el brazo robótico Mitsubishi", mismo que ha sido desarrollado para optar por el título de: Ingeniera Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autora me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....

Nombre: Andrea Liliana Asnalema Condo

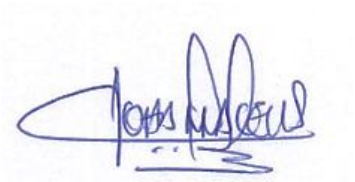
Cédula: 1724461205

Fecha: agosto 2017

CARTA DE DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, Desarrollo de un sistema de visión artificial a través de una tarjeta de desarrollo para controlar el brazo robótico Mitsubishi, realizado por Andrea Liliana Asnalema Condo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto 2017



_____

Carmen Johanna Celi Sánchez
CI: 1717437808

AGRADECIMIENTO

A Dios por todo lo que tengo y por todo lo que soy, porque tengo un techo donde dormir, una familia que me espera y una carrera que, con este proyecto técnico, inicia.

A mi madre Griselda Condo Ramírez porque día a día me enseñó a luchar por mis sueños, a preservar y a nunca rendirme, a pesar de todos los obstáculos que nos pone la vida, ella nos sacó adelante y con la culminación de mi carrera quiero agradecerle por haber estado en mis logros, festejándolos y en mis fracasos, animándome.

A mi hermano Maury, por su incondicional apoyo, por sus consejos y porque me ha permitido ver en él un ejemplo de superación y gracias a él pude acabar mis estudios.

A mi hermano Jimmy, por ser la luz que Dios nos regaló para salir adelante ya que todo lo que hacemos es para que él tenga un mejor futuro.

A mi familia, a mis tíos y primos, que con sus consejos y su apoyo incondicional hicieron posible la realización de esta meta.

Finalmente, a mis amigos, ya que ellos compartieron conmigo los mismos sueños, las mismas metas, un agradecimiento por el apoyo mostrado.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Existen tareas en las cuales se requiere de movimientos del brazo humano, pero por el peligro que representan tanto para la salud como para la vida del ser humano no son posibles realizarlas en el sitio, lo aconsejable es ejecutar todas las acciones necesarias a una distancia segura para las personas, para ello se puede hacer uso de Robots los cuales pueden comandarse remotamente.

El brazo robótico puede replicar los movimientos del brazo humano para ejecutar este tipo de tareas en donde existe un riesgo considerable para la salud e integridad de una persona. Este proyecto propone una herramienta para el reconocimiento de los movimientos de un brazo humano y su posterior réplica para dar un mayor alcance a las tareas que no son posibles realizarlas por personas.

El presente proyecto de titulación tiene como objetivo realizar la captura de movimientos simples del brazo humano para su posterior réplica en el brazo robótico MITSUBISHI MELFA RV-2AJ perteneciente a la Universidad Politécnica Salesiana.

Para ejecutar el proyecto, se construyó un cuarto obscuro, en donde se encuentran ubicadas las dos cámaras, que recogerán las imágenes de los movimientos que ejecute el brazo humano.

La tarjeta Raspberry Pi 3, procesará estas imágenes, realizará un proceso de filtrado y cálculos aritméticos para determinar los ángulos de las junturas (J1, J2, J3) que deberá describir el robot, para replicar el movimiento. Estos datos son enviados hacia la tarjeta Arduino, la misma que es la encargada de realizar la comunicación serial con el brazo MITSUBISHI.

Palabras clave: desarrollo, sistema, visión, artificial, tarjeta, brazo, robótico.

ABSTRACT

There are tasks in which human arm movements are required, but because of the danger they pose to the health and life of the human being, it is not possible to perform them on the site, it is advisable to take all necessary actions at a safe distance For people, for it can be made use of robots that can be commanded remotely.

The robotic arm can replicate the movements of the human arm to perform such tasks where there is a considerable risk to the health and integrity of a person. This project proposes a tool for the recognition of the movements of a human arm and its subsequent replication to give a greater scope to the tasks that are not possible to perform them by people.

The present titling project aims to capture simple movements of the human arm for later replication in the MITSUBISHI MELFA RV-2AJ robotic arm belonging to the Salesian Polytechnic University.

To execute the project, an obscure room was built, where the two chambers are located, which will collect the images of the movements executed by the human arm.
The Raspberry Pi 3 card will process these images, perform a filtering process and arithmetic calculations to determine the joints angles (J1, J2, J3) to be described by the robot, to replicate the movement. These data are sent to the Arduino card, which is responsible for performing serial communication with the MITSUBISHI arm.

Keywords: development, system, vision, artificial, card, arm, robotic.

INTRODUCCIÓN

En la actualidad existen tareas que, debido a su grado de toxicidad y peligro, para la integridad del ser humano, deben ser ejecutadas remotamente. Los movimientos del brazo humano no pueden realizarse en el sitio, debido al riesgo potencialmente alto que significaría para la persona.

Para realizar lo antes mencionado, hay que implementar un sistema que sea capaz de reconocer los movimientos que ejecuta el brazo humano y replicarlos en un brazo robótico, con un tiempo de respuesta adecuado para que las tareas puedan ser ejecutadas exitosamente.

El presente proyecto de titulación trata acerca del Desarrollo de un sistema de visión artificial a través de una tarjeta de desarrollo para controlar al brazo robótico MITSUBISHI. El robot puede realizar varias tareas a la vez tanto con algoritmos realizados en el software del fabricante, así como también ejecutando órdenes externas.

El proyecto abarca los siguientes capítulos:

Capítulo 1: Se presenta el planteamiento del problema, los objetivos, la justificación, la propuesta de solución, el alcance y las limitaciones del proyecto.

Capítulo 2: Se detalla la documentación técnica que fue necesaria para el desarrollo del presente proyecto técnico de titulación, con el objetivo de tener un mayor entendimiento acerca de cada etapa del proyecto.

Capítulo 3: Se describe el hardware construido para el sistema de visión artificial y el software implementado en la tarjeta de desarrollo Raspberry Pi 3 junto con Simulink de Matlab y en la tarjeta Arduino que es la encargada de la comunicación serial con el Robot.

Capítulo 4: Se detallan las pruebas realizadas al prototipo para comprobar su precisión, la eficiencia del proceso y el tiempo de respuesta del robot con respecto al cambio del movimiento, las conclusiones a las que se llegó y las recomendaciones para futuras investigaciones.

CAPÍTULO 1

ANTECEDENTES

En este capítulo se detalla el problema, los objetivos tanto general como específicos, la justificación, la propuesta de solución y el alcance que tendrá el presente proyecto técnico de titulación.

1.1. PLANTEAMIENTO DEL PROBLEMA

Existen tareas en las cuales se requiere de movimientos del brazo humano, pero por el peligro que representan tanto para la salud como para la vida del ser humano no son posibles realizarlas en el sitio, lo aconsejable es ejecutar todas las acciones necesarias a una distancia segura para las personas, para ello se puede hacer uso de Robots los cuales pueden comandarse remotamente. En este trabajo se propone usar, el brazo robótico perteneciente a la Universidad Politécnica Salesiana marca MITSUBISHI modelo MELFA RV-2AJ.

El uso del brazo robótico a través de visión artificial hace posible dar solución al problema controlándolo remotamente o a distancia del fenómeno.

1.2. OBJETIVOS

1.2.1. Objetivo General

Desarrollar un sistema de visión artificial que sea capaz de reconocer el movimiento de un brazo humano para replicarlo en un brazo robótico MITSUBISHI.

1.2.2. Objetivos Específicos

- Desarrollar algoritmos de visión artificial capaces de reconocer movimientos simples de un brazo humano a través de cámaras compatibles con la tarjeta de desarrollo para replicarlos en un brazo robótico MITSUBISHI.

- Implementar el software de visión artificial en una tarjeta de desarrollo a través de Matlab para la captura de imágenes y su posterior procesamiento.

- Establecer la comunicación entre la tarjeta de desarrollo y el brazo robótico MITSUBISHI para el envío de comandos de movimientos.

- Ejecutar las pruebas de campo para recoger datos, validar la efectividad del controlador y analizar los resultados.

- Establecer un documento técnico de todos los procesos fundamentales utilizados en el proyecto.

1.3. JUSTIFICACIÓN

El brazo robótico puede replicar los movimientos del brazo humano para ejecutar cualquier tipo de tareas en donde existe un riesgo considerable para la salud e integridad de una persona. El proyecto propone una herramienta para el reconocimiento de los movimientos de un brazo humano y su posterior réplica para dar un mayor alcance a las tareas que no son posibles realizarlas por personas.

1.4. PROPUESTA DE SOLUCIÓN

En el presente proyecto se detalla la aplicación de un sistema de visión artificial para el reconocimiento de movimientos simples del brazo humano para replicarlo en el brazo robótico MITSUBISHI. Este sistema a ser implementado contará con dos cámaras, las mismas que serán las encargadas de enviar las imágenes a la tarjeta de desarrollo que las procesará y entregará los comandos de movimiento al brazo robótico MITSUBISHI. El desarrollo de algoritmos de visión artificial será bajo la plataforma de Matlab como herramienta de desarrollo

1.5. ALCANCE

El sistema de visión artificial será capaz de realizar las siguientes acciones:

- Reconstruirá movimientos simples del brazo humano, tomando en cuenta las tres junturas principales del brazo robótico, las cuales en el cuerpo humano representan: base, hombro y codo, dejando el control de la muñeca para posteriores investigaciones.

- Capturará imágenes y las procesará mediante cámaras compatibles con la tarjeta de desarrollo.

- Enviará datos hacia el controlador del brazo mediante una comunicación serial desde la tarjeta de desarrollo.

- Realizará un proceso aritmético para calcular los ángulos descritos por el brazo humano para replicarlos en las junturas del brazo robótico Mitsubishi.

- Replicará los movimientos del brazo izquierdo, ajustándose a las limitaciones del hardware y del rango de movimiento de las junturas del Robot.

El sistema no podrá realizar:

- No será capaz de reconstruir movimientos complicados del brazo humano de más de tres grados de libertad ni muñeca.

- No será capaz de reconstruir los movimientos ejecutados en tiempo real.

- No podrá replicar movimientos del brazo humano derecho.

CAPÍTULO 2

MARCO TEÓRICO

2.1. TARJETA DE DESARROLLO RASPERRY PI

Raspberry PI es una mini computadora de placa reducida, es decir que centra todo en un solo circuito. Esta tarjeta es de bajo costo y fue desarrollada en Reino Unido por la fundación Raspberry Pi con objetivos académicos. Existen varias generaciones de Raspberry Pi desde su lanzamiento: 1-A, 1-A+, 1-B, 1-B+, 2-B, 3-B. En la tabla 2.1 se muestra una comparación de los diferentes modelos de Raspberry Pi. (RaspberryPi, 2014)

Tabla 2.1: Comparación entre diferentes modelos de Raspberry Pi

Todos los modelos de Raspberry Pi	Raspberry Pi A	Raspberry Pi A+	Raspberry Pi B	Raspberry Pi B+	Raspberry Pi 2 B
SoC (Chip)	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2836
CPU	ARM1176JZF-S a 700 MHz	ARM1176JZF-S a 700 MHz	ARM1176JZF-S a 700 MHz	ARM1176JZF-S a 700 MHz	ARM Cortex-A7 cuatro núcleos a 900 Mhz
GPU	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz
Memoria RAM	256 MB a 400 Mhz	256 MB a 400 Mhz	512 MB a 400 Mhz	512 MB a 400 Mhz	1 GB a 450 Mhz
Entradas de vídeo	Cámara CSI	Cámara CSI	Cámara CSI	Cámara CSI	Cámara CSI
Salidas de vídeo	HDMI 1.4, conector RCA	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS
Salidas de audio	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI
Conectores USB 2.0	1	1	2	4	4
Tarjetas de almacenamiento	SD	microSD	SD	microSD	microSD
Conexión a red	No	No	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg
Interfaz periféricos (GPIO)	8	17	8	17	17

Tabla comparativa entre tarjetas Raspberry, Fuente: (Pascual, 2015)

2.1.1. Raspberry Pi 3

El modelo que se usará para el presente proyecto de titulación es RASPERRY PI 3, ya que, al ser una mini computadora cuenta con características físicas que permiten conectar dos cámaras usb para la toma de imágenes y su posterior procesamiento,

también cuenta con una micro sd-card con gran capacidad de almacenamiento, en el caso de este proyecto se usará de 8GB.

