

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**TEMA:
DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA EL
RECONOCIMIENTO, CLASIFICACIÓN Y MAQUINADO DE PATRONES
CON UNA TARJETA ARM**

**AUTORES:
LUIS GONZALO ANDRADE FIERRO
LUIS ALFONSO CHULCA SIMBAÑA**

**TUTORA:
CARMEN JOHANNA CELI SÁNCHEZ**

Quito, marzo del 2018

CESIÓN DE DERECHOS DE LOS AUTORES

Nosotros, Luis Gonzalo Andrade Fierro y Luis Alfonso Chulca Simbaña, con documento de identidad N° 1720738721 y N° 1723723019 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO, CLASIFICACIÓN Y MAQUINADO DE PATRONES CON UNA TARJETA ARM”, mismo que ha sido desarrollado para optar por el título de INGENIEROS ELECTRÓNICOS , 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 nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Luis Gonzalo Andrade Fierro
C.I. 1720738721



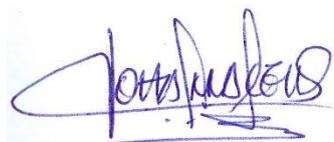
Luis Alfonso Chulca Simbaña
C.I. 1723723019

Quito, marzo del 2018.

DECLARATORIA DE COAUTORÍA DE LA DOCENTE TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Trabajo de Titulación “DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO, CLASIFICACIÓN Y MAQUINADO DE PATRONES CON UNA TARJETA ARM”, realizado por Luis Gonzalo Andrade Fierro y Luis Alfonso Chulca Simbaña, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, marzo del 2018.



Carmen Johanna Celi Sánchez

C.I. 1717437808

DEDICATORIA

A Dios, por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional. A mi madre, Victoria, por ser el pilar fundamental y por demostrarme siempre su cariño y apoyo desinteresado. A mi padre Luis, por su apoyo incondicional, que a pesar de ya no tenerlo a mi lado, sé que este momento hubiera sido tan especial y de enorme alegría para ti, como lo es para mí.

Luis Chulca

A mi hermana Mariela Andrade Fierro:

Siempre serás mi gran orgullo, estoy seguro que lograrás llegar tan alto como te propongas en tu vida profesional. Con este modesto logro pretendo brindarte a cambio, un poco de toda esa admiración que siento hacia ti. Te amo mí, eternamente, hermanita pequeña.

A mi hijo Luis Felipe Andrade Villalba:

Deseo, hijo mío, que cuando seas capaz de leer esto conozcas que con éste trabajo es mi voluntad el alentarte a forjar tu camino en la ciencia y la razón, para que alcances mejores y más altos logros profesionales, en bien tuyo, de la humanidad y para orgullo de tu familia.

A mi abuelo Gonzalo Fierro:

En tu memoria, padre de la familia, ejemplo, respeto y admiración de todos quienes te conocieron; mi primer mentor, espero de todo corazón haber sido digno de representar parte de tu recuerdo.

Luis Andrade

AGRADECIMIENTO

Le agradezco a Dios por haberme guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos difíciles y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Doy gracias a mis padres, Luis Chulca Casillas y Victoria Simbaña Pillajo por apoyarme en todo momento, por los valores inculcados, y por haberme brindado la oportunidad de tener una excelente educación en el transcurso de mi vida.

A mis hermanos Esperanza, Francisco y Paul por ser parte importante de mi vida, brindándome su apoyo en cada momento bueno o malo.

Luis Chulca

A mis padres Zobeyda Fierro Castro y Luis Andrade Álvarez, por el apoyo incondicional que me han brindado durante todos mis estudios, por guiarme con sus consejos y con su gran ejemplo, por siempre confiar en mí, y en especial, por todo el amor y comprensión que han derramado sobre mi vida. Sin ustedes éste logro no sería posible, logro que con humildad lo comparto con todos los miembros de mi familia.

A mi esposa Mayra Villalba Armendáriz, por toda la ayuda, paciencia y lealtad que me has demostrado, por ser el soporte que supiste dar a mi vida durante la elaboración del presente trabajo, y por todo el amor que me compartes en el día a día de nuestras vidas. Gracias por ser mi complemento. Te amo.

Luis Andrade

ÍNDICE

CESIÓN DE DERECHOS DE AUTOR.....	i
DECLARATORIA DE COAUTORÍA DE LA DOCENTE TUTORA	ii
DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE	v
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE TABLAS	xi
RESUMEN.....	xii
ABSTRACT.....	xiii
INTRODUCCIÓN	xiv
CAPÍTULO 1.....	1
ANTECEDENTES	1
1.1. Tema del proyecto	1
1.2. Justificación.....	1
1.3. Planteamiento del problema	1
1.4. Delimitación del problema	2
1.5. Objetivos	2
1.5.1. Objetivo general	2
1.5.2. Objetivos específicos.....	2
CAPÍTULO 2.....	4
MARCO TEÓRICO	4
2.1. Arquitectura ARM.....	4
2.1.1. Historia	4
2.1.2. Raspberry Pi	5
2.1.3. Raspberry pi3 modelo b	6
2.2. Visión artificial.....	7

2.2.1. Proceso de visión artificial	8
2.2.2. Formación de imágenes digitales	9
2.2.3. Extracción de regiones en una imagen digital.....	10
2.2.4. Técnica de Binarización por Detección de Umbral	10
2.2.5. Extracción de líneas y regiones	11
2.2.6. Propiedades métricas	11
2.2.7. Momentos invariantes de hu	12
2.2.8. Cámara digital	14
2.2.9. Cámara Logitech C170.....	14
2.3. Redes Neuronales Artificiales	15
2.3.1. Estructura de una red neuronal artificial	16
2.4. Robótica	18
2.4.1. Leyes de la robótica.....	19
2.5. Clasificación de los manipuladores robóticos	19
2.6. Brazo robot Mitsubishi RV – 2AJ.....	20
2.6.1. Características del robot RV-2AJ.....	20
2.7. Matlab.....	21
2.8. Simulink	22
2.9. Arduino.....	22
2.9.1. Características de la arduino mega.....	23
CAPÍTULO 3.....	24
DISEÑO E IMPLEMENTACIÓN	24
3.1. Programación de tarjetas de desarrollo	24
3.2. Programación de la Raspberry Pi 3	25
3.2.1. Paquete de soporte de Raspberry Pi para Simulink.....	25
3.2.2. Modelo programado en Simulink.....	27
3.2.3. Etapa de visión artificial.....	27

3.2.4. Etapa de adquisición de imágenes.....	27
3.2.5. Etapa de extracción de características	28
3.2.6. Reconocimiento.....	30
3.2.7. Red neuronal	31
3.2.8. Despliegue	35
3.2.9. Etapa de envío de datos	36
3.3. Programación de la Arduino Mega	37
3.3.1. Recepción de datos desde la Raspberry Pi 3	37
3.3.2. Envío de instrucciones para el movimiento del brazo Mitsubishi	38
3.4. Implementación de hardware	39
3.4.1. Mesa de trabajo	40
3.4.2. Brazo de cámara Logitech C170	40
3.4.3. Caja de tarjetas de control	41
3.4.4. Elaboración de modelos a maquinar	42
CAPÍTULO 4.....	44
PRUEBAS Y RESULTADOS	44
4.1. Pruebas de adquisición de características.....	44
4.1.1. Tiempos de respuesta de adquisición de imágenes	44
4.2. Pruebas de reconocimiento.....	45
4.2.1. Aciertos y desaciertos en el reconocimiento	46
4.2.2. Cálculo del error en el reconocimiento	47
4.3. Pruebas de maquinado.....	48
4.3.1. Obtención de las coordenadas de maquinado.....	48
4.3.2. Cálculo de error en el maquinado	50
CAPÍTULO 5.....	52
CONCLUSIONES Y RECOMENDACIONES.....	52
5.1. Conclusiones	52

5.2. Recomendaciones.....	54
REFERENCIAS BIBLIOGRÁFICAS.....	55

ÍNDICE DE FIGURAS

Figura 2.1 Acorn Archimedes	4
Figura 2.2 Apple Newton	5
Figura 2.3 Raspberry Pi 3	6
Figura 2.4 Proceso de Visión Artificial.....	8
Figura 2.5 Captura de una Imagen	9
Figura 2.6 Imagen Digital	9
Figura 2.7 Convención de Ejes	10
Figura 2.8 Binarización de objetos.....	11
Figura 2.9 Elementos de una cámara digital	14
Figura 2.10 Logitech C170	15
Figura 2.11 Red neuronal monocapa	17
Figura 2.12 Red neuronal multicapa	17
Figura 2.13 Red neuronal recurrente.....	18
Figura 2.14 Mitsubishi RV-2AJ.....	20
Figura 2.15 PID en Simulink	22
Figura 3.1 Diagrama de Comunicaciones	24
Figura 3.2 Get Hardware Support Packages	25
Figura 3.3 Select support package	26
Figura 3.4 Simulink Library Browser	26
Figura 3.5 Modelo principal en Simulink	27
Figura 3.6 Adquisición de imágenes.....	28
Figura 3.7 Extracción de características.....	29
Figura 3.8 Cálculo Momentos de Hu	30
Figura 3.9 Neural Network Start.....	31
Figura 3.10 Network Architecture	32
Figura 3.11 Algorithms and progress.....	33
Figura 3.12 Reconocimiento de figura.....	33
Figura 3.13 Etapa de reconocimiento.....	34
Figura 3.14 Etapa de despliegue	35
Figura 3.15 Ampliar imagen a desplegar	36
Figura 3.16 Etapa de envío de datos	37
Figura 3.17 Recepción de datos desde la Raspberry Pi	38

Figura 3.18 Movimiento del brazo al punto medio.....	39
Figura 3.19 Base de trabajo.....	40
Figura 3.20 Brazo de la cámara.....	41
Figura 3.21 Caja con tarjetas de control.....	41
Figura 3.22 Modelos a maquinar	43
Figura 4.1 Adquisición del centro geométrico	45
Figura 4.2 Reconocimiento de forma.....	46
Figura 4.3 Regresión para coordenada X.....	49
Figura 4.4 Regresión para coordenada Y	49
Figura 4.5 Maquinado de modelo	50

ÍNDICE DE TABLAS

Tabla 2.1 Características Raspberry Pi 3	7
Tabla 2.2 Características Logitech C170	15
Tabla 2.3 Características mitsubishi RV-2AJ	20
Tabla 2.4 Características de la Arduino Mega	23
Tabla 3.1 Permutaciones de modelos	30
Tabla 3.2 Valores que determinan la forma	34
Tabla 3.3 Instrucciones para mover brazo	38
Tabla 3.4 Mesa de trabajo	40
Tabla 3.5 Modelos elaborados para maquinar	42
Tabla 4.1 Tiempos de respuesta	44
Tabla 4.2 Pruebas de reconocimiento	45
Tabla 4.3 Error en el reconocimiento.....	47
Tabla 4.4 Errores por número de neuronas	47
Tabla 4.5 Coordenadas XY	48
Tabla 4.6 Error en la perforación en el centro geométrico	50

RESUMEN

El presente proyecto contiene el desarrollo de un sistema de Visión Artificial para dar solución al control de procesos industriales convencionales, mismos que tienen la principal desventaja de no ser autoadaptables en casos de existir diferentes patrones a maquinar.

Se realizó el estudio de la Visión Artificial que ayudó a adquirir los conocimientos básicos para desarrollar un proceso autoadaptable.

El uso de algoritmos de Visión Artificial permitió obtener información del entorno mediante una cámara web para posteriormente procesar la información obtenida en un modelo programado en Simulink de Matlab.

Con la ayuda de los Toolkit de visión artificial de Simulink, se obtuvieron las características principales de los modelos estudiados, como es el centro geométrico, además de lograr el reconocimiento de patrones utilizando una red neuronal junto al cálculo de los momentos invariantes de Hu.

Mediante el paquete de soporte de Raspberry para Simulink se compiló la programación de bloques logrando así dotar de autonomía a la tarjeta ARM, misma que procesa la información a 32 bits.

Finalmente los datos de coordenadas del centro geométrico y de forma del modelo son enviados mediante comunicación serial RS-232 al brazo robótico Mitsubishi RV-2AJ, para realizar el maquinado de los modelos haciendo una perforación en el centro geométrico de los mismos.

El sistema pudo reconocer la figura de los modelos así como sus características morfológicas; y el brazo robótico logró realizar una perforación en el centro geométrico con un error aproximado del 3%.

ABSTRACT

The present project contains the development of an Artificial Vision system to give solution to the control of conventional industrial processes, which have the main disadvantage of not self-adaptable in cases of different patterns to be machined.

The study of the Artificial Vision was carried out that helped to acquire the basic knowledge to develop a self-adapting process.

The use of Artificial Vision algorithms allowed us to obtain information from the environment through a webcam to later process the information obtained in a model programmed in Matlab Simulink.

With the help of Simulink's artificial vision toolkit, the main characteristics of the models studied were obtained, such as the geometrical center, as well as the recognition of patterns using a neural network together with the calculation of Hu's invariant moments.

Through the Raspberry support package for Simulink, the block programming was compiled, thus achieving autonomy for the ARM card, which processes the information at 32 bits.

Finally, the coordinates data of the geometrical center and the shape of the model are sent by RS-232 serial communication to the robotic arm Mitsubishi RV-2AJ, to perform the machining of the models by making a perforation in the geometrical center of the same.

The system was able to recognize the figure of the models as well as their morphological characteristics; and the robotic arm managed to make a hole in the geometric center with an error of approximately 3%.

INTRODUCCIÓN

El control de procesos en la industria tiene como problemática constante la falta de adaptabilidad en los casos en que existen diferentes patrones a maquinar, problemática que se ha venido solucionando convencionalmente con la utilización de sistemas de clasificación previo a los maquinados, pero dichas soluciones están expuestas a sufrir gran porcentaje de error.

Con la implementación de la inteligencia artificial en la industria se puede dar soluciones más eficaces al mencionado problema, y con sistemas dotados de visión artificial se puede obtener información más específica del entorno, como características morfológicas, posiciones, materiales, colores, entre otras.

El propósito de la realización del presente trabajo fue el ofrecer una alternativa a la industria local al momento de encontrarse con procesos que tengan la problemática expuesta anteriormente, así como el brindar a los estudiantes de todas las carreras afines una base de consulta en el ámbito de la visión artificial orientada al control industrial que funcione de manera autónoma.

