



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA
MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA,
OPCIÓN EN AUTOMATIZACIÓN

“Automatización de una Celda Robótica con Retroalimentación Visual”**

T E S I S

Presentada para obtener el título de:
Maestro en Ciencias de la Electrónica, Opción en Automatización

Presenta:

Ing. Edgar Hugo Mayoral Arzaba*

Directores:

Dra. Olga Guadalupe Félix Beltrán

Dr. Fernando Reyes Cortés

Dr. Jaime Cid Monjaraz

Puebla, México

Diciembre 2017

*BECARIO CONACYT

** TRABAJO FINANCIADO POR PROYECTO VIEP

BUAP®

Agradecimientos

En primera instancia agradezco a Dios por darme la vida, estar conmigo en todo momento, por ponerme pruebas día tras día que me demuestran que nunca estaré solo y por poner en mi camino a tanta gente que han sido pilar para afrontar la vida.

Agradezco a mi madre María Adela por ser mi apoyo moral cada día, a mi padre Carlos por demostrarme que la palabra rendirse no existe. A mi esposa Erika por cuidar mis desvelos y acompañarme en esta etapa de mi vida. A mi hermano quien me enseña el valor de la familia.

A todo el personal académico que conforma la Maestría en Ciencias de la Electrónica, opción Automatización por brindarme la oportunidad de continuar mi desarrollo profesional en sus instalaciones.

Agradezco a mis asesores de tesis, la Dra. Olga Guadalupe Félix Beltrán, al Dr. Jaime Cid Monjaraz y al Dr. Fernando Reyes Cortés por haberme brindado su amistad, su tiempo, compartir sus conocimientos y la paciencia necesaria para desarrollar esta tesis.

Agradezco al Consejo Nacional de Ciencias y Tecnología (CONACyT) por la beca de manutención otorgada para realizar mis estudios de maestría.

A mis compañeros y amigos de la Maestría por su amistad brindada, Manuel, Fernando, Gabino, Raúl, Francisco, Rigoberto, Geovanni, Daniel, Moya y Hermes.

Resumen

En este trabajo de tesis se realizó el diseño e instrumentación de una celda robótica que está conformada por un sistema transportador de cadena (banda transportadora), dos brazos robóticos de transmisión directa, un sistema de adquisición de datos y una cámara web convencional. Esta celda robótica se implementó para la identificación de tarjetas electrónicas usando algoritmos de aprendizaje automático. Se utilizó una tarjeta Arduino Mega para el control de posición de la banda transportadora y la tarjeta Arduino Due para el control de los ROTRADI's (Robot de transmisión directa). Además, se desarrolló el procesamiento digital de imágenes en conjunto con los algoritmos de aprendizaje automático, como parte del reconocimiento de las tarjetas electrónicas y la identificación de componentes de la tarjeta usando Visual Servoing.

Por otro lado, se desarrolló la instrumentación electrónica para los ROTRADI's de uno y tres grados de libertad. Adicionalmente, se desarrollaron los algoritmos de control de posición para cada uno de ellos. Para la transmisión de datos en la celda se implementó la tecnología WiFi para establecer la comunicación hacia una página de internet.

La integración del sistema se realizó por medio de una GUIDE (interfaz gráfica de usuario) en MATLAB. Como resultado se identificaron los componentes electrónicos de una PCB de densidad media, con un periodo de muestreo de 4 tarjetas por minuto, en caso de que la tarjeta identificada no sea la correcta o se encuentre incompleta el ROTRADI de 3 GDL la retira de la banda transportadora. La celda quedó instalada como una plataforma experimental para la validación de algoritmos de control para la banda transportadora o robots antropomorfos.

Índice general

Agradecimientos	II
Resumen	III
Introducción	X
1. Estado del Arte	1
1.1. Robot manipulador	2
1.1.1. Elementos que conforman al robot	2
1.1.2. Estructura mecánica	2
1.2. Instrumentación electrónica	5
1.3. Visual servoing	8
1.4. Procesamiento digital de imágenes	9
1.5. Bandas transportadoras	13
1.5.1. Banda transportadora de rodillos	13
1.5.2. Bandas modulares	14
2. Cinemática y dinámica de un ROTRADI	16
2.1. Cinemática y dinámica de un ROTRADI de un grado de libertad	18
2.1.1. Modelo dinámico del péndulo	21
2.2. Cinemática y dinámica de un ROTRADI de 3 grados de libertad	22
2.2.1. Modelo dinámico del ROTRADI de 3 GDL	25
3. Diseño mecánico e instrumentación electrónica	29
3.1. Banda transportadora	30
3.1.1. Diseño del soporte	32
3.2. Diseño electrónico de la banda transportadora	33
3.2.1. Caracterización del motor	33
3.2.2. Diseño de driver electrónico	34
3.2.3. Firmware del controlador	39
3.2.4. Algoritmo de temperatura	40
3.3. Diseño electrónico del ROTRADI	44

3.3.1. Lectura del encoder	44
3.3.2. Convertidor digital a analógico	48
4. Diseño de algoritmos de detección de movimiento, reconocimiento de características y detección de componentes	53
4.1. Algoritmo de detección de movimiento	55
4.2. Algoritmo de reconocimiento de características críticas	57
4.2.1. Algoritmo de aprendizaje automático	58
4.3. Algoritmo de búsqueda de componentes	71
4.3.1. Altium Designer	72
4.3.2. Implementación de algoritmo	73
5. Internet de las cosas	77
5.1. Internet de las cosas	77
6. Resultados	84
6.1. Resultados de la instrumentación electrónica	85
6.2. Estructura de control	87
6.2.1. Implementación de Guide	97
6.2.2. Resultados de ejecución	99
7. Conclusiones	109
A. Hojas de diseño del soporte del motor	112
B. Hoja de datos	115
C. Firmware para tarjetas Arduinos	119
C.1. Firmware PWM variable para control de la banda transportadora	119
C.2. Firmware para DS1920	121
C.3. Firmware conexión ThingSpeak	122
D. Códigos de MATLAB	124
D.1. Algoritmo de aprendizaje automatico	124
D.2. Algoritmo de la instrucción CarFinderLive	125
E. Comandos AT-ESP8266	127
F. Congresos y publicaciones	129
G. Solicitud de patente	156
Bibliografía	158

Índice de figuras

1.	Celda robótica implementada en el Laboratorio de Robótica de la Facultad de Ciencias de la Electrónica, BUAP.	XI
2.	(a) Cámara en mano, (b) cámara fija.	XII
3.	Diagrama de flujo de implementación de la celda robótica.	XIV
1.1.	Esquema general de la electrónica y la estructura mecánica que integran un robot.	3
1.2.	(a) Cinemática abierta y (b) cinemática cerrada.	3
1.3.	Articulación rotacional y articulación prismática.	4
1.4.	Espacio de trabajo de un robot antropomorfo de 3 grados de libertad.	4
1.5.	Relación brazo humano con brazo robótico.	5
1.6.	Clasificación de robots industriales.	6
1.7.	Elementos que integran el sistema electrónico.	7
1.8.	Motor de transmisión directa.	7
1.9.	Encoder.	8
1.10.	Indexado de una imagen.	10
1.11.	Obtención de una imagen binaria a partir de una imagen en escala de grises.	10
1.12.	Imagen binaria.	11
1.13.	Representación de una imagen a color RGB.	11
1.14.	Filtros para detección de bordes.	12
1.15.	Reconocimiento de objetos dentro de una imagen.	12
1.16.	Banda transportadora con rodillos.	13
1.17.	Bandas con rodillos motorizados.	14
1.18.	Bandas modulares.	15
2.1.	Esquema referencial del péndulo.	18
2.2.	Representación gráfica para obtener la cinemática directa del péndulo.	20
2.3.	Diagrama del ROTRADI para determinar la cinemática directa.	23
2.4.	Vista superior del robot, la posición final se encuentra desplazada por la proyección de los motores.	23
3.1.	Diagrama a bloques del funcionamiento de la celda robótica.	29

3.2. Soporte mecánico del motor.	30
3.3. Motor Crouzet.	32
3.4. Ensamble de motor.	33
3.5. Curva de eficiencia.	34
3.6. Caracterización del motor a 260 Hz.	34
3.7. Topología básica del convertidor Buck-Boost.	36
3.8. Módulo regulador de voltaje XL6009.	36
3.9. Esquemático de puente H.	37
3.10. Esquemático de fuente de voltaje.	38
3.11. Diseño de la tarjeta de control.	38
3.12. Diagrama de flujo del controlador.	39
3.13. Señal PWM de frecuencia de 260 Hz.	40
3.14. Sensor DS1920.	41
3.15. Protocolo de comunicación 1-Wire para el sensor DS1920.	42
3.16. Pulso de reset de sensor DS1920.	43
3.17. Diagrama de tiempo de comando de lectura de sensor DS1920.	43
3.18. Diagrama de electrónico de ROTRADI.	44
3.19. Diagrama de estados del encoder.	45
3.20. Señal del encoder proporcionada por el servoamplificador.	45
3.21. Esquemático para la lectura de la señal del encoder.	46
3.22. Esquemático de PCB para la lectura de la señal del encoder.	47
3.23. Diseño de PCB para la lectura de la señal del encoder.	47
3.24. Señal del encoder filtrada.	48
3.25. MCP4725 convertidor digital a analógico.	48
3.26. Protocolo de comunicación I^2C para escritura de datos.	49
3.27. Esquemático del DAC.	50
3.28. Diseño de la PCB del DAC.	50
3.29. Diseño de la PCB del DAC para proporcionar la señal de control hacia el ROTRADI.	51
3.30. Respuesta de tarjeta DAC 1.	51
3.31. Respuesta de tarjeta DAC 2.	52
3.32. Respuesta de tarjeta DAC 3.	52
4.1. Cámara web.	54
4.2. Colocación de la cámara web y la banda transportadora.	54
4.3. Diagrama de bloques de algoritmo de identificación de movimiento.	55
4.4. Algoritmo detector de movimiento 1.	56
4.5. Algoritmo detector de movimiento 2.	57
4.6. Algoritmo de Paul Viola.	58
4.7. Algoritmo de aprendizaje automático.	59

4.8. Extracción de características.	60
4.9. Tarjeta colocada sobre la banda transportadora.	60
4.10. Búsqueda de características.	61
4.11. Filtrado de características críticas.	61
4.12. Tarjeta identificada dentro de la imagen.	62
4.13. Guide de captura de imágenes.	63
4.14. Tarjeta Arduino Mega y Motum.	63
4.15. Tarjeta selector y programador.	64
4.16. Tarjeta controlador y programador.	64
4.17. Carpetas de imágenes de tarjetas electrónicas.	64
4.18. Vector de características críticas.	65
4.19. Matriz de características críticas.	65
4.20. Selección de matriz de características críticas.	66
4.21. Entrenadores de Clasificador.	67
4.22. Ejemplo de aplicación del algoritmo K-NN.	68
4.23. Tabla de confusión.	69
4.24. Diagrama de flujo del comando CarFinderLive.	69
4.25. Reconocimiento de tarjeta Controlador.	70
4.26. Reconocimiento de tarjeta Motum.	70
4.27. Reconocimiento de tarjeta Programador.	71
4.28. Reporte de componentes.	73
4.29. Algoritmo de reconocimiento de componentes electrónicos.	74
4.30. Proceso de filtrado para reconocimiento de componentes electrónicos.	74
4.31. PCB de referencia.	75
4.32. PCB completa.	75
4.33. PCB incompleta.	76
5.1. Módulo ESP8266.	78
5.2. Esquemático de tarjeta de comunicación Arduino Mega - ESP8266.	79
5.3. Shield Arduino Mega - ESP8266.	79
5.4. ESP8266 búsqueda de redes inalámbricas.	80
5.5. ESP8266 conexión a red inalámbrica.	81
5.6. Página web de Thingspeak.	81
5.7. APIKeys de conexión.	82
5.8. Diagrama de flujo del algoritmo de comunicación entre el módulo ESP82266 y el servidor ThingSpeak.	82
5.9. Envío de datos a ThingSpeak.	83
6.1. Integración de celda robótica.	84
6.2. Instrumentación electrónica para el control del ROTRADI.	85
6.3. Esquemático de fuente de voltaje lineal.	86

6.4. Fuente de voltaje lineal.	87
6.5. Control de posición PD aplicado al ROTRADI de 1 GDL.	89
6.6. Torque aplicado al ROTRADI de 1 GDL, utilizando el controlador PD.	90
6.7. Control de posición tanh aplicado al ROTRADI de 1 GDL.	92
6.8. Torque aplicado al ROTRADI de 1 GDL, utilizando el controlador tanh.	93
6.9. Control de posición PD aplicado al codo del ROTRADI.	94
6.10. Controlador PD, torque aplicado al codo del ROTRADI.	94
6.11. Control de posición PD aplicado al hombro del ROTRADI.	95
6.12. Controlador PD, torque aplicado al hombro del ROTRADI.	95
6.13. Control de posición PD aplicado a la base del ROTRADI.	96
6.14. Controlador PD, torque aplicado a la base del ROTRADI.	96
6.15. Guide implementada.	97
6.16. Posición de trabajo del ROTRADI de 3 GDL.	99
6.17. Activación del ROTRADI de 1 GDL.	100
6.18. ROTRADI de 1 GDL sujetando la tarjeta electrónica de prueba.	100
6.19. ROTRADI de 1 GDL colocando tarjeta de prueba sobre banda transportadora.	101
6.20. Reconocimiento de banda transportadora.	101
6.21. Tarjeta Entrenador, tarjeta lateral izquierda incompleta, tarjeta lateral derecha completa.	102
6.22. Reconocimiento de tarjeta Entrenador incompleta.	102
6.23. Tarjeta Entrenador incompleta.	103
6.24. Tarjeta Entrenador incompleta.	103
6.25. ROTRADI retirando la tarjeta Entrenador incompleta.	104
6.26. Tarjeta Entrenador incompleta.	105
6.27. Reconocimiento de tarjeta Entrenador completa.	105
6.28. ROTRADI colocando tarjeta Arduino sobre la banda transportadora.	106
6.29. Reconocimiento de tarjeta Arduino Mega.	106
6.30. ROTRADI retirando la tarjeta Arduino de la banda transportadora.	107
6.31. Guide implementada.	107
A.1. Soporte 1 del motor Crouzet.	113
A.2. Soporte 2 del motor Crouzet.	114

Introducción

La *automatización* es el estudio de los métodos y procedimientos cuya finalidad es la sustitución del operador humano por el operador artificial en la generación de una tarea física previamente programada. Así, la automatización es el estudio y la aplicación del control de procesos industriales y de gestión de la producción [1].

La investigación actual en el control de robots manipuladores se orienta fundamentalmente en el diseño de nuevos algoritmos de control, es una de las actividades en la que constantemente se está trabajando dentro del área de robótica, el fin es mejorar el desempeño de controladores experimentalmente aprobados. Es bien sabido que la mayoría de los manipuladores industriales que se usan actualmente están equipados con controladores simples, tales como el Proporcional-Derivativo (PD, por sus siglas en inglés) y el Proporcional-Integral-Derivativo (PID, por sus siglas en inglés), los cuales son efectivos para alcanzar el propósito de posicionamiento del efector final en una tarea específica [2].

En el área de control de robots manipuladores se incorpora información sensorial externa que con la utilización de algoritmos avanzados de control, permite la realización de tareas en medios parcialmente estructurados o con grandes incertidumbres. Dentro de los sensores utilizados en la actualidad, los de visión se convierten en una interesante alternativa gracias a la cantidad de información que proporcionan. El conjunto de brazos robóticos y bandas transportadoras combinados con sistemas de visión se han incluido en los últimos años en los procesos de automatización no sólo con fines de investigación, sino también a nivel industrial [3].

El problema del posicionamiento de los sistemas que usan la información visual ha sido un área de investigación en los últimos 30 años [2]. En años recientes, la atención a este tópico tiene un interés creciente. La información con retroalimentación visual puede resolver muchos problemas que limitan las aplicaciones. *Visual Servoing* hace referencia a un sistema en lazo cerrado para el control de posición de un sistema usando directamente la retroalimentación visual. Este término se introdujo por Hill y Park en 1979 [4] y fue empleado posteriormente por Peter I. Corke en su trabajo *Real-Time Vision, Tracking and Control* [5].

Actualmente existen sistemas constituidos por bandas transportadoras y cámaras de video de alta definición, para la identificación de objetos. Uno de los usos más comunes actualmente se encuentra en el sector alimenticio, estos sistemas son normalmente utilizados para la clasificación de frutas [6]. El principio de funcionamiento de estos sistemas está basado en un algoritmo de identificación de colores, tamaño o un conjunto de ambos. La principal problemática del algoritmo de identificación por color radica en que es necesario establecer un umbral de comparación, para poder identificar el producto. Si se desea utilizar este sistema para otro tipo de producto es necesario realizar un cambio en el algoritmo e implementar una nueva calibración del sistema. Por lo tanto el costo de producción incrementa.

En el Laboratorio de Robótica de la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla se encuentra una celda robótica conformada por 2 ROTRADI (Robot de transmisión directa), una banda transportadora de la empresa Rexroth y una cámara web, tal como se muestra en la figura 1.



Figura 1: Celda robótica implementada en el Laboratorio de Robótica de la Facultad de Ciencias de la Electrónica, BUAP.

La transmisión directa consiste en que el rotor del motor funciona como una fuente de par, por lo tanto, el sistema de engranes es eliminado, reduciendo el cascabeleo y la fricción; la construcción mecánica es más simple y la exactitud en el posicionamiento de robots manipuladores es mayor [2]. Para el control de posición del ROTRADI se implementan estructuras de control que garanticen un excelente desempeño, es decir, que el movimiento sea suave y sin oscilaciones. Para implementar el algoritmo de control se desarrolló una interfaz electrónica que permita la interacción entre el hardware del ROTRADI y el hardware de la estructura de control.

La banda transportadora está conformada de la cinta transportadora, la estructura física y el sistema de transmisión. Se selecciono un motor que permite el correcto funcionamiento de la banda transportadora, para el control de velocidad y cambio de giro se desarrolló una interfaz electrónica que permite la interacción con el usuario.

La cámara web tiene como función ser el instrumento de adquisición de imágenes para el sistema de Visual Servoing. Existen dos modos de configuración, cámara en mano o cámara fija, tal como se muestra en la figura 2. El sistema de cámara en mano, la cámara se coloca en el extremo final del brazo robótico acompañado de la herramienta de trabajo, tiene como objetivo posicionar el brazo en el punto deseado sin considerar elementos externos. La configuración cámara fija se coloca la cámara en una posición que permita visualizar todo el área de trabajo del brazo robótico, considerando aspectos externos [7].

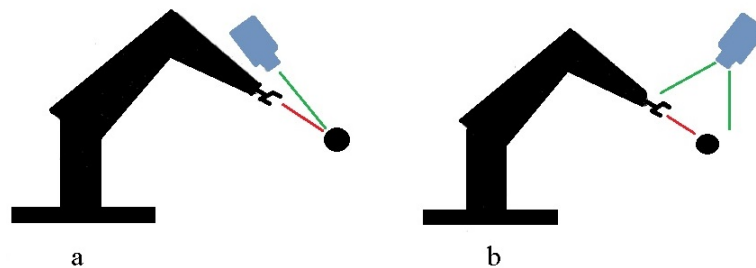


Figura 2: (a) Cámara en mano, (b) cámara fija.

La importancia de este trabajo radica en la marcada carencia de sistemas constituidos por celdas robóticas y sistemas de visión a nivel internacional y sobre todo a nivel nacional. Con este trabajo se realiza un aporte de carácter científico tecnológico, combinando tecnologías de vanguardia y de bajo costo.