La Raspberry Pi 3 es la tercera generación de Raspberry Pi, reemplazó a la tarjeta Raspberry Pi 2 modelo B+ en febrero del 2016. (RaspberryPi, 2014)
Las características que presenta esta tarjeta son:

- 4 puertos USB
- 40 pines GPIO
- Puerto Full HDMI
- Puerto Ethernet
- Interfaz para cámara (CSI)
- Interfaz para Display (DSI)
- Micro SD card
- Video Core IV 3D
- Una CPU ARMv8 quad-core de 64 bits de 64 bits y 1.2 GHz
- LAN inalámbrica 802.11n
- Bluetooth 4.1
- Bluetooth baja energía (BLE)

Las ventajas sobre la generación anterior son:

- ARM Cortex-A7 CPU a 900MHz
- 1GB de RAM

Debido a que tiene un procesador ARMv7, esta tarjeta puede ejecutar toda la gama de distribuciones ARM GNU / Linux, incluyendo Snappy Ubuntu Core, así como Microsoft Windows 10. (Upton, 2015)

2.2. ROBÓTICA

La robótica es una disciplina científica que aborda la investigación y desarrollo de una clase particular de sistemas mecánicos. (Reyes, 2016)

Un robot es una máquina programable que realiza acciones en sustitución de las personas humanas y se rige bajo tres principios establecidos por Isaac Asimov, los cuales son: (Batz, 2015)

1. Un robot no puede actuar contra un ser humano, o mediante la inacción, permitir que un humano sufra daños. (Batz, 2015)

2. Un robot debe obedecer las órdenes dadas por los seres humanos, salvo que estén en conflicto con la primera ley. (Batz, 2015)

3. Un robot debe proteger su propia existencia, a no ser que esté en conflicto con las tres leyes. (Batz, 2015).

2.2.1. Robots manipuladores

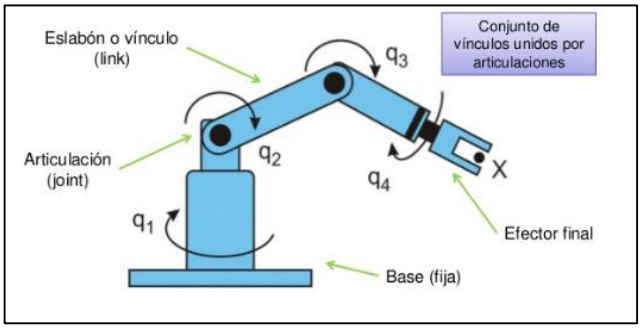
Estos robots también son conocidos como robots industriales, ya que sus aplicaciones son en empresas de manufactura y procesos automatizados. Según la ISO, lo define como un manipulador multipropósitos, reprogramable y controlado automáticamente en tres o más ejes. (Reyes, 2016)

Estos robots se distinguen de los sistemas mecatrónicos, los cuales realizan solo una función como en el caso de los electrodomésticos, es decir la estructura mecánica de los sistemas mecatrónicos no puede realizar multifunciones, en contraste los robots manipuladores si pueden realizar estas acciones cambiando únicamente la herramienta. (Reyes, 2016)

- **Estructura de los manipuladores**

Un robot manipulador consta de eslabones y articulaciones tal como se puede observar en la figura 2.1. Los eslabones son estructuras rígidas conectadas mediante las articulaciones o junturas, de diferentes grados de libertad. (Reyes, 2016)

Figura 2. 1: Estructura de un robot manipulador



Fuente: (Rosales, 2012)

Una articulación puede ser:

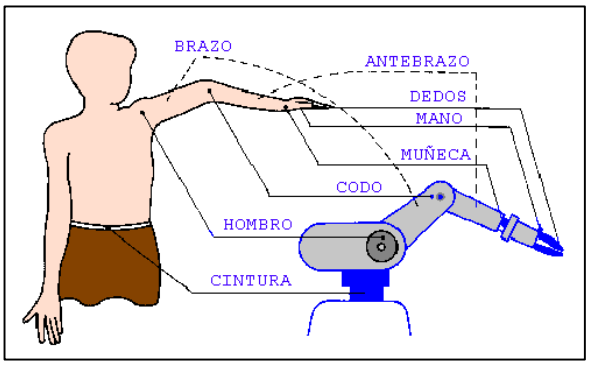
Lineal (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior. (Reyes, 2016)

Rotacional, en caso de que un eslabón gire en torno a un eje solidario al eslabón anterior. (Reyes, 2016)

A los manipuladores robóticos se les suele denominar también brazos de robot por la analogía que se puede establecer, en muchos casos, con las extremidades superiores del cuerpo humano. (Reyes, 2016)

En la figura 2.2 se puede observar la semejanza del brazo robótico con el brazo humano.

Figura 2. 2: Semejanza robot manipulador con brazo humano



Fuente: (Reyes, 2016)

Tipos de Manipuladores.

a. Robot cartesiano

Los robots presentan una estructura articulada, es decir que se encuentra conformado por una serie de elementos o eslabones que facilitan el movimiento, el cual puede ser de translación vertical, horizontal y transversal, los cuales se pueden combinar con movimiento de giro sobre el mismo eje. (Segovia, Alamilla, & Domínguez, 2015).

b. Robot cilíndrico

La estructura determinada para el robot cilíndrico facilita un desplazamiento vertical, transversal y giratorio sobre el eje que gira sobre su propia base. Se ha de reconocer también la forma de la estructura del robot SCARA que facilita movimientos y desplazamiento vertical, giratorio sobre su propio eje de la base, como el de la articulación, lo cual facilita cualquier tipo de ensamble de un producto, que se relacione con el mundo moderno. Se encuentra otros tipos de estructuras como la esférica ó polar, la angular o antropomórfico y el cilíndrico. (Segovia, Alamilla, & Domínguez, 2015)

- La posición del punto en el espacio se encuentra en las coordenadas cilíndricas (a,r,z). (Segovia, Alamilla, & Domínguez, 2015)
- Ofrecen ventajas cuando las tareas o máquinas servidas se encuentran radialmente al Robot. (Segovia, Alamilla, & Domínguez, 2015)

c. Robot angular

A continuación, veremos las principales características de este tipo de Robots.

- La posición del punto se ubica en el espacio en las coordenadas angulares (a,b,g). (Segovia, Alamilla, & Domínguez, 2015).

- Sin embargo, obliga a un esfuerzo suplementario en el sistema de control para el seguimiento de trayectorias rectilíneas. (Segovia, Alamilla, & Domínguez, 2015)

d. Robot SCARA

Especialmente para realizar tareas de ensamblado electrónico, y en general de manipulación vertical. (Segovia, Alamilla, & Domínguez, 2015)

e. Robot paralelo

- Elemento que se encuentra conectado a la base, por lo menos, a dos cadenas cinemáticas independientes.

- Inicialmente utilizada en los simuladores de vuelo.
- La carga se reparte entre los eslabones.
- La rigidez de los eslabones asegura con mayor precisión el punto deseado.
- Bajo coste relativo y montaje preciso. (Segovia, Alamilla, & Domínguez, 2015)

2.3. ROBOT MITSUBISHI MELFA RV-2AJ

El brazo robótico RV-2AJ de Mitsubishi se puede ver en la Figura 3. Se trata de un robot de brazo articulado vertical con 5 grados de libertad, que se utiliza para el transporte de piezas. El diseño del RV-2AJ es para aplicaciones con movimiento de cargas de hasta 2 Kg de peso. Este robot tiene un alcance de 410mm, y combina una velocidad máxima de 2,100mm/s con una precisión de ±0.02mm. (Guaranga, 2014)

Los servomotores de corriente alterna, unidos a codificadores de posición absolutos, garantizan fiabilidad y bajo mantenimiento. Estos codificadores permiten apagar el robot en cualquier momento y al conectar de nuevo la alimentación, el robot podrá continuar trabajando desde la posición actual. También tiene integrado un conector para cuatro sensores, y la posibilidad de utilizar pinzas de accionamiento eléctrico. (Guaranga, 2014).

Figura 2. 3: Brazo robótico Mitsubishi Melfa RV 2AJ

Fuente: (Batz, 2015)

2.3.1. Especificaciones técnicas

El brazo robótico Mitsubishi cuenta con las siguientes especificaciones técnicas, mostradas en la tabla 2.2:

Tabla 2. 2: Especificaciones técnicas Robot Mitsubishi RV-2AJ

Modelo RV-2AJ		
Grados de libertad		5
Motores		Servomotores AC (Ejes J1, J3 y J5 con freno)
Detección de posición		Encoders absolutos
Máxima carga (Kg)		2
Longitud del brazo (mm)		250+160
Alcance radial máximo (mm)		410
Límites (grados)	J1	±150
	J2	180(-60~+120)
	J3	230 (-110~+120)
	J4	--
	J5	±90
	J6	±200
Velocidad máxima (grados/s)	J1	180
	J2	90
	J3	135
	J4	--
	J5	180
	J6	210
Velocidad máxima (mm/s)		2100
Repetibilidad (mm)		±0.02
Peso (Kg)		17
Cableado		4 entradas (pinza) / 4 salidas (base)
Conexiones aire		ø4mm x 4 en la base
Instalación		Suelo o techo

Fuente: (Freire Muñoz Santiago, 2012)

De la tabla 2.2 se obtienen las siguientes características: el brazo robot tiene integrados en su interior una serie de conductos que permiten la conexión de pinzas y ventosas neumáticas para su movimiento. Dispone de un puerto RS232 y de 16 entradas y salidas digitales que son el equipamiento de serie para comunicar el robot con el exterior. Un adaptador permite comunicar el robot mediante red Ethernet (Protocolo TCP/IP). Otra opción, la red CC-Link, de Mitsubishi, permite el intercambio rápido de datos, sobre todo entre el robot y un PLC. Para el control del brazo es necesario utilizar una controladora CR1-571 y una botonera para grabar y ejecutar los movimientos deseados. (Guaranga, 2014)

2.3.2. Grados de libertad del Robot Melfa RV-2AJ

Corresponde al número mínimo de variables independientes que es necesario de unir para determinar la situación en el espacio de todos los eslabones del manipulador. La regla de Kutzbach-Grübler. permite determinar el número de grados de libertad de un robot con mecanismos especiales en términos del número de eslabones y de articulaciones. (Segovia, Alamilla, & Domínguez, 2015)

$$GDL = 3\,(n - 1) - 2j1 - 2j2 \qquad \text{Ec. (2. 1)}$$

Donde:

GDL=Grados de libertad.

n=Número de eslabones.

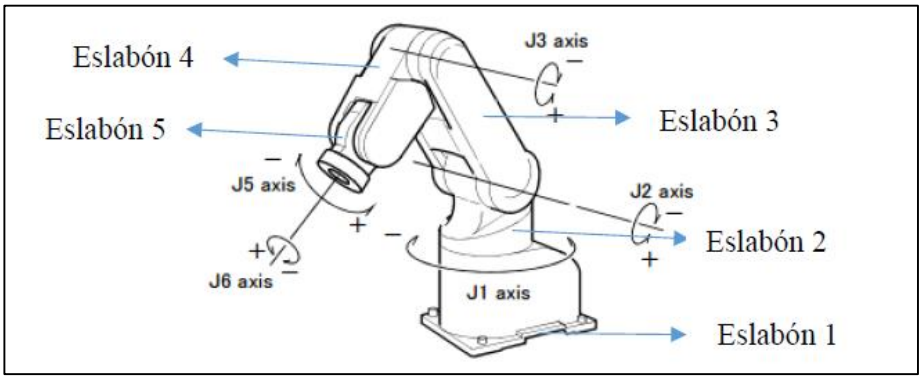
j1=Número de articulaciones de 1 grado de libertad

j2= Número de articulaciones de 2 grados de libertad

Pasos a seguir:

1. El primer paso para calcular los grados de libertad es identificar el número de junturas y el número de eslabones del brazo robótico.

Figura 2. 4: Eslabones y junturas del brazo robótico RV-2AJ



Fuente: (Mena Lisbeth, 2012)

De la figura 2.4 Se pueden obtener los siguientes datos:

n=5

j1=3

j2=1

Reemplazando estos datos en la ecuación 1 se tiene:

m=3(n−1)−2j1−j2

m=3(5−1)−2(3)−1

m=12−6−1

m=5 GDL

El robot MITSUBISHI rv-2aj, tiene 5 grados de libertad.

2.3.3. Controlador Del Robot (CR1-571)

En el controlador radica el sistema de control del Robot y también se conecta la botonera o teach. Tiene su propio lenguaje de programación llamado Melfa Basic IV y los datos se transmiten por puerto serial o TCP/IP desde una computadora. (Mena Lisbeth, 2012)

El software que permite el envío de datos desde la PC a la controladora se llama Ciros Studio, mediante dicho programa se puede controlar el movimiento de los brazos en sus diferentes ejes y además se posee el control de los sensores del entorno. El corazón del controlador es una CPU de 64 bits que permite la ejecución en paralelo de hasta 32 programas en modo multitarea, es decir, mientras se está moviendo puede recibir datos, activar o desactivar entradas y salidas, hacer cálculos y 28 tareas más. (Mena Lisbeth, 2012).