Se empieza con una sinopsis de todos los conocimientos básicos necesarios para el diseño y desarrollo del actual proyecto, tales como conceptos, definiciones, información técnica de los equipos y demás herramientas a utilizar.

Continuamos exponiendo la ideación y diseño del proceso a utilizar, las técnicas necesarias para la fabricación del hardware, los algoritmos utilizados para la programación del software, y por último la implementación del proyecto tanto en hardware como en software conjuntamente.

Finalmente, se presenta la toma de datos realizado para la calibración del sistema, y las pruebas efectuadas junto con los resultados obtenidos.

CAPÍTULO 1

ANTECEDENTES

1.1. Tema del proyecto

Desarrollo de un Sistema de Visión Artificial para el Reconocimiento, Clasificación y Maquinado de Patrones con la Tarjeta ARM.

1.2. Justificación

El control convencional en los procesos industriales realiza un trabajo concreto, único y repetitivo. Aplicar conocimientos de visión artificial e inteligencia artificial hace la diferencia para dar soluciones adaptativas y obtener resultados exitosos cuando en el sistema se han presentado entradas no establecidas.

Por tal motivo se requiere realizar un sistema que integre el procesamiento de imágenes con la inteligencia artificial, para satisfacer los requerimientos de la industria que solo pueden ser resueltos con el uso de control moderno.

Este sistema inicia el estudio de la visión artificial, campo que no ha sido explorado en la carrera de Electrónica y que es la base fundamental para la adquisición de información y procesamiento de imágenes.

Los sistemas que trabajan en conjunto tanto con visión como con inteligencia artificial son capaces de producir decisiones semejantes a las de un ser humano debido a la capacidad de recibir información y características del entorno, siendo una gran ventaja si se compara con sistemas de control convencionales.

El uso de la tarjeta de control Raspberry Pi3 Modelo B con arquitectura ARM tiene la ventaja del procesamiento a 32 bits, tecnología que ya se ha venido implementando en computadores de bolsillo y en teléfonos celulares, así como también en tarjetas de desarrollo y que eventualmente vendrá a reemplazar a los microcontroladores de 8 y 10 bits que se han venido utilizando hasta la actualidad.

1.3. Planteamiento del problema

El control convencional en procesos industriales tiene la desventaja de no ser autoadaptable, teniendo que el operador recalibrar y reprogramar el proceso

dependiendo de las características de los modelos a maquinar. Trabajo que resulta en una pérdida de recursos como tiempo, dinero y mano de obra.

El uso de la visión artificial permite obtener información del entorno, a la vez aplicar soluciones inteligentes a través de redes neuronales y sistemas de lógica difusa, dando respuesta al problema de falta de adaptabilidad en el maquinado y procesamiento industrial.

1.4. Delimitación del problema

El desarrollo de un sistema de visión artificial está orientado a ayudar a los investigadores en innovación en la automatización y los procesos industriales, dándoles una alternativa al momento de resolver problemas propios del control industrial.

Además permitirá a los estudiantes a desarrollar e implementar sus propios proyectos basados en la visión e inteligencia artificial, y a dotar dichos proyectos de autonomía con tarjetas de arquitectura ARM.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar un sistema de visión artificial para el reconocimiento, clasificación y maquinado de patrones con una tarjeta ARM para dar solución a procesos industriales no autoadaptables.

1.5.2. Objetivos específicos

- Adquirir conocimientos de las tecnologías de visión artificial para encontrar el proceso más idóneo en el desarrollo del sistema.
- Analizar las condiciones del entorno para solucionar la problemática de iluminación, resolución de imágenes y posicionamiento de modelos.
- Establecer un algoritmo que permita procesar imágenes bidimensionales para el reconocimiento de las características relevantes de modelos geométricos regulares.

- Comprobar el algoritmo a través de Matlab para definir su funcionamiento previo a la implementación en la tarjeta ARM.
- Implementar el proceso de visión e inteligencia artificial en una tarjeta de arquitectura ARM para que el sistema sea autónomo e integrar las características de auto adaptabilidad.
- Adecuar un sistema robótico que trabaje en conjunto con la visión artificial para el maquinado de los modelos previamente reconocidos.
- Validar el sistema mediante pruebas en diferentes condiciones para un posterior análisis de los datos obtenidos en dichas pruebas.

CAPÍTULO 2

MARCO TEÓRICO

2.1. Arquitectura ARM

2.1.1. Historia

En los años 80 la compañía ACORN COMPUTERS tenía su famoso CBC MICRO¹, pero pensando en la creación de sus nuevos modelos no les satisfacían ninguno de los procesadores existentes en esa época, por lo que decidieron crear un microprocesador a partir del proyecto RISC² de la Universidad de Berkeley, desarrollando así en 1985 el primer ACORN RISC MACHINE o conocido por sus iniciales ARM, este primer procesador no se comercializa, sino más bien serviría como base para la creación del ARM2, un procesador con algunas mejoras de rendimiento, el cual se incorpora como CPU en el computador doméstico ACORN ARCHIMEDES que se ve en la Figura 2.1, misma que salió a la venta a mediados de 1987. (Caprile, 2012)

Figura 2.1 Acorn Archimedes



Computador diseñado con un procesador ARM2, Fuente: (Caprile, 2012)

¹ Uno de los primeros computadores personales, fabricado para la Critish Broadcasting Corporation (CBC).

² Reduced Instruction Set Computer (RISC), un procesador concebido para tener solamente dos características fundamentales: Instrucciones de tamaño fijo, y Acceso a memoria solo de las instrucciones de carga y almacenamiento.

Años más tarde APPLE COMPUTERS forma con ACORN y VLSI³ la compañía ADVANCED RISC MACHINES, que luego se llamaría ARM HOLDINGS LTD., con el fin de introducir al mercado la PDA⁴ APPLE NEWTON, que se observa en la Figura 2.2, que tenía como CPU una ARM6; aunque las PDA no tuvieron gran trascendencia, los procesadores de arquitectura ARM fueron poco a poco incorporándose en los teléfonos celulares y en las tarjetas de desarrollo microcontroladas. (Caprile, 2012)

Figura 2.2 Apple Newton



Computador de bolsillo diseñado con ARM6, Fuente: (Caprile, 2012)

2.1.2. Raspberry Pi

La Raspberry Pi es un micro computador, que incluye en una sola placa todos los componentes básicos para un ordenador de escritorio, CPU, RAM, GPU, USB, HDMI, Ethernet y puertos GPIO⁵. Para su funcionamiento es necesario nada más conectar los periféricos básicos, de entrada teclado y de salida un monitor, como sistema operativo de preferencia utiliza el Raspbian que es una adaptación de Debian, que a la vez es una distribución del sistema Linux. (Raspberry Pi Foundation, 2017)

³ Very Large Scale Integration (VLSI), Integración a escala muy grande, circuitos integrados con decenas de miles de transistores o más.

⁴ Personal Digital Assistant (PDA), ordenador de bolsillo que incluía prestaciones para ejecutivos como agenda, calendario, procesador de texto, hoja de cálculos, etc.

⁵ General Purpose Input/Output (GPIO), Puertos de entrada y salida de propósito general, utilizados para desarrollo de sistemas micro controlados.

El objetivo de la Raspberry es el de dar la opción de acceder a la tecnología en sectores de bajos recursos, también para la enseñanza de primeros pasos en programación, sobre todo en lenguaje Python⁶. (Raspberry Pi Foundation, 2017)

La Raspberry Pi es también muy utilizada como tarjeta de desarrollo por lo que ha sido incluida en las librerías de varios softwares de ingeniería como son Matlab, Simulink, LabVIEW. (Raspberry Pi Foundation, 2017)

2.1.3. Raspberry pi3 modelo b

La versión 3 Modelo B de la Raspberry Pi, que se ve en la Figura 2.3, es la tercera generación de la placa, entró en el mercado a inicios del año 2016 y posee las características enumeradas en la Tabla 2.1.

Figura 2.3 Raspberry Pi 3



Micro computadora de placa única, utilizada también como tarjeta de desarrollo, Fuente: (Raspberry Pi Foundation, 2017)

⁶ Lenguaje de programación interpretativo y de multiplataforma que posee licencia libre y hace énfasis en un código legible e interactivo

Tabla 2.1 Características Raspberry Pi 3

Raspberry Pi3 Modelo B	
Procesador	QuadCore, 1.2 GHz, Broadcom, BCM2837 de 64bit
Memoria RAM	1GB
Comunicación	Wireless LAN y Bluetooth Low Energy (BLE) BCM43438
Pines de propósito general	40
Puertos USB 2.0	4
Audio y Video	Audio estéreo de 4 polos y video compuesto
Puerto HDMI	1
Puerto CSI para cámara Raspberry Pi	1
Puerto DSI para pantalla táctil Raspberry Pi	1
Puerto Micro SD	1

Características Raspberry Pi 3, Fuente: (Raspberry Pi Foundation, 2017)

2.2. Visión artificial

Los componentes que forman los sistemas dotados de inteligencia artificial tienen que relacionarse unos con otros y a la vez con el mundo exterior. (Pajares & Santos, 2006)

Del mundo exterior se reciben datos que serán utilizados para acciones propias o relacionadas con otros componentes. (Pajares & Santos, 2006)

El ser humano percibe estos datos de manera muy desarrollada, con lo que se llama sentidos: olfato, gusto, tacto, oído y vista; la información recibida luego es procesada por el cerebro de manera casi inmediata para actuar en consecuencia de la misma. (Pajares & Santos, 2006)

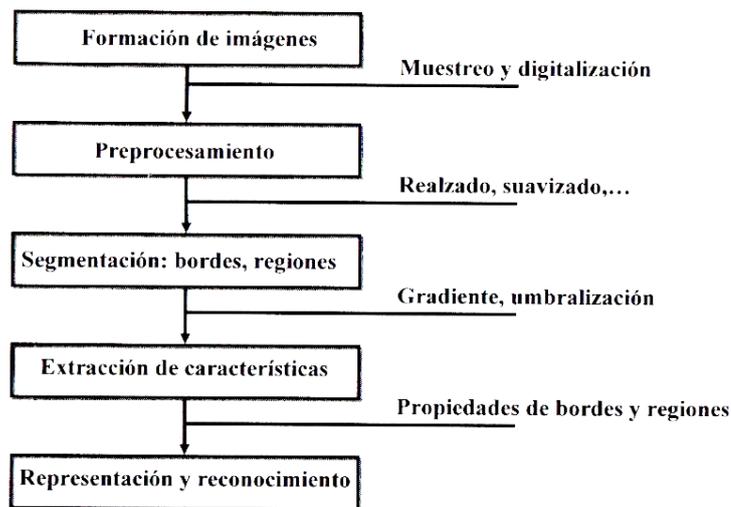
La visión artificial trata de imitar el sentido de la vista adquiriendo información del entorno por medio de tecnologías computacionales y luego procesando esa

información mediante algoritmos genéricos de inteligencia artificial. (Pajares & Santos, 2006)

2.2.1. Proceso de visión artificial

En la Figura 2.4, se observan las etapas que conforman el proceso de percepción de la visión artificial.

Figura 2.4 Proceso de Visión Artificial



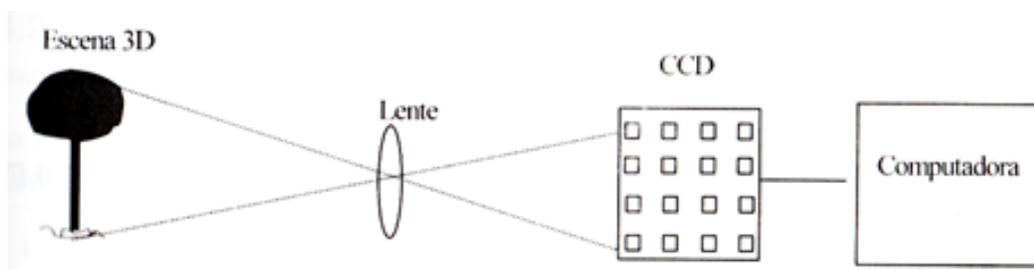
Etapas que conforman el proceso de percepción visual artificial, Fuente (Pajares & Santos, 2006)

- Formación de imágenes digitales: A través de cámaras electrónicas se obtienen imágenes del entorno en formato digital.
- Pre procesamiento: Como un paso previo, con técnicas como el realzado y el suavizado se mejora las imágenes digitales obtenidas.
- Segmentación: A las imágenes tratadas en la etapa de pre procesamiento se les extrae información importante como bordes, esquinas, centroides, etc.
- Representación y conocimiento: Con el uso de técnicas de aprendizaje y algoritmos matemáticos, se procesa la información obtenida para reconocer o catalogar los objetos del entorno. (Pajares & Santos, 2006)

2.2.2. Formación de imágenes digitales

Las imágenes se forman por la luminosidad reflejada por un objeto y dependiendo de la longitud de onda se reciben las diferentes intensidades que a la vez son procesadas para reconocer colores o formas. (Pajares & Santos, 2006)

Figura 2.5 Captura de una Imagen



Captura digital de una imagen, Fuente: (Pajares & Santos, 2006)

Mediante dispositivos fotosensibles como por ejemplo los CCD⁷ que constan a la vez de muchos elementos fotosensibles dispuestos en forma de matriz, se recibe luz en cada uno de ellos como se ve en la Figura 2.5, recibiendo diferentes intensidades dependiendo de la escena. (Pajares & Santos, 2006)

Figura 2.6 Imagen Digital



Imagen digital con una calidad de 16 píxeles, Fuente: (Pajares & Santos, 2006)

Las intensidades de la matriz del CCD son almacenadas en forma de valores en una matriz matemática a la cual se llama imagen digital, a cada valor se le llama píxel⁸.

En la Figura 2.6 (b), por ejemplo se capta en una CCD de 4x4 diferentes valores de intensidad que serán almacenados en una matriz de 16 píxeles, por lo que mientras

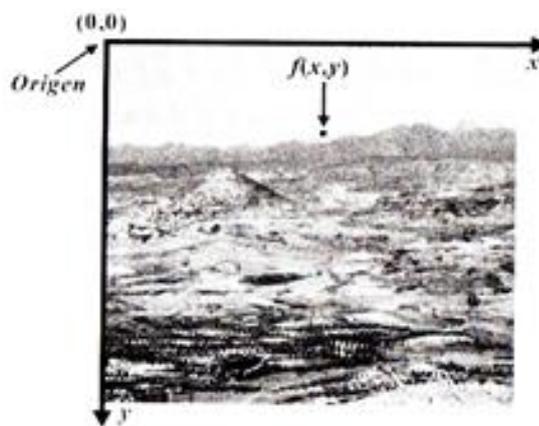
⁷ Charge Coupled Devices (CCD), dispositivos fotosensibles de acoplamiento de carga.