El objetivo general para el presente trabajo de tesis es el siguiente:

Desarrollar la automatización de una celda robótica compuesta por una banda transportadora y dos robots de transmisión directa usando Visual Servoing.

Los objetivos particulares tenemos los siguientes:

1. Desarrollar la automatización de una banda transportadora que interactúe con una serie de robots de transmisión directa mediante el uso de una computadora, un entorno de desarrollo capaz de realizar la adquisición de datos y control por medio de una interfaz con puertos de entradas y salidas, estableciendo de esta forma el control deseado.

2. Proponer un algoritmo de control para la banda transportadora utilizando como elemento de retroalimentación fundamental la información visual.
3. La estrategia deberá de tener sustento teórico con un análisis matemático formal de estabilidad y validación experimental del desempeño de forma práctica para verificar su funcionamiento.
4. Generar al menos una publicación internacional en la cual se presenten las aportaciones realizadas en este trabajo de tesis.

Se utiliza el sistema *Visual Servoing* para el diseño y construcción de una celda robótica, el cual está constituido por una banda transportadora, dos robots de transmisión directa *ROTRADI* y un sistema de adquisición de datos (cámara web, tarjetas electrónicas de adquisición y una computadora) para la identificación y selección de tarjetas electrónicas. Por medio de un brazo robótico se colocan las tarjetas electrónicas sobre la banda transportadora. Cuando la tarjeta electrónica se encuentra en el área visible de la cámara web, se ejecuta el algoritmo de reconocimiento, este algoritmo extrae las características más importantes de cada tarjeta electrónica y son comparadas con una base de datos. Esta base de datos es previamente entrenada y contiene todas las características de cada una de las tarjetas que se desean reconocer. El proceso de reconocimiento se realiza de manera automática, obteniendo como resultado que no exista la necesidad de cambiar algún parámetro en el algoritmo o realizar una nueva calibración del sistema. Posteriormente se verifica que la tarjeta electrónica contiene todos sus componentes físicos en la posición correcta. Cuando la tarjeta identificada no corresponde a la tarjeta deseada o presenta ausencia de algunos de sus componentes, es retirada de la banda transportadora por medio un *ROTRADI*.

En la figura 3 se muestra el diagrama de flujo del sistema propuesto, en el cual se describen paso a paso las actividades que se realizaron para la automatización de la celda robótica.

Considerando la estructura física de una tarjeta electrónica, el índice de complejidad de este proyecto resulta demasiado alto, en cambio se garantiza que el sistema pueda funcionar de manera adecuada con otros procesos más simples. Los algoritmos de aprendizaje automático permiten identificar diferentes tipos de objetos, dando como resultado una gran variedad de aplicaciones.

Este documento está estructurado de la siguiente manera: en el capítulo 1 se incluye el estado del arte, definiciones y trabajos realizados previamente conforme al tema. En el capítulo 2 se presenta el estudio del modelo dinámico de un robot de 3 grados de libertad, en donde se obtienen los parámetros necesarios para los esquemas de control. En el capítulo 3 se presenta el diseño mecánico y la instrumentación electrónica para la banda transportadora y el *ROTRADI*, se realizan las tarjetas de control para cada elemento. En el capítulo 4 se

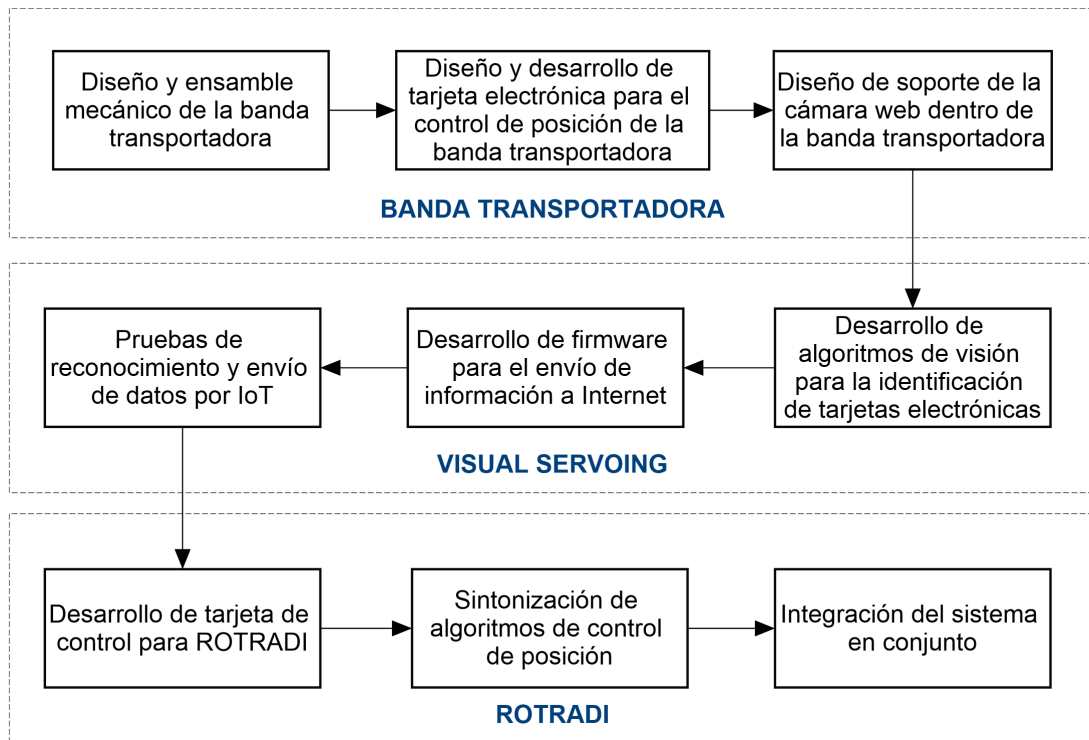


Figura 3: Diagrama de flujo de implementación de la celda robótica.

presentan los algoritmos de detección de movimiento, reconocimiento de características y detección de componentes electrónicos, estos algoritmos son implementados en el plataforma de MATLAB. En el capítulo 5 se muestran los pasos necesarios para conectar una tarjeta de adquisición a una página web, utilizando la metodología del Internet de las Cosas. En el capítulo 6 se muestran los resultados obtenidos de este trabajo, mostrando las gráficas de control de posición del ROTRADI, la GUIDE desarrollada en MATLAB para la integración de la celda robótica y la publicación de resultados en una página web. En el capítulo 7 se muestran las conclusiones del trabajo desarrollo. En el apéndice A se muestra los dibujos técnicos de las piezas diseñadas para la banda transportadora. En el apéndice B se incluyen las hojas de datos de los componentes electrónicos utilizados en el diseño de las tarjetas electrónicas. En el apéndice C se muestra el firmware desarrollado para la tarjeta Arduino Due para la comunicación con sensores y envío de información por medio de IoT. En el apéndice D se muestran los algoritmos desarrollados en la plataforma de MATLAB para la implementación del algoritmo de aprendizaje automático. En el apéndice E se incluyen los comandos AT para la comunicación con el módulo ESP8266. En el apéndice F se muestra los reconocimientos obtenidos en la participación de congresos nacionales e internacionales. En el apéndice G se muestra el registro de patente *Sistema supervisor de motores eléctricos en tiempo real*.

Capítulo 1

Estado del Arte

Hoy en día la ciencia y la tecnología representan áreas estratégicas para el avance de la humanidad. Los niveles de competitividad para poder incidir en los mercados nacionales o internacionales son de aquellos países los cuales tienen la capacidad de producir tecnología de vanguardia, originando a su paso una mayor calidad de vida a sus habitantes. Para que un país sea capaz de desarrollar tecnología es necesario que cuente con sectores de carácter científico y tecnológico, para poder generar respuesta a las exigencias que demanda el mercado [2].

La Mecatrónica es una área multidisciplinaria que involucra las siguientes áreas: sistemas electrónicos, robótica, sistemas de control, programación, informática, sistemas mecánicos, diseños de CAD, componentes mecánicos y administración de proyectos [8]. Esta fuertemente ligada a la automatización, la cual denota una tecnología dirigida a suplir actividades de los seres humanos por máquinas en un proceso de fabricación rutinario o de alto riesgo, considerando no sólo la ejecución de operaciones físicas sino también el procesamiento adecuado de información sobre el estado del proceso. Es un área clave del desarrollo en la sociedad moderna, sinónimo de modernización tecnológica, por su naturaleza multidisciplinaria, científico-práctica, permite generar y aplicar conocimientos y/o tecnologías a problemas reales del entorno [9].

La robótica es un área de investigación y desarrollo de sistemas mecánicos conocidos como robots, que de acuerdo con el Robot Institute of America (RIA) un robot es un manipulador multifuncional reprogramable, diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para la ejecución de una variedad de tareas [10].

En este capítulo se presenta una descripción de la robótica, las bandas transportadoras y los sistemas de visión por computadora.

1.1. Robot manipulador

La palabra robot deriva de la palabra “*robot*” de origen eslavo, contiene dos connotaciones, en ruso significa “*trabajo*” y en checo significa “*trabajo forzado*”. El término robot fue introducido por primera vez por el escritor Karol Capek en su obra *Rossums Universal Robots*, mientras que el término robótica se debe a Isaac Asimov, famoso escritor de ciencia ficción [11].

El primer brazo articulado fue construido en 1938 por Harol Roselund, de la compañía Devilbiss, ahora desaparecida. Usado para pintar, no tuvo mucho éxito debido a la dificultad para programar una tarea. La revolución de la robótica vino por la revolución en la computación, y de esta forma, en 1954 George Devol diseñó el primer robot programable, razón por la cual es llamado el padre de la robótica [11]. En 1961 se diseñó el primer robot con retroalimentación; para 1963 se integró el primer sistema de visión robótica, y en 1971 se desarrolló el primer robot articulado en la Universidad de Stanford, misma en la que se desarrolló el primer lenguaje de programación para robots, el WAVE.

El primer robot controlado por microcomputador fue producido por Cincinnati Milacron en 1973. En 1978 Unimation desarrolló el PUMA (inicales de Programmable Universal Machine for Assembling), usado en la automatización en los procesos de ensamble automotriz [12].

En 1981 se desarrolló el primer robot de transmisión directa en la Universidad de Carnegie-Mellon. En la actualidad, muchas empresas de Estados Unidos, Alemania y Japón desarrollan robots industriales comerciales como: Hitachi, Kuka, Asea, Fanuc, ABB, entre otros [11].

1.1.1. Elementos que conforman al robot

Un esquema general del sistema del robot manipulador se presenta en la figura 1.1, donde se observan las partes básicas que lo conforma, las cuales se agrupan en: la estructura mecánica y la instrumentación electrónica.

1.1.2. Estructura mecánica

Los robots manipuladores son esencialmente brazos articulados; en forma más precisa es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante juntas. Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones. A continuación se describen algunos conceptos básicos necesarios para comprender el diseño de un robot antropomorfo.

- **Eslabón:** Es un cuerpo rígido¹ que posee por lo menos dos nodos, que son los puntos

¹Un cuerpo rígido es aquel en el cual ninguna parte de su masa sufre cambios de posición con respecto a

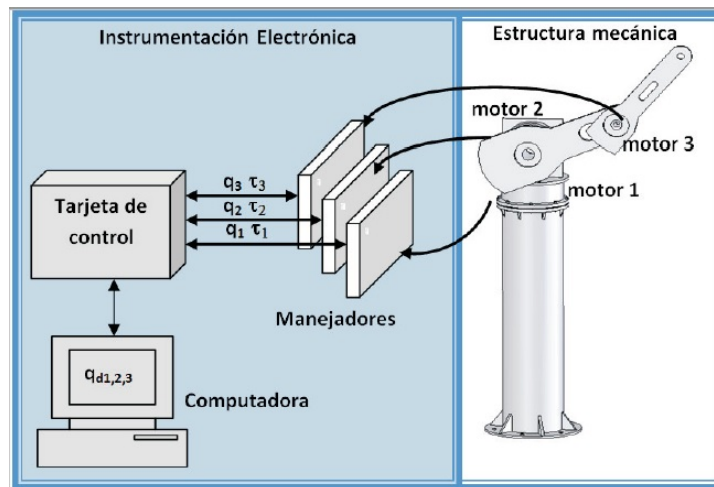


Figura 1.1: Esquema general de la electrónica y la estructura mecánica que integran un robot.

de unión con otros elementos.

- **Articulación:** Es una conexión entre dos o más eslabones, la cual permite un movimiento rotacional o movimiento lineal de traslación.
- **Cadena cinemática:** Un robot manipulador está compuesto por una serie consecutiva de eslabones y articulaciones. La cinemática abierta se dice que existe solo cuando hay una secuencia de eslabones sin que las dos puntas terminales de la cadena desde la base hasta el extremo final se unan, es decir, que no formen un lazo cerrado, de otra manera se considera una cadena de cinemática cerrada, tal como se muestra en la figura 1.2.

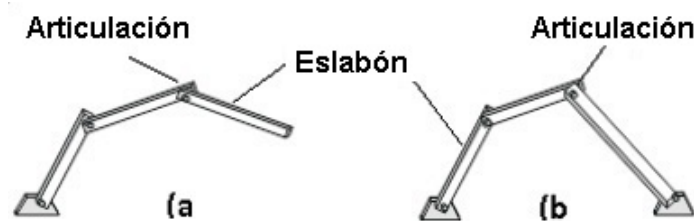


Figura 1.2: (a) Cinemática abierta y (b) cinemática cerrada.

- **Grados de libertad (g.d.l):** Cada articulación contribuye con un grado de libertad, siendo n el número de articulaciones que corresponde al número de grados de libertad.
- **Articulaciones:** Las articulaciones pueden producir movimiento rotacional o movimiento lineal de traslación. Las articulaciones que producen movimiento giratorio o las demás partes, independientemente de la fuerza a que esté sometido.

rotacional se les denomina *articulaciones rotacionales*. Por otro lado, las que producen movimiento lineal se les denomina *articulaciones prismáticas o lineales*. En la figura 1.3 se muestran los dos tipos de articulaciones.

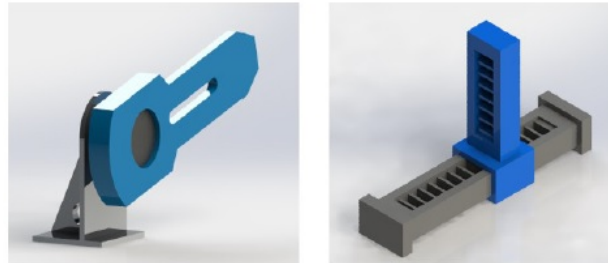


Figura 1.3: Articulación rotacional y articulación prismática.

- **Espacio de trabajo de un robot:** Es el espacio o lugar donde el robot puede realizar todos sus posibles movimientos. El espacio de trabajo está determinado por la geometría del robot y la naturaleza de sus articulaciones, en caso del robot antropomorfo su espacio de trabajo se muestra en la figura 1.4.

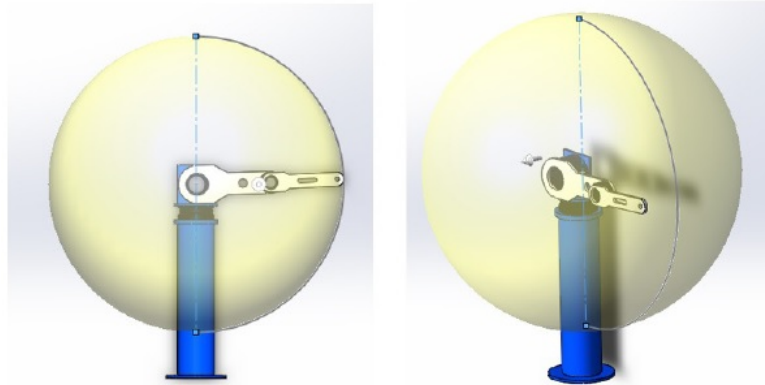


Figura 1.4: Espacio de trabajo de un robot antropomorfo de 3 grados de libertad.

- **Efactor final:** Es el último eslabón, destinado a colocar la herramienta adecuada para realizar una tarea específica. La posición del extremo final se representa por $[x, y, z]^T$ y su orientación se denota a través de los ángulos de Euler.
- **Analogía brazo humano y brazo robótico:** En la figura 1.5 se muestra la analogía entre el brazo humano y un brazo robot, la articulación de la base corresponde a la cintura, la articulación del hombro debe ser la de mayor capacidad con respecto a las otras articulaciones, debido a que soporta el peso de la articulación del codo y de la

herramienta de trabajo, así como también el peso de objetos que realice durante una determinada aplicación.

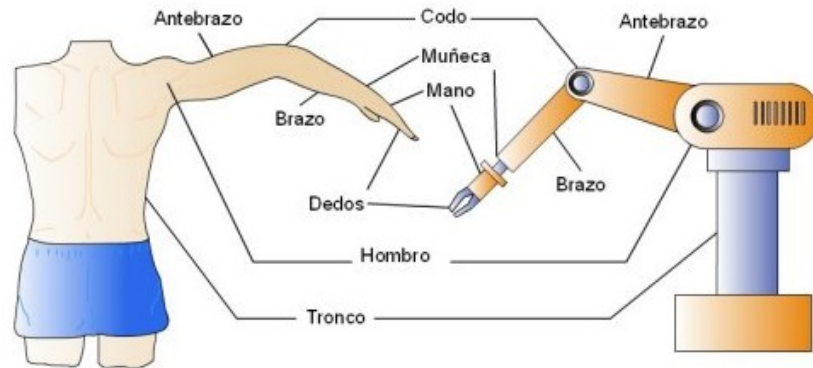


Figura 1.5: Relación brazo humano con brazo robótico.

- Cinemática:** Los robots presentan diferentes formas y tamaños y su clasificación puede estar basada en sus movimientos de configuración geométrica [13]. Dentro de los cinco diseños básicos de robots industriales se pueden identificar las configuraciones: cartesiana o rectangular, robot tipo Scara, configuración cilíndrica, configuración polar o esférica y configuración antropomorfo, tal como se muestra en la figura 1.6. La gran mayoría de los robots industriales tienen la configuración articular debido a que presenta mayor versatilidad en sus movimientos.

1.2. Instrumentación electrónica

Este apartado tiene como objetivo la descripción de los dispositivos de mando y control que integran el sistema electrónico de control de posición de un brazo robótico de tres grados de libertad, como el que se muestra en el diagrama en la figura 1.7.

- Computadora:** En la PC se lleva a cabo la programación de los algoritmos de control utilizando una interfaz gráfica la cual se encuentra desarrollada en la plataforma de MATLAB. Con esta aplicación se realiza el monitoreo y revisión de las variables de posición, velocidad y par aplicado del robot.
- Interfaz de comunicación:** Para gestionar la transferencia de datos entre los amplificadores de potencia y el programa de control es necesario contar con una tarjeta de control y la electrónica necesaria. Debe procesar la señal del control de la computadora y proporcionarla a los controladores, al mismo tiempo interpreta la señal de los codificadores de posición.