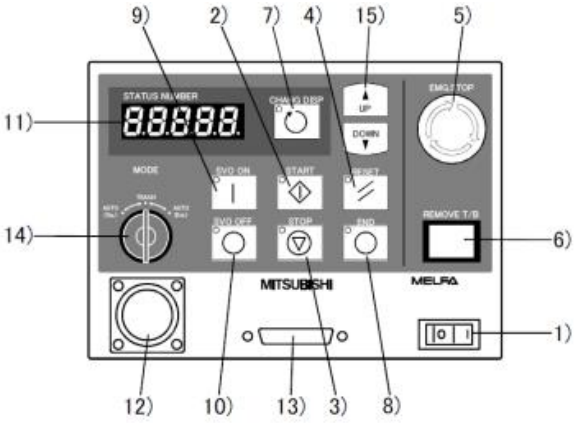
En la figura 2.5 se puede observar la controladora del robot Mitsubishi.

Figura 2. 5: Controladora del Robot (CR1-571)

Elaborado por Liliana Asnalema

A continuación, en la tabla 2.3, se presenta un diagrama de todos los botones disponibles en la controladora.

Tabla 2. 3: Botones de la controladora

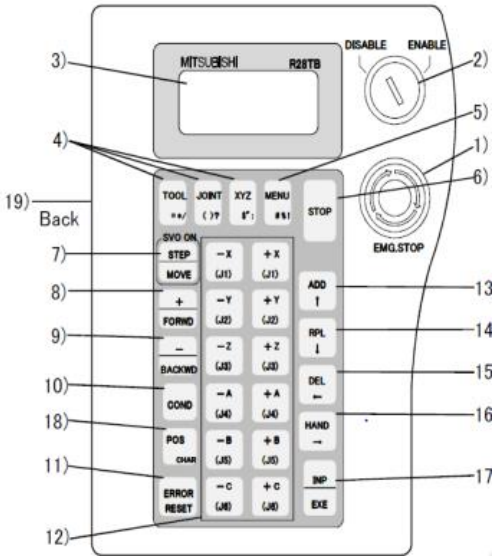
CONTROLADORA CR1-571	BOTONES DE LA CONTROLADORA
	1)Botón de encendido/apagado
	2) Botón de inicio
	3) Botón de paro
	4) Botón de reinicio
	5) Botón de paro de emergencia
	6) Interruptor del T/B (sin apagar el controlador)
	7) Botón de configuración de la pantalla de estado (display)
	8) Botón de "terminar"
	9) Botón de encendido de los servomotores
	10) Botón de apagado de los servomotores
	11) Pantalla de estado (display)
	12) Conexión dedicada del T/B
	13) Conexión para PC (RS 232, DB25)
	14) Interruptor de modos de operación
	15) Botón de arriba/abajo

Elaborado por Liliana Asnalema

2.3.4. Botonera De Mando (T/B)

También llamado teachpendant, se utiliza para determinar las posiciones del brazo robot. Para ayudar en la programación y para el control del robot, tiene integrado un visualizador LCD. En la tabla 2.4 se pueden observar las características del mismo. (Mena Lisbeth, 2012)

Tabla 2. 4: Botones de la Teaching Box

TEACH PENDANT	BOTONES
	1) Botón de paro de emergencia
	2) Botón de activar/desactivar la botonera
	3) Pantalla LCD (display)
	4) Modo de movimiento, Herramienta (tool), por ejes (joint) o cartesiano (X,Y,Z)
	5) Botón del Menú principal
	6) Botón de paro
	7) Botón de paso/movimiento
	8) Botón de +/adelante
	9) Botón de -/atrás
	10) Botón de edición de programa
	11) Botón de reinicio de errores
	12) Botones de acción
	13) Botón de agregar/arriba
	14) Botón de abajo
	15) Botón de borrar/izquierda
	16) Botón de mano/derecha
	17) Botón de registro/ejecutar
	18) Botón de caracteres numéricos/alfabéticos
	19) Botón de accionamiento
	20) Botón de contraste de pantalla

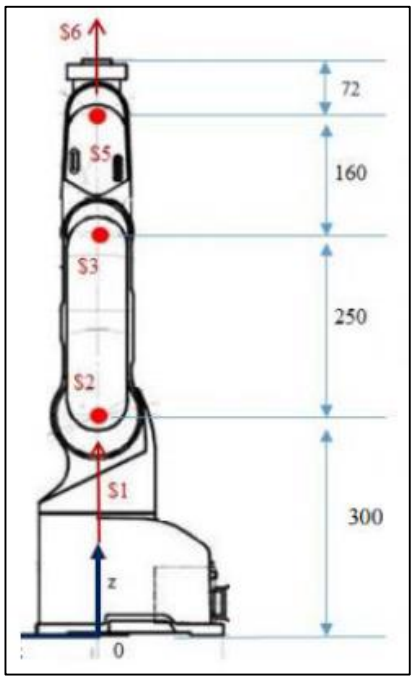
Elaborado por Liliana Asnalema

2.3.5. Cinemática directa

El modelo cinemático directo permite determinar el espacio de trabajo del brazo robot. Para el cálculo es necesario obtener los parámetros de Denavit-Hartenberg, estos datos son propios de cada mecanismo dependiendo de la geometría y dimensiones de cada robot. (Cajamarca Andrés, 2016)

En la figura 2.6 se observa las dimensiones del brazo MITSUBISHI, las mismas que serán usadas en el cálculo para los parámetros de Denavit-Hartenberg.

Figura 2. 6: Dimensiones del Brazo rv 2aj



Fuente: (Cajamarca Andrés, 2016)

Se ha escogido una posición para el cálculo y los datos son los que se muestran en la tabla 2.5.

Tabla 2. 5: Parámetros de Denavit-Hartenberg

	Θ	**d**	**a**	**α**
1	Θ1	300	0	90º
2	Θ2	0	250	0
3	Θ3	160	0	90º

Elaborado por Liliana Asnalema

Para el presente proyecto de titulación se tomarán solo tres grados de libertad debido a que ese es alcance del mismo.

Por lo tanto, la ecuación general para el cálculo es:

$$A_0^3 = A_0^1 * A_1^2 * A_2^3 \qquad \text{Ec. (2. 2)}$$

Como siguiente paso, se encuentran las matrices de transformación, para lo cual se encuentran utiliza la siguiente matriz estándar. (Cajamarca Andrés, 2016)

$$ {}^{n}_{n-1}A = \begin{pmatrix} cos\,\theta_n & -sin\,\theta_n cos\,\propto_n & sin\,\theta_n sin\,\propto_n & a_n cos\theta_n \\ sin\,\theta_n & cos\,\theta_n cos\,\propto_n & -cos\,\theta_n sin\,\propto_n & a_n sin\theta_n \\ 0 & sin\,\propto_n & cos\,\propto_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Ec. (2. 3)

La ecuación 2.4 muestra la transformación del punto 0 a 1

$$ {}^{1}_{0}A = \begin{pmatrix} cos\,\theta_1 & 0 & sin\,\theta_1 & 0 \\ sin\,\theta_1 & 0 & -cos\,\theta_1 & 0 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Ec. (2. 4)

La ecuación 2.5 muestra la transformación del punto 1 a 2

$$ {}^{2}_{1}A = \begin{pmatrix} cos\,\theta_2 & -sin\,\theta_2 & 0 & 250\,cos\,\theta_2 \\ sin\,\theta_2 & cos\,\theta_2 & 0 & 250\,sin\,\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Ec. (2. 5)

La ecuación 2.6 muestra la transformación del punto 2 a 3

$$ {}^{3}_{2}A = \begin{pmatrix} cos\,\theta_3 & 0 & sin\,\theta_3 & 0 \\ sin\,\theta_3 & 0 & -cos\,\theta_3 & 0 \\ 0 & 1 & 0 & 160 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Ec. (2. 6)

Al multiplicar estas tres matrices se obtiene una resultante de 4x4, de la misma solo nos interesa la última columna ya que es la que se asocia las coordenadas X, Y, Z, las mismas que son:

$$ X = -160\sin\theta 1 + 250 cos\theta 1 cos\theta 2 $$

$$ Y = 160\cos\theta 1 + 250 sin\theta 1 sin\theta 2 $$

$$ Z = 250\sin\theta 2 + 300 $$

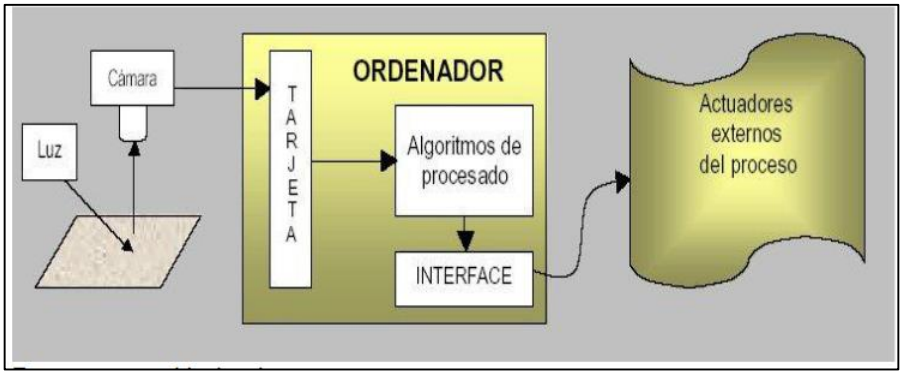
2.4. VISIÓN ARTIFICIAL

La visión artificial o llamada también visión por computadora, es una rama de la inteligencia artificial que incluye métodos para adquirir, procesar, analizar y comprender las imágenes captadas por medio de cámaras. (Lozano Germán, 2015)

2.4.1. Componentes de la visión artificial

En la figura 2.7 se pueden observar los componentes que conforman un sistema de visión artificial.

Figura 2. 7: Componentes de la visión artificial



Fuente: (Lozano Germán, 2015)

- **Iluminación**

Esta es la parte más importante del sistema, debido a que es necesario resaltar las características más relevantes de los objetos a analizar. Esto a su vez ayuda a que las cámaras obtengan una buena captura de la imagen que posteriormente será procesada. Los objetos siempre deben estar iluminados de forma adecuada y estos pueden variar dependiendo del resultado que se desee obtener. (Lozano Germán, 2015)

- **Adquisición**

Esta tarea se describe como la captura de los rayos reflejados por los objetos, los cuales son convertidos en señales eléctricas para poder ser procesados. Es realizada por la cámara, y en ocasiones por otros dispositivos como rayos X o ultrasonidos. (Lozano Germán, 2015)

- **Procesamiento**

En este punto del proceso, es cuando la imagen previamente capturada pasa a ser introducida en memoria, y el sistema de adquisición o el procesador pueden aplicar distintas técnicas de pre procesamiento que permitan la extracción de características importantes, necesarias y requeridas por el usuario para su interpretación. (Lozano Germán, 2015)

- **Actuadores**

Debido a que los sistemas de visión artificial, generalmente forman parte de grandes procesos en grandes sistemas, sus resultados son entregados a robots, cintas transportadoras y todo tipo de actuadores los cuales se encargan de realizar el procedimiento mecánico del sistema. (Lozano Germán, 2015)

CAPÍTULO 3
DISEÑO E IMPLEMENTACIÓN

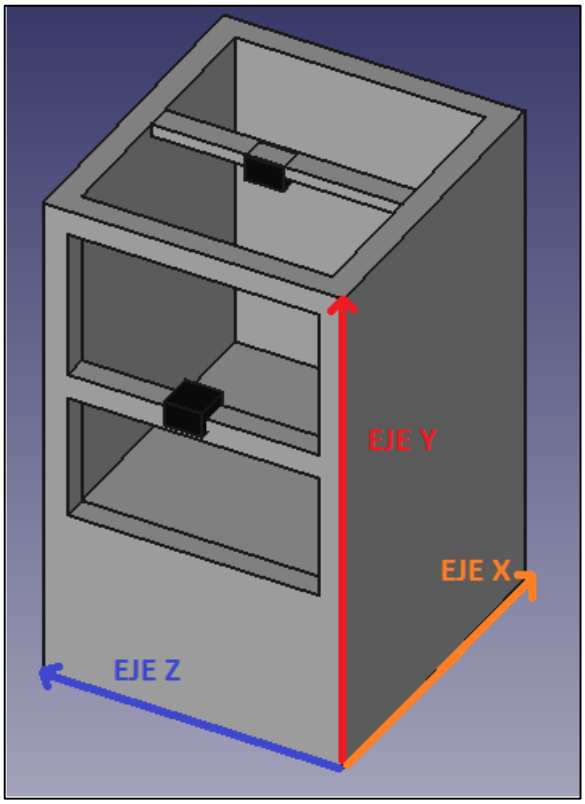
En el presente capítulo se va a detallar el hardware y software utilizado para el presente proyecto de titulación. Se describe el procedimiento realizado por la tarjeta de desarrollo Raspberry Pi 3 junto con Matlab para ejecutar las técnicas de visión artificial, también se describirá la comunicación hacia el brazo robótico.