⁸ Acrónimo de Picture Element, es la unidad elemental que forma parte de una imagen digital.

más píxeles recibe el dispositivo, mayor será la calidad de la imagen digital; en la Figura 2.6 (a) se puede ver la representación digital del árbol de la Figura 2.5 con una resolución de 16 píxeles. (Pajares & Santos, 2006)

La imagen digital entonces no es otra cosa más que una función de coordenadas, teniendo cada píxel una posición (x, y) ; en la Figura 2.7 se grafican los ejes de coordenadas con el origen y dirección comúnmente utilizados. (Pajares & Santos, 2006)

Figura 2.7 Convención de Ejes



Coordenadas (x,y) con origen y dirección, Fuente: (Pajares & Santos, 2006)

2.2.3. Extracción de regiones en una imagen digital

Las imágenes que se extraen del entorno cuentan con características específicas en diferentes zonas, las cuales son notables y están dadas por la agrupación de píxeles de propiedades similares o comunes, por ejemplo, píxeles del mismo color. A estas agrupaciones se las conoce con el nombre de regiones. (Pajares & Santos, 2006)

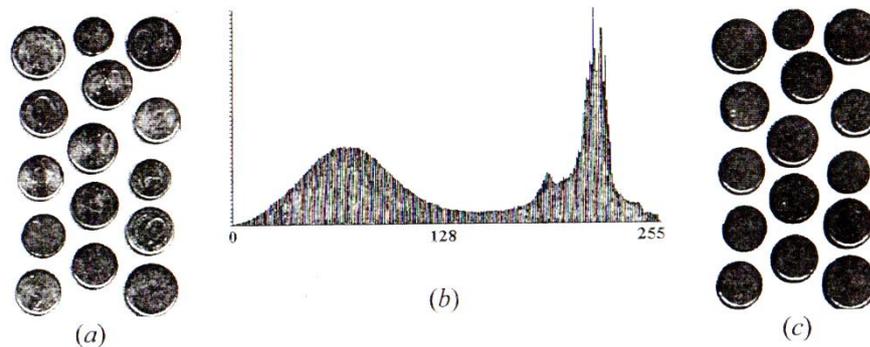
2.2.4. Técnica de Binarización por Detección de Umbral

La Detección de Umbral es una técnica común usada en los sistemas de visión artificial para la detección de objetos, primordialmente cuando se requiere procesar una gran cantidad de datos. (Pajares & Santos, 2006)

En la Figura 2.8 (b), se muestra el histograma de intensidad que corresponde a la Figura 2.8 (a) en la cual se muestran objetos oscuros sobre una superficie clara. En

el histograma se puede apreciar dos diferentes niveles de intensidades que son dominantes y que pueden ser separadas por un valor X, es decir, el pixel que tenga un valor menor a X corresponderá al objeto; en caso contrario el pixel que tenga un valor mayor a X corresponderá a la superficie. Como resultado en la Figura 2.8 (c) se obtiene las regiones por detección de umbral de la Figura 2.8 (a). (Pajares & Santos, 2006)

Figura 2.8 Binarización de objetos



(a) Imagen original, (b) Histograma de intensidades, (c) Imagen resultante por detección de umbral, Fuente: (Pajares & Santos, 2006)

2.2.5. Extracción de líneas y regiones

Existen varias propiedades de las regiones, lo que permitirá identificar cada una de ellas. Las cuales se citan a continuación: las propiedades métricas, las texturas y los momentos invariantes de Hu. (Pajares & Santos, 2006)

2.2.6. Propiedades métricas

Son Propiedades de las regiones caracterizadas por una métrica. Entre las principales se tiene: área, perímetro, centro de gravedad, compatibilidad, elongación, ejes, orientación, excentricidad, redondez, compacidad. (Pajares & Santos, 2006)

El área se conoce como el número de píxeles contenidos dentro de su frontera. El perímetro P es la longitud de su frontera, que se puede obtener a partir de las coordenadas (x_i, y_i) de los puntos que lo definen como sigue:

$$P = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad \text{Ec. (2.1)}$$

Es más común usar la compatibilidad de una región que es definida como P^2/A . Este es un valor sin dimensiones que resulta ser mínimo para una región circular. (Pajares & Santos, 2006)

El centro de gravedad (x,y) es un punto muy representativo de la región y se obtiene como sigue:

$$\bar{x} = \sum_i x_i / A, \quad \bar{y} = \sum_i y_i / A \quad \text{Ec. (2.2)}$$

En los cálculos de la Ecuación 2.2 del centro de gravedad se tiene pesos específicos a todos los pixeles que intervienen; sin embargo, a veces puede resultar de interés calcular el centro de gravedad teniendo en cuenta el valor de la intensidad en cada uno de los puntos, en cuyo caso la ecuación se transforma como sigue:

$$\bar{x} = \frac{\sum_i f(x_i, y_i) x_i}{\sum_i f(x_i, y_i)}, \quad \bar{y} = \frac{\sum_i f(x_i, y_i) y_i}{\sum_i f(x_i, y_i)} \quad \text{Ec. (2.3)}$$

Otra propiedad importante es la elongación o razón de aspecto, que en el caso de un rectángulo es la razón de su longitud respecto de su ancho. Generalizando este concepto a las regiones de cualquier tipo, se circunscribe un rectángulo a la figura y se calcula la razón de aspecto del rectángulo circunscrito. (Pajares & Santos, 2006)

2.2.7. Momentos invariantes de hu

Se toma $f(x,y)$ como la intensidad de punto (x,y) en una región. El momento de orden (p+q) para la región se define como:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad \text{Ec. (2.4)}$$

Donde el sumatorio se toma sobre todas las coordenadas especiales(x,y) de puntos de la región. El momento central de orden (p+q) viene dado por:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \text{ con } \bar{x} = \frac{m_{10}}{m_{00}}; \bar{y} = \frac{m_{01}}{m_{00}} \quad \text{Ec. (2.5)}$$

Los momentos centrales normalizados de orden (p+q) se definen como:

$$n_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad \text{donde } \gamma = \frac{p+q}{2} + 1 \quad \text{para } (p+q) = 2,3, \dots \dots \quad \text{Ec. (2.6)}$$

El siguiente conjunto de momentos invariantes a traslaciones, rotaciones y cambios de escala, propuesto por Hu(1962), se puede obtener como sigue:

$$\phi_1 = n_{20} + n_{02} \quad \text{Ec. (2.7)}$$

$$\phi_2 = (n_{20} + n_{02})^2 + 4n_{11}^2$$

$$\phi_3 = (n_{30} + 3n_{12})^2 + (3n_{21} + n_{03})^2$$

$$\phi_4 = (n_{30} + n_{12})^2 + (n_{21} + n_{03})^2$$

$$\begin{aligned} \phi_5 = & (n_{30} + 3n_{12})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] \\ & + (3n_{21} - n_{03})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 \\ & - (n_{21} + n_{03})^2] \end{aligned}$$

$$\begin{aligned} \phi_6 = & (n_{20} - n_{02})[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] \\ & + 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03}) \end{aligned}$$

$$\begin{aligned} \phi_7 = & (3n_{21} + n_{03})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] \\ & + (3n_{121} - n_{03})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 \\ & - (n_{21} + n_{03})^2] \end{aligned}$$

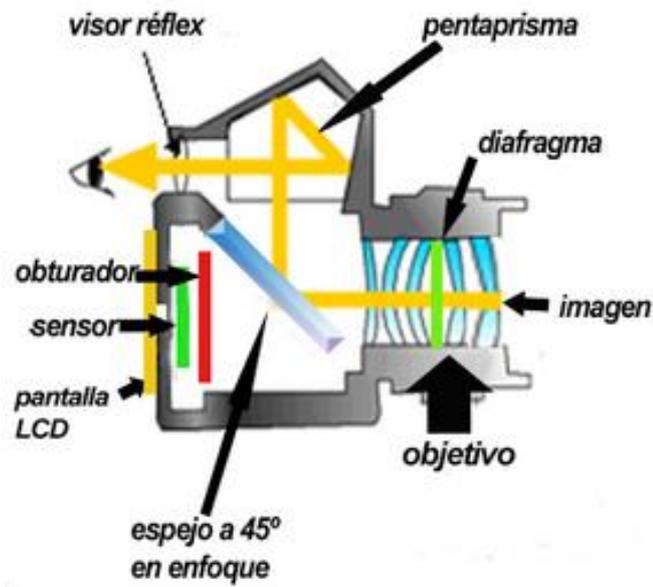
Para que todos los momentos contribuyan por igual en la función de discriminación γ , por tanto, estén dentro del mismo orden de magnitud, a veces es necesario proceder a una normalización con la Ecuación 2.8. (Pajares & Santos, 2006)

$$\phi_{n=abs}[\ln(abs(\phi_n))] \quad \text{Ec. (2.8)}$$

2.2.8. Cámara digital

Las cámaras digitales son dispositivos que almacenan fotografías en dispositivos digitales⁹, en éste grupo las cámaras web son dispositivos que transmiten en directo las imágenes al computador para que se puedan utilizar en tiempo real. (López, 2014)

Figura 2.9 Elementos de una cámara digital



Elementos en común de toda cámara digital, Fuente: (López, 2014)

Toda cámara digital dispone de los siguientes elementos:

- Diafragma.- Limita la cantidad de luz que entra hacia los lentes.
- Obturador.- Controla el tiempo en el cuál llega la luz hacia el sensor.
- Sensor.- Dispositivo que capta la imagen. (López, 2014)

2.2.9. Cámara Logitech C170

En el presente proyecto de titulación se utilizó la cámara web Logitech C170, que se ve en la Figura 2.10, presenta las características de la Tabla 2.2.

⁹ Son dispositivos de almacenamiento electrónico, basados en memorias tipo ROM, como son tarjetas SD (Secure Digital), M2 (Memory Stick Pro), Flash Memory, entre otras.

Figura 2.10 Logitech C170



Cámara web Logitech, con tecnología USB de transmisión directa de imágenes al computador, Fuente: (Logitech, 2017)

Tabla 2.2 Características Logitech C170

Cámara Logitech C170	
Resolución Videoconferencias	640x480 pixeles
Resolución captura de video	1024x768 pixeles
Tecnología	Logitech Fluid Crystal
Modelo	960-000946/70
Resolución Fotografía	5 megapíxeles
Micrófono	Integrado con reducción de ruido
Certificación USB 2.0	De alta velocidad
Compatibilidad	Windows 7 o superior, iOS, Linux

Características Logitech C170, Fuente: (Logitech, 2017)

2.3. Redes Neuronales Artificiales

Las redes neuronales artificiales son parte del conjunto de nuevos métodos para el procesamiento de información que son desarrollados en la rama de las ciencias de la computación y basan su comportamiento en sistemas biológicos. (Martin del Peso, 2001)

La inteligencia artificial (IA) tiene como objetivo principal la modelización del conocimiento, se tiene dos divisiones de acuerdo a la forma empleada en la

representación del conocimiento: simbólica o cognoscitiva en lo que se refiere a sistemas expertos y neuronal o conexionista en lo que se refiere a redes neuronales. Las redes neuronales representan al modelo conexionista de aprendizaje, es decir, tienen la respuesta a un estímulo. (Martin del Peso, 2001)

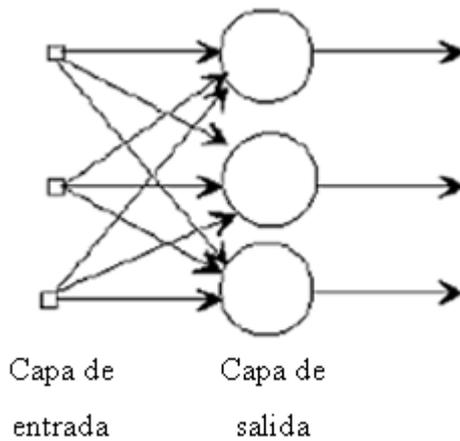
Las redes neuronales artificiales basan su funcionamiento en el sistema nervioso animal debido a que se tienen respuestas más rápidas a las características de la organización y el funcionamiento del cerebro. Los principales conceptos que intentar imitar son:

- Procesamiento paralelo: permiten procesar grandes cantidades de información en el menor tiempo posible.
- Memoria distribuida: la información procesada se almacena en la sinapsis en forma distribuida.
- Adaptabilidad del entorno: la sinapsis modifica su valor de acuerdo a la variación del tiempo y a la experiencia adquirida. (Martin del Peso, 2001)

2.3.1. Estructura de una red neuronal artificial

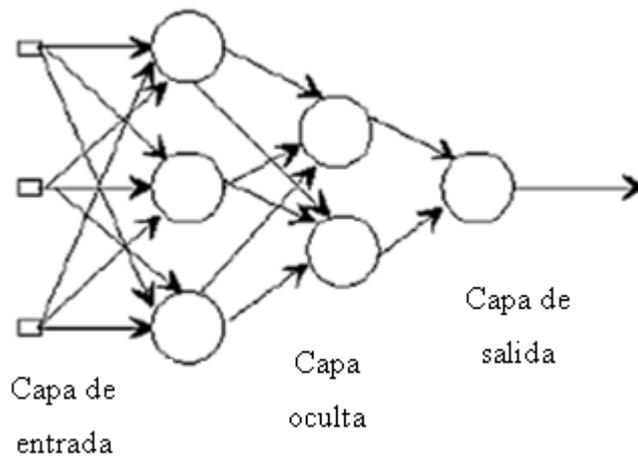
La estructura de la neurona dentro de la red se basa en formar capas o niveles. Teniendo así tres tipos de capas: la capa de entrada, que se encarga de la recepción directa de la información proporcionada por el exterior de la red, la capa oculta, es la que se encarga de procesar la información y se encuentra inmersa en la red, y la capa de salida, que tiene como misión es transmitir la información obtenida del proceso hacia el exterior. (Martin del Peso, 2001)