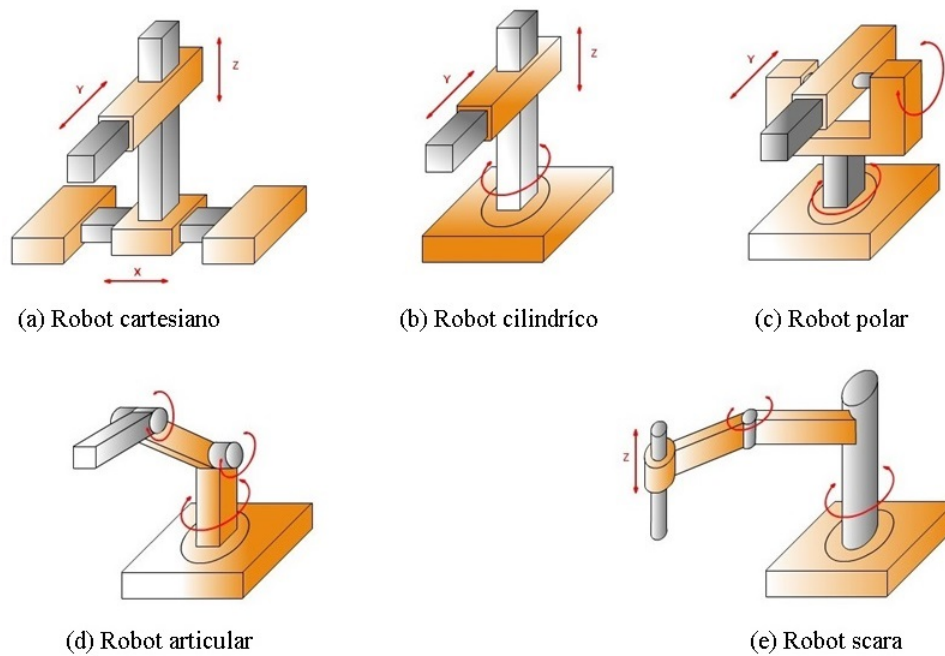


Figura 1.6: Clasificación de robots industriales.

- **Amplificador de potencia:** Los amplificadores se encargan de recibir las señales de retroalimentación de los decodificadores de posición (*encoder*) integrados en el interior de cada motor, así como de proporcionar las señales de potencia necesarias para excitar las fases del motor, de esta manera provocar un movimiento que se refleje en el cambio de posición del robot.
- **Motores:** El servomotor es el actuador que proporciona movimiento rotacional a la articulación, los motores pueden ser de corriente directa o corriente alterna, según las necesidades de torque y velocidad. Los motores de transmisión directa representan una fuente de par, es decir, el par que proporciona a la salida se mantiene en magnitud y signo independiente de la carga que desplace. Los servomotores de transmisión directa tienen tres modos de operación: posición, velocidad y par. El modo posición permite mover al motor a una posición preestablecida, sin embargo no puede desplazar cargas o aplicar una fuerza determinada. El modo velocidad controla el movimiento del motor sobre una velocidad deseada, tampoco puede ejercer fuerza como en el caso del modo de posición. Estos modos tienen la característica de ser arquitectura cerrada, es decir, no permiten programar otro tipo de controladores, por lo que sus aplicaciones en robótica se encuentran limitadas. El modo par es el que se emplea en robótica y su principal característica es que se encuentra en arquitectura abierta, lo que hace posible evaluar experimentalmente el desempeño y robustez de cualquier estrategia de control, en la figura 1.8 se muestra un servomotor de transmisión directa.

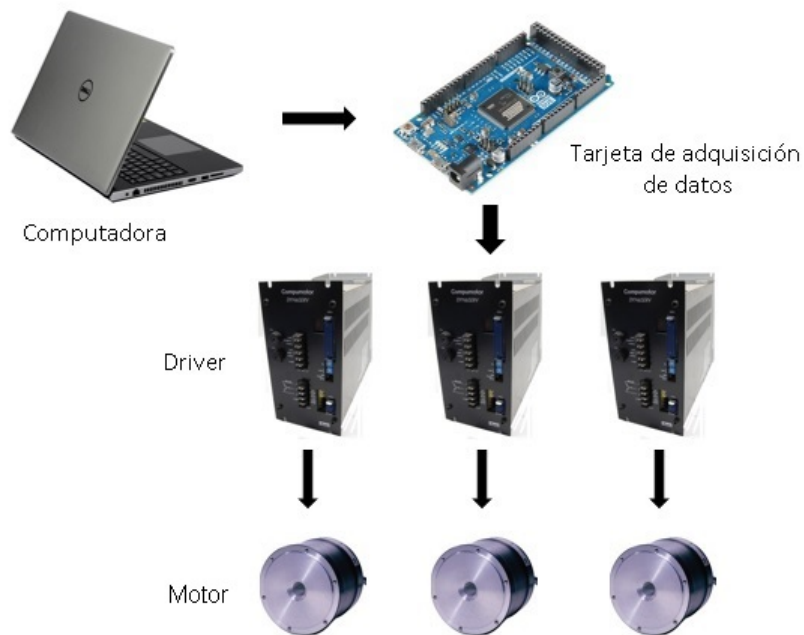


Figura 1.7: Elementos que integran el sistema electrónico.



Figura 1.8: Motor de transmisión directa.

- Decodificador de posición:** El decodificador de posición es un dispositivo que convierte movimiento en pulsos digitales. Este puede dar la posición relativa por medio de una secuencia de bits o la posición absoluta por medio de bits codificados, por ejemplo el código *Grey*. La tecnología empleada en los encoders es de tipo optoelectrónica por lo que se les denomina encoders ópticos, los encoders puede clasificarse en encoders incrementales o encoders absolutos. Un encoder óptico consta de una fuente de luz conformada por un arreglo de diodos LED que se encuentran enfrente de un disco giratorio con ranuras separadas y bien distribuidas, tal y como se muestra en la figura 1.9.

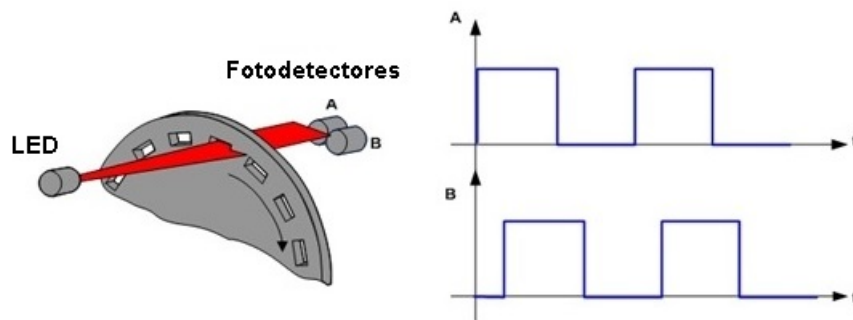


Figura 1.9: Encoder.

Como fuente de luz se utilizan diodos LED, los cuales emiten rayos infrarrojos, que a su vez pasa por las ranuras del disco giratorio hacia los elementos foto-detectores que se encuentran en el respaldo del disco. El disco giratorio está acoplado en el rotor del servomotor para que gire de la misma forma y por lo tanto al girar el disco interrumpe el haz de luz, cambiando de estado de salida, de ahí que el disco se le conoce como disco foto-interruptor, entonces el encoder óptico puede producir como señal de salida un tren de pulsos proporcional al ángulo de giro del rotor (encoder incremental) o producir una palabra digital codificada en formato binario (encoder absoluto).

1.3. Visual servoing

Visual Servoing hace referencia a un sistema en lazo cerrado para el control de posición del extremo final de un robot usando directamente la retroalimentación visual [14]. Este término se introdujo por Hill y Park en 1979 [4]. Representa una solución atractiva para posicionar y mover robots manipuladores autónomos que evolucionan en los ambientes no estructurados.

En *Visual Servoing* de Weiss [15] y Williams [16] se han identificado dos clases de control con base en visión para el robot: el de retroalimentación visual basada en posición y el de imagen. Dentro de las características de este último se extrae las características principales de la imagen y se estima la posición del blanco con respecto a la cámara. Usando estos valores, se crea una señal de error entre la posición actual y la posición deseada del robot dentro del espacio de trabajo.

En la configuración entre la cámara y robot, puede tenerse como cámara fija o cámara en mano. Los sistemas robóticos de cámara-fija tienen un sistema de visión fijo en el marco de la coordenada general donde captura imágenes del robot y su ambiente. El objetivo del control es mover el extremo final del robot de tal manera que alcance un punto deseado. En la configuración de la cámara-en-mano, a menudo llamada también ojo-en-mano, generalmente

una cámara está montada en el extremo final del robot proporcionando información visual del ambiente. En esta configuración, el objetivo del control es mover el efector-final del robot de tal manera que la proyección del blanco estático siempre se encuentre en una situación deseada en la imagen dado por la cámara.

Desde que los primeros sistemas de *Visual Servoing* que se reportaron en los inicios de 1980s estos han ido en aumento y en los últimos años se incrementó también en los trabajos publicados. Una apreciación global excelente de los problemas principales en *Visual Servoing* en el control de robots manipuladores lo estudia Corke [17].

Sin embargo, pocos estudios resultan ser rigurosos, donde incorporen la dinámica no lineal del robot en su contenido. La primera solución explícita al problema formulada en una publicación es debida a Miyazaki y Masutani en 1990, donde el esquema de control para entregar acciones acotadas se basa en un nuevo algoritmo de control mediante retroalimentación visual, utilizando la filosofía del Jacobiano transpuesto introducida por Takegaki y Arimoto [18].

Rafael Kelly se orienta al estudio de *Visual Servoing* de los robots manipuladores planares bajo la configuración cámara-fija [19]. Park y Lee en 2003 presentan un control por retroalimentación visual para una pelota en un plato para rastrear su trayectoria deseada [20]. Rafael Kelly propone un nuevo acercamiento, se dirige en el campo de la aplicación del control en la velocidad, utilizando la retroalimentación visual en la configuración cámara-fija para el robot manipulador [21].

Malis y Benhimane [22] (2005) presentan un sistema genérico y flexible para el control del robot basado en visión, el sistema integra el rastreo visual y la retroalimentación visual se acerca a un cuadro de trabajo unificado.

1.4. Procesamiento digital de imágenes

Una imagen es una representación de un objeto de tres dimensiones [23]. Una imagen digital es una imagen representada en un plano de dos dimensiones conformada por filas y columnas [23]. En MATLAB una imagen a escala de grises es representada por medio de una matriz bidimensional de $m \times n$ elementos en donde n representa el número de píxeles de ancho y m el número de píxeles de largo. El elemento $1,1$ corresponde al elemento de la esquina superior izquierda (ver figura 1.10), donde cada elemento de la matriz de la imagen tiene un valor de 0 (negro) a 255 (blanco) [23].

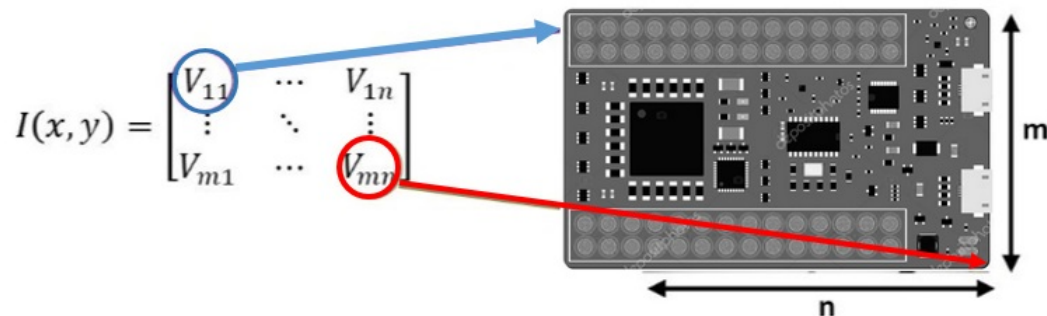


Figura 1.10: Indexado de una imagen.

Una imagen binaria es una imagen en la cual cada píxel puede tener solo uno de dos valores posibles 1 o 0. En visión por computadora la obtención de imágenes binarias es mediante la utilización del valor del umbral de una imagen a escala de grises, es decir, se elige un valor límite a partir del cual todos los valores de intensidad mayores serán codificados como 1, mientras que los que estén por debajo serán codificados a cero. En la figura 1.11 se muestra la obtención de una imagen binaria a partir de una imagen en escala de grises [24].



Figura 1.11: Obtención de una imagen binaria a partir de una imagen en escala de grises.

Una imagen binaria de un mismo objeto puede presentar diferentes resultados dependiendo del valor del umbral seleccionado, a este proceso se le denomina segmentación. En la figura 1.12 se muestra la imagen de una tarjeta electrónica, se aplica el proceso de segmentación, cuando el valor del umbral seleccionado es muy bajo, la imagen obtenida se asemeja a la imagen original. Cuando el valor es demasiado alto, la imagen obtenida se va distorsionando. Este proceso es utilizado para eliminar el ruido dentro de una imagen, pero es necesario elegir un valor correcto para no eliminar datos importantes para el procesamiento.

Una imagen de color RGB (por sus siglas en inglés Red, Green y Blue) se encuentra representada por una matriz tridimensional m, n, p en MATLAB, donde m y n tienen la misma representación para el caso de las imágenes en escala de grises, mientras p representa un plano

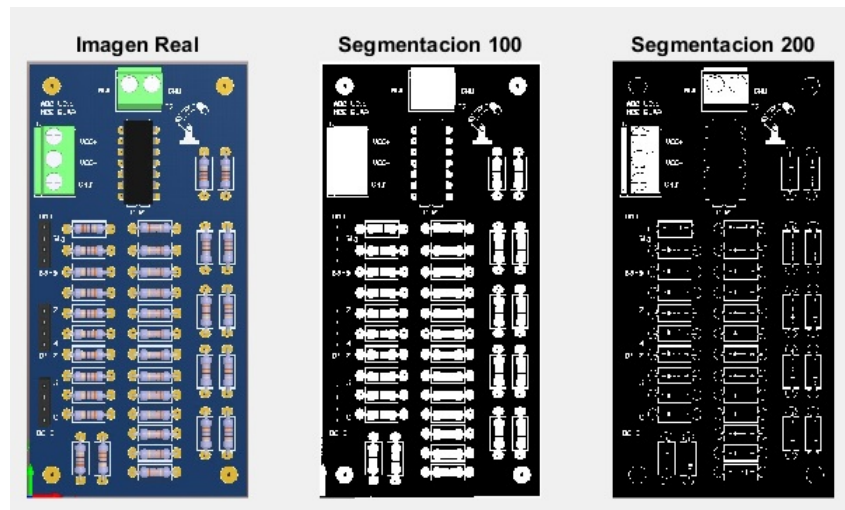


Figura 1.12: Imagen binaria.

de la imagen RGB, donde el plano 1 corresponde a los valores del color rojo, el plano 2 para el color verde y el plano 3 para el azul, el conjunto de estos planos permiten formar una imagen a color. La figura 1.13 se muestra una imagen a color y su representación en cada una de las capas [24].

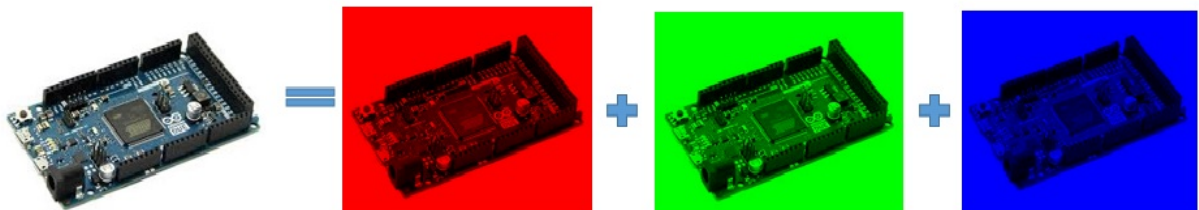


Figura 1.13: Representación de una imagen a color RGB.

En visión computacional es de gran utilidad realizar el reconocimiento de objetos por medio de la extracción de bordes. Existen diferentes algoritmos que permiten realizar este tipo de operaciones, entre los algoritmos más reconocidos en MATLAB se encuentran los siguientes: *filtro Canny*, *filtro Prewitt*, *filtro Sobel* y *filtro Roberts*. En la figura 1.14 se muestran los filtros Canny y Prewitt. El filtro Canny se basa en el cambio de contraste, por lo que resulta ser sensible a cambios de tonalidad. El filtro Prewitt tiene como base reconocer la continuidad de puntos de forma horizontal o vertical, descartando aquellos que se presenten en forma diagonal, este filtro es menos sensible que el filtro Canny [23].

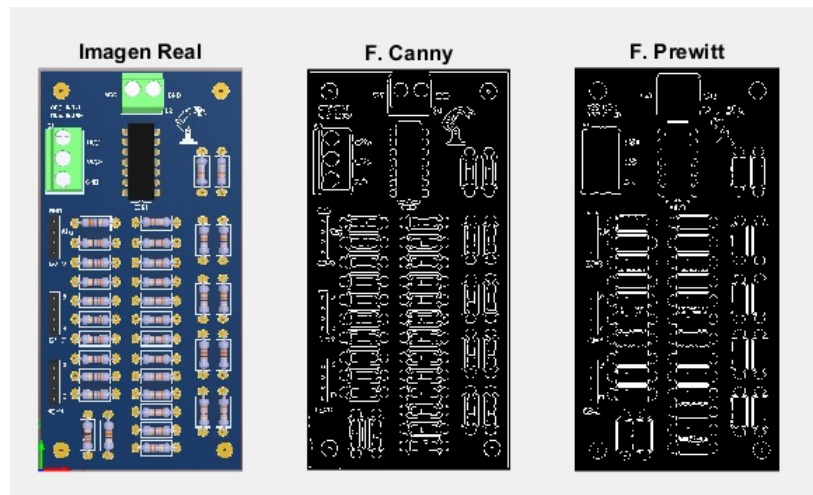


Figura 1.14: Filtros para detección de bordes.

En visión por computador resulta de utilidad poder aislar objetos de una imagen binaria, puede definirse un objeto como un conjunto de píxeles conectados con valor a 1. Para muchas operaciones, la distinción de objetos depende de la convención utilizada de conectividad, que es la forma en la que se considera si dos píxeles tienen relación como para considerarse que forman parte del mismo objeto. De esta manera se puede distinguir diferentes objetos dentro de una imagen. En la figura 1.15 se muestra una imagen, la cual contiene dos objetos dentro de ella [23].



Figura 1.15: Reconocimiento de objetos dentro de una imagen.

Estas son algunas de las propiedades básicas que rigen los algoritmos de procesamiento de imágenes, a partir de las cuales se pueden desarrollar diferentes algoritmos para dar solución a una gran variedad de problemas. Todos estos algoritmos se encuentran implementados y desarrollados en la plataforma de MATLAB, por lo que resulta un software bastante eficiente para el diseño, desarrollo e implementación de algoritmos basados en el procesamiento de imágenes.

1.5. Bandas transportadoras

Una banda transportadora es un sistema de transporte continuo formado básicamente por una banda continua que se mueve entre dos tambores, es arrastrada por fricción por uno de los tambores, que a su vez es accionado por un motor. El otro tambor suele girar libre, sin ningún tipo de accionamiento, y su función es servir de retorno a la banda. Algunas veces la banda es soportada por rodillos entre los dos tambores. Las bandas y rodillos transportadores son elementos auxiliares de las instalaciones cuya misión es la de recibir un producto de forma relativamente continua y conducirlo a otro punto.

Se han inventado muchas formas para el transporte de materiales, materias primas, minerales y diversos productos, pero una de las más eficientes es el transporte por medio de bandas y rodillos transportadores tal como se muestra en la figura 1.16, ya que estos elementos son de una gran sencillez de funcionamiento y una vez instalados en condiciones normales suelen dar pocos problemas mecánicos y de mantenimiento.