3.1. HARDWARE

Para poder implementar la visión artificial se construyó un cuarto obscuro de dimensiones: 95x210x140 cm, con la finalidad de que las cámaras detecten solo los puntos de referencia.

Dentro del cuarto obscuro están ubicadas las dos cámaras a 90 grados, como se puede ver en la figura 1. Las cámaras son las encargadas de tomar la imagen de su plano con los dos puntos referenciados, que posteriormente servirán para vectorizarlos.

Figura 3. 1: Esquema del cuarto obscuro



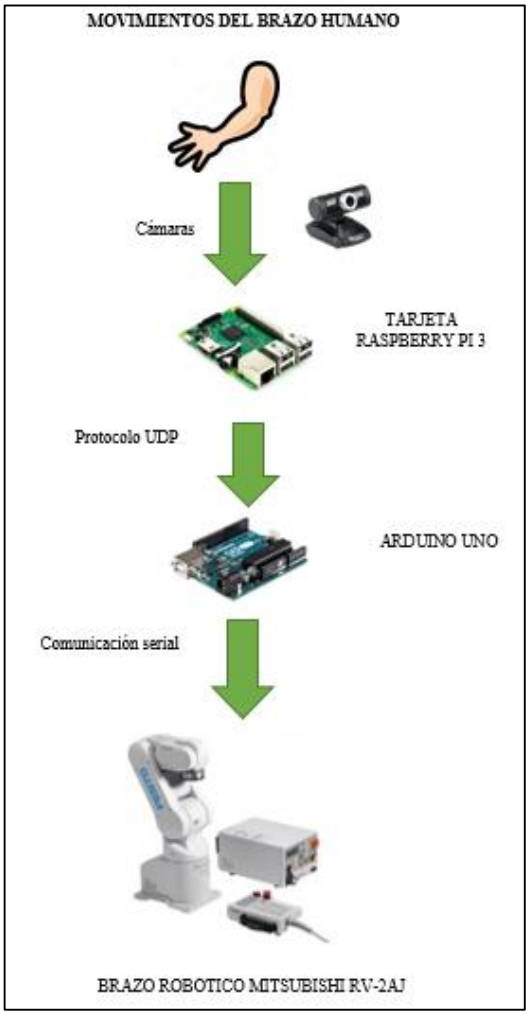
Elaborado por Liliana Asnalema

La calibración se realiza con una persona de estatura 1,50 cm, con el brazo de las siguientes medidas: de codo a hombro 20 cm, antebrazo 25 cm.

Los puntos de referencia son dos, uno en el codo y otro en la muñeca, y de color verde, el resto del brazo está forrado con un guante de color negro para facilitar el reconocimiento de los puntos.

En la figura 3.2 se observa el diagrama de conexionado. La tarjeta Raspberry Pi es la encargada de procesar las imágenes tomadas por las cámaras y enviar los datos de los centros de los puntos de referencia tomados, en pixeles. Se envía los datos mediante UDP por el puerto ethernet hacia el shield de arduino. Esta tarjeta procesa los datos recibidos, los vectoriza, realiza el proceso de lógica inversa, y envía las posiciones para el brazo.

Figura 3. 2: Diagrama de conexionado



Elaborado por Liliana Asnalema

3.2. SOFTWARE

A continuación, se describe el software desarrollado para el presente proyecto de titulación.

3.2.1. Instalación del sistema operativo en la tarjeta Raspberry

Antes de utilizar la tarjeta Raspberry, se debe instalar el sistema operativo propio que es Raspbian, en la tarjeta SD que viene en el kit de la tarjeta. El SO a escogerse es RASPBIAN JESSIE que es compatible con Matlab.

1. Se ingresa a la página oficial de Raspberry-downloads, se descarga el SO, que para la compatibilidad con Matlab es RASPBIAN JESSIE.

2. El archivo descargado es una imagen ISO, por lo tanto, se necesita un software que permita grabar esta imagen en la SD Card de la tarjeta. Se ha escogido Win 32 Disk Imager.

3. En la figura 3.3 se puede observar el sistema operativo siendo cargado en la SD Card mediante este software.

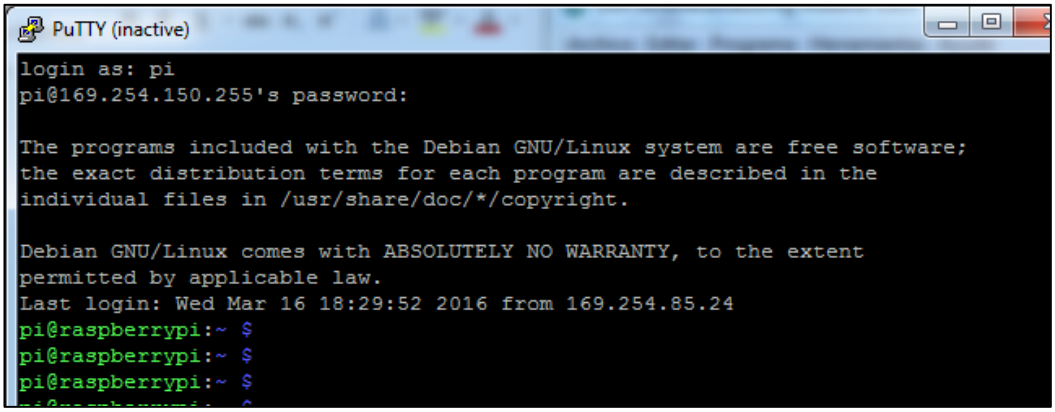
Figura 3.3: Cargando imagen a SD Card de Raspberry

Elaborado por Liliana Asnalema

Por último, se inserta la tarjeta SD en la ranura para la misma, se alimenta a la corriente eléctrica y para la comunicación, se conecta un cable de red hacia la computadora.

Para verificar la ip de la tarjeta es necesario conectar una pantalla por el puerto HDMI y un teclado por un puerto USB. Una vez obtenida la ip se puede realizar la conexión a través del terminal Putty por medio de sesión SSH, como se observa en la figura 3.4.

Figura 3.4: Raspberry Pi a través de terminal Putty



Elaborado por Liliana Asnalema

3.3. PROCESO PARA RECONOCIMIENTO DE RASPBERRY EN MATLAB

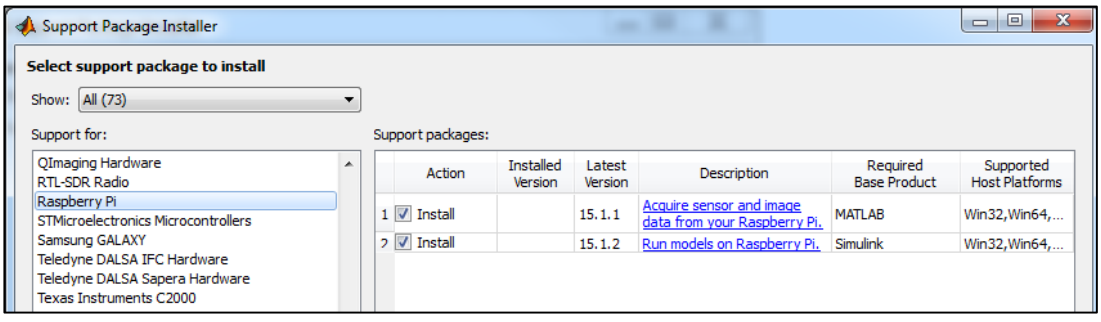
Para enlazar la tarjeta de desarrollo Raspberry Pi 3 con Matlab es necesario descargar de la página oficial de Matlab las librerías necesarias e instalarlas. A continuación, se muestra parte del proceso mencionado.

1. Ingresar en Matlab y en la barra HOME-ENVIROMENT, seleccionar la opción ADD ONS-GET HARDWARE SUPPORT PACKAGE.

2. En la siguiente pantalla seleccionar la opción INSTALL FROM THE INTERNET. Para realizar todo este proceso hay que estar enlazado en la cuenta de Mathworks.

3. En la siguiente ventana que se despliega, como se muestra en la figura 3.5 hay que seleccionar la tarjeta de la cual se desean instalar las librerías.

Figura 3.5: Seleccionar el dispositivo del cual se desean las librerías.

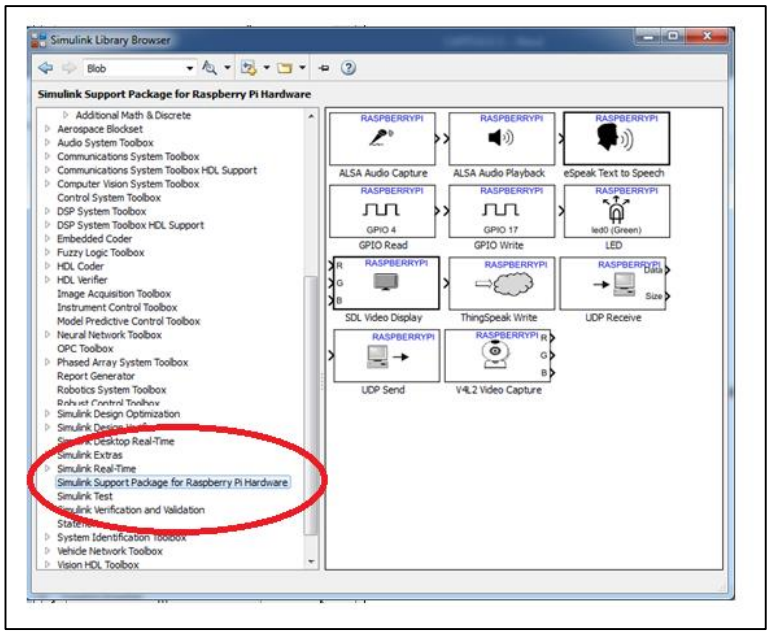


Elaborado por Liliana Asnalema

4. A continuación, se selecciona el tipo de conexión que va a tener la tarjeta con la computadora en la que esté instalado Matlab.

5. A continuación, en la figura 3.6, se verifica las librerías instaladas en SIMULINK.

Figura 3.6: Bloques de programación de Raspberry en SIMULINK



Elaborado por Liliana Asnalema

Se utilizará simulink para desarrollar el sistema de visión artificial que será procesado por la tarjeta de desarrollo Raspberry Pi 3.

3.4. BLOQUES DE PROGRAMACIÓN DE SIMULINK USADOS

En el presente proyecto técnico se utilizaron los bloques de programación de Simulink para la tarjeta Raspberry Pi 3, instalados previamente.

Luego de esto aparecerá una ventana en la cual se puede arrastrar los bloques de programación que se necesiten.

3.5. BLOQUES UTILIZADOS PARA LA PROGRAMACIÓN DE LA VISIÓN ARTIFICIAL EN LA TARJETA RASPBERRY

Para configurar a la tarjeta Raspberry Pi 3 para que realice el proceso de visión artificial, se utilizaron los siguientes bloques de las librerías previamente instaladas.

3.5.1. Matlab function

Este bloque es usado para ser configurado según las necesidades requeridas. Puede ser estructurado con varias entradas y una salida, también realiza operaciones matemáticas.

En la figura 3.7 se puede observar la definición de las entradas y salidas del bloque, así como también los parámetros de la matriz RGB y HSV. Los puntos a usarse de referencia son de color verde, se escoge este color debido a que resalta con el fondo negro del guante.