Figura 2.11 Red neuronal monocapa



Esquema de una red monocapa, Fuente: (Martin del Peso, 2001)

Figura 2.12 Red neuronal multicapa



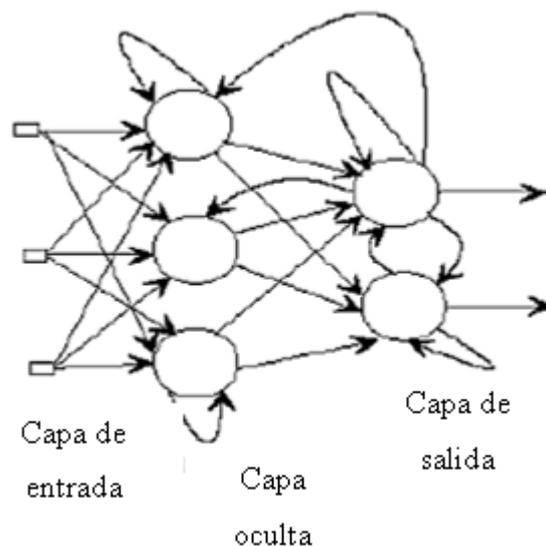
Esquema de una red neuronal multicapa, Fuente: (Martin del Peso, 2001)

Las capas ocultas pueden ser muy elevadas como lo consideren necesario para el proceso. En este sentido pueden clasificarse en dos grupos de acuerdo al número de capas en que se organice la neurona: redes monocapa, que constan de un solo nivel de neuronas, lo que se aprecia en la Figura 2.11, y redes multicapa, que constan de dos o más capas, como se puede observar en la Figura 2.12. (Martin del Peso, 2001)

Existen diferentes tipos de conexiones entre las neuronas que vienen dadas por la dirección o flujo que toman los datos de salida y el origen de las señales o datos de entrada. Las neuronas que reciben las señales de entrada de capas anteriores y a su vez envían las salidas a capas posteriores se conocen como redes unidireccionales o feedforward. (Martín del Peso, 2001)

Así también existen conexiones hacia atrás conocidas como feedback que permiten la circulación de información al contrario de la expuesta anteriormente. Las redes con conexiones hacia adelante como hacia atrás son conocidas como redes recurrentes o feedforward / feedback que permiten que circule la información en los dos sentidos lo que se aprecia en la Figura 2.13. (Martín del Peso, 2001)

Figura 2.13 Red neuronal recurrente



Esquema de una red neuronal recurrente, Fuente: (Martín del Peso, 2001)

2.4. Robótica

La automatización industrial y la robótica se encuentran relacionadas entre sí, puesto que la automatización está vinculada con los sistemas eléctricos, electrónicos, hidráulicos, neumáticos y mecánicos además son capaces de operar de manera autónoma. La robótica es una combinación de estas disciplinas con el fin de realizar trabajos y de esta manera sustituir el trabajo humano. (Alonso, 2012)

La robótica se encarga del estudio en particular de los robots que son máquinas programables, capaces de manipular objetos y realizar trabajos antes realizado por el ser humano. Es la combinación de varias disciplinas como son la electricidad, la electrónica, la mecánica, la informática, las matemáticas, la física, la biología, la neurociencia, etc. (Alonso, 2012)

2.4.1. Leyes de la robótica

Isaac Asimov¹⁰ propuso en sus cuentos y novelas tres leyes o normas básicas que un autómata debería cumplir:

1ra Ley. – “Un robot no puede hacer daño a un ser humano, o, por medio de la inacción, permitir que un ser humano sea lesionado” (Alonso, 2012, pág. 68)

2da Ley. – “Un robot debe obedecer las órdenes recibidas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la primera Ley” (Alonso, 2012, pág. 68)

3ra Ley. – “Un robot debe proteger su propia existencia en la medida en que esta protección no sea incompatible con la primera o la segunda Ley” (Alonso, 2012, pág. 68)

2.5. Clasificación de los manipuladores robóticos

Los manipuladores robóticos la mayoría de autores los clasifican por sus movimientos en el espacio, teniendo como principales tipos:

- Robot cartesiano
- Robot cilíndrico
- Robot esférico
- Robot antropomórfico
- Robot SCARA¹¹.

¹⁰ Escritor y bioquímico ruso, que se caracterizó por su gran influencia en la divulgación de conceptos científicos por medio de sus libros así como también por su participación como asesor científico en obras de cine.

¹¹ Selective Compliant Assembly Robot Arm (SCARA), brazo robótico articulado obediente y selectivo.

2.6. Brazo robot Mitsubishi RV – 2AJ

El brazo RV-2AJ, es un manipulador robótico programable y de operación remota diseñado y producido por Mitsubishi de tipo antropomórfico, es decir que emula un brazo humano. Se lo puede observar en la Figura 2.14.

Figura 2.14 Mitsubishi RV-2AJ



Robot manipulador antropomórfico programable y de operación remota, Fuente: (Robotics Exchange, 2017)

2.6.1. Características del robot RV-2AJ

El Robot Mitsubishi de tipo antropomórfico tiene las características que se detallan en la Tabla 2.3.

Tabla 2.3 Características 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 de brazo (mm)	250+160	
Alcance radial máximo (mm)	410	
Limites (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	

Características Mitsubishi RV-2AJ, Fuente: (Robotics Exchange, 2017)

2.7. Matlab

Matrix Laboratory, su primera versión se lanzó en el año 1989 para el sistema operativo MS-DOS, Mac y Workstations, empezó siendo un software de cálculo interactivo sobre todo para matrices, y se popularizó rápidamente debido a su sencillez, potencia y a sus herramientas gráficas en 2 y 3 dimensiones. (Rodríguez, 2003)

MatLab está equipado con funciones de cálculo, de Bessel, de complejos, transformadas e inversas de Laplace y Furier, entre otras, así como también soporte para un sinnúmero de tarjetas de desarrollo y periféricos electrónicos, que facilitan el desarrollo de sistemas de control. (Rodríguez, 2003)

2.8. Simulink

Simulink es una aplicación dentro de la plataforma de desarrollo Matlab, se basa en el desarrollo y simulación en tiempo real de sistemas por medio de programación con diagramas de bloques. (Colgren, 2007)

En la Figura 2.15 se muestra un modelo de bloques programado en Simulink para el control PID de un sistema.

Figura 2.15 PID en Simulink

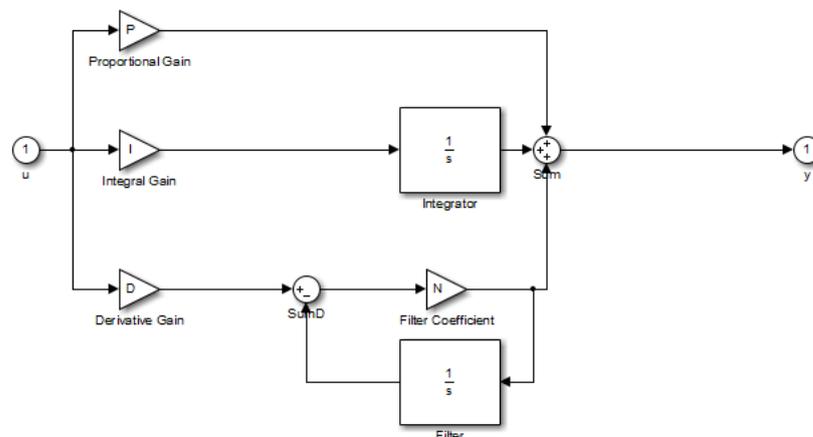


Diagrama de bloques de un proceso PID de lazo cerrado, Fuente: (Colgren, 2007)

Simulink se usa para representar sistemas dinámicos mediante diagramas de bloques. Estos bloques son copiados de la biblioteca de bloques. En el software Matlab se escribe el comando Simulink para acceder a la biblioteca de bloques estándar, dicha biblioteca se encuentra dividida en diferentes subsistemas, agrupando los bloques de acuerdo al comportamiento. (Colgren, 2007).

En las bibliotecas de bloques estándar de Simulink se encuentran bloques lineales, bloques no lineales, bloques discretos y conexiones. (Colgren, 2007)

2.9. Arduino

Arduino cuenta con una plataforma de código abierto y hardware libre, que está compuesto por un sencillo microprocesador que permite recibir información del entorno y enviar información hacia los actuadores lo que facilita el desarrollo de los proyectos. (Pérez, 2012)

La placa de Arduino se basa en un microcontrolador Atmel AVR, cuenta con puertos de entrada/salida y un entorno de desarrollo (IDE) lo que facilita el uso de la electrónica en proyectos de diferentes áreas. (Pérez, 2012)

El lenguaje de programación se basa en una implementación de Wiring, plataforma de computación física y a su vez está basada en Processing, un entorno de programación multimedia. (Pérez, 2012)

2.9.1. Características de la arduino mega

El Arduino Mega 2560 es una tarjeta de desarrollo, basada en un microcontrolador ATmega2560, con características muy superiores a su hermana menor la Arduino Uno, dichas características se las detalla en la Tabla 2.4.

Tabla 2.4 Características de la Arduino Mega

ATmega2560	
Voltaje de operación	5V
Voltaje de entrada(recomendado)	7-12v
Voltaje de entrada (limites)	6-20V
Pines de entrada/salida digital	54(15 salidas PWM)
Pines de entrada análogos	16
Pines de entrada de corriente continua DC	20 mA
Pines para corriente continua de 3.3V	50 mA
Flash Memory	256 KB (8 KB para bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
Largo	101.52 mm
Ancho	53.3 mm
Peso	37 g

Características de la Arduino Mega , Fuente: (Pérez, 2012)

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN

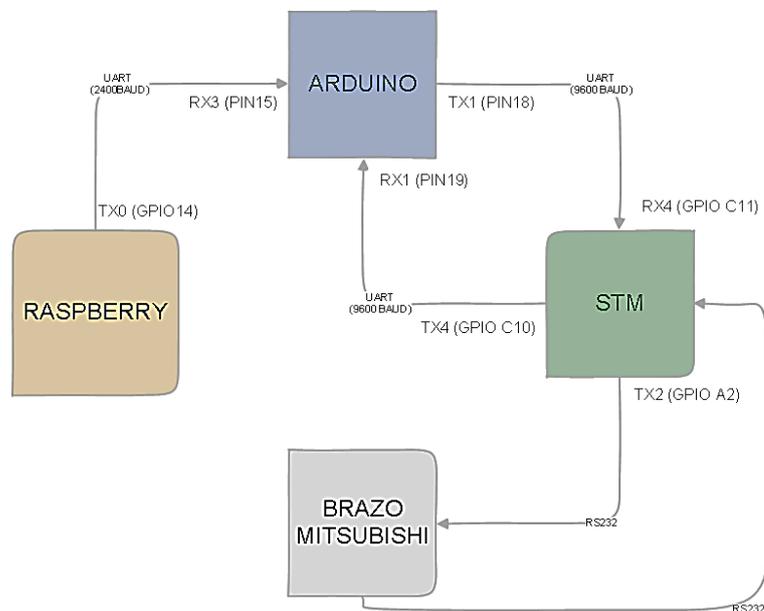
En este capítulo se presenta el proceso de diseño del sistema de visión artificial, en sus diferentes etapas: la programación del software utilizando algoritmos de visión artificial para la detección de regiones y el reconocimiento de patrones; la compilación de las tarjetas de desarrollo para dotarlas de autonomía; y la implementación del hardware necesario para el maquinado de los modelos.

3.1. Programación de tarjetas de desarrollo

Para el funcionamiento del sistema se necesitó la ayuda de tres tarjetas de desarrollo para las fases de: adquisición y reconocimiento de imágenes; control e interfaz gráfica; y manejo del brazo robótico.

Se utilizó comunicación serial¹² para la interacción de las diferentes tarjetas, así como el brazo robótico como se ve en la Figura 3.1.

Figura 3.1 Diagrama de Comunicaciones



Representación de las conexiones, los pines y puertos utilizados, Elaborado por: Andrade & Chulca

¹² La comunicación serial envía un pulso a la vez sobre un único canal, cada pulso representa un bit.

Los puertos utilizados en la comunicación entre cada dispositivo fueron:

- Raspberry PI a Arduino: UART0 (TX) a UART3(RX)
- Arduino a STM: UART1 (TX) a UART4 (RX)
- STM a Arduino: UART4 (TX) a UART1 (RX)
- STM a Brazo Robótico: UART2 (TX y RX)

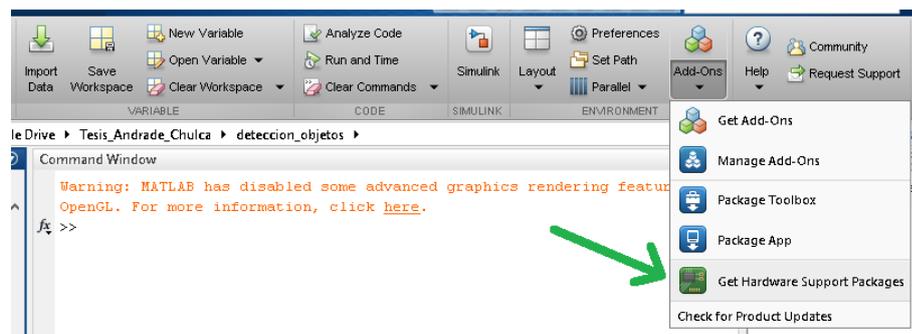
3.2. Programación de la Raspberry Pi 3

Se desarrolló la programación de la etapa de Visión Artificial en la tarjeta de desarrollo o mini computadora Raspberry Pi 3, con la ayuda de la herramienta Simulink perteneciente al software MATLAB R2016a. Se utilizó el Paquete de Soporte de Raspberry Pi para MATLAB y Simulink, mismo que contiene los bloques de función básicos para la compilación de la tarjeta.

3.2.1. Paquete de soporte de Raspberry Pi para Simulink

Para agregar cualquiera de los bloques de soporte se accedió a ‘Get Hardware Support Packages’ dentro de la pestaña ‘Add-Ons’ de la barra de herramientas de MATLAB, como se ve en la Figura 3.2.