Figura 1.16: Banda transportadora con rodillos.

Existen diferentes tipos de bandas transportadoras, de acuerdo a sus características físicas pueden ser clasificadas de manera diferente.

1.5.1. Banda transportadora de rodillos

- **Bandas con rodillos:** Generalmente estos transportadores son más resistentes y duraderos que las bandas transportadoras, son capaces de transportar cargas más pesadas, cargas que podrían desgarrar, generar abrasión o derretir una banda transportadora textil convencional. Estos sistemas se constituyen como una forma alternativa a la cinta transportadora pues pueden utilizarse en diversas aplicaciones. Para la motorización de los rodillos existen diversas alternativas. Existen tres topologías básicas de sistemas motorizados, cada uno con determinadas ventajas.

- **Banda de rodillos por correa deslizante:** Cada rodillo se conecta a un eje motriz mediante una correa textil, de rodillo a rodillo en bucle o tangencial a todos los rodillos, de modo que el movimiento se produce al deslizarse las cajas sobre una cama de rodillos giratorios. Este tipo de transportador se adecúa a las cargas más ligeras, tiene una ventaja interesante en la medida que se constituye como un transportador de acumulación automática.
- **Banda de rodillos por cadena y piñón:** Cada rodillo está conectado a una cadena que se extiende a lo largo del transportador tangencial a todos los rodillos o de rodillo a rodillo en bucle que se conecta finalmente, al equipo motriz. Esto asegura que cada rodillo gira a la misma velocidad y es impulsado con firmeza. Es un método habitual para mover cargas pesadas que no harían más que sobrecargar un transportador por correa deslizante y continuamente provocaría solo deslizamiento de la correa sin desplazar la carga. Una variante de este sistema sería el constituido por piñón y correa dentada.
- **Banda de rodillos motorizados:** Se utilizan para el transporte de materiales muy pesados u objetos voluminosos. En este sistema, cada rodillo tiene un motor interno, de modo que no sólo pueden entregar más potencia que un sistema basado en la cadena de transmisión, sino también permitir “zonas de acumulación” xpor conmutación electrónica. Es posible manejar diferentes velocidades en diferentes secciones del transportador o incluso detenerlos por completo de forma independiente, tal como se muestra en la figura 1.17.



Figura 1.17: Bandas con rodillos motorizados.

1.5.2. Bandas modulares

Se trata de bandas no continuas, es decir, formadas por módulos y cuya característica principal es la utilización de piñones de accionamiento. Las bandas modulares están fabricadas con módulos termoplásticos, moldeados por inyección y unidos por sólidas varillas plásticas,

el diseño de todos los componentes plásticos prolongan la vida útil de las bandas y permiten obtener un rendimiento superior en múltiples aplicaciones. Entre las características más importantes de las bandas modulares se encuentran las siguientes:

- Buena tracción y alineación debido al arrastre por piñones.
- Bajo coste de mantenimiento y número de paradas por avería (no necesita mano de obra especializada).
- Todas las referencias son aptas para alimentación.
- Gracias a la superficie abierta permiten el drenaje en procesos húmedos.
- Posibilidad de transporte curvo.
- Determinar el número de piñones por eje en función de la tensión de trabajo.
- Fijación del piñón central del eje motriz para alinear la banda.

Algunos modelos de bandas transportadoras modulares son las que se muestran en la figura 1.18, donde se puede observar que las bandas están constituidas por la unión de varios elementos.

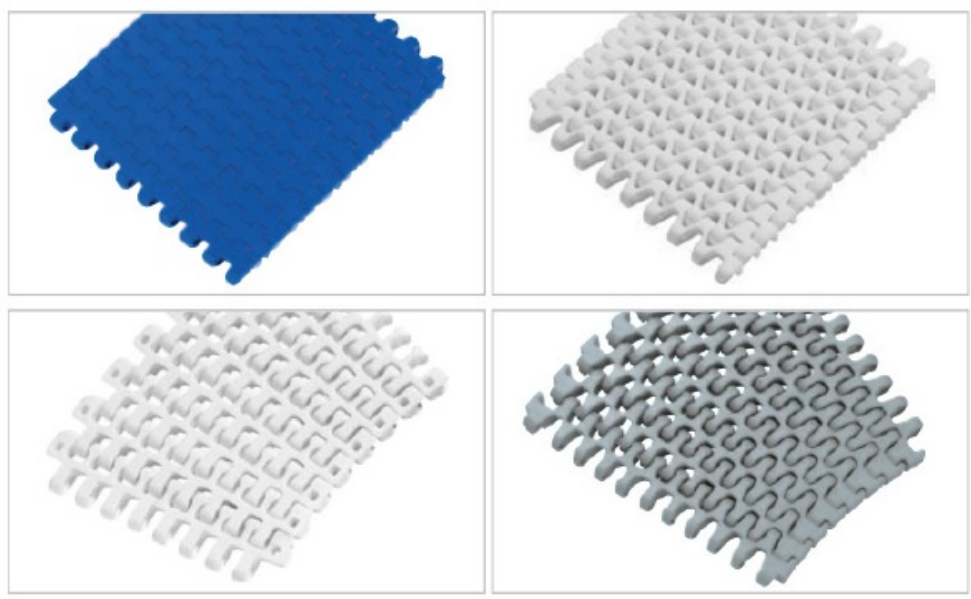


Figura 1.18: Bandas modulares.

La banda transportadora que se utilizó en el desarrollo de esta tesis corresponde a una banda modular de la compañía Rexroth.

Capítulo 2

Cinemática y dinámica de un ROTRADI

En este capítulo se presenta la deducción del modelo dinámico de un robot de un grado de libertad, empleando las ecuaciones de movimiento de Euler Lagrange. Por otra parte, también se presenta el modelo dinámico de un robot antropomorfo 3 grados de libertad[2]. Este modelo ya ha sido publicado por lo que esta revisión es parte del estudio del sistema.¹

El modelo dinámico de un robot manipulador permite explicar todos los fenómenos físicos que se encuentran en su estructura mecánica, tales como efectos inerciales, fuerzas centrípetas y de Coriolis, par gravitacional y fricción.

Un método estándar para obtener el modelo dinámico de un robot está basado en las *ecuaciones de movimiento de Euler Lagrange* [25] [26].

La energía total \mathcal{E} (*hamiltoniano*) del robot manipulador está dada por la suma de la energía cinética $K(q, \dot{q})$ más la energía potencial $U(q)$ [27].

$$\mathcal{E}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{U}(\mathbf{q}) \quad (2.1)$$

donde $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n$ representando los vectores de posición y velocidad articular, respectivamente. Se observa que la energía cinética $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$ tiene una dependencia de la posición y velocidad articular, mientras que la energía potencial $\mathcal{U}(\mathbf{q})$ está relacionada con el campo conservativo de la gravedad y por lo tanto depende solo de la posición [27].

El **lagrangiano** $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ de un robot manipulador de n grados de libertad se define como la diferencia entre la energía cinética y la energía potencial [27]:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}) \quad (2.2)$$

¹A. Mancilla, Tesis de Maestría, MCEA, FCE-BUAP, 2010; E. Gonzalez Tlaxco, Tesis de Maestría, MCEA, FCE-BUAP, 2010; J. Necochea Chamorro, Tesis de Maestría, MCEA, FCE-BUAP, 2012.

Las ecuaciones de movimiento de Euler Lagrange de un robot manipulador de n grados de libertad están dadas por:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \boldsymbol{\tau} - \mathbf{v}(\dot{\mathbf{q}}, \boldsymbol{\tau}) \quad (2.3)$$

donde $\mathbf{q} = [q_1, q_2, \dots, q_n]^T \in \mathbb{R}^n$ representa el vector de posiciones articulares o coordenadas generalizadas, $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T \in \mathbb{R}^n$ es el vector de velocidades articulares, $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_n]^T \in \mathbb{R}^n$ es el vector de pares aplicados, donde el i -ésimo par τ_i se encuentra asociado con la i -ésima coordenada generalizada q_i , y $\mathbf{v}(\dot{\mathbf{q}}, \mathbf{f}_e) \in \mathbb{R}^n$ es el vector de fuerzas o pares de fricción que depende de la velocidad articular $\dot{\mathbf{q}}$ y de la fricción estática f_e que se encuentra presente en las articulaciones del mismo [2].

La energía cinética tiene una estructura matemática bien definida en función de la velocidad articular:

$$\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \quad (2.4)$$

donde $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ es la matriz de inercia del manipulador, y es una matriz definida positiva (por lo tanto simétrica). Por otro lado la energía potencial $\mathcal{U}(\mathbf{q})$ no presenta forma específica. Sin embargo tiene una dependencia exclusivamente del vector de posición \mathbf{q} [28].

Las ecuaciones de movimiento de Euler Lagrange pueden escribir en su forma compacta de la siguiente manera:

$$\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = \frac{\partial}{\partial \dot{\mathbf{q}}} \left[\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right] - \frac{\partial \mathcal{U}(\mathbf{q})}{\partial \dot{\mathbf{q}}} = M(\mathbf{q}) \dot{\mathbf{q}} \quad (2.5)$$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right] = M(\mathbf{q}) \ddot{\mathbf{q}} + \dot{M}(\mathbf{q}) \dot{\mathbf{q}} \quad (2.6)$$

$$\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \left[\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right] - \frac{\partial \mathcal{U}(\mathbf{q})}{\partial \mathbf{q}} \quad (2.7)$$

Las ecuaciones de movimiento de Euler Lagrange para un robot de n grados de libertad adquieren la siguiente forma:

$$\boldsymbol{\tau} = M(\mathbf{q}) \ddot{\mathbf{q}} + \dot{M}(\mathbf{q}) \dot{\mathbf{q}} - \frac{\partial}{\partial \mathbf{q}} \left[\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right] + \frac{\partial \mathcal{U}(\mathbf{q})}{\partial \mathbf{q}} + f_f(\dot{\mathbf{q}}, \mathbf{f}_e) \quad (2.8)$$

El modelo dinámico de un robot manipulador de n grados de libertad está dado por la ecuación 2.8 que en su forma compacta y con la notación más ampliamente utilizada en el área de la robótica se encuentra descrito de la siguiente forma:

$$\boldsymbol{\tau} = M(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + f_f(\dot{\mathbf{q}}, \mathbf{f}_e) \quad (2.9)$$

donde

- $\mathbf{q} \in \mathbb{R}^n$ es el vector de coordenadas generalizadas o posiciones articulares,
- $\dot{\mathbf{q}} \in \mathbb{R}^n$ es el vector de velocidades articulares,
- $\ddot{\mathbf{q}} \in \mathbb{R}^{n \times n}$ es el vector de aceleraciones articulares,
- $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ es la matriz de inercia, la cual es simétrica y definida positiva.
- $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ es matriz de fuerzas centrípétas y de Coriolis:

$$C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \dot{M}(\mathbf{q})\dot{\mathbf{q}} - \frac{\partial}{\partial(\mathbf{q})} \left[\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right] \quad (2.10)$$

- $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ es el vector de fuerzas o pares gravitacionales obtenido como el gradiente de la energía potencial:

$$\mathbf{g}(\mathbf{q}) = \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}} \quad (2.11)$$

- $f_f(\dot{\mathbf{q}}, f_e)$ es el vector de pares de fricción que incluye la fricción viscosa, de Coulomb y estática (f_e) de cada articulación del robot.

2.1. Cinemática y dinámica de un ROTRADI de un grado de libertad

En esta sección se describe la cinemática y dinámica de un robot de 1 grado de libertad, tal como se observa en la figura 2.1.

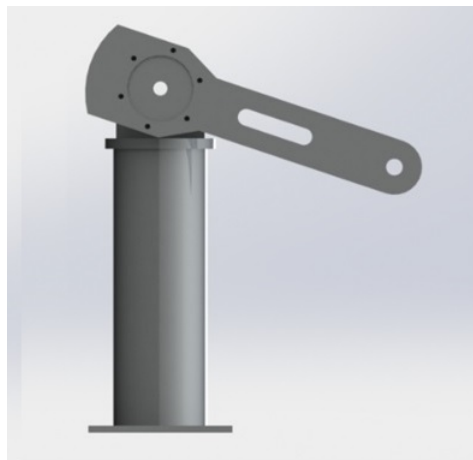


Figura 2.1: Esquema referencial del péndulo.

El método de Denavit-Hartenberg es una herramienta ampliamente conocida en el área de ingeniería, ya que ofrece un procedimiento sencillo para obtener el modelo cinemático directo cuya estructura queda en términos de las transformaciones homogéneas. La cinemática directa del robot proporciona las coordenadas cartesianas del extremo final del robot relativo a un sistema de referencia cartesiano fijo $\sum_0(x_0, y_0, z_0)$; el procedimiento para la asignación de coordenadas se realiza de la siguiente manera:

- El eje z_i se asigna rígidamente a la articulación $i+1$. Es decir, z_0 es el eje de la articulación 1, z_1 es el eje de la articulación 2, y así sucesivamente.
- Localizar el origen o_i del sistema de referencia $\sum_i(x_i, y_i, z_i)$ en la intersección del eje z_i con la normal común a los ejes z_{i-1} y z_i .
- Seleccionar el eje x_{i-1} sobre la normal que une los ejes z_{i-1} y z_i en dirección de la articulación $i-1$ hacia la articulación i .
- Definir el ángulo de torsión α_i , este es el ángulo entre los ejes z_i y z_{i-1} y se mide con valor positivo en el sentido de las manecillas del reloj sobre el eje x_i .
- Seleccionar el eje y_i por la regla de la mano derecha.

La convención Denavit-Hartenberg proporciona una representación no única para los siguientes casos:

- Para el sistema de referencia $\sum_0(x_0, y_0, z_0)$ sólo la dirección del eje z_0 es especificada, entonces su origen o_0 y el eje x_0 pueden ser seleccionados de manera arbitraria.
- Cuando dos ejes consecutivos z_i y z_{i-1} son paralelos entre sí, la normal común entre ellos no es única.
- Cuando dos ejes consecutivos z_i y z_{i-1} se interceptan, la dirección del eje x_i es arbitraria.
- Cuando la articulación i -ésima es lineal o prismática, entonces la dirección de z_i es arbitraria.

De acuerdo con la convención previamente descrita, a continuación se resumen los parámetros del i -ésimo eslabón:

- l_i es la longitud del i -ésimo eslabón, es la distancia del eje z_{i-1} hacia el eje z_i , medida sobre el eje x_{i-1} .
- α_i es el ángulo de torsión, el cual representa el ángulo entre los ejes z_{i-1} a z_i , medido en el sentido de las manecillas del reloj sobre el eje x_i .

- d_i se emplea en articulaciones lineales o prismáticas y representa el desplazamiento lineal. Cuando la articulación es rotacional, entonces representa el *offset* o espesor del servomotor (la distancia de x_{i-1} a x , medio sobre el eje z_{i-1}), se denota por β_i .
- θ_i es el desplazamiento rotacional de x_{i-1} a x_i medidor alrededor del eje z_{i-1} . El signo positivo de θ_i , es el sentido contrario a las manecillas del reloj.

El procedimiento Denavit-Hartenberg permite obtener el modelo cinemático directa de robots manipuladores con eslabones en serie a través de la tabla 2.1.

$Eslabón_i$	l_i	α_i	d_i	θ_i
-------------	-------	------------	-------	------------

Tabla 2.1: Parámetros de Denavit-Hartenberg.

En la figura 2.2 se muestra los ejes de referencia para obtener la cinemática directa por medio de la metodología de Denavit-Hartenberg.

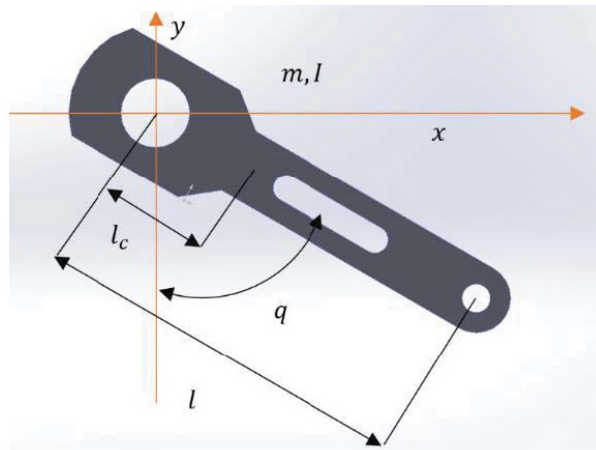


Figura 2.2: Representación gráfica para obtener la cinemática directa del péndulo.

La representación de la cinemática del péndulo por medio de la metodología de Denavit-Hartenberg se observa en la tabla 2.2.

l	d	α	θ
l	β	0	q

Tabla 2.2: Parámetros de DH del péndulo

La matriz de Denavit-Hartenberg aplicada al péndulo queda representada de la siguiente manera:

$$H_0^1 = \begin{bmatrix} \cos(q) & -\text{sen}(q) & 0 & l \cos(q) \\ \text{sen}(q) & \cos(q) & 0 & l \text{sen}(q) \\ 0 & 0 & 1 & \beta \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

Las coordenadas del extremo final del péndulo se encuentran determinadas por

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} l \cos(q) \\ l \text{sen}(q) \\ \beta \end{bmatrix} \quad (2.13)$$

Las coordenadas cartesianas del extremo final del péndulo dependen de la ubicación de la posición de casa, la cual es el punto de reposo u origen del péndulo que corresponde a $q_1 = 0$, la posición de casa está colocada sobre el eje $-y$. Por lo tanto se aplica una matriz de rotación para cambiar el punto de referencia.

$$H_0^1 = \begin{bmatrix} R_{z_0} \left(-\frac{\pi}{2}\right) \begin{bmatrix} \cos(q) & -\text{sen}(q) & 0 \\ \text{sen}(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} R_{z_0} \left(-\frac{\pi}{2}\right) \begin{bmatrix} l \cos(q) \\ l \text{sen}(q) \\ \beta \end{bmatrix} \\ 0^T \quad 1 \end{bmatrix} \quad (2.14)$$

$$H_0^1 = \begin{bmatrix} \text{sen}(q) & \cos(q) & 0 & l \text{sen}(q) \\ -\cos(q) & \text{sen}(q) & 0 & -l \cos(q) \\ 0 & 0 & 1 & \beta \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

Las coordenadas del extremo final del péndulo, considerando que la posición de casa se encuentra en el eje $-y$, quedan expresadas de la siguiente manera:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l \text{sen}(q) \\ -l \cos(q) \\ 0 \end{bmatrix} \quad (2.16)$$

donde l es la distancia al centro de masa y q es el ángulo proyectado.

2.1.1. Modelo dinámico del péndulo

Utilizando el método de Euler Lagrange, se obtiene el modelo dinámico del péndulo, considerando el centro de masa del eslabón, se encuentra representado por:

$$v = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_c \cos(q) \dot{q} \\ l_c \text{sen}(q) \dot{q} \\ 0 \end{bmatrix} \quad (2.17)$$

La rapidez de traslación se encuentra expresada como $v^T v = \|v\|^2 = l_c^2 \dot{q}^2$.