Figura 3. 7: Programación del bloque Matlab Function

```
1    %Se crean las entradas y salidas del bloque
2    function BW  = fcn(R, G, B, Hmin, Hmax, Smin, Smax, Vmin, Vmax)
3
4    %Se define una matriz para la imagen en formato RGB
5    % De tamaño 320x240x3
6    rgbImage = uint8(zeros(320,240,3));
7
8    %Primer plano para R (RED)
9    rgbImage(:,:,1) = R;
10
11   %Segundo plano para G (GREEN)
12   rgbImage(:,:,2) = G;
13
14   %Tercer plano para B (Blue)
15   rgbImage(:,:,3) = B;
16
17   %Se transforma a formato HSV
18   hsvImage = rgb2hsv(rgbImage);
19
20   %Primer plano para H(HUE)
21   hImage = hsvImage(:,:,1);
22
23   %Segundo plano para S(SATURATION)
24   sImage = hsvImage(:,:,2);
25
26   %Tercer plano para V(VALUE)
27   vImage = hsvImage(:,:,3);
```

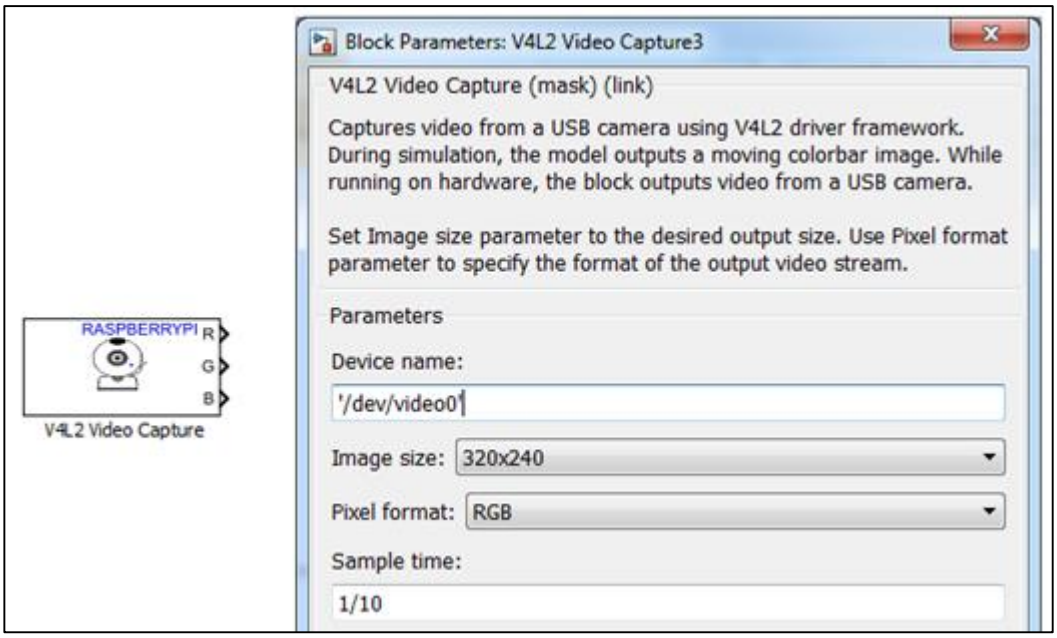
Elaborado por Liliana Asnalema

3.5.2. V4L2 Video Capture

En el área de programación de Simulink debe ser colocado el bloque VLS2 Video Capture, que es el encargado de reconocer las cámaras que se conectan los puertos USB de la tarjeta Raspberry.

En la figura 3.8 se muestra el bloque de programación y sus parámetros. El primer parámetro a configurarse es el nombre con el que se guardará a la cámara conectada, el segundo es el tamaño de la imagen a tomarse, para el presente proyecto se ha escogido 320x240 pixeles. También se puede configurar el formato de la imagen, que para nuestro caso es RGB.

Figura 3.8: Bloque de programación V4L2 Video Capture



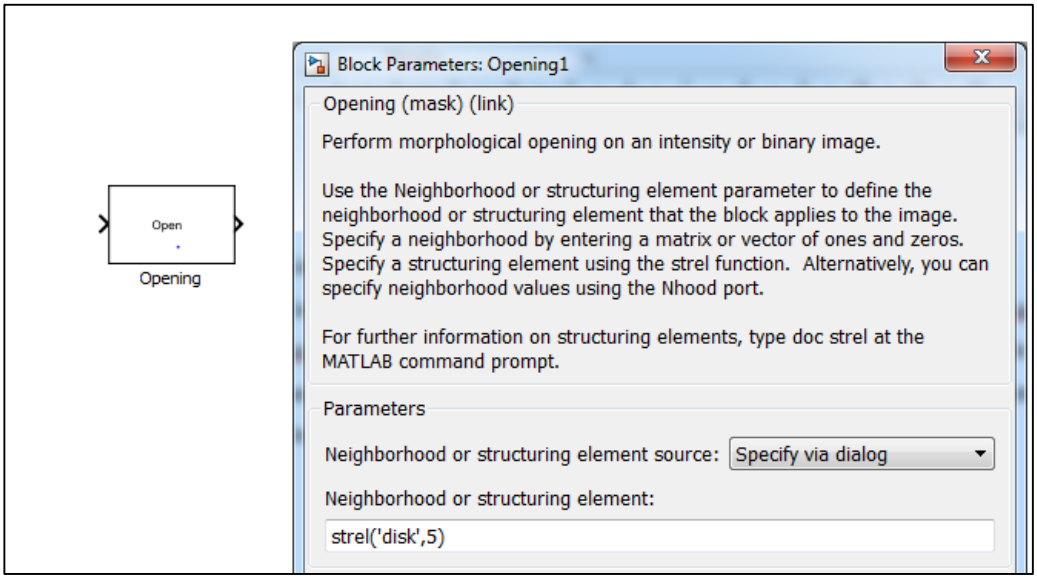
Elaborado por: Liliana Asnalema

3.5.3. Bloque de programación OPENING

Este bloque de programación se encuentra dentro de la librería Computer vision system toolbox / Morphological Operations de Simulink. Realiza una operación morfológica de filtrado a una imagen capturada. Su función es la de retirar los objetos pequeños manteniendo la forma y el tamaño de los objetos más grandes, tiende a eliminar algunos de los píxeles de primer plano es decir los objetos brillantes.

En la figura 3.9 se pueden visualizar las configuraciones para este bloque, en el presente proyecto técnico se usaron por default ya que los únicos parámetros son: el del área a filtrarse y la fuente.

Figura 3. 9: Bloque de programación OPENING



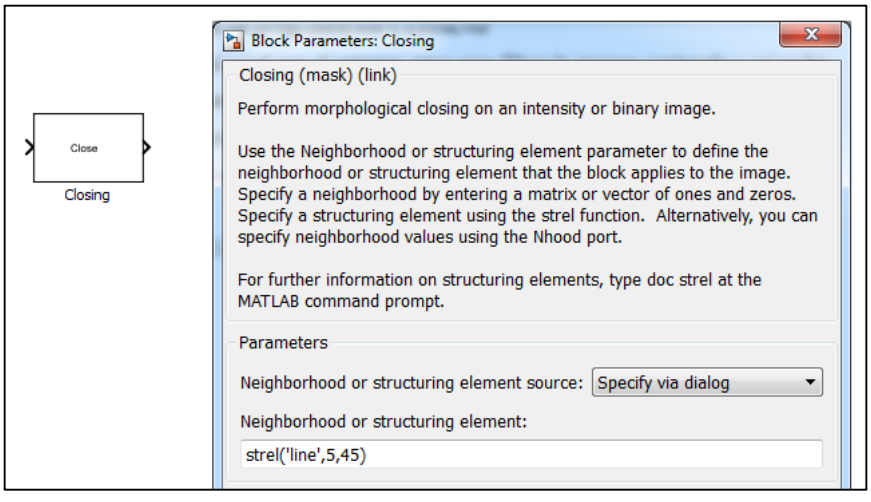
Elaborado por: Liliana Asnalema

3.5.4. Bloque de programación CLOSING

Este bloque, al igual que el anterior, sirve para filtrar la imagen capturada y así poder definir los puntos de interés. La función principal es la de agrandar los límites de la región de primer plano en la imagen capturada y reducir los agujeros de color de fondo en dichas regiones.

En la figura 3.10 se puede observar los parámetros a configurarse del bloque closing. Se utilizan los parámetros por defecto ya que seleccionan la zona de la imagen capturada

Figura 3.10: Bloque de programación CLOSING

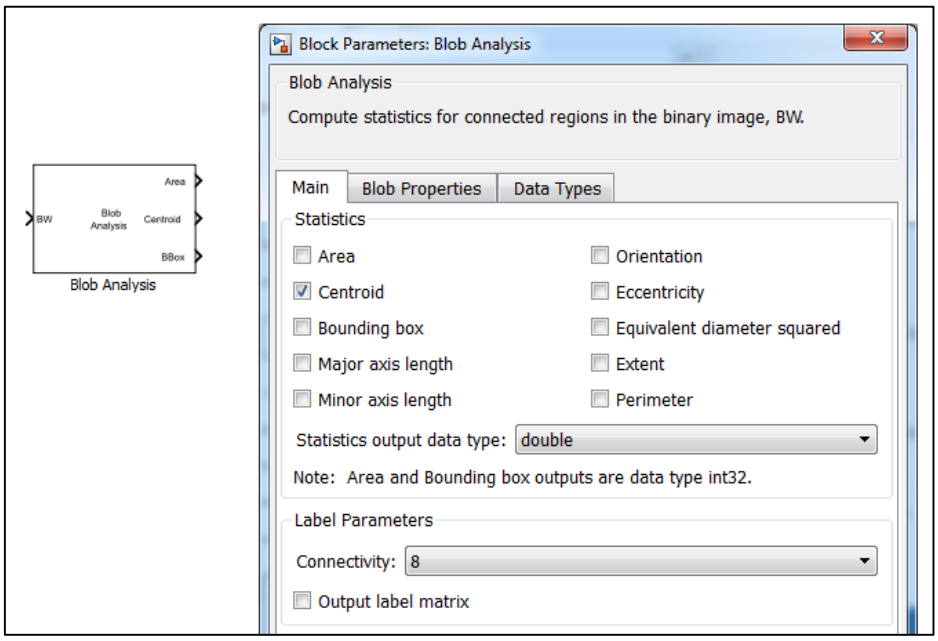


Elaborado por: Liliana Asnalema

3.5.5. Bloque Blob Analysis

Este bloque se encuentra en la librería Computer vision system toolbox/Statistics. La función que desempeña en el presente proyecto de titulación es la de encontrar el centro de los puntos de referencia.

Figura 3. 11: Bloque Blob Analysis

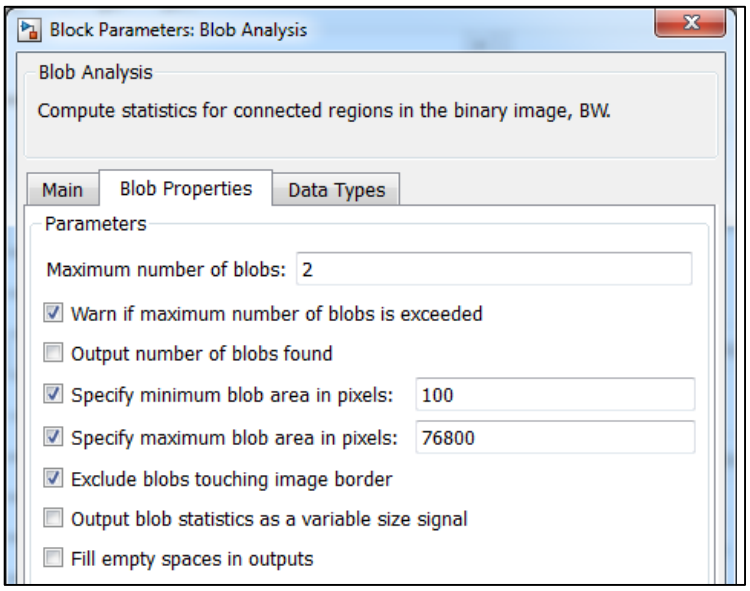


Elaborado por: Liliana Asnalema

En la figura 3.11 se observan los parámetros que han sido configurados para este bloque. Se selecciona solamente centroid en la pantalla main.

En la figura 3.12 se pueden verificar las propiedades que se han configurado, las cuales son las necesarias para que este bloque desempeñe correctamente sus funciones.

Figura 3. 12: Propiedades del bloque Blob Analysis

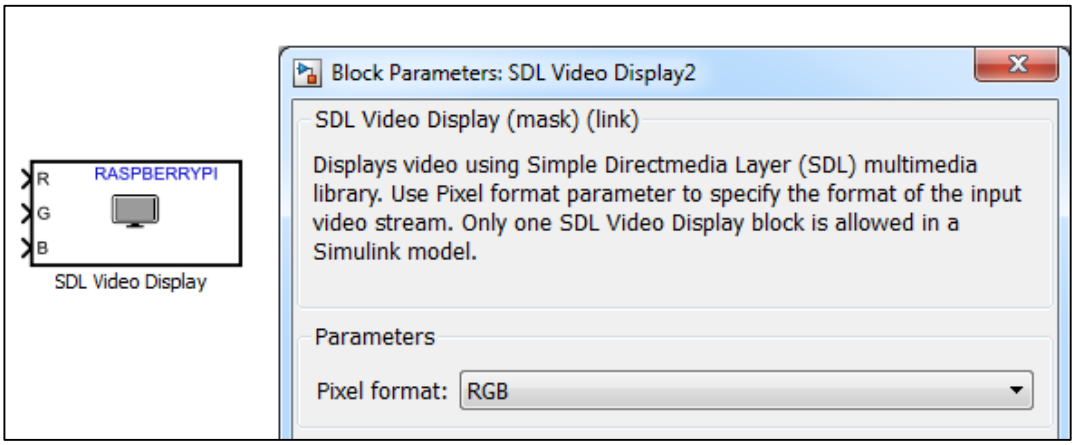


Elaborado por: Liliana Asnalema

3.5.6. SDL Video display

Este bloque se lo usa para poder visualizar imágenes. En la figura 3.13 se observa que el único parámetro a configurarse es el del tipo de formato el cual para este proyecto es RGB.