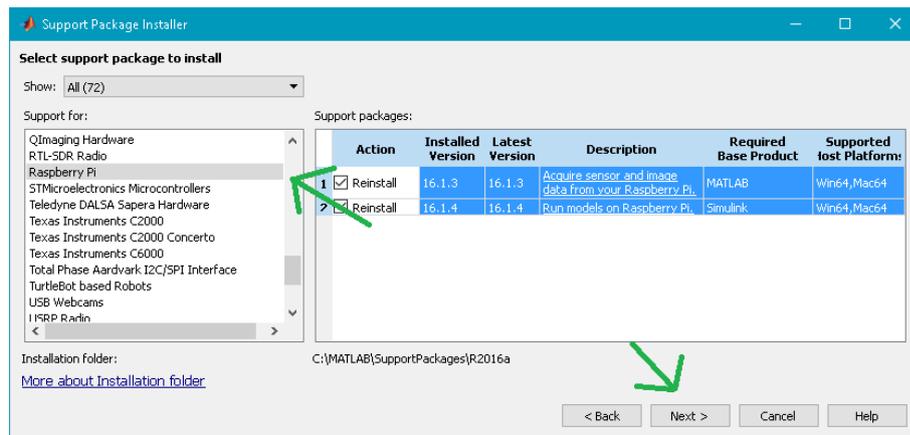
Figura 3.2 Get Hardware Support Packages



Acceso a la herramienta de instalación de Paquetes de Soporte para MATLAB, Elaborado por: Andrade & Chulca

Se seleccionó en la ventana de Support for (Soporte para) la opción Raspberry Pi, en el área de selección a la derecha de éste menú aparecerá los paquetes de soporte existentes para la tarjeta, se marcó las dos existentes, la una para Matlab y la otra para Simulink; como se muestra en la Figura 3.3.

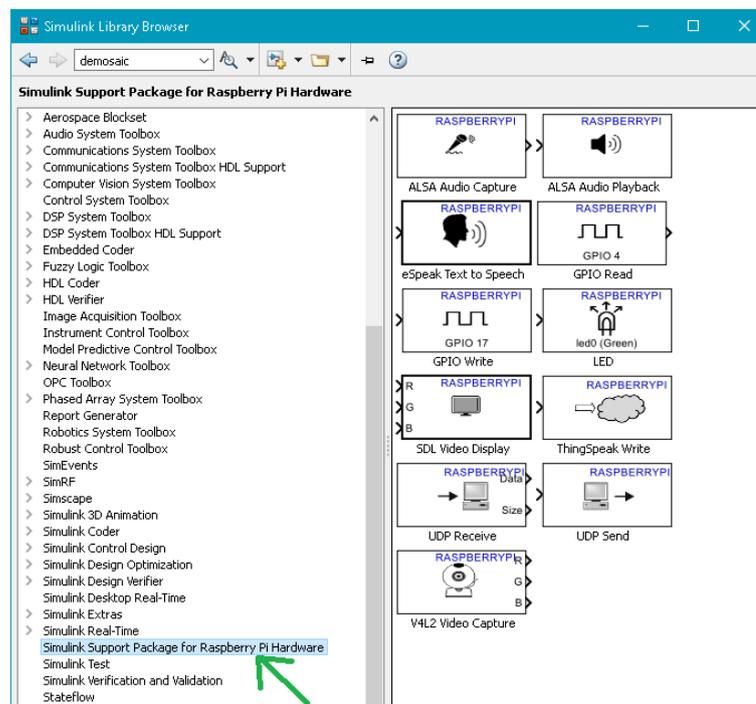
Figura 3.3 Select support package



Selección de los paquetes de soporte para Raspberry Pi, Elaborado por: Andrade & Chulca

Luego de la instalación se tiene los bloques para Simulink correspondientes a Raspberry Pi en el Simulink Library Browser, como se ve en la Figura 3.4.

Figura 3.4 Simulink Library Browser

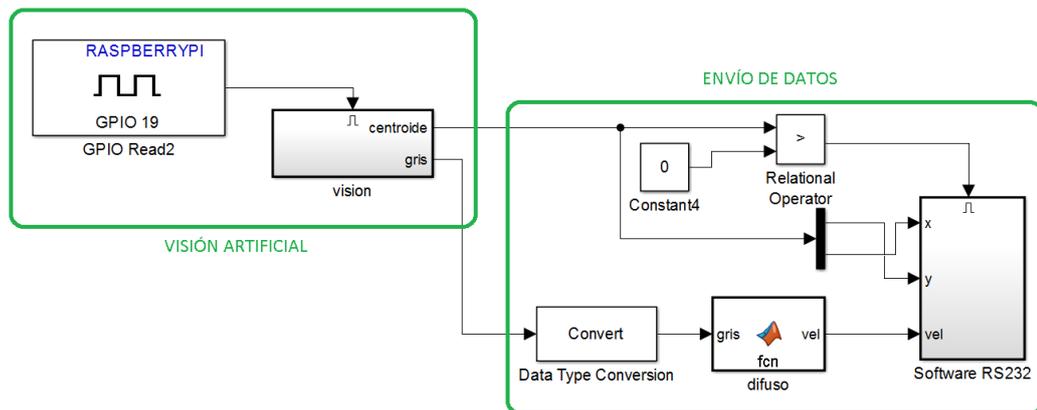


Bloques de desarrollo para la Raspberry Pi 3 en Simulink, Elaborado por: Andrade & Chulca

3.2.2. Modelo programado en Simulink

El modelo programado para desarrollar el sistema de reconocimiento y maquinado en Simulink tiene como base dos etapas: Visión Artificial y Envío de datos, mismas que se pueden apreciar en la Figura 3.5.

Figura 3.5 Modelo principal en Simulink



Etapas principales del modelo programado en Simulink, Elaborado por: Andrade & Chulca

La etapa de Visión Artificial es la encargada de la adquisición, procesado y reconocimiento de las imágenes; la etapa de envío de datos es la encargada del tratamiento, conversión y transmisión de datos por medio de comunicación serial.

3.2.3. Etapa de visión artificial

Es la etapa principal del presente proyecto, y como tal la más extensa y compleja, por lo que se la ha separado en cuatro subetapas, que son:

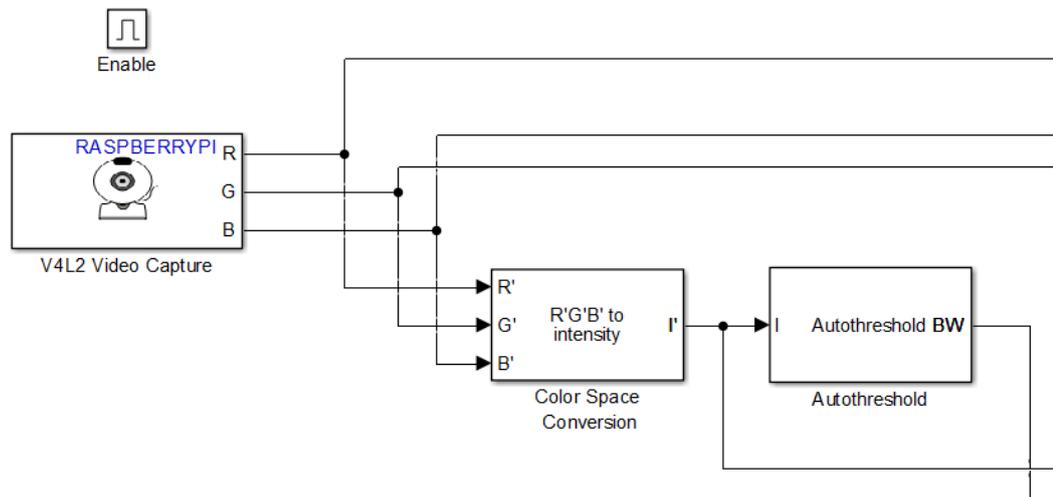
- Adquisición de imágenes
- Extracción de características
- Reconocimiento
- Despliegue

3.2.4. Etapa de adquisición de imágenes

La etapa de adquisición de imágenes consta de tres bloques, el bloque de captura de video incluido en el paquete de soporte de Raspberry para Simulink, el bloque de conversión de RGB a Intensidad (escala de grises), y por último el bloque de

conversión de escala de grises a imagen binaria (blanco y negro), mismos que se los puede ver en la Figura 3.6.

Figura 3.6 Adquisición de imágenes



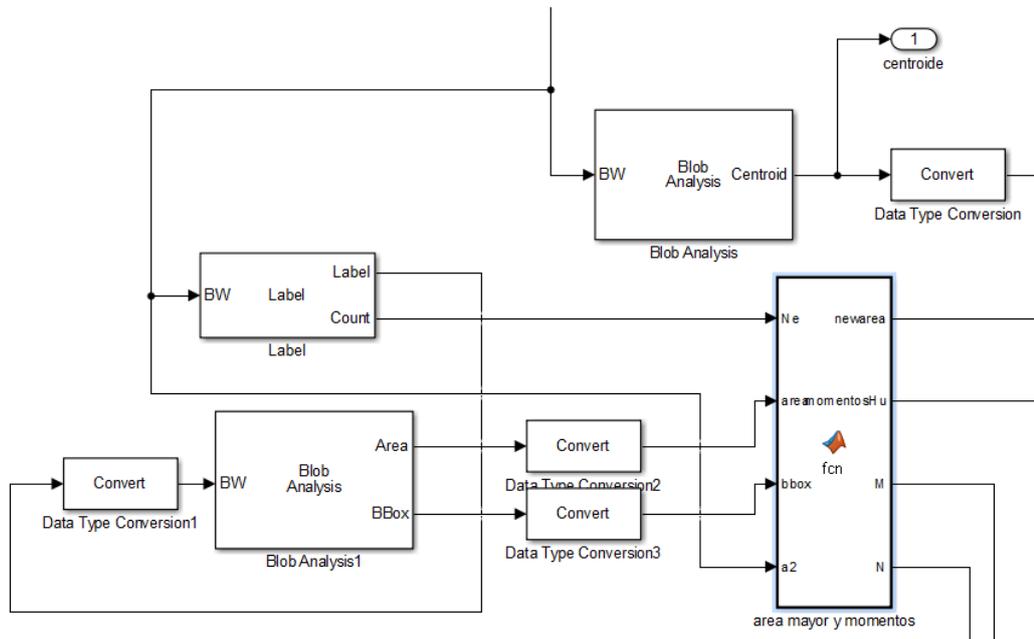
Bloques de adquisición y tratamiento de imágenes, Elaborado por: Andrade & Chulca

Se configuró la adquisición para obtener imágenes con una resolución de 160x120 píxeles y con un tiempo de muestreo de 100 ms, es decir, 10 imágenes por segundo, el bloque Autothreshold que convierte la imagen en binaria, está configurada de manera que los objetos queden en blanco (valor 0) y el espacio libre en negro (valor 1), teniendo así una matriz de 19200 datos (160x120) con valores binarios.

3.2.5. Etapa de extracción de características

La etapa de extracción de características se vale principalmente del bloque de análisis de características Blob, mismo que extrae a partir de la imagen binaria características básicas como son centroide o centro geométrico, área, contorno, entre otras; así mismo se utiliza el bloque de función de Matlab, que permite la elaboración de un código de programación, en este caso utilizado para establecer el objeto con mayor área, en caso de existir más de un objeto, y el cálculo de los momentos de Hu para el reconocimiento de diferentes modelos a partir de patrones establecidos, el bloque Label es usado como ayuda para etiquetar cada objeto con un número, así el primer objeto será etiquetado como '1' el segundo como '2', etc.; se pueden ver los bloques en la Figura 3.7.

Figura 3.7 Extracción de características



Bloques utilizados para la extracción de características relevantes para el reconocimiento de las imágenes, Elaborado por: Andrade & Chulca

En el bloque de función de Matlab que tiene como nombre ‘area mayor y momentos’ se desarrolla el cálculo matemático de los Momentos de Hu, basándose en las Ecuaciones 2.7 y 2.8 descritas en el Capítulo 2. El cambio fundamental que se hace en el código escrito para Matlab es el de cambiar a un cálculo por matrices, ya que el compilador incluido en el paquete de soporte para Raspberry es basado en lenguaje C, y no acepta cálculos por arreglos puntuales.

Lo anteriormente mencionado da a lugar un cambio en el orden de los momentos invariantes como por ejemplo: $\phi_1 = n_{20} + n_{02}$ se traduce en código como: `phi(1) = e(3,1) + e(1,3)`; se evidencia lo expuesto, el valor puntual $n(0,2)$, se lo traduce en valores de matriz como `e(3,1)`, ya que en una matriz no existe una posición 0, y la primera posición en una matriz de cualquier orden empieza con el valor almacenado en (1,1); parte del código del bloque de función se lo puede ver en la Figura 3.8.