El modelo de energía del robot está compuesto de la energía cinética $\mathcal{K}(q, \dot{q})$ y de la energía potencial $\mathcal{U}(q)$, donde la energía cinética se encuentra representada por:

$$\mathcal{K}(q, \dot{q}) = \frac{1}{2} m v^T v + \frac{1}{2} I \dot{q}^2 = \frac{1}{2} [m l_c^2 + I] \dot{q}^2 \quad (2.18)$$

La energía potencial tiene la siguiente forma:

$$\mathcal{U}(q) = m g l_c [1 - \cos(q)] \quad (2.19)$$

Por lo tanto el langrangiano para el caso del péndulo está dado por:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q) = \frac{1}{2} [m l_c^2 + I] \dot{q}^2 - m g l_c [1 - \cos(q)] \quad (2.20)$$

Las ecuaciones de movimiento de Euler Lagrange para el caso particular del péndulo adquieren la forma de una ecuación escalar con la siguiente estructura [2]

$$\tau = \frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] - \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} \right] + f_f(f_e, \dot{q}) \quad (2.21)$$

donde τ es el par aplicado y f_e es el fenómeno de fricción modelado por la ecuación siguiente:

$$f_f(f_e, \dot{q}) = b \dot{q} + f_c \text{signo}(\dot{q}) + f_e [1 - |\text{signo}(\dot{q})|] \quad (2.22)$$

donde f_c es la fricción de Coulomb, f_e es la fricción estática, b es el coeficiente de fricción viscosa y la función $\text{signo}(\dot{q})$ está definida como:

$$\text{signo}(\dot{q}) \begin{cases} 1 & \text{si } \dot{q} > 0 \\ -1 & \text{si } \dot{q} < 0 \end{cases} \quad (2.23)$$

Por lo tanto el modelo dinámico de un robot péndulo incluyendo el fenómeno de fricción está dado por:

$$\tau = [m l_c^2 + I] \ddot{q} - m g l_c \text{sen}(q) + b \dot{q} + f_c \text{signo}(\dot{q}) + f_e [1 - |\text{signo}(\dot{q})|] \quad (2.24)$$

2.2. Cinemática y dinámica de un ROTRADI de 3 grados de libertad

A continuación se presenta el procedimiento para obtener la cinemática y dinámica de un ROTRADI de tres grados de libertad. En la figura 2.3 se muestra el ROTRADI empleado para el desarrollo de esta tesis.

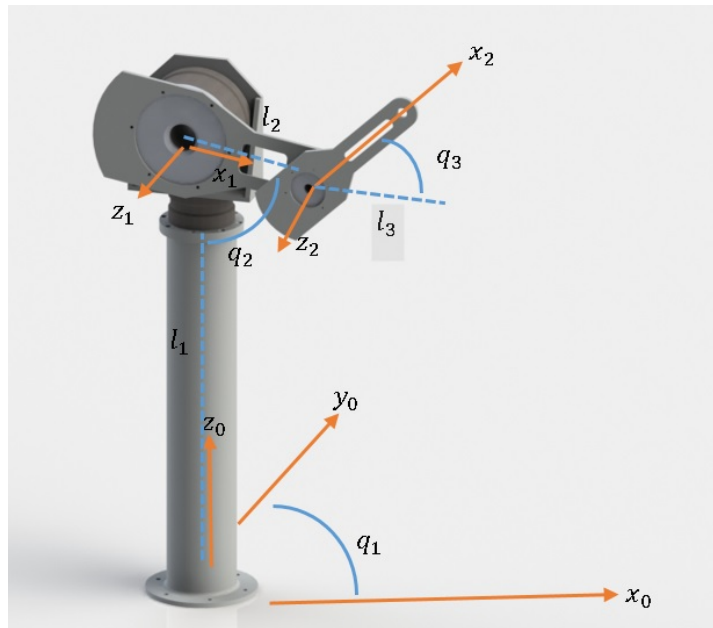


Figura 2.3: Diagrama del ROTRADI para determinar la cinemática directa.

Considerando la vista planar superior del robot, se encuentra que la posición real está desplazada debido al ancho de los motores, por lo que en la expresión de la posición se deben de incluir las proyecciones del ancho del motor, tal como se muestra en la figura 2.4.

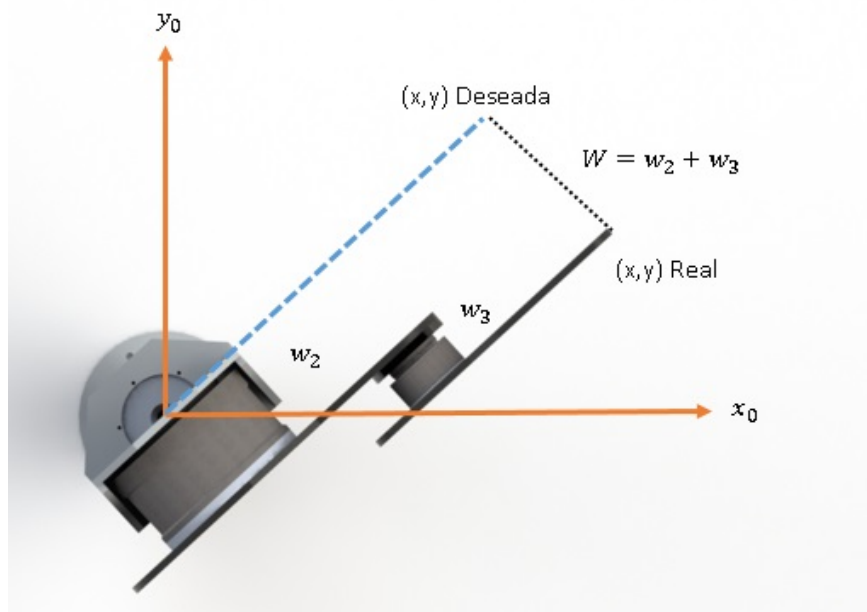


Figura 2.4: Vista superior del robot, la posición final se encuentra desplazada por la proyección de los motores.

La descripción de los parámetros se muestra en la tabla 2.3.

Eslabón	Significado	Notación	Valor
Base	Masa del eslabón 1	m_1	26.9 kg
	Longitud del eslabón 1	l_1	1.20 m
	Inercia del eslabón 1	I_{z1}	1.266 Nms ² /rad
	Inercia del eslabón 1	I_{y1}	0.089 Nms ² /rad
	Inercia del eslabón 1	I_{x1}	0.03 Nms ² /rad
	Coefficiente de fricción viscosa eslabón 1	b_1	2.288 Nms ² /rad
	Coefficiente de fricción de Coulomb 1	f_{c1}	7.17 Nm
	Posición articular del eslabón 1	q_1	rad
	Grosor del motor 1	β_1	0.26 m
Hombro	Masa del eslabón 2	m_2	30 kg
	Longitud del eslabón 2	l_2	0.63 m
	Centro de masa del eslabón 2	l_{c2}	0.11 m
	Inercia del eslabón 2	I_{z2}	0.084 Nms ² /rad
	Inercia del eslabón 2	I_{y2}	0.003 Nms ² /rad
	Inercia del eslabón 2	I_{x2}	0.05 Nms ² /rad
	Coefficiente de fricción viscosa eslabón 2	b_2	0.2 Nms/rad
	Coefficiente de fricción de Coulomb 2	f_{c2}	1.9 Nm
	Posición articular del eslabón 2	q_2	rad
Codo	Grosor del motor 2	β_2	0.22 m
	Masa del eslabón 3	m_3	3.88 kg
	Longitud del eslabón 3	l_3	0.5 m
	Centro de masa del eslabón 3	l_{c3}	0.055 m
	Inercia del eslabón 3	I_{z3}	0.056 Nms ² /rad
	Inercia del eslabón 3	I_{y3}	0.0012 Nms ² /rad
	Inercia del eslabón 3	I_{x3}	0.009 Nms ² /rad
	Coefficiente de fricción viscosa eslabón 3	b_3	0.175 Nms/rad
	Coefficiente de fricción de Coulomb 1	f_{c3}	1.734 Nm
Posición articular del eslabón 3	q_3	rad	
Grosor del motor 3	β_3	0.14 m	

Tabla 2.3: Parámetros del robot de 3 GDL.

La tabla 2.4 contiene los parámetros de Denavit-Hartenberg para el robot antropomórfico de tres grados de libertad.

Eslabón	l	d	α	θ
1	l_1	β_1	$\frac{\pi}{2}$	q_1
2	l_2	β_2	0	q_2
3	l_3	β_3	0	q_3

Tabla 2.4: Parámetros de DH del péndulo

La cinemática directa correspondiente a la base del robot se encuentra representada de la siguiente manera:

$$\begin{bmatrix} x_{11} \\ y_{11} \\ z_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_1 + \beta_1 \end{bmatrix} \quad (2.25)$$

La cinemática directa correspondiente al hombro del robot con respecto a la base se encuentra representada de la siguiente manera:

$$\begin{bmatrix} x_{21} \\ y_{21} \\ z_{21} \end{bmatrix} = \begin{bmatrix} \beta_2 \text{sen}(q_1) + l_{c2} \cos(q_1) \text{sen}(q_2) \\ l_{c2} \text{sen}(q_1) \text{sen}(q_2) - \beta_2 \cos(q_1) \\ \beta_1 + l_1 - l_{c2} \cos(q_2) \end{bmatrix} \quad (2.26)$$

La cinemática directa correspondiente al codo del robot con respecto a la base se encuentra representada de la siguiente manera:

$$\begin{bmatrix} x_{31} \\ y_{31} \\ z_{31} \end{bmatrix} = \begin{bmatrix} [\beta_2 + \beta_3] \text{sen}(q_1) + \cos(q_1) [l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)] \\ -[\beta_2 + \beta_3] \cos(q_1) + \text{sen}(q_1) [l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)] \\ \beta_1 + l_1 - l_2 \cos(q_2) - l_{c3} \cos(q_2 + q_3) \end{bmatrix} \quad (2.27)$$

2.2.1. Modelo dinámico del ROTRADI de 3 GDL

A continuación se presenta el modelo dinámico del ROTRADI de 3 GDL, utilizando el método de Euler Lagrange. La cinemática diferencial permite obtener la velocidad lineal con respecto al centro de masa de cada eslabón.

$$v_1 = \frac{d}{dt} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.28)$$

$$v_2 = \frac{d}{dt} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} [\beta_2 \cos(q_1) - l_{c2} \text{sen}(q_1) \text{sen}(q_2)] \dot{q}_1 + l_{c2} \cos(q_1) \cos(q_2) \dot{q}_2 \\ [l_{c2} \cos(q_1) \text{sen}(q_1) + \beta_2 \text{sen}(q_1)] \dot{q}_1 + l_{c2} \text{sen}(q_1) \cos(q_2) \dot{q}_2 \\ l_{c2} \text{sen}(q_2) \dot{q}_1 \end{bmatrix} \quad (2.29)$$

$$v_3 = \frac{d}{dt} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} [(\beta_2 + \beta_3)\cos(q_1) - \text{sen}(q_1)(l_2\text{sen}(q_2) + l_{c3}\text{sen}(q_2 + q_3))] \dot{q}_1 \\ + [\cos(q_1)(l_2\cos(q_2) + l_{c3}\cos(q_2 + q_3))] \dot{q}_2 \\ + l_{c3}\cos(q_1)\cos(q_2 + q_3)\dot{q}_3 \\ [(\beta_2 + \beta_3)\text{sen}(q_1) + \cos(q_1)(l_2\text{sen}(q_2) + l_{c3}\text{sen}(q_2 + q_3))] \dot{q}_1 \\ + [\text{sen}(q_1)(l_2\cos(q_2) + l_{c3}\cos(q_2 + q_3))] \dot{q}_2 \\ + l_{c3}\cos(q_2 + q_3)\text{sen}(q_1)\dot{q}_3 \\ [l_2\text{sen}(q_2) + l_{c3}\text{sen}(q_2 + q_3)] \dot{q}_2 \\ + l_{c3}\cos(q_2 + q_3)\cos(q_1)\dot{q}_3 \end{bmatrix} \quad (2.30)$$

La rapidez de traslación se encuentra expresada como $v^T v = \|v\|^2$ y queda representada de la siguiente manera.

$$v_1^T v_1 = 0 \quad (2.31)$$

$$v_2^T v_2 = \beta_2^2 \dot{q}_1^2 + l_{c2}^2 \text{sen}^2(q_2) \dot{q}_1^2 + 2\beta_2 l_{c2} \cos(q_2) \dot{q}_1 \dot{q}_2 + l_{c2}^2 \dot{q}_2^2 \quad (2.32)$$

$$\begin{aligned} v_3^T v_3 = & [\dot{q}_1[(\beta_2 + \beta_3)\text{sen}(q_1) + \cos(q_1)(l_{c3}\text{sen}(q_2 + q_3) + l_2\text{sen}(q_2))] \\ & + \dot{q}_2\text{sen}(q_1)(l_{c3}\cos(q_2 + q_3) + l_2\cos(q_2)) + l_{c3}\dot{q}_3\cos(q_2 + q_3)\text{sen}(q_1)]^2 \\ & + [\dot{q}_2(l_{c3}\text{sen}(q_2 + q_3) + l_2\text{sen}(q_2)) + l_{c3}\dot{q}_3\cos(q_2 + q_3)\cos(q_1)]^2 \\ & + [\dot{q}_1[\cos(q_1)(\beta_2 + \beta_3) - \text{sen}(q_1)(l_{c3}\text{sen}(q_2 + q_3) + l_2\text{sen}(q_2))] \\ & + \dot{q}_2\cos(q_1)(l_{c3}\cos(q_2 + q_3) + l_2\cos(q_2)) + l_{c3}\dot{q}_3\cos(q_2 + q_3)\cos(q_1)]^2 \end{aligned} \quad (2.33)$$

La energía cinética para cada articulación está determinada de la siguiente manera:

$$K_1(q, \dot{q}) = \frac{1}{2} I_{z1} \dot{q}_1^2 \quad (2.34)$$

$$K_2(q, \dot{q}) = \frac{1}{2} m_2 v_2^T v_2 + \frac{1}{2} [I_{x2} \dot{q}_2^2 \cos^2(q_1) + I_{y2} \dot{q}_2^2 \text{sen}^2(q_1) + I_{z2} \dot{q}_1^2] \quad (2.35)$$

$$\begin{aligned} K_3(q, \dot{q}) = & \frac{1}{2} m_3 v_3^T v_3 + \frac{1}{2} I_{x3} \cos^2(q_1) [\dot{q}_2^2 + 2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2] \\ & + \frac{1}{2} I_{y3} \text{sen}^2(q_1) [\dot{q}_2^2 + 2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2] + \frac{1}{2} I_{z3} \dot{q}_1^2 \end{aligned} \quad (2.36)$$

$$\begin{aligned} K(q, \dot{q}) = & \frac{1}{2} I_{z1} \dot{q}_1^2 + \frac{1}{2} m_2 v_2^T v_2 + \frac{1}{2} [I_{x2} \dot{q}_2^2 \cos^2(q_1) + I_{y2} \dot{q}_2^2 \text{sen}^2(q_1) + I_{z2} \dot{q}_1^2] \\ & + \frac{1}{2} m_3 v_3^T v_3 + \frac{1}{2} I_{x3} \cos^2(q_1) [\dot{q}_2^2 + 2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2] \\ & + \frac{1}{2} I_{y3} \text{sen}^2(q_1) [\dot{q}_2^2 + 2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2] + \frac{1}{2} I_{z3} \dot{q}_1^2 \end{aligned} \quad (2.37)$$

Por otra parte, la energía potencial para cada articulación está determinada como:

$$\begin{aligned}
 U_1(q) &= m_1 g l_{c1} \\
 U_2(q) &= m_2 g [l_1 - l_{c2} \cos(q_2)] \\
 U_3(q) &= m_3 g [l_1 - l_2 \cos(q_2) - l_{c3} \cos(q_2 + q_3)]
 \end{aligned} \tag{2.38}$$

La energía potencial total del robot está dada como:

$$U(q) = m_1 g l_{c1} + m_2 g [l_1 - l_{c2} \cos(q_2)] + m_3 g [l_1 - l_2 \cos(q_2) - l_{c3} (q_2 + q_3)] \tag{2.39}$$

Por lo tanto el modelo de energía se encuentra definido de la ecuación de 2.2, obteniendo como resultado:

$$\begin{aligned}
 \mathcal{L}(q, \dot{q}) &= \frac{1}{2} I_{z1} \dot{q}_1^2 + \frac{1}{2} m_2 [\beta_2^2 \dot{q}_1^2 + l_{c2}^2 \text{sen}^2(q_2) \dot{q}_1^2 + 2\beta_2 l_{c2} \cos(q_2) \dot{q}_1 \dot{q}_2 + l_{c2}^2 \dot{q}_2^2] \\
 &+ \frac{1}{2} [I_{x2} \dot{q}_2^2 \cos^2(q_1) + I_{y2} \dot{q}_2^2 \text{sen}^2(q_1) + I_{z2} \dot{q}_1^2] + \frac{1}{2} I_{x3} \cos^2(q_1) [\dot{q}_2^2 + 2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2] \\
 &+ \frac{1}{2} I_{y3} \text{sen}^2(q_1) [\dot{q}_2^2 + 2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2] + \frac{1}{2} I_{z3} \dot{q}_1^2 \\
 &+ \frac{1}{2} m_3 [[\dot{q}_1 [(\beta_2 + \beta_3) \text{sen}(q_1) + \cos(q_1) (l_{c3} \text{sen}(q_2 + q_3) + l_2 \text{sen}(q_2))] \\
 &+ \dot{q}_2 \text{sen}(q_1) (l_{c3} \cos(q_2 + q_3) + l_2 \cos(q_2)) + l_{c3} \dot{q}_3 \cos(q_2 + q_3) \text{sen}(q_1)]^2 \\
 &+ [\dot{q}_2 (l_{c3} \text{sen}(q_2 + q_3) + l_2 \text{sen}(q_2)) + l_{c3} \dot{q}_3 \cos(q_2 + q_3) \cos(q_1)]^2 \\
 &+ [\dot{q}_1 [\cos(q_1) (\beta_2 + \beta_3) - \text{sen}(q_1) (l_{c3} \text{sen}(q_2 + q_3) + l_2 \text{sen}(q_2))] \\
 &+ \dot{q}_2 \cos(q_1) (l_{c3} \cos(q_2 + q_3) + l_2 \cos(q_2)) + l_{c3} \dot{q}_3 \cos(q_2 + q_3) \cos(q_1)]^2 \\
 &- [m_1 g l_{c1} + m_2 g (l_1 - l_{c2} \cos(q_2)) + m_3 g (l_1 - l_2 \cos(q_2) - l_{c3} (q_2 + q_3))]
 \end{aligned} \tag{2.40}$$