Figura 3. 13: Bloque de programación SDL Video Display



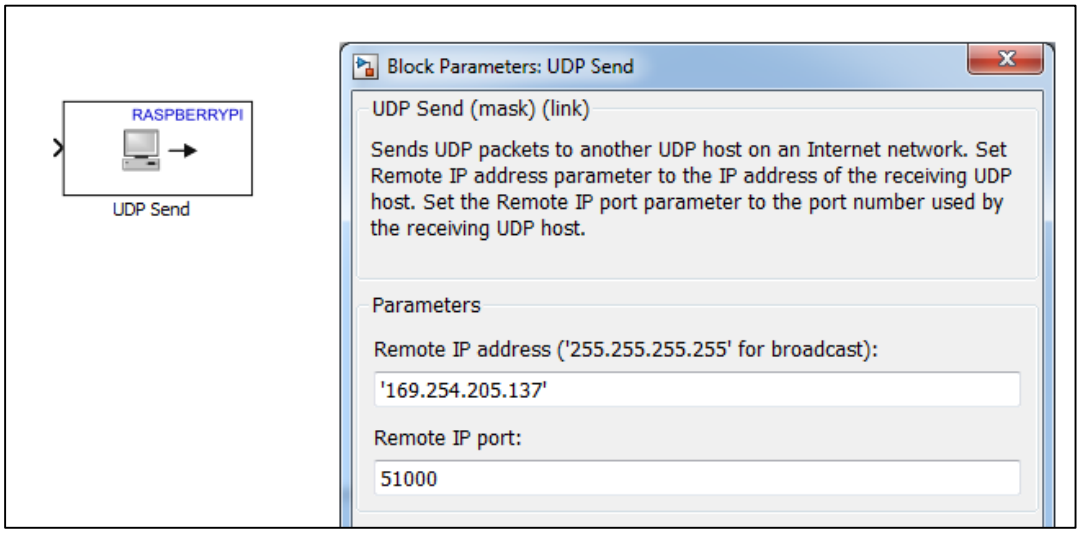
Elaborado por: Liliana Asnalema

3.5.7. Bloque RaspberryPi UDP Send

Este bloque se utiliza para enviar los datos recolectados de Matlab hacia la tarjeta que es la encargada de enviar los datos de posición al robot.

En este bloque, que se observa en la figura 3.14, se configura la dirección ip del host remoto al cual se va a conectar la tarjeta raspberry, en este proyecto se configura el shield ethernet de arduino con la ip 169.254.205.137 y el puerto para la comunicación es el 51000.

Figura 3.14: Bloque de configuración UDP send



Elaborado por: Liliana Asnalema

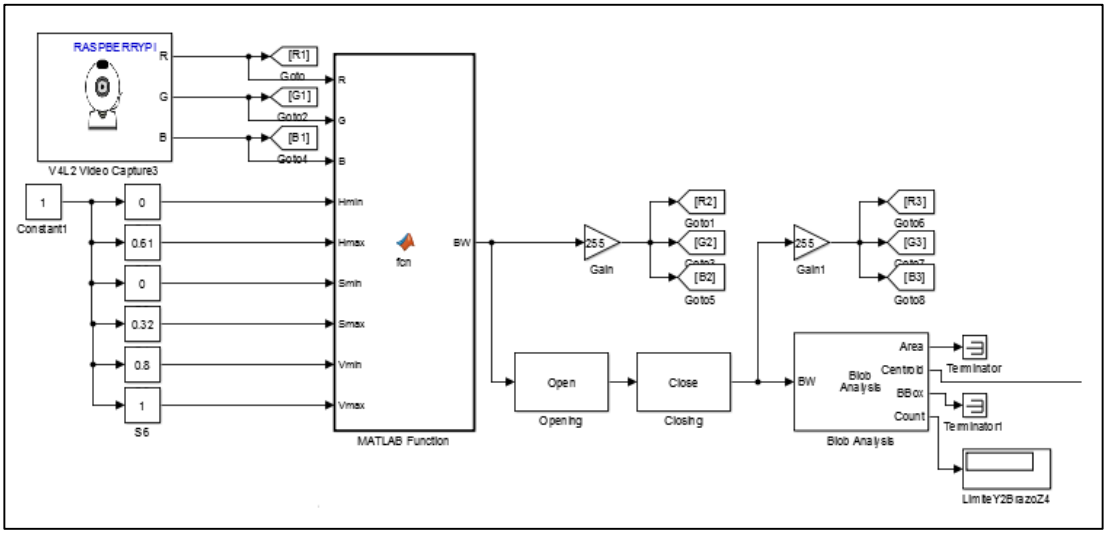
3.6. PROGRAMACIÓN DE LA TARJETA RASPBERRYPI 3

Para programar la tarjeta RaspberryPi 3, se usan los bloques anteriormente descritos, en dos programas: uno el cual será el encargado de la calibración de las cámaras, y el otro el que enviará los datos hacia Arduino.

3.6.1. Bloque de adquisición de datos y filtrado

En la figura 3.15 se puede visualizar este bloque el cual es el encargado de adquirir la imagen en formato RGB, lo transforma a HSV y filtra la misma, esto con el fin de obtener una imagen binarizada y filtrada, de la cual se pueda obtener dos objetos de referencia de los cuales se determinará su centro en pixeles.

Figura 3. 15: Bloque de adquisición y filtrado de imagen

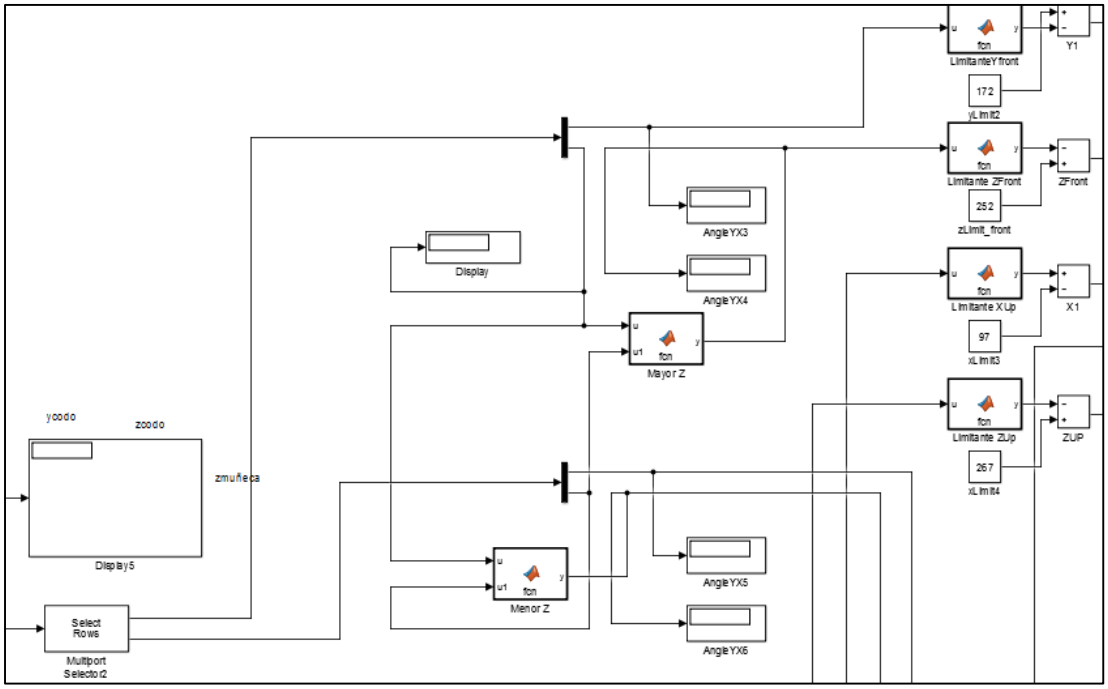


Elaborado por: Liliana Asnalema

3.6.2. Bloque de referenciado

En la figura 3.16 se observa parte del bloque de referenciado. Aquí se dan las limitaciones dentro del rango de visión de las cámaras, así como también se fijan los ejes de referencia relativos que se utilizan para calcular los puntos dentro de los planos XZ, YZ, XY.

Figura 3. 16: Bloque de referenciado



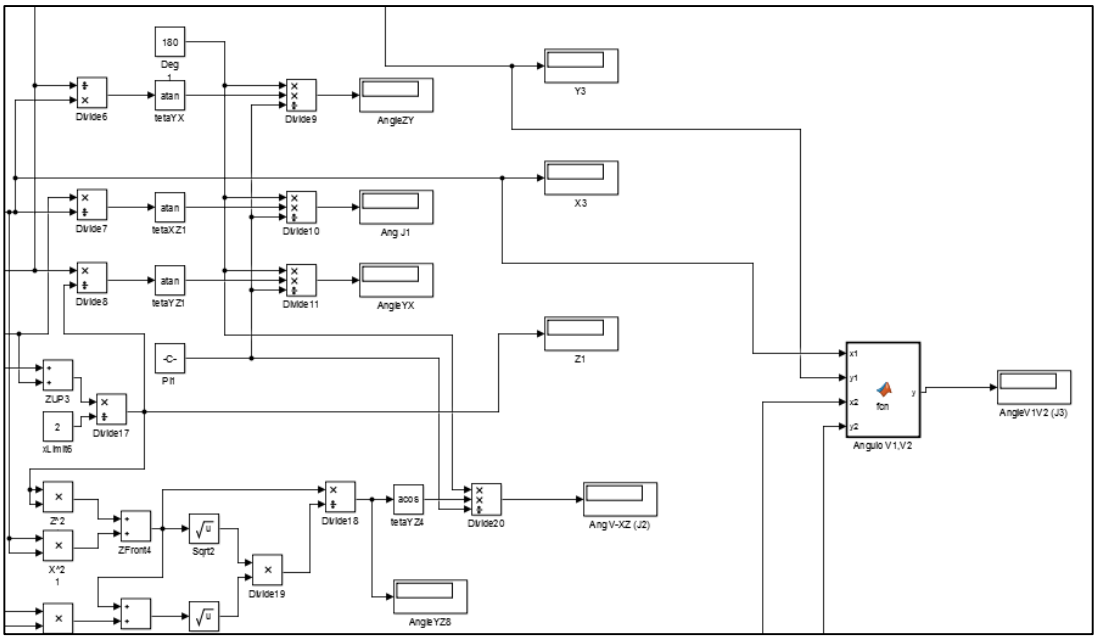
Elaborado por: Liliana Asnalema

3.6.3. Bloque de cálculo de ángulos

En la figura 3.17 se visualiza parte del bloque de cálculo de ángulos. En esta parte de la programación se realizan operaciones matemáticas para encontrar los ángulos que cada juntura debe moverse.

Para la juntura 1 (J1) del robot se tomará en cuenta el ángulo que se han desplazado el punto de referencia del codo en el plano XZ. Para la juntura 2 (J2) se realiza el producto punto entre el vector en tres dimensiones del codo y su proyección en el plano XZ. Finalmente, para la juntura 3 (J3), se calcula la pendiente de la recta que forman los dos puntos de referencia (codo y muñeca).

Figura 3. 17: Bloque de cálculo de ángulos

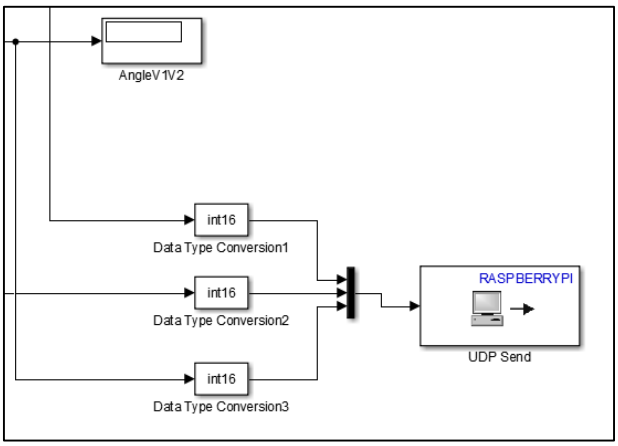


Elaborado por: Liliana Asnalema

3.6.4. Bloque de envío de datos hacia Arduino

En la figura 3.18 se puede observar el envío de los tres ángulos hacia la tarjeta arduino. Antes de enviar los datos se los transforma a enteros.