Figura 3.8 Cálculo Momentos de Hu

```

Editor - Block: tesis_vision/vision/area mayor y momentos*
vision/calc_gris  vision/area mayor y momentos*
39 % Momentos centrales
40 m(2,1) = sum(x .* F);
41 m(1,2) = sum(y .* F);
42 m(2,2) = sum(x .* y .* F);
43 m(3,1) = sum(x.^2 .* F);
44 m(1,3) = sum(y.^2 .* F);
45 m(4,1) = sum(x.^3 .* F);
46 m(1,4) = sum(y.^3 .* F);
47 m(2,3) = sum(x .* y.^2 .* F);
48 m(3,2) = sum(x.^2 .* y .* F);
49 %Momentos de tercer orden
50 xbar = m(2,1) / m(1,1);
51 ybar = m(1,2) / m(1,1);
52 e(2,2) = (m(2,2) - ybar*m(2,1)) / m(1,1)^2;
53 e(3,1) = (m(3,1) - xbar*m(2,1)) / m(1,1)^2;
54 e(1,3) = (m(1,3) - ybar*m(1,2)) / m(1,1)^2;
55 e(4,1) = (m(4,1) - 3 * xbar * m(3,1) + 2 * xbar^2 * m(2,1)) / m(1,1)^2.5;
56 e(1,4) = (m(1,4) - 3 * ybar * m(1,3) + 2 * ybar^2 * m(1,2)) / m(1,1)^2.5;
57 e(3,2) = (m(3,2) - 2 * xbar * m(2,2) - ybar * m(3,1) + 2 * xbar^2 * m(1,2)) / m(1,1)^2.5;
58 e(2,3) = (m(2,3) - 2 * ybar * m(2,2) - xbar * m(1,3) + 2 * ybar^2 * m(2,1)) / m(1,1)^2.5;
59 %Resultados de los momentos invariantes de Hu
60 phi=zeros(1,7);
61 phi(1) = e(3,1) + e(1,3);
62 phi(2) = (e(3,1) - e(1,3))^2 + 4*e(2,2)^2;
63 phi(3) = (e(4,1) - 3*e(2,3))^2 + (3*e(3,2) - e(1,4))^2;
64 phi(4) = (e(4,1) + e(2,3))^2 + (e(3,2) + e(1,4))^2;
65 phi(5) = (e(4,1) - 3*e(2,3)) * (e(4,1) + e(2,3)) * ((e(4,1) + e(2,3))^2 - 3* (e(3,2) + e(1,4)));
66 phi(6) = (e(3,1) - e(1,3)) * ( (e(4,1) + e(2,3))^2 - (e(3,2) + e(1,4))^2 ) + 4 * e(2,2) * (e(3,2) + e(1,4));
67 phi(7) = (3*e(3,2) - e(1,4)) * (e(4,1) + e(2,3)) * ((e(4,1) + e(2,3))^2 - 3* (e(3,2) + e(1,4)));
68 momentosHu = (abs(-sign(phi) .* (log10(abs(phi)))))*100;

```

Código en Matlab para el cálculo de Momentos de Hu, Elaborado por: Andrade & Chulca

3.2.6. Reconocimiento

En la etapa de reconocimiento se utilizan los bloques de Simulink de Fuzzy que controla las salidas dependiendo del área del modelo, así se tiene las diferentes posibilidades como se muestra en la Tabla 3.1;

Tabla 3.1 Permutaciones de modelos

Modelos a utilizar en el reconocimiento	
Forma	Tamaño
Estrella	Pequeño
	Grande
Pentágono	Pequeño
	Grande
Ángulo T	Pequeño
	Grande

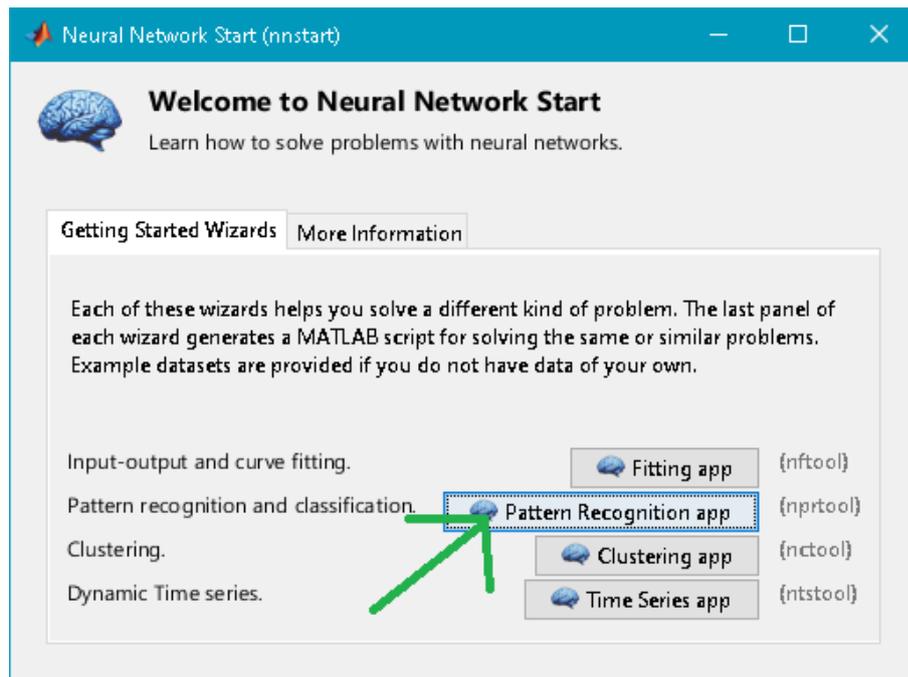
Diferentes posibilidades, en función de la forma y el área, Elaborado por:

Andrade & Chulca

3.2.7. Red neuronal

Para el entrenamiento de la red neuronal se utilizó la herramienta de Matlab 'Neural Network Start', que se activa con el comando 'nnstart'; se utilizó en ésta ocasión el 'Pattern recognition and classification' incluido en la 'nnstart', como se ve en la Figura 3.9.

Figura 3.9 Neural Network Start

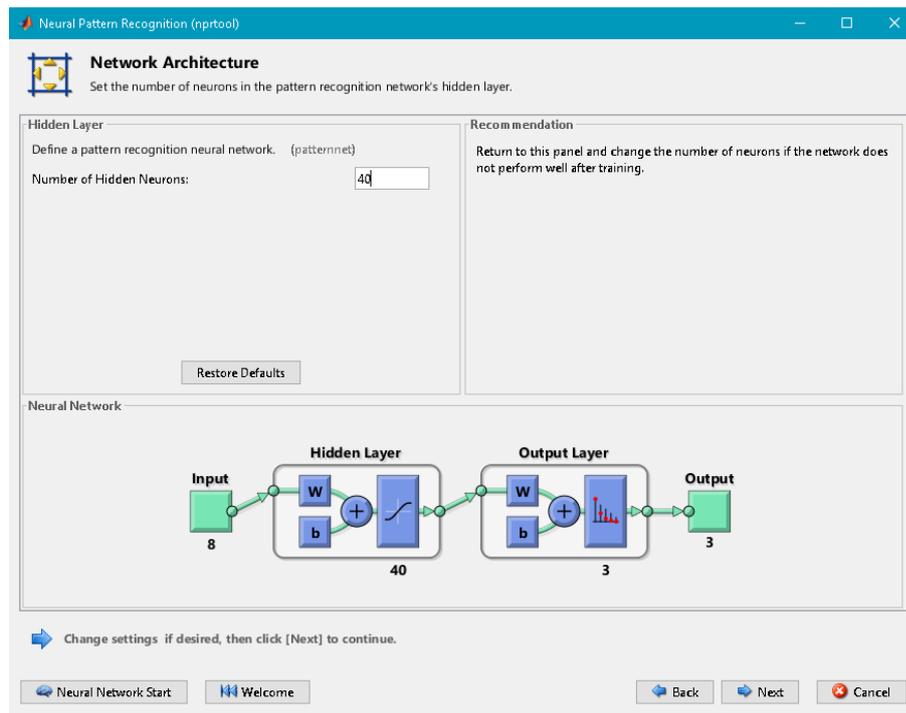


Herramienta para creación de redes neuronales en Matlab, Elaborado por: Andrade & Chulca

Como variables de entrada, se utilizó las características morfológicas anteriormente obtenidas, como son los momentos de Hu, y una octava entrada, perteneciente a la forma y tamaño, obtenidos de la Tabla 3.1; como variables de salida se utilizó tres vectores de tres posiciones, uno para cada forma, $\{1,0,0\}$ para estrella, $\{0,1,0\}$ para ángulo, y $\{0,0,1\}$ para pentágono.

Luego de las pruebas, cuyos resultados se muestran en el Capítulo 4, se configuró la red neuronal con 40 neuronas en la capa oculta, ya que se obtuvo un mejor resultado el momento de 'Entrenar' a la neurona con 51 iteraciones. Como se muestra en la Figura 3.10.

Figura 3.10 Network Architecture



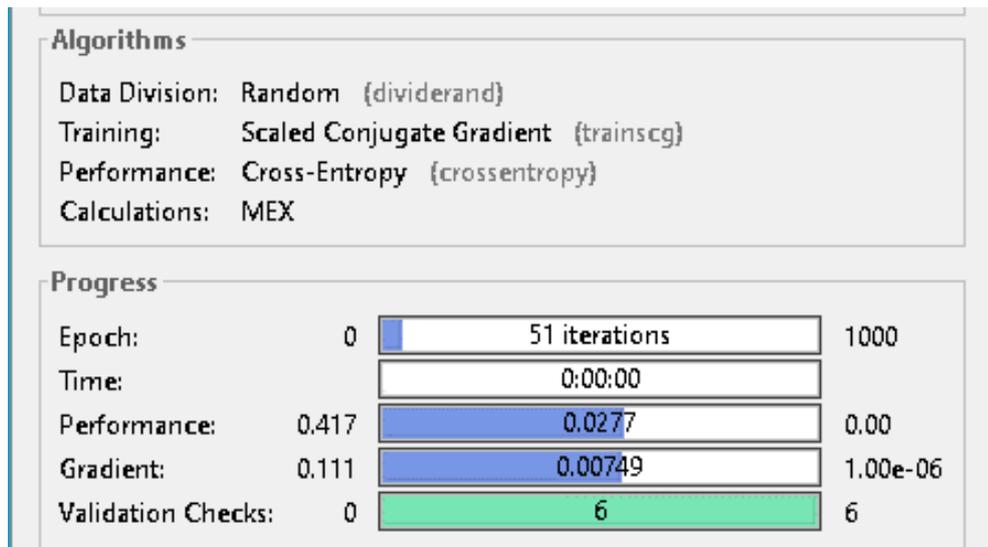
Arquitectura de la red neuronal, configurada con 40 neuronas en la capa oculta, Elaborado por: Andrade & Chulca

Los algoritmos utilizados fueron los que por defecto tiene configurados la herramienta de creación de redes neuronales para el caso de reconocimiento de patrones, como son:

- División de datos: 'Random' (aleatorio)
- Entrenamiento: 'Scaled Conjugate Gradient' (gradiente conjugado escalado)
- Rendimiento: Cross-Entropy (entropía cruzada)

Los resultados de Rendimiento, gradiente y validación con 40 neuronas se pueden observar en la Figura 3.11.

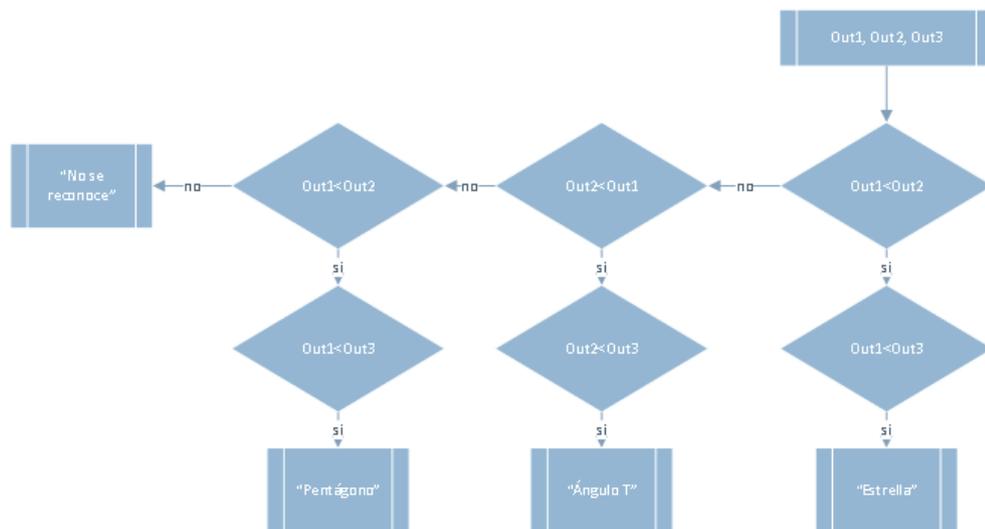
Figura 3.11 Algorithms and progress



Algoritmos y progreso del entrenamiento de la neurona, Elaborado por: Andrade & Chulca

En el bloque de función de Matlab con nombre RedIN, se utiliza la salida Fuzzy como entrada para la red neuronal además de los momentos de Hu, el código describe el flujograma que se puede ver en la Figura 3.12.

Figura 3.12 Reconocimiento de figura



Reconocimiento de figura a partir de las tres salidas de la red neuronal, Elaborado por: Andrade & Chulca

La red neuronal devuelve como salida un vector de cuatro valores en el que la posición con el menor valor es el que determina la forma del modelo, esta relación es cambiada a un valor entero, de acuerdo a la Tabla 3.2.

Tabla 3.2 Valores que determinan la forma

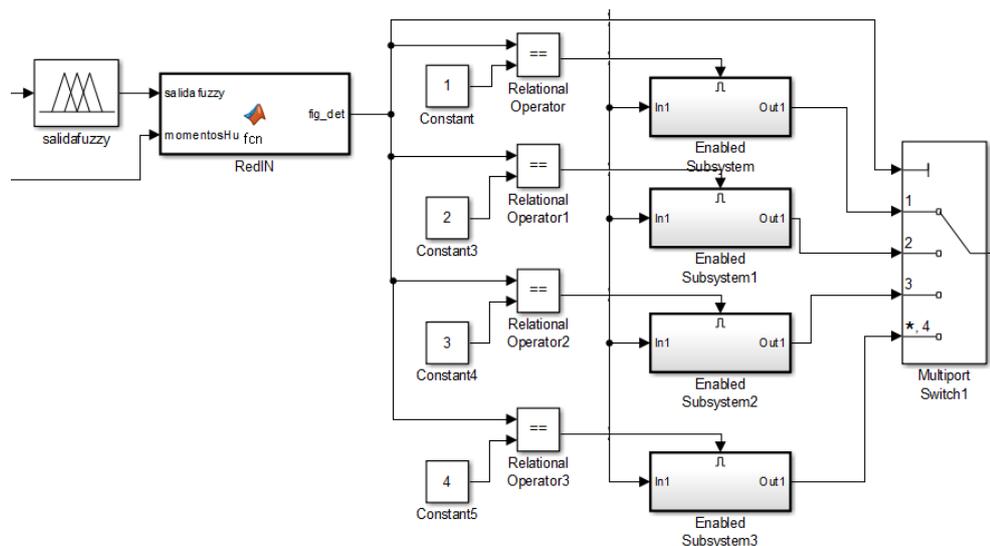
Posición menor del vector de salida	Forma
1	Estrella
2	Ángulo T
3	Pentágono
4	No reconocido

La menor posición del vector de salida determina la forma del modelo,

Elaborado por: Andrade & Chulca

Los bloques siguientes tienen por objeto el comparar el valor entero de salida del bloque de función 'RedIN' y compararlo con los valores correspondientes a cada forma en concordancia con la Tabla 3.2, así en un subsistema independiente se devuelve una cadena de texto dependiendo del caso, se puede observar toda la etapa de reconocimiento programada en Simulink en la Figura 3.13.