Aplicando las ecuaciones de movimiento de Euler Lagrange (ver ecuación 2.3) para cada una de las coordenadas angulares, obtenemos las expresiones de τ para cada articulación:

$$\begin{aligned}
& \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} I_{y_2} \text{sen}^2(q_2) + I_{y_3} \text{sen}^2(q_2 + q_3) + I_{z_1} & 0 & 0 \\ I_{z_2} \text{cos}^2(q_2) + I_{z_3} \text{cos}^2(q_2 + q_3) + m_2 l_{c_2}^2 \text{cos}^2(q_2) & 0 & 0 \\ +m_3 [2\text{cos}(q_2) + l_{c_3} \text{cos}(q_2 + q_3)]^2 & I_{x_2} + I_{x_3} + m_3 l_{c_3}^2 + m_2 l_{c_3}^2 + m_3 l_{c_3}^2 & I_{x_3} + m_3 l_{c_3}^2 + m_3 l_2 l_{c_3} \text{cos}(q_3) \\ 0 & +2m_3 l_2 l_{c_3} \text{cos}(q_3) & I_{x_3} + m_3 l_{c_3}^2 \\ 0 & I_{x_3} + m_3 l_{c_3}^2 + m_3 l_2 l_{c_3} \text{cos}(q_3) & I_{x_3} + m_3 l_{c_3}^2 \\ 0 & I_{x_3} + m_3 l_{c_3}^2 + m_3 l_2 l_{c_3} \text{cos}(q_3) & I_{x_3} + m_3 l_{c_3}^2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (2.41) \\
& + \begin{bmatrix} [I_{y_2} - I_{z_2} - m_2 l_{c_2}^2] \text{cos}(q_2) \text{sen}(q_2) \dot{q}_2 \\ + [I_{y_3} - I_{z_3}] \text{cos}(q_2 + q_3) \text{sen}(q_2 + q_3) \dot{q}_2 \\ - m_3 [l_2 \text{cos}(q_2) + l_{c_3} \text{cos}(q_2 + q_3)] l_2 \text{sen}(q_2) \dot{q}_2 \\ - m_3 [\beta_2 l_2 \text{cos}(q_2) + l_{c_3} \text{cos}(q_2 + q_3)] l_{c_3} \text{sen}(q_2 + q_3) \dot{q}_2 \\ + [I_{y_3} - I_{z_3}] \text{cos}(q_2 + q_3) \text{sen}(q_2 + q_3) \dot{q}_3 \\ - m_3 l_{c_3} \text{sen}(q_2 + q_3) l_2 \text{sen}(q_2) \dot{q}_3 \\ - m_3 l_{c_3} \text{sen}(q_2 + q_3) l_{c_3} \text{sen}(q_2 + q_3) (\beta_2 + \beta_3) \dot{q}_3 \\ [I_{z_2} - I_{y_2} + m_2 l_{c_1}^2] (\beta_2 + \beta_3) \text{cos}(q_2) \text{sen}(q_2) \dot{q}_1 \\ + [I_{z_3} - I_{y_3}] \text{cos}(q_2 + q_3) \text{sen}(q_2 + q_3) \dot{q}_1 \\ + m_3 [l_1 \text{cos}(q_2) + l_{c_2} \text{cos}(q_2 + q_3)] \beta_3 l_1 \text{sen}(q_2) \dot{q}_1 \\ + m_3 [l_1 \text{cos}(q_2) + l_{c_2} \text{cos}(q_2 + q_3)] (\beta_2 + \beta_3) l_{c_2} \text{sen}(q_2 + q_3) \dot{q}_1 \\ [I_{z_3} - I_{y_3}] \text{cos}(q_2 + q_3) \text{sen}(q_2 + q_3) \dot{q}_1 (\beta_2 + \beta_3) \\ + m_3 l_{c_2} \text{sen}(q_2 + q_3) [l_1 \text{cos}(q_2) + l_{c_2} \text{cos}(q_2 + q_3)] \dot{q}_1 \\ 0 \\ l_{c_2} m_2 \text{sen}(q_2) + m_3 l_2 \text{sen}(q_2) + m_3 l_{c_3} \text{sen}(q_2 + q_3) \\ m_3 l_{c_3} \text{sen}(q_2 + q_3) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \\
& + g \begin{bmatrix} 0 \\ b_1 \dot{q}_1 + f_{c_1} \text{signo}(\dot{q}_1) + f_{c_1} [1 - |\text{signo}(\dot{q}_1)|] \\ b_2 \dot{q}_2 + f_{c_2} \text{signo}(\dot{q}_2) + f_{c_2} [1 - |\text{signo}(\dot{q}_2)|] \\ b_3 \dot{q}_3 + f_{c_3} \text{signo}(\dot{q}_3) + f_{c_3} [1 - |\text{signo}(\dot{q}_3)|] \end{bmatrix} \\
& + \begin{bmatrix} [I_{y_3} - I_{z_3}] \text{cos}(q_2 + q_3) \text{sen}(q_2 + q_3) \dot{q}_1 \\ - m_3 l_{c_2} \text{sen}(q_2 + q_3) l_{c_1} \text{cos}(q_2) \dot{q}_1 (\beta_2 + \beta_3) \\ - m_3 l_{c_2}^2 \text{sen}(q_2 + q_3) \text{cos}(q_2 + q_3) \dot{q}_1 \\ 0 \\ - (\beta_2 + \beta_3) l_1 m_3 l_{c_2} \text{sen}(q_3) (\dot{q}_2 + \dot{q}_3) \\ 0 \end{bmatrix}
\end{aligned}$$

Capítulo 3

Diseño mecánico e instrumentación electrónica

En este capítulo se realiza el análisis de la instrumentación electrónica desarrollada para la banda transportadora y para las señales de control del ROTRADI. Para la banda transportadora se implementa un controlador utilizando una tarjeta Arduino Mega, una tarjeta de control, constituida por un puente H y sistema de comunicación WiFi. Para el ROTRADI se implementa la instrumentación electrónica para la lectura de las señales de los encoders y proporcionar las señales analógicas para el control del ROTRADI por medio de una tarjeta Arduino Due. En la figura 3.1 se muestra el diagrama a bloques del funcionamiento de la celda robótica.

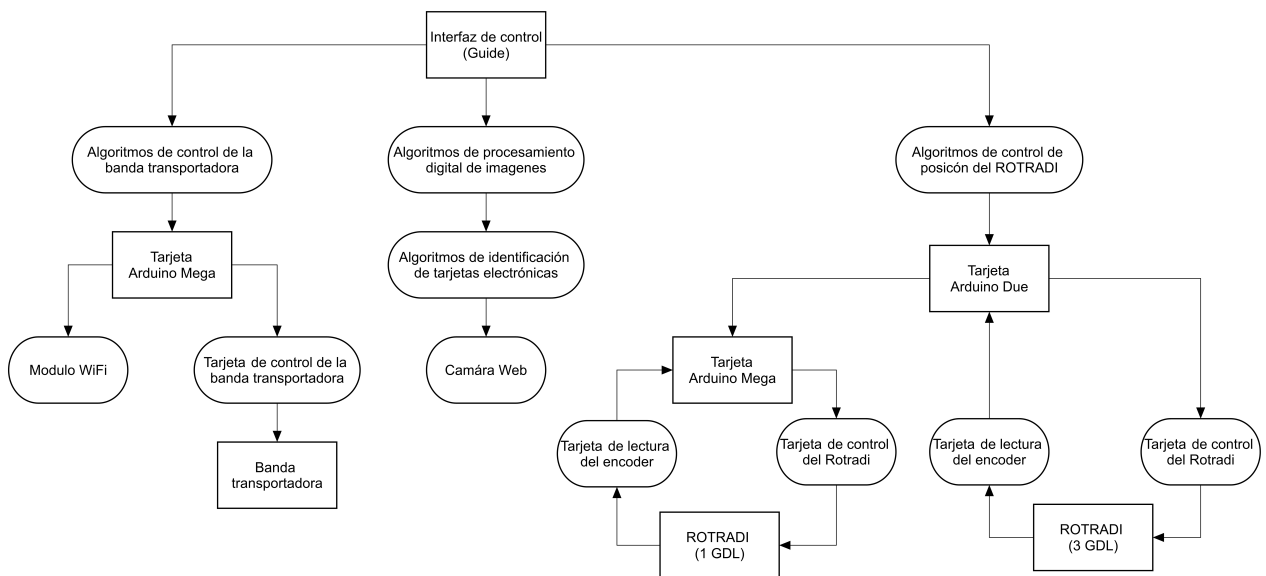


Figura 3.1: Diagrama a bloques del funcionamiento de la celda robótica.

El sistema de control de desplazamiento lo efectúa la tarjeta Arduino Mega, el sistema de control de posición del ROTRADI se implementa en la tarjeta Arduino Due. La interfaz tiene como finalidad realizar la sincronización entre la banda transportadora y el ROTRADI por medio de la información obtenida de la cámara web.

3.1. Banda transportadora

En la figura 3.2 se muestra la estructura mecánica utilizada para fijar el motor con la banda transportadora, para garantizar un mejor funcionamiento se propone el diseño de un sistema mecánico.



Figura 3.2: Soporte mecánico del motor.

La primera acción a realizar consiste en proponer un motor que cuente con las características necesarias para realizar el desplazamiento de la banda transportadora, considerando el peso del producto a transportar, se desea transportar un conjunto de tarjetas electrónicas. En la tabla 3.1 se mencionan las características de la banda transportadora y de la tarjeta electrónica de mayor tamaño y peso.

Parámetros	Descripción	Parámetros	Descripción
Ancho de la cadena	160 mm	Dimensión PCB	10 cm x 10 cm
Velocidad nominal	$10 \frac{\text{m}}{\text{min}}$	Peso de PCB	250 gr
Peso de la cadena	$8 \frac{\text{kg}}{\text{m}^2}$	Dimensiones del soporte	15 cm x 15 cm
Coefficiente de fricción	0.18	Peso del soporte	200 gr
Peso transportado	25 kg	Peso total	450 gr

Tabla 3.1: Características físicas de la banda transportadora y tarjeta electrónica.

Se desea transportar 1 tarjeta electrónica cada 30 s, considerando que la tarjeta mide 10 cm de ancho, por lo que la velocidad de la banda transportadora debe de ser:

$$Velocidad = \frac{0.15 \text{ m}}{0.5 \text{ min}} = 0.3 \frac{\text{m}}{\text{min}} = 0.984 \frac{\text{ft}}{\text{min}} \quad (3.1)$$

Por lo tanto la velocidad requerida de la transportadora es de $0.984 \frac{\text{ft}}{\text{min}}$. La capacidad de carga distribuida es el peso que puede transportar la banda transportadora sobre una área definida y se encuentra definida de la siguiente manera:

$$Carga \text{ distribuida} = \frac{Peso \text{ a transportar}}{Area \text{ de carga}} \quad (3.2)$$

El máximo peso de la tarjeta con su base es de 0.45 kg y las dimensiones del soporte es de 0.15 m^2 , por lo tanto:

$$Carga \text{ distribuida} = \frac{0.45 \text{ kg}}{0.15 \text{ m}^2} = 3 \frac{\text{kg}}{\text{m}^2} \quad (3.3)$$

Para el calculo de la fuerza motriz del transportador (potencia necesaria para poder mover la carga y la banda transportadora), tomando en cuenta el coeficiente de fricción dado por el fabricante y el tipo de transportador a utiliza, se obtiene a partir de la siguiente ecuación:

$$Fuerza \text{ motriz} = \frac{(W + w)fs}{K} \quad (3.4)$$

donde

W : Peso de carga

w : Peso de banda

f : coeficiente de fricción

s : Velocidad

K : Constante de conversión fuerza Hp a $\frac{\text{ft} \cdot \text{lb}}{\text{min}}$ (3300)

Por lo tanto obtenemos que

$$W = \left(3 \frac{\text{kg}}{\text{m}^2} \right) \left(\frac{\frac{2.20462 \text{ lb}}{1 \text{ kg}}}{\frac{10.76 \text{ ft}^2}{1 \text{ m}^2}} \right) = 0.614 \frac{\text{lb}}{\text{ft}^2} \quad (3.5)$$

$$w = \left(8 \frac{\text{kg}}{\text{m}^2} \right) \left(\frac{\frac{2.20462 \text{ lb}}{1 \text{ kg}}}{\frac{10.76 \text{ ft}^2}{1 \text{ m}^2}} \right) = 1.639 \frac{\text{lb}}{\text{ft}^2} \quad (3.6)$$

$$Fuerza \text{ motriz} = \frac{(0.614 \frac{\text{lb}}{\text{ft}^2} + 1.639 \frac{\text{lb}}{\text{ft}^2}) (0.18) (0.984 \frac{\text{ft}}{\text{min}})}{3300} = 0.00012 \frac{\text{ft} \cdot \text{lb}}{\text{min}} = 0.4 \text{ HP} \quad (3.7)$$

Por lo tanto se considera utilizar un motor que proporcione un fuerza motriz de 0.4 HP para realizar el desplazamiento de la banda transportadora considerando el peso de la carga

a transportar, también es necesario considerar la fuente de alimentación del motor. El motor seleccionado pertenece al fabricante *Crouzet* con número de parte 82 – 867 – 011, en la figura 3.3 se muestra el motor junto con sus características eléctricas y mecánicas.

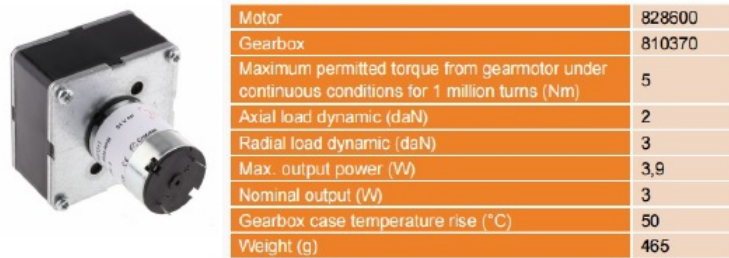


Figura 3.3: Motor *Crouzet*.

Es necesario conocer si el motor seleccionado cumple con las características necesarias, se realiza la conversión de Nm a HP, considerando que el motor proporciona un torque máximo de 5 Nm a 34 RPM, dando como resultado la siguiente expresión:

$$P = M\omega \quad (3.8)$$

donde

M = Momento de inercia

ω = Velocidad angular

$$P = 5 \text{ Nm } 232 \text{ min} = 1162 \text{ W} \quad (3.9)$$

$$P = 1162 \text{ W} = 1.56 \text{ HP} \quad (3.10)$$

Por lo tanto realizando la comparación entre el torque obtenido (ecuación 3.10) y el torque necesario (ecuación 3.7) se considera que el motor cumple con las características adecuada para realizar el desplazamiento de la banda transportadora considerando el peso de la carga a transportar.

3.1.1. Diseño del soporte

Una vez considerado el motor a utilizar, es necesario implementar un soporte que permita colocar el motor en la banda transportadora. Para realizar el diseño de las piezas mecánicas se utiliza la plataforma de *Solidworks*. Se diseñan dos piezas que se complementan entre sí, una de ellas tiene como función servir de soporte al motor, mientras la otra pieza funciona como

base de sujeción del motor con la banda transportadora, en el apéndice A se encuentra la hoja técnica de cada una de las piezas diseñadas, en la figura 3.4 se muestra el diseño propuesto y la pieza real.



Figura 3.4: Ensamble de motor.

3.2. Diseño electrónico de la banda transportadora

La siguiente acción a realizar consiste en la implementación de la parte electrónica en la banda transportadora. Para esto es necesario caracterizar el motor con la finalidad de conocer la frecuencia de trabajo que nos permita manipular el motor en su zona lineal. Posteriormente se desarrolla el controlador, está conformado por un driver y un firmware.

3.2.1. Caracterización del motor

El proceso de caracterización consiste en encontrar una frecuencia de trabajo, que permita manipular el motor en su zona lineal, con la finalidad de utilizar el motor como un sistema de transmisión directa. Para realizar esta prueba se fuerza el motor a trabajar en zona máxima, posteriormente se selecciona un rango de voltaje que nos brinde el mayor torque con la menor potencia.

En la figura 3.5 se muestra la gráfica obtenida a partir de la realización de un barrido de voltaje en el motor, en esta gráfica se muestra la eficiencia del motor, es decir la comparación entre el torque obtenido y el voltaje aplicado, en este caso se observa que al aplicar un voltaje de 21 Vcd, se produce una corriente de 1.5 A, considerando una eficiencia de 0.21, en este nivel de voltaje el torque proporcionado por el motor es de 6.615 Nm, el cual resulta suficiente para poder desplazar la banda transportadora.

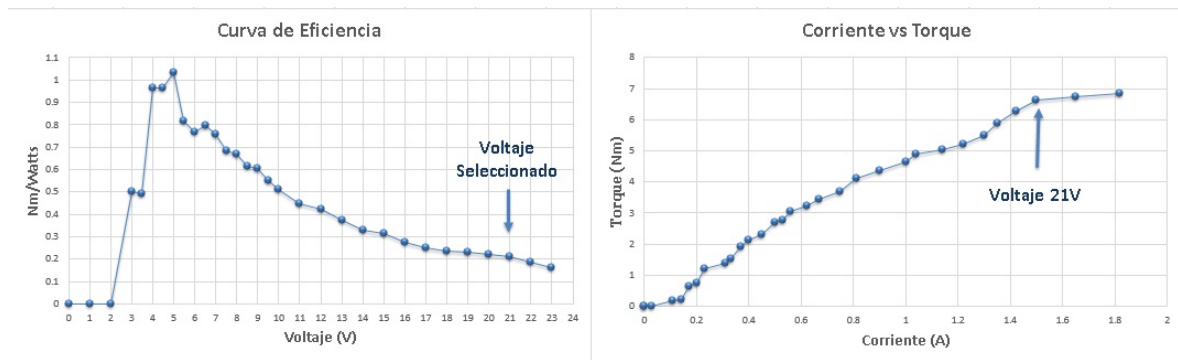


Figura 3.5: Curva de eficiencia.

El siguiente paso es encontrar una frecuencia tal que cuando se realice un barrido de una señal PWM al motor, el torque proporcionado sea lineal y el torque máximo sea de 6.615 Nm. En la figura 3.6 se muestra el resultado obtenido al aplicar una frecuencia de 260 Hz, el comportamiento del motor se considera aproximadamente lineal, esta frecuencia deberá de ser considerada para el diseño del controlador.

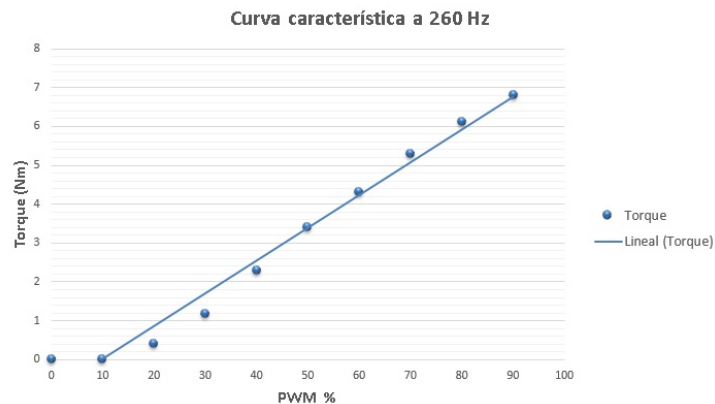


Figura 3.6: Caracterización del motor a 260 Hz.

3.2.2. Diseño de driver electrónico

En esta sección se realiza el diseño del driver electrónico para el control del motor. Se considera el diseño de una estructura conocida como puente H. El controlador tendrá implementado el cambio de giro y modulación por ancho de pulso a una frecuencia de 260 Hz.

Para la implementación del puente H, es necesario considerar que el voltaje de alimentación del motor es de 24 Vcd y una corriente de consumo de 2 A. Por medio de un arreglo de transistores Darlington se realiza la estructura del puente H. Los transistores seleccionados se muestran en la tabla 3.2.

Transistor	Tipo	I_c	$\beta_{Forzada}$
TIP41C	NPN	6 A	8
TIP42C	PNP	-6 A	8
2N2222A	NPN	800 mA	15
BC327	PNP	-800 mA	15

Tabla 3.2: Características eléctricas de transistores.

Los transistores utilizados se manejan en el modo de saturación, el cual considera al transistor como un interruptor abierto o cerrado, en este modo de funcionamiento el transistor disipa la mínima cantidad de energía, obteniendo la máxima eficiencia del dispositivo. Para ello se considera la *beta* del transistor (el valor máximo de amplificación), la corriente de base del transistor y la corriente del colector. A continuación se muestran los cálculos realizados para el diseño del puente H.