Figura 3. 18: Bloque de envío de datos



Elaborado por: Liliana Asnalema

3.7. PROGRAMACIÓN DE LA TARJETA ARDUINO UNO

La tarjeta de desarrollo Arduino uno es la encargada de recibir los datos enviados desde la tarjeta Raspberry Pi 3 mediante comunicación UDP, además envía los comandos que habilitan el movimiento de las junturas (J1, J2, J3) del robot MITSUBISHI MELFA RV-2AJ.

En la figura 3.19, se visualizan todos los parámetros utilizados para que sea factible la comunicación entre la tarjeta RASPERRY PI 3 y ARDUINO.

Figura 3. 19: Parámetros de comunicación UDP

```
//LIBRERIAS USADAS PARA LA COMUNICACION UDP
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>

//MAC SHIELD ETHERNET
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};

//IP SHIELD ETHERNET
IPAddress ip(169, 254, 205, 137);

//PUERTO PARA LA COMUNICACION UDP
unsigned int localPort = 51000;

// buffers PARA LA RECEPCION DE DATOS
byte packetBuffer[UDP_TX_PACKET_MAX_SIZE];  //BUFFER DE RECEPCION DE DATOS
byte  ReplyBuffer[] = "acknowledged";       // RESPUESTA ACK
```

Elaborado por: Liliana Asnalema

En la figura 3.20 se observa la recepción de datos en los buffers de arduino, estos datos se envían mediante comunicación serial hacia el robot MITSUBISHI MELFA RV-2AJ. Los comandos usados fueron extraídos desde el programa compatible de MITSUBISHI COSIROP.

Figura 3. 20: Datos de los ángulos

```
  //DATOS DE LOS ANGULOS CALCULADOS PARA LAS TRES JUNTURAS
  x1 =  packetBuffer[0];
  x2= 90 - packetBuffer[2];
  x3 =  packetBuffer[4];
pinMode(8, OUTPUT);
digitalWrite(8,HIGH);
  //POSICION DEL BRAZO
Serial.println("1;1;STATE");
delay(100);
Serial.println("1;1;CNTLON");
delay(100);
Serial.println("1;1;EXECJOVRD 50");
delay(100);
Serial.print("1;1;EXECJCOSIROP=(");
Serial.print("-");Serial.print(x1);
Serial.print(",");Serial.print(x2);
Serial.print(",");Serial.print("-");Serial.print(x3);
Serial.print(",");Serial.print(0);
Serial.print(",");Serial.print(0);
Serial.print(",");Serial.print(0);
Serial.println(")");
delay(100);
Serial.println("1;1;EXECMOV JCOSIROP");
delay(100);
Serial.println("1;1;CNTLOFF");
digitalWrite(8,LOW);
```

Elaborado por: Liliana Asnalema

Una vez que el robot recibe los datos desde Arduino, replicará los movimientos realizados.

CAPÍTULO 4
PRUEBAS Y RESULTADOS

En el presente capítulo se presentan las pruebas realizadas al proyecto técnico para verificar la precisión de los movimientos replicados y el tiempo de respuesta que tiene el robot. Además, se presentan las conclusiones a las que se llega y las recomendaciones para futuras investigaciones.

4.1. PRUEBA DE EXACTITUD DE LOS ÁNGULOS CALCULADOS POR LA TARJETA RASPBERRY

Para esta prueba se conecta la tarjeta Raspberry Pi 3 a la computadora y se ejecuta el programa de calibración con ello se puede visualizar los ángulos que está calculando y enviando hacia Arduino.

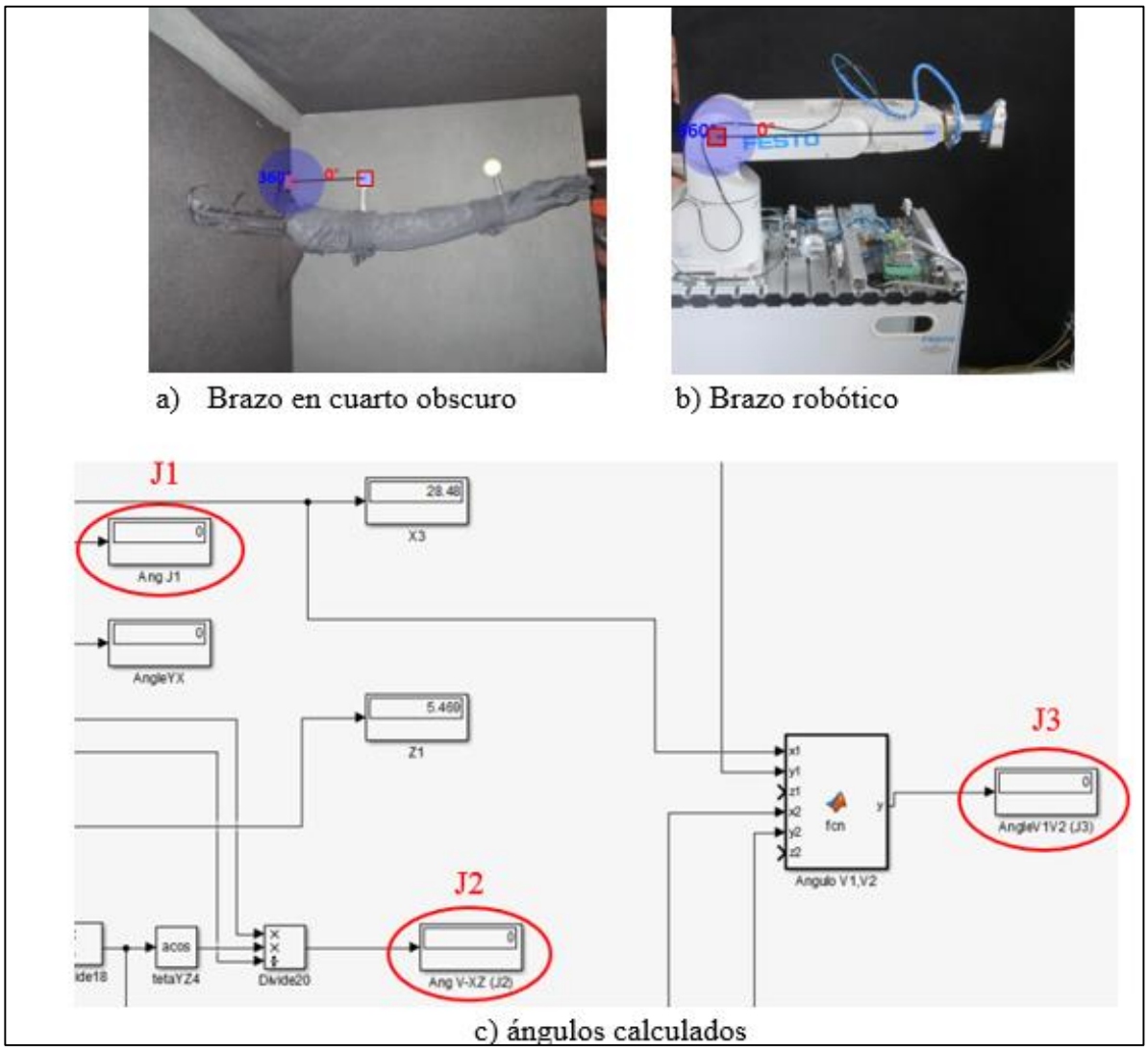
Se toman cuatro posiciones del brazo para realizar la prueba, una en la posición X, otra en la posición Y, otra en la posición Z y una en una posición en el espacio. De estas posiciones se toman los grados calculados por la tarjeta Raspberry Pi 3 y también se mide físicamente con un graduador los ángulos del brazo, se realiza una comparación para verificar el error que tiene.

Para la medición de ángulos se toman fotos de algunas posiciones y se miden los ángulos con el programa GEOGEBRA, el cual permite medir ángulos en fotografías. No se puede realizar las mediciones directamente con un graduador dentro del cuarto obscuro debido a que la luz refleja sobre el mismo lo que es sensado por las cámaras y podría darse un error en los centroides que calcula la tarjeta Raspberry.

4.1.1. Posición de prueba 1

En la figura 4.1a, se puede observar el brazo dentro del cuarto obscuro en la posición de prueba 1. Los ángulos desplazados en los tres grados de libertad son cero. Es decir, la tarjeta Raspberry Pi 3 envía lo siguiente: J1:0, J2:0, J3:0 tal como se puede observar en la figura 4.1c. En la figura 4.1b, se observa al Robot con sus junturas: J1:0, J2:90, J3:0.

Figura 4. 1: Brazo en posición en X

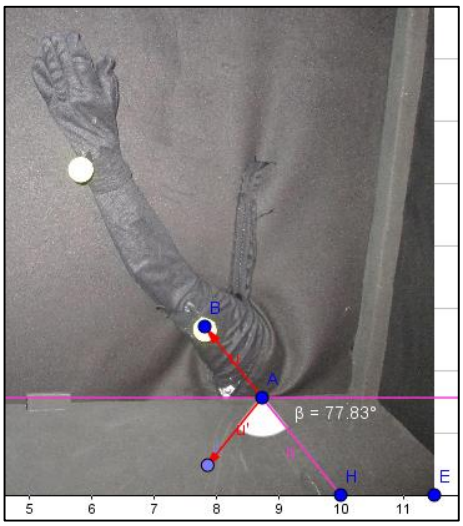


Elaborado por Liliana Asnalema

4.1.2. Posición de prueba 2

En la figura 4.2 se puede observar el brazo en el cuarto obscuro en la posición de prueba 2. En esta imagen se ha medido en ángulo que se ha tomado de referencia para la juntura 1. El cual es 77, 83º

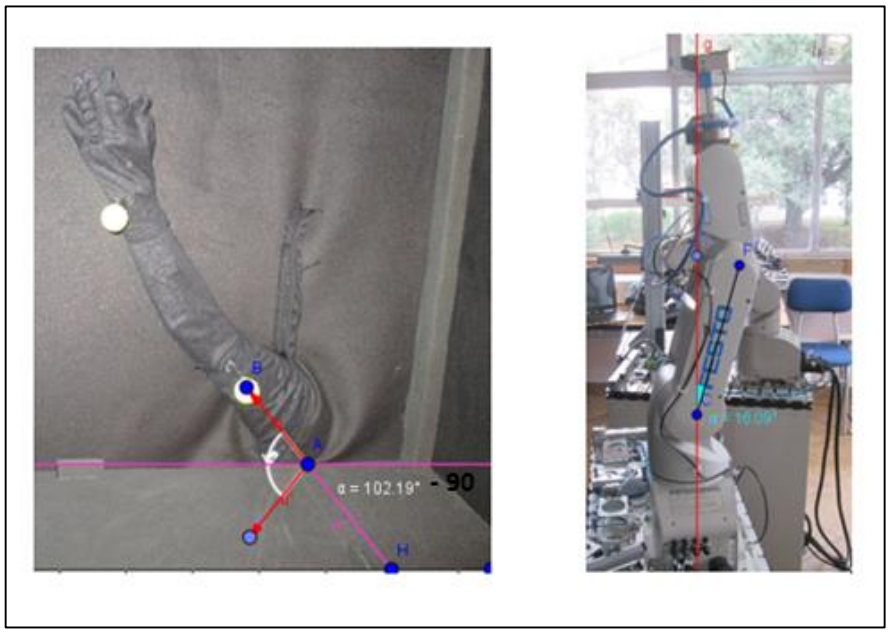
Figura 4. 2: Brazo en posición de prueba 1



Elaborado por Liliana Asnalema

En la figura 4.3a se puede observar el brazo dentro del cuarto obscuro, con Geogebra se realiza la medición del ángulo de elevación que se ha tomado en cuenta para la juntura 2 del robot. Cabe recalcar que la medición se realizó con la proyección del vector A del codo sobre el plano XZ, el ángulo que forman entre los dos es el que será enviado a la tarjeta arduino, la cual restará 90º por la posición y la referencia del robot.