Figura 3.13 Etapa de reconocimiento

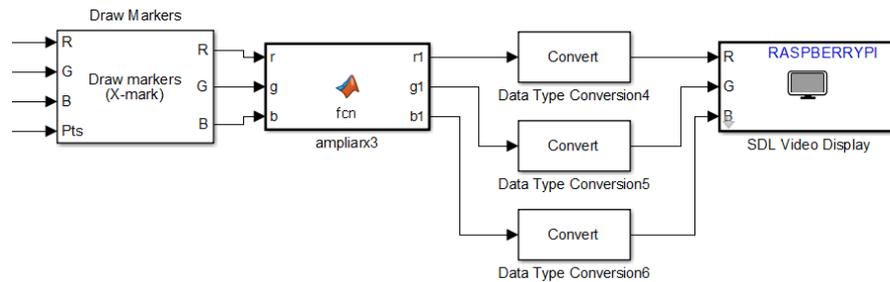


Bloques de Simulink para el reconocimiento de los modelos, Elaborado por: Andrade & Chulca

3.2.8. Despliegue

La etapa de despliegue de resultados tiene como bloque principal el ‘SDL Video Display’ del paquete de soporte de Raspberry para Simulink, que permite desplegar video en un monitor conectado el puerto HDMI de la Raspberry Pi, Figura 3.14.

Figura 3.14 Etapa de despliegue

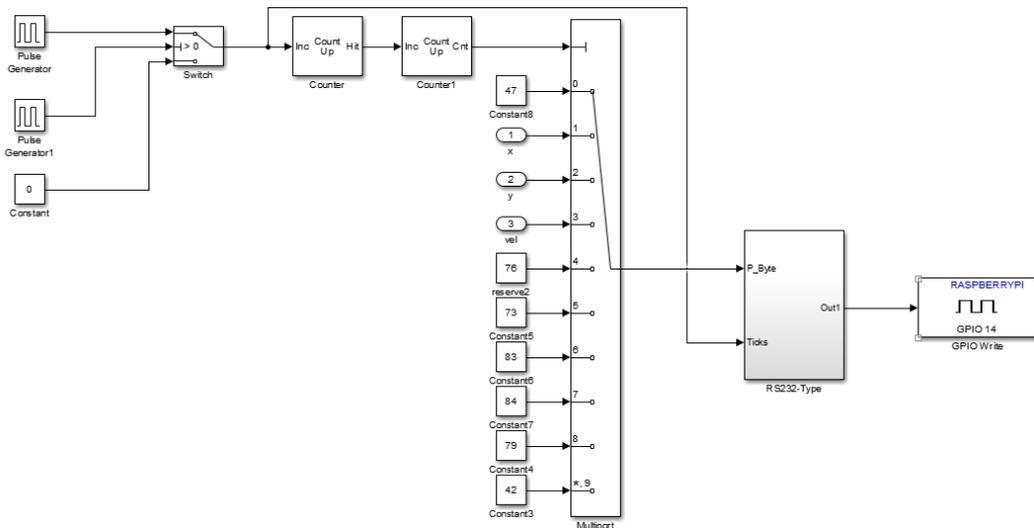


Bloques para desplegar en pantalla el video en tiempo real, Elaborado por: Andrade & Chulca

El bloque ‘Draw Markers’ es utilizado para agregar una marca en el lugar en donde se encuentre el centro geométrico del modelo, los puntos son obtenidos de la etapa de extracción de características.

En esta etapa es necesario un bloque de función de Matlab para ampliar la imagen que se muestra en el monitor, debido a que la adquisición original se la hizo con una resolución de 160x120 por motivos de procesamiento, motivo por el cual la imagen que se observó en pantalla resulta muy pequeña.

Figura 3.16 Etapa de envío de datos



Bloques de Simulink que permiten el envío de datos por el pin 14 que corresponde al transmisor serial, Elaborado por: Andrade & Chulca

Los datos se envían por caracteres separados uno a la vez para lo que se utiliza un bloque selector junto con bloques de conteo, los caracteres a su vez son enviados en código ASCII, se puede observar los bloques de la comunicación serial programada por software en la Figura 3.16.

3.3. Programación de la Arduino Mega

La tarjeta de desarrollo Arduino Mega se utilizó para el procesamiento de las coordenadas del centro geométrico encontradas con la visión artificial, y mediante la interfaz desarrollada anteriormente por (Cajamarca & Portilla, 2016) se utilizó la cinemática inversa para el maquinado de los modelos con el brazo Mitsubishi.

3.3.1. Recepción de datos desde la Raspberry Pi 3

Se agregó al código existente la programación necesaria para la adquisición de datos provenientes de la Raspberry Pi configurando el puerto serial 3 a una velocidad de 2400 bits por segundo, de acuerdo a la configuración de la etapa de envío de datos ya desarrollada en el punto 3.2.9 del presente documento.

Figura 3.18 Movimiento del brazo al punto medio

```
TESIS $
//POSICION MEDIA

Serial1.print("X ");
Serial1.print(150);
Serial1.println("\t");
delay(200);

Serial1.print("Y ");
Serial1.print(-250);
Serial1.println("\t");
delay(200);

Serial1.print("Z ");
Serial1.print(-250);
```

Parte del código para mover el brazo Mitsubishi hacia el punto medio, Fuente: (Cajamarca & Portilla, 2016)

Luego se envía las coordenadas para un traslado en el plano XY hacia el punto donde se encuentra el centro geométrico utilizando las coordenadas obtenidas con las ecuaciones 4.1 y 4.2.

Posteriormente se moverá con el taladro encendido en el eje Z hacia abajo quedando a 5 mm de la base seguido de un movimiento en el mismo eje hacia arriba, taladrando de esta manera el modelo en el centro geométrico.

Por último se moverá nuevamente el brazo hasta el punto medio y al punto de reposo.

3.4. Implementación de hardware

Como hardware del sistema se tiene una mesa de trabajo metálica y de color blanco, sobre la cual irán los modelos a maquinar adheridos a la misma por imanes, un brazo en el que se adaptará la cámara Logitech necesario para la adquisición de imágenes, y el brazo robótico Mitsubishi RV-2AJ, mismo que tendrá adaptado en su pinza un taladro para el maquinado de los modelos.

3.4.1. Mesa de trabajo

Para el funcionamiento del sistema, se utilizó como área de trabajo la mesa adjunta al Brazo Robótico Mitsubishi, sobre la cual se adaptó una base de fondo sobre la cual se ubicarán los modelos a maquinar, se la puede observar en la Figura 3.19.

Figura 3.19 Base de trabajo



Base de trabajo de color blanca sobre la mesa adjunta al brazo Mitsubishi, se observan imanes utilizados como puntos de referencia, Elaborado por: Andrade & Chulca

Las características de la mesa de trabajo son las incluidas en la Tabla 3.4.

Tabla 3.4 Mesa de trabajo

Mesa de trabajo	
Dimensiones	35 x 45 cm
Material	Acero
Color	Blanco
Grosor	2 mm

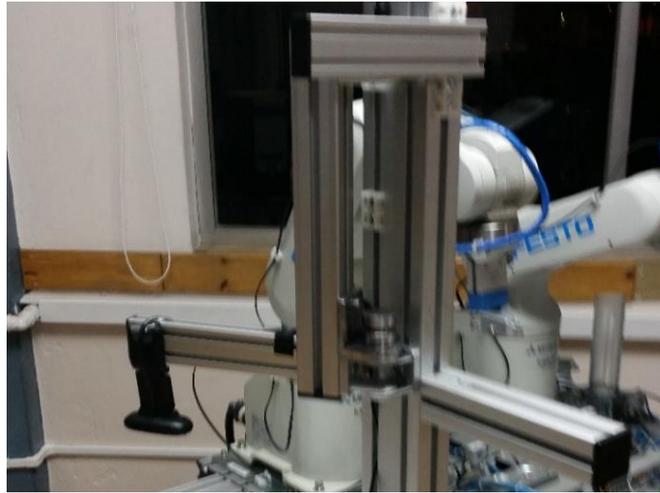
Características de la mesa de trabajo, Elaborado por: Andrade & Chulca

3.4.2. Brazo de cámara Logitech C170

Se implementó un brazo con movimiento angular en el plano XY, al cual se le adaptó en su extremidad la cámara Logitech C170, para lo cual se instaló un juego de motor y polea que permita el movimiento del mismo.

El brazo se lo ubicó en una de las esquinas de la mesa de trabajo con un movimiento de 90 grados que le permite ingresar al área de la base de trabajo para adquirir las imágenes y posteriormente salir a la posición de reposo, permitiendo que ingrese al área de trabajo el brazo robótico. Se puede observar el brazo en la Figura 3.20.

Figura 3.20 Brazo de la cámara



Brazo con movimiento de 90 grados, se observa la cámara Logitech en el extremo del mismo, Elaborado por: Andrade & Chulca

3.4.3. Caja de tarjetas de control

Se elaboró una caja en acrílico, para instalar en la misma las tarjetas y demás elementos electrónicos utilizados para el control del sistema, como se ve en la Figura 3.21.

Figura 3.21 Caja con tarjetas de control



Caja de control con las tarjetas utilizadas en el sistema, Elaborado por: Andrade & Chulca

La caja incluye aberturas en las cuales encaja los puertos de la Raspberry Pi, Arduino Mega, STM Discovery y Pantalla Touch TFT, de igual manera puerto Serial DB9, y plugs entrada, salida y alimentación.

Dentro de la caja se realizó conexiones físicas entre las tarjetas de desarrollo y las tarjetas de comunicación, además de las tarjetas de control de motor.

3.4.4. Elaboración de modelos a maquinar

Se elaboró los modelos que se utilizaron para el maquinado en formas, tamaño, materiales y colores como se especifica en la Tabla 3.5.

Tabla 3.5 Modelos elaborados para maquinar

Modelos a maquinar		
Forma	Tamaños	Color / Material
Estrella	Pequeño	Café / Balsa
		Negro / Parafina
	Grande	Amarillo / Espuma Flex
		Café / Balsa
Ángulo T	Pequeño	Amarillo / Espuma Flex
		Café / Balsa
	Grande	Amarillo / Espuma Flex
		Negro / Parafina
Pentágono	Pequeño	Negro / Parafina
		Café / Balsa
	Grande	Amarillo / Espuma Flex
		Café / Balsa

Formas, tamaños y materiales en los que se elaboró los modelos a maquinar por el brazo robótico, Elaborado por: Andrade & Chulca

Se dio color a los modelos con pintura a base agua y se incrustó en su base tres imanes a cada uno para que se adhieran a la mesa de trabajo, minimizando los movimientos el momento del taladrado por parte del brazo robótico, se muestran algunos de los modelos en la Figura 3.22.

Figura 3.22 Modelos a maquinar



Modelos elaborados en espuma flex de color amarillo, se puede observar los imanes incrustados en las bases de los mismos, Elaborado por: Andrade & Chulca

CAPÍTULO 4

PRUEBAS Y RESULTADOS

Como parte del proyecto se realizaron varias pruebas de funcionamiento, incluso antes del desarrollo completo del sistema, debido a que se utilizó el método de prueba y error en la calibración de las etapas de visión artificial y de maquinado. Posteriormente se sometió al sistema a nuevas pruebas para calcular la precisión y el margen de error.

4.1. Pruebas de adquisición de características

Para el mejorar el funcionamiento del sistema fue necesario la calibración del programa en Simulink, en específico de los bloques de adquisición y procesamiento.

4.1.1. Tiempos de respuesta de adquisición de imágenes

Se realizó pruebas de funcionamiento con la tarjeta Raspberry Pi 3, conectada únicamente al monitor y con la cámara Logitech C170, para medir el tiempo de respuesta para la obtención del centro geométrico, con diferentes resoluciones y tiempos de muestreo, obteniendo los resultados de la Tabla 4.1.

Tabla 4.1 Tiempos de respuesta

Tiempos de Respuesta		
Resolución	Muestreo (ms)	Retraso (ms)
480x360	10	5000
	100	2000
320x240	10	3000
	100	1000
160x120	10	2000
	100	500

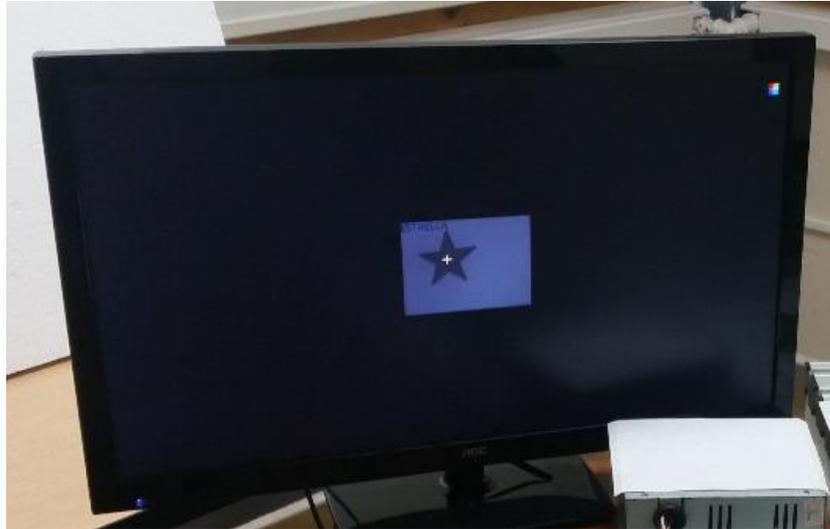
Retrasos antes de que se obtenga el centro geométrico del modelo, Elaborado

por: Andrade & Chulca

En concordancia con los resultados obtenidos, se configuró la adquisición de imágenes con una resolución de 160x120 pixeles y a un tiempo de muestreo de 100

ms. Se puede observar el funcionamiento de la adquisición de características en la Figura 4.1.

Figura 4.1 Adquisición del centro geométrico



Se observa una cruz en el sitio exacto del centro geométrico del modelo, que en este caso es una estrella, Elaborado por: Andrade & Chulca

Debido a la resolución de 160x120, el error en el punto del centro geométrico se redujo a un valor menor al 1%.

4.2. Pruebas de reconocimiento

Las pruebas de reconocimiento de la forma del modelo estuvieron relacionadas con la calibración del balance de blancos y la distancia a la cual se ubicó la cámara Logitech C170.