Se considera un arreglo de transistor Darlington conformado por dos TIP41C en modo de saturación con una beta forzada de 5 cada uno, la beta total es de 25 (la beta total es el producto de cada una de las betas), la corriente del motor es de 2 A. La corriente de base necesaria para activar el arreglo Darlington se muestra en la ecuación 3.11, en la ecuación 3.12 se muestra el valor de la resistencia obtenida.

$$I_{base} = \frac{I_{colector}}{B_{forzada}} = \frac{2 \text{ A}}{25} = 80 \text{ mA} \quad (3.11)$$

$$R = \frac{V_{Entrada}}{I_{base}} = \frac{24 \text{ Vcd}}{80 \text{ mA}} = 300 \Omega \quad (3.12)$$

En la ecuación 3.13 se muestra el valor de disipación de la resistencia.

$$W_{Resistencia} = VI = 1.92 \text{ W} \quad (3.13)$$

Para activar ambas partes del puente H se utiliza un transistor 2n2222A, el cual posee una beta forzada de 10, la corriente necesaria en la base para activar este transistor en modo de saturación se muestran en la ecuación 3.14. En la ecuación 3.15 se calcula el valor de la resistencia para suministrar 8 mA a la base del transistor y en la ecuación 3.16 se muestra la capacidad de disipación de la resistencia calculada.

$$I_{base} = \frac{I_{colector}}{B_{forzada}} = \frac{80 \text{ mA}}{10} = 8 \text{ mA} \quad (3.14)$$

$$R = \frac{V_{Alimentacin}}{I_{activacin}} = \frac{5 \text{ Vcd} - 0.7 \text{ Vcd}}{8 \text{ mA}} = 537.5 \Omega \quad (3.15)$$

$$W_{Resistencia} = VI = 0.0344 \text{ W} \quad (3.16)$$

Con este cálculo se garantiza que no se presente un sobrecalentamiento en la resistencia de activación de los transistores.

Dado que el motor de la banda transportadora requiere de una alimentación de 24 Vcd, es necesario utilizar una fuente de voltaje con la capacidad necesaria. Para solucionar este problema se implementa una fuente conmutada amplificadora, este tipo de fuente permite elevar el voltaje de entrada a un voltaje mayor. Su principio de funcionamiento se basa en fuente conmutada con topología básica Buck-Boost, conformada por un control detector de cruce por cero, un dispositivo conmutador (mosfet), una diodo, un capacitor y una bobina; estos dos últimos elementos permiten almacenar energía dependiendo de la velocidad de conmutación tal como se muestra en la figura 3.7.

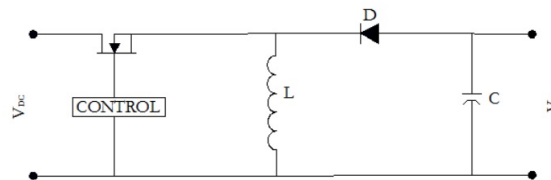


Figura 3.7: Topología básica del convertidor Buck-Boost.

Se utiliza el módulo XL6009, es una fuente conmutada basada en el diagrama que se muestra en la figura 3.7. Este dispositivo proporciona una corriente máxima de 3 A y el voltaje de salida depende del voltaje de entrada y de la velocidad de conmutación, que es seleccionado por medio de una resistencia variable. El voltaje que se suministra a este módulo es de 12 Vcd y se regula la fuente de alimentación para obtener un voltaje de salida de 24 Vcd, el cual corresponde al voltaje de alimentación del motor de la banda transportadora, en la figura 3.8 se muestra el módulo utilizado y su diagrama eléctrico.

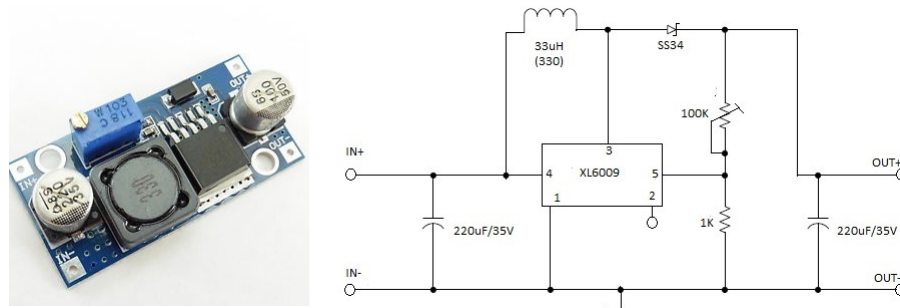


Figura 3.8: Módulo regulador de voltaje XL6009.

En la figura 3.9 se muestra el diseño del controlador para la banda transportadora, el cual presenta la estructura de un puente H, por medio de transistores en configuración Darlington. Por medio de las compuertas digitales es posible seleccionar el sentido del giro del motor utilizando un voltaje de 5 Vcd y una corriente de 1 mA. Cuando se activan las dos entradas de control al mismo tiempo se induce un corto circuito al motor, para evitar esta acción, se considera esta acción dentro de la programación. Cuando el circuito se energiza es necesario que las entradas de activación se encuentren en estado bajo para que no se accione el motor, por esta razón se colocan resistencias *pull-down* en cada una de las entradas de las compuertas digitales. El circuito RC que se coloca en paralelo al motor se utiliza como circuito de protección a la corriente inversa que se produce cuando el motor cambia de sentido de giro.

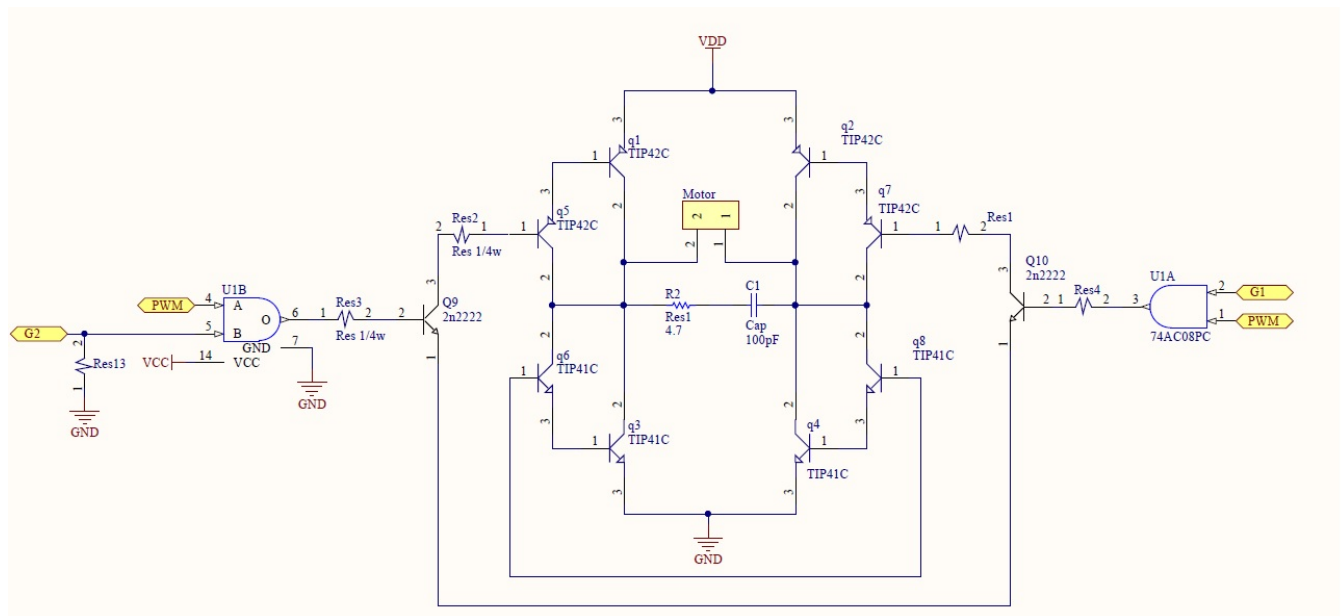


Figura 3.9: Esquemático de puente H.

En la figura 3.10 se muestra el esquemático de la fuente de alimentación para el motor de la banda transportadora, incluye dos fuentes conmutadas utilizando los módulos XL6009 representados por los bloques F1 y F2 (el diagrama eléctrico de este componente se muestra en la figura 3.8), la fuente F1 es regulada 5 Vcd para alimentar los circuitos integrados y la fuente F2 se regula a 24 Vcd para alimentar al motor. El diodo D1 funciona como protección ante corriente inversa. Los capacitores C2 y C3 se colocan en paralelo para eliminar los picos de voltaje y proporcionar la corriente de arranque al motor.

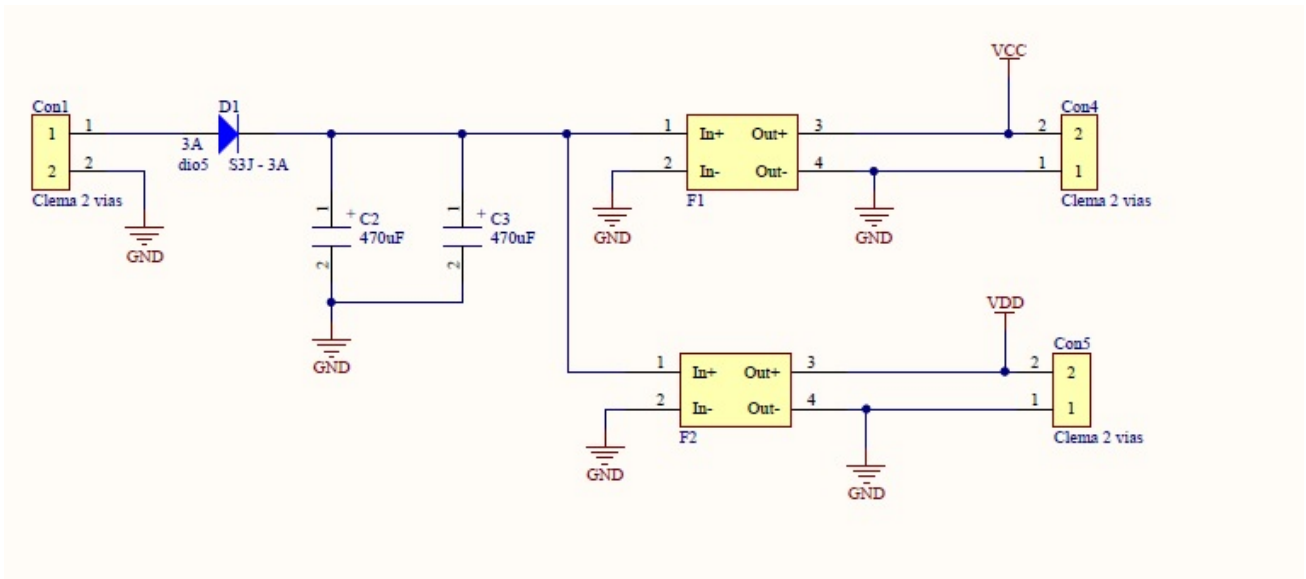


Figura 3.10: Esquemático de fuente de voltaje.

En la figura 3.11 se muestra el diseño de la tarjeta electrónica para el control de la banda transportadora, en la sección de la izquierda se encuentra la fuente de voltaje y en la sección de la derecha el diseño del puente H. Para realizar el diseño de manera impresa es necesario considerar el espacio de cada uno de los componentes electrónicos y la serigrafía que facilite la conexión y uso de cada una de las tarjetas.

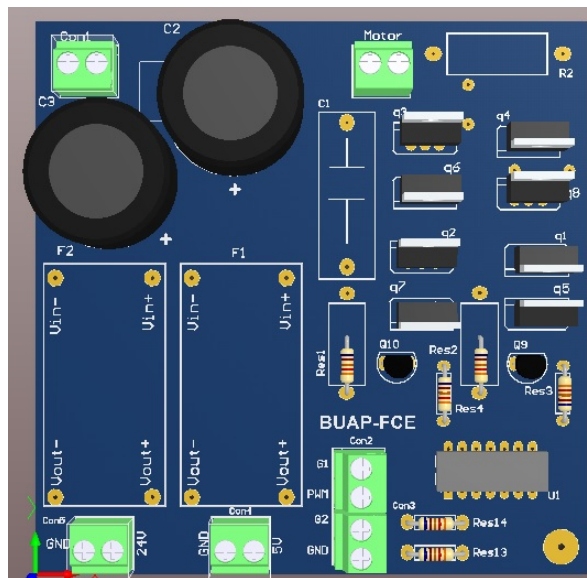


Figura 3.11: Diseño de la tarjeta de control.

3.2.3. Firmware del controlador

En esta sección se muestra el diseño del controlador, para ellos se emplea la plataforma de Arduino y la tarjeta Arduino Mega, el propósito general consiste en implementar una señal PWM, con frecuencia fija y periodo de trabajo variable, el cual puede ser configurado por medio de un comando serial. La señal generada se inserta a la tarjeta de control para cambiar la velocidad del motor.

Para el desarrollo del programa en la plataforma de Arduino es necesario considerar la frecuencia del motor donde se comporte como un driver de transmisión directa, la cual es de 260 Hz, a pesar de que la plataforma Arduino cuenta con una librería llamada PWM para generar la señal de ancho de trabajo variable, la frecuencia dada por esta librería es de 1.2 KHz, es necesario realizar el diseño de un firmware que sea capaz de proporcionar una señal PWM con una frecuencia fija de 260 Hz. El diagrama de flujo del firmware diseñado se muestra en la figura 3.12.

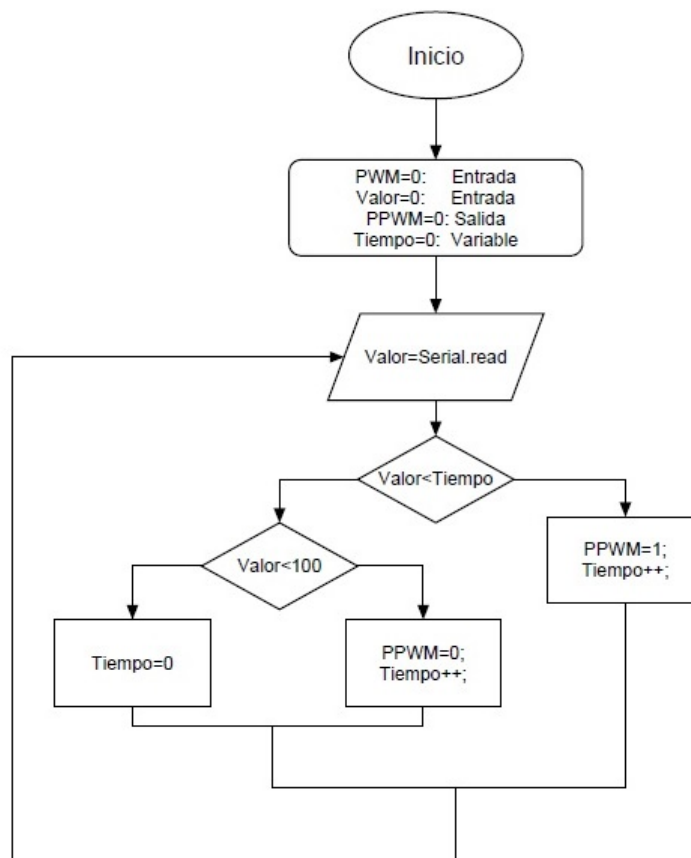


Figura 3.12: Diagrama de flujo del controlador.

Para la implementación del firmware en la tarjeta *Arduino Mega* es necesario el uso de interrupciones para llevar poder tener la variable del tiempo, dado que el microprocesador ATMEGA2560 contiene 4 interrupciones por hardware, una de ellas se utiliza para contar el número de ciclos de reloj y realizar una conversión a tiempo. Posteriormente es necesario realizar la lectura del PWM deseada, está se realiza por medio del protocolo serial. El programa elaborado en la plataforma Arduino se muestra en el apéndice C.1.

De esta manera se implementa un firmware capaz de generar una señal PWM con un periodo de trabajo variable. Al algoritmo anterior se implementa un función adicional que permite realizar el cambio de la frecuencia de la señal PWM, dando como resultado un firmware capaz de generar una señal PWM con frecuencia variable y periodo de trabajo variable. En la figura 3.13 se muestra el resultado de la señal PWM generada por el microprocesador, la cual tiene una frecuencia de 260 Hz y un ancho de pulso del 20 %.

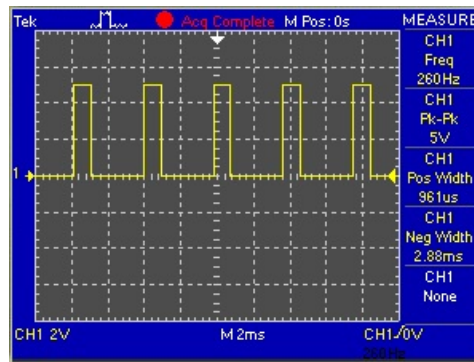


Figura 3.13: Señal PWM de frecuencia de 260 Hz.

3.2.4. Algoritmo de temperatura

Se realiza el monitoreo de temperatura en el motor de la banda transportadora, con el objetivo de asegurar un buen funcionamiento en la celda robótica, en caso de que el motor sobrepase los 40° este se desactiva y se genera una alerta notificando una posible falla sobre el motor. Para realizar la medición de temperatura se utiliza el sensor DS1920 de la compañía Maxim el cual va conectado al pin 2 de la tarjeta Arduino Mega, por medio de un algoritmo utilizando el protocolo 1-Wire se realiza la comunicación entre el sensor y la tarjeta Arduino Mega. Este sensor cuenta con una presentación iButton, tal como se muestra en la figura 3.14, la hoja de datos de este componente se encuentra en el apéndice B.

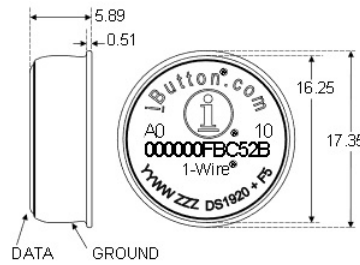


Figura 3.14: Sensor DS1920.

El sensor DS19b20 tiene un rango de medición de -55°C a 100°C , con una resolución de 0.5°C . Es un sensor digital que utiliza el protocolo de comunicación 1-Wire. Una de las principales ventajas es que sólo requiere de dos cables para establecer comunicación, un cable de referencia (cable de tierra) y un cable de datos. Sobre la línea de datos es posible conectar hasta 255 sensores de diferente. El protocolo 1-Wire de comunicación ha tenido una gran aceptación en la sector industrial, gracias a la gran cantidad de sensores conectados en una misma línea y la distancia que soporta, la cual es de 300 m sin energía parásita, hasta 1 km de distancia implementando un amplificador de corriente.

Cada sensor cuenta con un único número de identificación, el cual se encuentra formado por 8 bits correspondiente al código de la familia, 48 bit del número de serie y 8 bits de comprobación CRC. Estos últimos permiten asegurar que la comunicación entre el dispositivo maestro-esclavo sea correcta, brindando mayor seguridad que otros protocolos de comunicación. El protocolo de comunicación 1-Wire para el sensor DS1920 se muestra en la figura 3.15, el cual consiste en un pulso de reset, seguido por un pulso de presencia y un conjunto de comandos para solicitar el valor de la temperatura del sensor.

El maestro envía un pulso de reset para saber si existen sensores conectados, cuando un sensor se encuentra conectado envía un pulso de presencia, el maestro envía el comando $0x55$ seguido de la dirección física del sensor y del comando de conversión de temperatura $0x44$. Existe un tiempo de espera de 0.75 s. El maestro envía nuevamente el pulso de reset, el comando $0x55$, la dirección del dispositivo y el comando $0xBE$ que corresponde a la solicitud de temperatura. El sensor envía la información solicitada seguido del CRC. Por último el maestro envía un pulso de reset.