Figura 4. 3:Brazo en posición de prueba 2(Juntura 2)



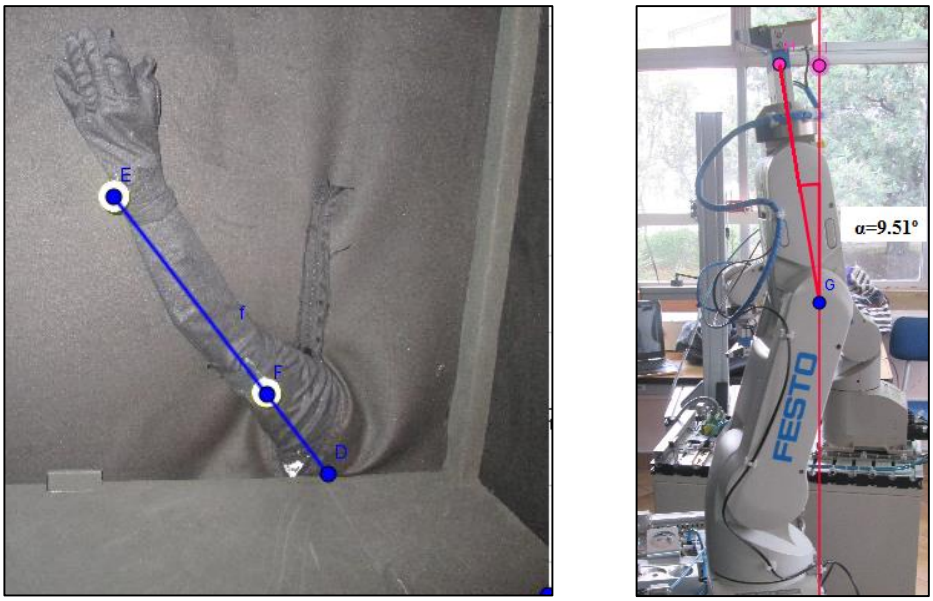
a) Medición del ángulo de elevación J2 b) ángulo de J2 en el Robot

Elaborado por Liliana Asnalema

En la figura 4.3b, se observar el ángulo medido en el Robot el cual es 16,09º.

En la figura 4.4a se puede observar que los dos puntos de referencia son colineales, por lo tanto, no existe ningún ángulo de inclinación entre ellos, es decir que el ángulo de la juntura 3 del brazo robótico es 0º

Figura 4. 4: Posición de prueba 2(Juntura 3)



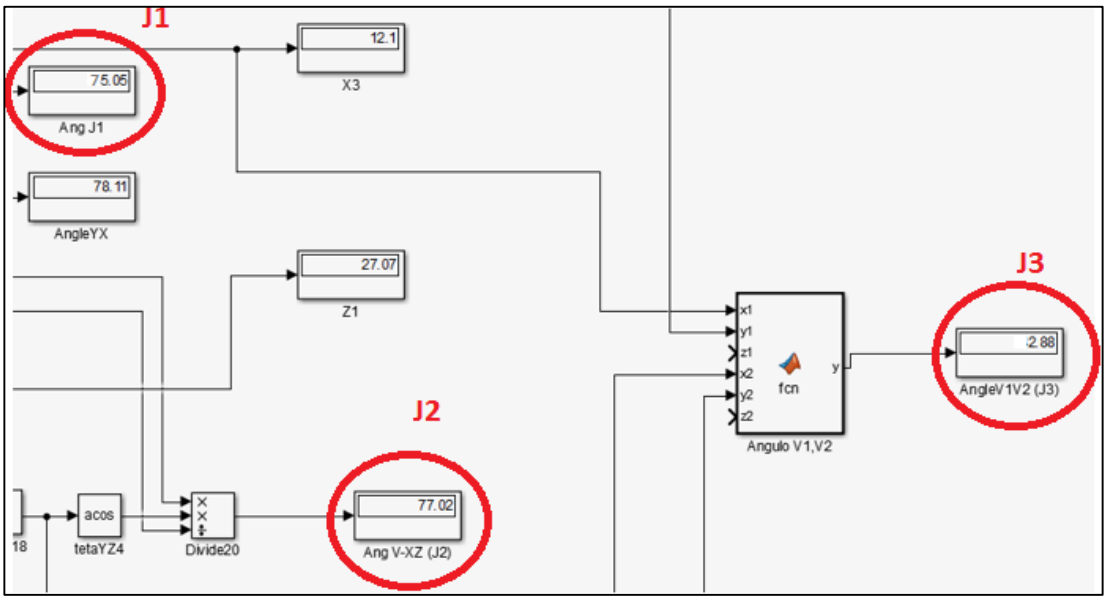
a) Brazo dentro del cuarto obscuro b) Brazo robótico

Elaborado por Liliana Asnalema

En la figura 4.4b, se observa el ángulo que ha descrito el robot que es 9, 51º

En la figura 4.5 se pueden observar los tres ángulos calculados por la tarjeta Raspberry, los cuales serán enviados hacia Arduino.

Figura 4. 5: Ángulos calculados en la tarjeta Raspberry Pi 3



Elaborado por Liliana Asnalema

- **Cálculo del error JUNTURA 1 (J1)**

En la tabla 4.1 se puede observar todas las mediciones realizadas junto con su valor calculado y su error que en el caso de la juntura 1 es de 5.04%

Tabla 4. 1: Cálculo del error para la juntura 1

Posición	Ángulo medido (grados)	Ángulo calculado (grados)	Error (%)
1	0	0	0
2	15,65	13,63	12,91
3	19,98	20,32	1,70
4	23,56	24,48	3,90
5	32,9	30,1	8,51
6	46,12	44,67	3,14
7	52,38	55,32	5,61
8	65,34	63,98	2,08
9	77,83	75,62	2,84
10	89,51	88,5	1,13
PROMEDIO DEL ERROR			**5,04 %**

Elaborado por Liliana Asnalema

- **Cálculo del error JUNTURA 2 (J2)**

En la tabla 4.2 se puede observar todas las mediciones realizadas junto con su valor calculado y su error que en el caso de la juntura 1 es de 7.93%

Tabla 4. 2: Cálculo del error para la juntura 2

Posición	Ángulo medido (grados)	Ángulo calculado (grados)	Error (%)
1	90	90	0,00
2	87,63	85,34	2,61
3	74,3	72,21	2,81
4	62,12	59,54	4,15
5	55,28	52,19	5,59
6	50,12	47,24	5,75
7	46,23	44,12	4,56
8	32,89	30,16	8,30
9	23,45	22,13	5,63
10	12,19	16,09	31,99
PROMEDIO DEL ERROR			**7,93 %**

Elaborado por Liliana Asnalema

- **Cálculo del error JUNTURA 3 (J3)**

En la tabla 4.3 se puede observar todas las mediciones realizadas junto con su valor calculado y su error que en el caso de la juntura 1 es de 9.86%

Tabla 4. 3: Cálculo del error para la juntura 3

Posición	Ángulo medido (grados)	Ángulo calculado (grados)	Error (%)
1	0	0	0,00
2	10,9	12,32	13,03
3	23,87	27,34	14,54
4	35,89	36,87	2,73
5	47,6	50,32	5,71
6	56,23	60,65	7,86
7	60,78	67,12	10,43
8	69,65	72,3	3,80
9	72,35	85,98	18,84
10	80,01	89,45	11,80
PROMEDIO DEL ERROR			**9,86 %**

Elaborado por Liliana Asnalema

El promedio de las tres junturas 7.61%

4.2. PRUEBAS DE TIEMPO DE RESPUESTA

En esta prueba se mide el tiempo de respuesta que tiene el robot con respecto al movimiento del brazo humano en el cuarto obscuro. Se toman diez muestras, y se calcula el tiempo promedio.

De la tabla 4.4, se puede obtener que el tiempo de respuesta promedio es 4.23 segundos.

Tabla 4. 4: Tiempo de respuesta de 10 posiciones

Nro.	TIEMPO DE RESPUESTA (s)
1	4.9
2	4.8
3	4.5
4	3.5
5	3
6	4.1
7	4.4
8	3.9
9	4.7
10	4.5
PROMEDIO	**4.23**

Elaborado por Liliana Asnalema

4.3. MEDICIÓN DE LUXES

Para la calibración de la intensidad de luz dentro del cuarto obscuro se realizó un proceso de prueba y error, utilizando el programa de calibración y para determinar cuanta luz requiere el sistema. Se mide con un luxómetro la intensidad de luz en diferentes puntos dentro del cuarto obscuro. En la figura 4.6 se observa la medición de luxes en el eje z.

Figura 4. 6: Luxes en el eje z

Elaborado por Liliana Asnalema

En la figura 4.7 se observa la medición en el eje y.

Figura 4. 7:Luxes en el eje y

Elaborado por Liliana Asnalema

En la figura 4.8. se observa la medición de luxes en el centro del plano XZ

Figura 4. 8:Luxes en el plano XZ

Elaborado por Liliana Asnalema

CONCLUSIONES

Se desarrolló un sistema de visión artificial capaz de reconocer los movimientos de un brazo humano para replicarlo en el brazo robótico MITSUBISHI al construir un cuarto obscuro en donde se detectaron dos puntos de referencia para procesar su posición, realizar cálculos de los ángulos descritos y enviar los comandos correspondientes hacia el robot para su ejecución.

Se implementó el software de visión artificial en una tarjeta de desarrollo Raspberry Pi3, a través de Matlab para la captura de imágenes y su posterior procesamiento ya que se desarrollaron algoritmos de tratamiento de imágenes en Simulink usando las librerías instaladas para Raspberry.

Se estableció la comunicación entre la tarjeta de desarrollo Arduino y el brazo robótico MITSUBISHI para el envío de comandos de movimientos, para ello se usaron las librerías de la comunicación serial propia de la tarjeta, que junto con los comandos recopilados del programa COSIROP perteneciente al fabricante del Robot, hicieron posible el intercambio de datos entre ellos.

Se ejecutaron pruebas de campo para recoger datos, validar la efectividad del controlador y analizar los resultados ya que se realizaron comparaciones entre los ángulos medidos del brazo humano en el cuarto obscuro y los ángulos que el robot describe, mediante el software Geogebra, con lo cual se obtuvo el 7.96% de error entre el movimiento real del brazo humano y el del robot, además, se midió el tiempo promedio de respuesta en que el robot Mitsubishi empieza a replicar los ángulos, el cual fue de 4,23 segundos, este retardo se debe a todo el proceso que tiene que realizar la tarjeta.

RECOMENDACIONES

Para realizar las pruebas de transmisión de datos con arduino se debe tomar en cuenta que los jumpers del shield serial estén en modo down, ya que arduino trabaja en sus pines RX/TX con niveles lógicos TTL (5V o 3.3V) y el puerto serial RS232 del robot trabaja con voltajes de +/- 12V, lo que puede causar daños en la tarjeta si se conectan los jumpers en modo UP.

Se puede aumentar los grados de libertad del proyecto utilizando más puntos de referencia y cálculos aritméticos más elaborados.

Este tipo de proyectos es mejor realizarlo con encoders absolutos o usando un sensor Kinect, ya que con el sistema de visión artificial existen muchas limitaciones, debido al rango de visión de las cámaras, a la luminosidad y a los grados de libertad que puede tener el brazo.

REFERENCIAS BIBLIOGRÁFICAS

Batz, C. (01 de Julio de 2015). *Universidad de San Carlos de Guatemala.* Recuperado el 06 de Diciembre de 2016, de http://biblioteca.usac.edu.gt/tesis/08/08_0286_CS.pdf

Cajamarca Andrés, P. J. (agosto de 2016). *Repositorio Digital UPS.* Recuperado el 2017 de junio de 2017, de http://dspace.ups.edu.ec/handle/123456789/13683

Freire Muñoz Santiago, M. C. (Enero de 2012). *Repositorio Digital UPS.* Recuperado el 16 de Marzo de 2017, de dspace: http://dspace.ups.edu.ec/bitstream/123456789/1627/6/UPS-ST000928.pdf

Guaranga, R. (Octubre de 2014). *REPOSITORIO UTE.* Recuperado el 15 de Marzo de 2017, de http://repositorio.ute.edu.ec/bitstream/123456789/5791/1/58769_1.pdf

Lozano Germán, O. J. (23 de Enero de 2015). *Repositorio UAC.* Recuperado el 21 de Marzo de 2017, de http://repositorio.uac.edu.co/bitstream/handle/11619/1371/TMCT%200013C.pdf?sequence=1&isAllowed=y

Mena Lisbeth, S. P. (Febrero de 2012). *Repositorio Digital UPS.* Recuperado el 16 de Marzo de 2017, de dspace.ups.edu.ec/bitstream/123456789/1794/7/UPS-ST000865.pdf

RaspberryPi. (04 de 2014). Recuperado el 22 de Octubre de 2016, de https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Reyes, F. (2016). *ROBÓTICA:CONTROL DE MANIPULADORES.* México D.F: Alfaomega. Recuperado el 21 de 04 de 2017, de https://books.google.com.ec/books?hl=es&lr=&id=cULVDQAAQBAJ&oi=fnd&pg=PT5&dq=manipuladores&ots=LP6Jtou_8Y&sig=bZTzeVcuZP__i-nzWYdbV4nuaTI&redir_esc=y#v=onepage&q=manipuladores&f=false

Segovia, J., Alamilla, M., & Domínguez, J. F. (Noviembre de 2015). *Universidad Autónoma del Estado de Hidalgo.* Recuperado el 14 de Febrero de 2016, de https://repository.uaeh.edu.mx/bitstream/handle/123456789/10529

Upton, E. (02 de Febrero de 2015). *RaspberryPi*. Recuperado el 12 de 06 de 2016, de https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/