Tabla 4.2 Pruebas de reconocimiento

Pruebas de reconocimiento			
Balance de blancos	Distancia (cm)	Aciertos	Desaciertos
0	20	10	10
	25	12	8
	30	12	8
-1	20	16	4

	25	17	3
	30	19	1
-2	20	5	14
	25	8	12
	30	10	10

Datos obtenidos en pruebas con variables de balance de blancos y distancia de la cámara, Elaborado por: Andrade & Chulca

4.2.1. Aciertos y desaciertos en el reconocimiento

Se realizó 20 pruebas de reconocimiento con 3 niveles de balance de blancos¹³ y 3 niveles de altura de la cámara, los datos obtenidos se los resumió en la Tabla 4.2.

Se puede observar una prueba exitosa en el reconocimiento de un modelo en la Figura 4.2.

Figura 4.2 Reconocimiento de forma



Se observa el reconocimiento de la estrella, Elaborado por: Andrade & Chulca

¹³ Ajuste realizado mediante software para representar imágenes sin dominantes de color independientemente de la iluminación, es decir una configuración automática de contraste para evitar obtener imágenes con resplandecencia.

4.2.2. Cálculo del error en el reconocimiento

Luego de obtener los datos de la Tabla 4.2 se configuró el sistema con un balance de blancos de -1 y se ubicó la cámara a una distancia de 30 cm; luego de lo cual se procedió a realizar nuevamente la toma de datos, esta vez probando 30 veces cada forma, obteniendo los resultados resumidos en la Tabla 4.3.

Tabla 4.3 Error en el reconocimiento

Error en el reconocimiento			
Figura	Aciertos	Desaciertos	Error
Estrella	28	2	6,67%
Ángulo	30	0	0,00%
Pentágono	29	1	3,33%

Cálculo de errores por la forma del modelo a reconocer, Elaborado por: Andrade & Chulca

Con los valores de error de la Tabla 4.3, se calculó un error promedio del sistema y se obtuvo el valor de 3,33% de error.

Se realizó pruebas de reconocimiento modificando la cantidad de neuronas en la capa oculta de la red neuronal antes entrenada, como se expuso en el punto 3.2.7. Obteniendo los resultados resumidos en la Tabla 4.4.

Tabla 4.4 Errores por número de neuronas

Errores con diferente número de neuronas		
Neuronas en capa oculta	Aciertos / desaciertos	Error
10	15/30	50%
20	20/30	33%
40	30/30	0%
60	30/30	0%

Errores obtenidos modificando el número de neuronas presentes en la capa oculta de la red neuronal de reconocimiento, Elaborado por: Andrade & Chulca

4.3. Pruebas de maquinado

Para el maquinado se realizaron pruebas para la calibración de los puntos de coordenadas de maquinado a partir de las coordenadas en pixeles obtenidas de la visión artificial; y luego se realizó pruebas de funcionamiento de campo, luego de lo cual se realizaron ajustes en el software y hardware para la reducción del error.

4.3.1. Obtención de las coordenadas de maquinado

Fue necesario realizar pruebas para cambiar de las coordenadas virtuales obtenidas en la visión artificial hacia las coordenadas XY reales que serán enviadas al brazo robótico, para lo cual se necesitó tomar los datos resumidos en la Tabla 4.5.

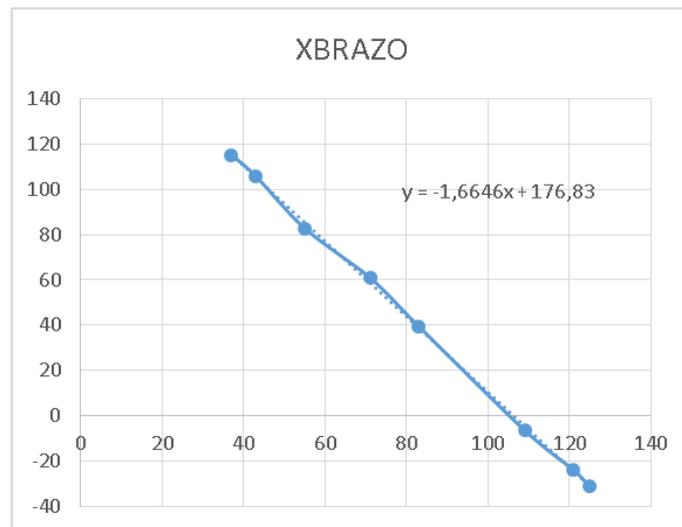
Tabla 4.5 Coordenadas XY

Coordenadas XY Visión Artificial y Brazo Robótico			
X Visión	X Brazo	Y Visión	Y Brazo
37	115,28	37	-369,84
43	105,73	45	-362,29
55	82,75	59	-332,54
71	61	67	-320,85
83	39,5	83	-291,58
109	-6,4	89	-281,8
121	-23,91	91	-275,89
125	-31,26	91	-273,62

Datos de las coordenadas obtenidos de la visión artificial y a la par de las coordenadas de la posición del brazo robótico, Elaborado por: Andrade & Chulca

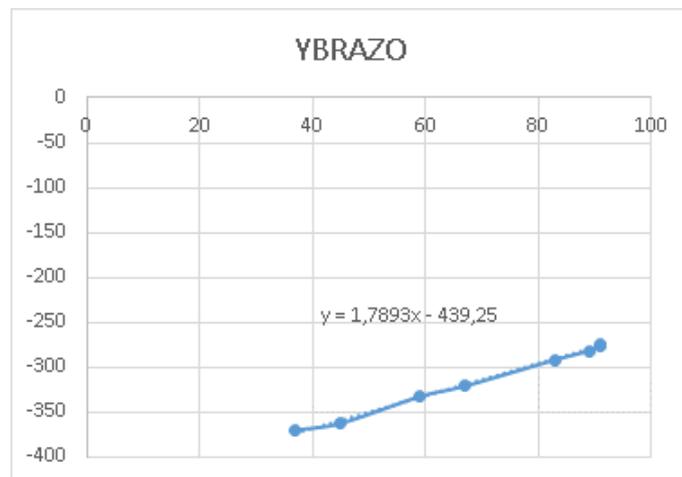
Con los datos obtenidos se procedió a realizar una regresión cuadrática mediante el software Microsoft Excel, obteniendo las Figuras 4.3 y 4.4.

Figura 4.3 Regresión para coordenada X



Gráfica obtenida con los datos de la Tabla 4.5 para X, Elaborado por: Andrade & Chulca

Figura 4.4 Regresión para coordenada Y



Gráfica obtenida con los datos de la Tabla 4.5 para Y, Elaborado por: Andrade & Chulca

Con las gráficas de las Figuras 4.3 y 4.4 obtenidas en Excel se obtiene las ecuaciones 4.1 y 4.2 para el cambio de coordenadas de pixeles a coordenadas de posición del brazo, mismas que son incluidas en el código de programación de la Arduino Mega.

$$X_{real} = -1,6646 * X_{virtual} + 176,83 \quad \text{Ec. (4.1)}$$

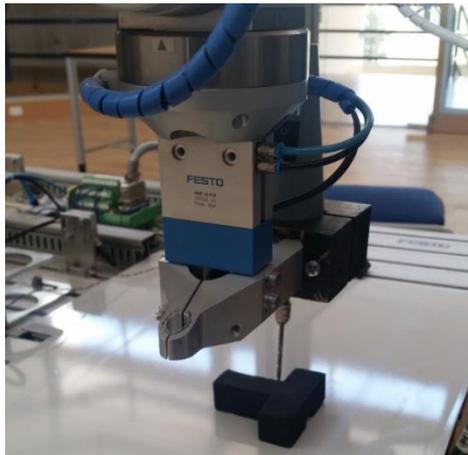
$$Y_{real} = 1,7893 * Y_{virtual} - 439,25$$

Ec. (4.2)

4.3.2. Cálculo de error en el maquinado

Por último se procedió a la prueba de maquinado de los modelos, como se puede observar en la Figura 4.5.

Figura 4.5 Maquinado de modelo



Perforación en el centro geométrico de un ángulo T, Elaborado por: Andrade & Chulca

Para el cálculo del error se realizaron 10 pruebas de perforación con cada forma y se obtuvo los datos resumidos en la Tabla 4.6, donde se muestran los errores promediados.

Tabla 4.6 Error en la perforación en el centro geométrico

Error en la perforación			
Forma	Diámetro (mm)	Error (mm)	Error (%)
Estrella	100	2	2,00
Ángulo T	80	1	1,25
Pentágono	80	2	2,50

Errores obtenidos en la perforación en el centro geométrico, Elaborado por:
Andrade & Chulca

Con los datos obtenidos de la Tabla 4.6 y promediando los errores individuales de cada forma se obtiene un error global del sistema en la perforación en el centro geométrico de cada modelo del 1,92%.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Los momentos invariantes de Hu permitieron dotar al sistema de flexibilidad en el proceso de reconocimiento, porque las variaciones de traslación, rotación y tamaño no afectan a los resultados.

Se solucionó el problema de la iluminación instalando una mesa de trabajo completamente blanca, además con la ayuda del paquete de soporte de Matlab para la adquisición de imágenes vía webcam se reguló el balance de blancos. El sistema no funciona con una iluminación menor a los 50 lux, y se presentan errores aleatorios con iluminación mayor a los 500 lux.

Se redujo el tiempo en cada ciclo de reconocimiento de patrones a 500ms utilizando una resolución de imagen de 160x120 píxeles, debido a que a menor tamaño de imagen mayor velocidad en el procesamiento, como se muestra en los datos experimentales de la Tabla 4.1.

Se logró la autonomía del sistema ya que se obtuvo un ejecutable de la programación con la ayuda del paquete de soporte de Matlab y Simulink mismo que se cargó en la tarjeta de arquitectura ARM Raspberry Pi 3, lo que permitió al sistema arrancar en cuanto se energiza, sin la ayuda de un computador externo.

Se logró coincidir la posición del maquinado con el centro geométrico real, ya que se obtuvo las coordenadas cartesianas reales del entorno de trabajo relacionándolas con las coordenadas virtuales de la visión artificial a través del método de regresión lineal.

El sistema de visión artificial respondió con un aproximado del 3% de error en la fase de pruebas de maquinado y se determinó que mientras la imagen es más grande el error disminuye, debido a que se tiene un desplazamiento de entre el 0 y 2 mm con respecto al centro geométrico real y éste se relaciona con el diámetro circunscrito del modelo.

En las pruebas de campo se encontró un error en el reconocimiento de los patrones aproximadamente del 1%, esto se debe a la presencia de una intensa luz externa inesperada en el entorno del trabajo.

La red neuronal fue entrenada con una capa oculta y 40 neuronas en dicha capa, con este número de elementos la red tiene comportamiento estable, ya que en 50 pruebas realizadas el error es del 1% en el reconocimiento de los patrones.

Con el tiempo obtenido de 500 ms en el procesamiento se concluye que el sistema se podría aplicar eficientemente en la industria, ya que no se aleja de lo que se considera tiempo real.

5.2. Recomendaciones

Los sistemas de visión artificial con variables externas fijas como son iluminación, distancia y estabilidad angular, trabajan mejor debido a que sus fluctuaciones hacen experimentar grandes cambios en los resultados de los algoritmos de reconocimiento.

Las tarjetas de desarrollo con arquitectura ARM que no tienen un amplio uso a nivel mundial no son muy afines a trabajar autónomamente con adquisición de imágenes, por lo que se recomienda el uso de tarjetas más conocidas como es el caso de la Raspberry Pi.

Las comunicaciones seriales resultan en una mayor fluidez cuando se trata de compartir información entre múltiples tarjetas de desarrollo, como en el caso particular del presente trabajo entre Raspberry Pi, Arduino, STM y brazo Mitsubishi

Para la ecuación obtenida con regresión lineal aumenta su precisión mientras más puntos se registren en las pruebas de campo, ya que el método de ajuste de curvas funciona mejor si se tiene una gran cantidad de datos para su cálculo.

REFERENCIAS BIBLIOGRÁFICAS

- Alonso, J. L. (2012). Montaje y puesta en marcha de sistemas robóticos y sistemas de visión en bienes de equipo y maquinaria industrial. Andalucía: INNOVA.
- Cajamarca, J., & Portilla, A. (2016). *IMPLEMENTACIÓN DE UN CONTROLADOR PARA CINEMÁTICA INVERSA DEL BRAZO ROBÓTICO MITSUBISHI RV-2AJ A TRAVÉS DE UNA TARJETA ARM Y MATLAB*. Quito: Universidad Politécnica Salesiana.
- Caprile, S. R. (2012). Desarrollo con microcontroladores ARM. Buenos Aires: Puntolibro.
- Colgren, R. (2007). Basic MATLAB, Simulink, and Stateflow. Virginia: American Institute of Aeronautics and Astronautics, Inc.
- Logitech. (13 de 10 de 2017). *Logitech Webcam C170*. Obtenido de <https://www.logitech.com/es-mx/product/webcam-c170>
- López, J. A. (2014). Fotografía Digital. Madrid: Ministerio de Educación, Cultura y Deporte.
- Madrigal, R. I. (2002). Robots Industriales Manipuladores. Barcelona: Edicions UPS.
- Martin del Peso, M. (2001). APLICACIONES DE LAS REDES NEURONALES ARTIFICIALES A PROBLEMAS DE PREDICCIÓN FINANCIERA. 14-27.
- Pajares, G., & Santos, M. (2006). Ingeniería Artificial e Ingeniería del Conocimiento. Madrid: RA-MA Editorial.
- Pérez, R. (2012). Prototipo de adquisición de señales biológicas utilizando Arduino. La Habana: Editorial Universitaria del Ministerio de Educación Superior.
- Raspberry Pi Foundation. (12 de 10 de 2017). *Raspberry Pi 3 Model B*. Obtenido de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Raspberry Pi Foundation. (10 de 10 de 2017). *Raspberry Pi FAQs*. Obtenido de <https://www.raspberrypi.org/help/faqs/#introWhatIs>

Robotics Exchange. (11 de 12 de 2017). *Mitsubishi Robotic Arm RV-2AJ*. Obtenido de <http://www.mitsubishirobots.com/RV2AJ.html>

Rodríguez, M. G. (2003). *Introducción rápida a Matlab y Simulink para ciencia e ingeniería*. Madrid: Días de Santos S.A.