La comunicación de este sensor se encuentra basada en intervalos de tiempos, por ejemplo el pulso de reset, el maestro configura la línea de comunicación en estado bajo durante un intervalo menor de $480\ \mu\text{s}$ y después libera la línea en el estado alto. Cuando existe un sensor coloca la línea en estado bajo por un intervalo de tiempo de $60\ \mu\text{s}$ a $240\ \mu\text{s}$, en caso de no existir ningún sensor la línea se queda en alto. La lectura de este sensor consiste en leer el

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Reset pulse(480–960 μ s)
Rx	Presence	Presence pulse
Tx	55h	Issue “Match ROM” command
Tx	<64-bit ROM code>	Issue address for DS1920
Tx	44h	Issue “Convert Temperature” command
Tx	<DATA LINE HIGH>	Data line is held high for at least 0.75 seconds by bus master to allow conversion to complete.
Tx	Reset	Reset pulse
Rx	Presence	Presence pulse
Tx	55h	Issue “Match ROM” command
Tx	<64-bit ROM code>	Issue address for DS1920
Tx	BEh	Issue “Read Scratchpad” command.
Rx	<9 data bytes>	Read entire scratchpad plus CRC; the master now recalculates the CRC of the eight data bytes received from the scratchpad, compares the CRC calculated and the CRC read. If they match, the master continues; if not, this read operation is repeated.
Tx	Reset	Reset pulse
Rx	Presence	Presence pulse, done

Figura 3.15: Protocolo de comunicación 1-Wire para el sensor DS1920.

estado de la línea de comunicación en intervalos de tiempos específicos, en la figura 3.16 se muestra el comportamiento del pulso de reset.

El comportamiento es similar para realizar la lectura de un dato, el dispositivo esclavo coloca la línea en bajo 15 μ s, después de 60 μ s el maestro realiza la lectura del bus de datos, si la línea está en alto corresponde a un 1 lógico, caso contrario corresponde a 0 lógico, en la figura 3.17 se muestra el diagrama de tiempos del comando de lectura. En la implementación del algoritmo de lectura y escritura para este tipo de sensores, se debe de tener cuidado conforme a los tiempos, porque se puede leer o escribir un dato incorrecto. La obtención del CRC es fundamental tanto para la lectura y escritura del sensor, dado que garantiza la veracidad del dato. Cuando solo se utiliza un sensor con este protocolo, por medio del comando 0x44, no es necesario conocer la dirección del dispositivo; en caso de tener más de un sensor esta instrucción queda invalidada y es necesario proporcionar cada una de las direcciones.

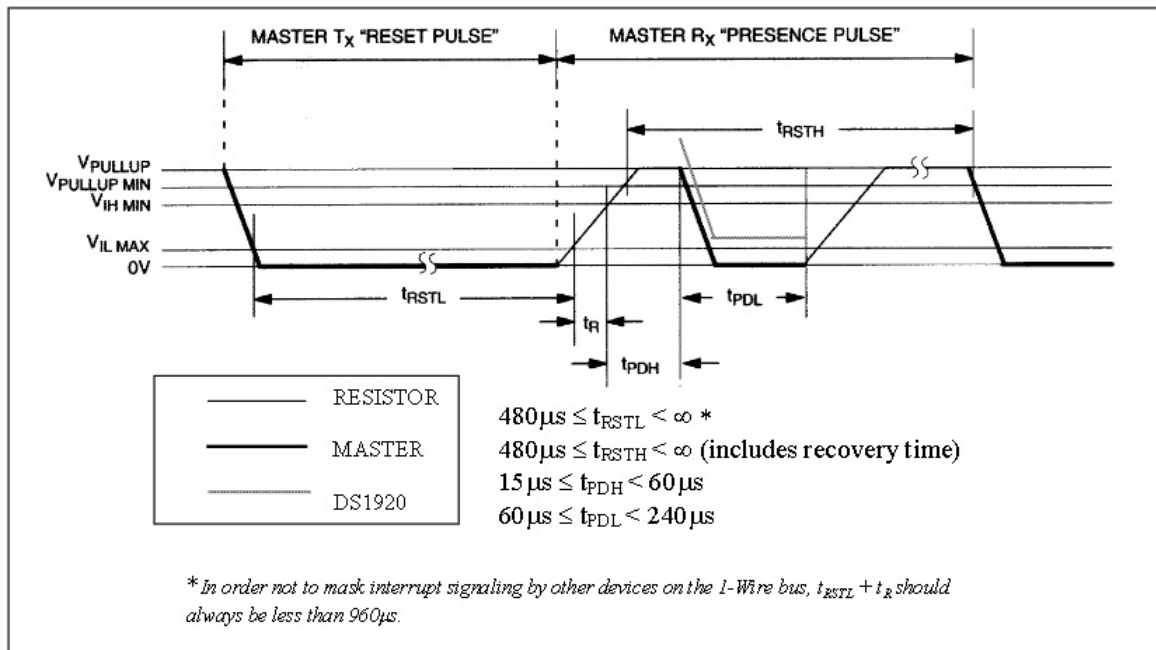


Figura 3.16: Pulso de reset de sensor DS1920.

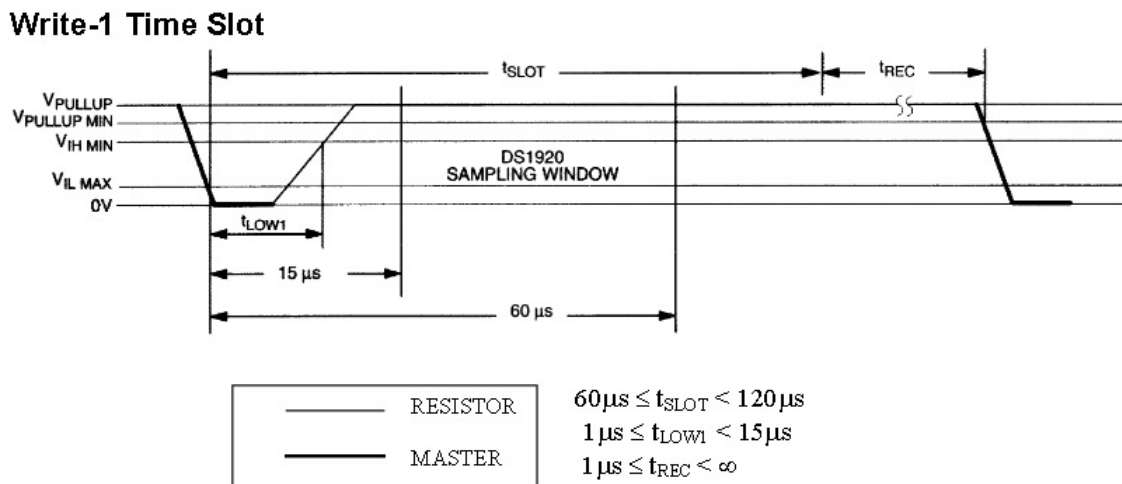


Figura 3.17: Diagrama de tiempo de comando de lectura de sensor DS1920.

El código implementado en la tarjeta Arduino Mega para realizar la lectura de este sensor se muestra en el apéndice C.2, este algoritmo es implementado para realizar la medición de temperatura del motor de la banda transportadora.

3.3. Diseño electrónico del ROTRADI

Para el sistema de control y adquisición de datos del ROTRADI es necesario considerar la velocidad y resolución de cada driver del ROTRADI, los cuales están configurados a una resolución de 10260 pulsos por revolución y una velocidad máxima de 1.5 Hz, obteniendo 46170 pulsos considerando los tres motores de transmisión directa. Es necesario utilizar una tarjeta de control que tenga la capacidad de leer estos pulsos, por lo tanto se propone la tarjeta Aduino Due, esta basada en un procesador de 32 bit ARM y a una velocidad de 84 MHz, permitiendo realizar operaciones a una velocidad de 12 ps. Se considera una tarjeta de bajo costo y con altas prestaciones.

Para implementar un algoritmo de control de posición en el ROTRADI es necesario realizar la lectura de cada uno de los encoder del ROTRADI, implementar un algoritmo de control y contar con una señal que represente la energía requerida para posicionar el ROTRADI en la posición deseada. En la figura 3.18 se muestra el diagrama general del sistema descrito.



Figura 3.18: Diagrama de electrónico de ROTRADI.

3.3.1. Lectura del encoder

El programa de control para el posicionamiento angular del ROTRADI consiste en la obtención de la posición actual del robot a través de la lectura de las señales del encoder del servoamplificador, utilizando los recursos de interrupción de la tarjeta Arduino Due, los cuales son recursos del microcontrolador para responder a eventos de alta prioridad, suspendiendo temporalmente el programa principal para ejecutar una tarea de interrupción [29]. La tarjeta Arduino Due posee 24 interrupciones por hardware y cada encoder requiere de dos interrup-

ciones para realizar una correcta lectura del encoder. La combinación de las señales A-B del encoder determina el sentido de movimiento en base al estado anterior, tal como se muestra en la figura 3.19.

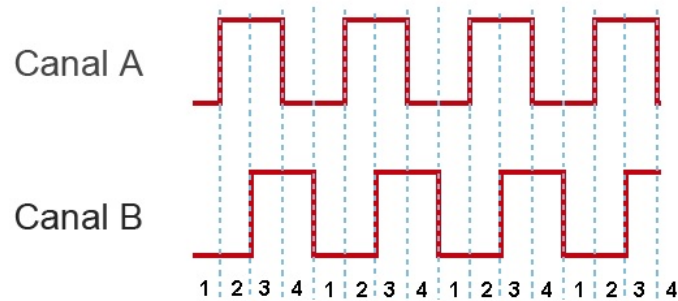


Figura 3.19: Diagrama de estados del encoder.

Analizando la forma en que se combinan la señales mostradas en la figura 3.19, se puede observar que tanto la señal A como la señal B se encuentran defasadas 90° . En equivalencia en los estados lógicos (0 y 1), en las fases 1 y 3 la señales A y B se encuentran en el mismo estado lógico, mientras que en las fases 2 y 4 son diferentes. Considerando estos cambios se puede determinar el sentido de giro de cada motor, por ejemplo, estando en la fase 1 ambas señales se encuentran en 0, cuando el motor gira en sentido horario la siguiente fase del encoder cambia a la fase 2, en caso de girar en sentido antihorario las señales cambian al estado 4 [29]. El servoamplificador proporciona una señal de cuadratura del encoder a una resolución de 10260 pulsos por revolución, tal como se muestra en la figura 3.20 [2].

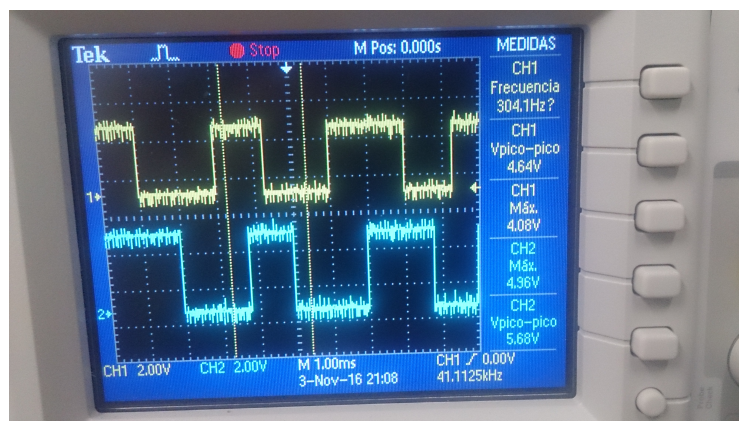


Figura 3.20: Señal del encoder proporcionada por el servoamplificador.

Es necesario filtrar la señal para eliminar el ruido que se observa en la figura 3.20. Se considera que el voltaje de salida de los encoders del ROTRADI corresponde a 5 Vcd, es

necesario reducir el voltaje a 3.3 Vcd, debido a que es el voltaje de entrada de los pines de la tarjeta Arduino DUE; si el voltaje es superior a este valor es posible que la tarjeta presenta lecturas erróneas. Se implementa un circuito electrónico constituido de una compuerta AND y un amplificador operacional en su modo de comparador. Se introducen las señales del encoder A y B hacia la compuerta AND, la entrada A a la compuerta A1 y la entrada B a la compuerta A3, ambas señales se encuentran conectadas en paralelo con capacitores de acoplamiento, con el objetivo de eliminar el ruido de la señal de entrada. Las señales de salida de la compuerta AND se conectan en la terminal negativa del amplificador operacional U2 y U4 en su configuración de comparador, por medio de un voltaje de referencia se descartan señales no deseadas con niveles inferiores al voltaje de referencia, esta señal se conecta a la terminal positiva. La salida de los comparadores es enviada a unas resistencias *pull-down*. El voltaje de alimentación del comparador, define el voltaje de entrada hacia el Arduino DUE. El diseño esquemático se muestra en la figura 3.21

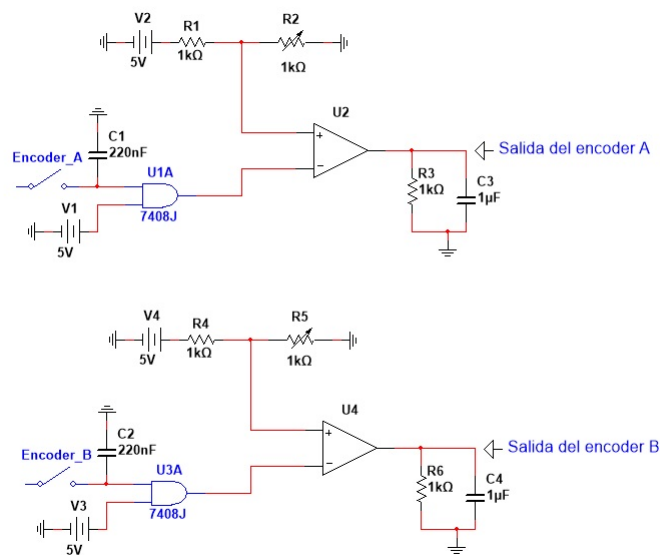


Figura 3.21: Esquemático para la lectura de la señal del encoder.

En la figura 3.22 se muestra el esquemático desarrollado en el programa Altium Designer para el diseño de la tarjeta electrónica. En este diagrama se muestran las conexiones de entrada del encoder del ROTRADI, las cuales se conectan en la terminal con2, la señal filtrada se proporciona a la tarjeta Arduino Due por medio de la terminal con3, en la terminal con1 se conectan los voltaje de alimentación para la tarjeta. Las terminales con4 y con5 sirven para proporcionar el voltaje de alimentación a las tarjeta que se colocarán en la parte superior.

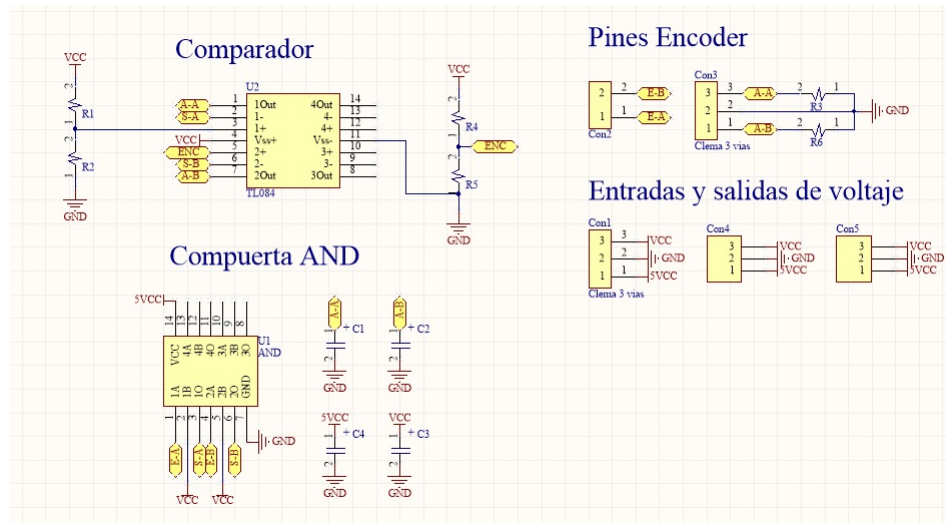


Figura 3.22: Esquemático de PCB para la lectura de la señal del encoder.

El diseño de la tarjeta desarrollada para la lectura de los encoders del ROTRADI se observa en la figura 3.23.

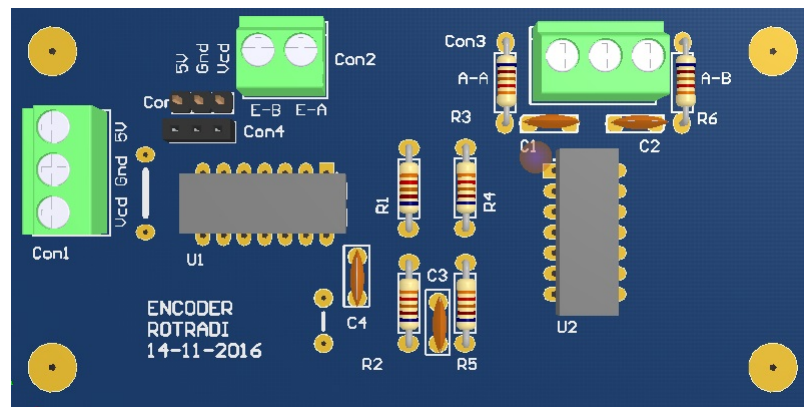


Figura 3.23: Diseño de PCB para la lectura de la señal del encoder.

El resultado obtenido del circuito anterior se presenta en la figura 3.24, se observa que se ha eliminado el ruido y el nivel de voltaje de salida corresponde a 3.5 Vcd, de esta manera se puede realizar la lectura del encoder utilizando la tarjeta Arduino Due. El voltaje de salida puede ser modificado desde los 1.7 Vcd hasta los 5 Vcd cambiando el voltaje de entrada en el conector con1 en la terminal Vcd. Los capacitores que se observan deben de ser de tantalio a $1\mu\text{F}$ para eliminar el ruido de la señal del voltaje de entrada.

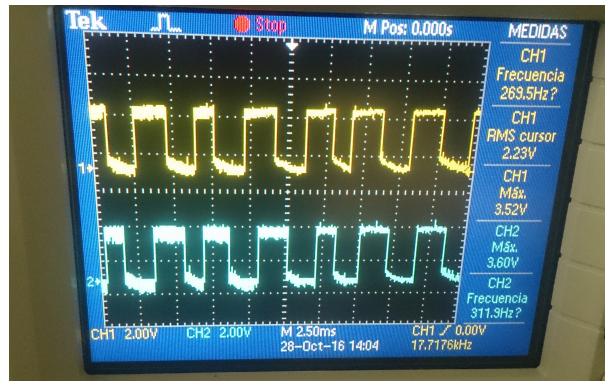


Figura 3.24: Señal del encoder filtrada.

3.3.2. Convertidor digital a analógico

Para el control de la plataforma se precisa de la aplicación de voltaje, el cual debe estar en un rango de ± 5 Vcd y cuyo valor va a definir el par aplicado. Tomando en cuenta lo anterior se implementa un DAC. Se utiliza el integrado MCP4725, el cual es un convertidor digital a analógico, su rango de voltaje de operación se encuentre entre los 2.7 Vcd hasta los 5.5 Vcd, proporciona un voltaje de salida de hasta 5.5 Vcd, trabaja en un rango de temperatura desde los -40°C a 125°C , contiene una memoria EEPROM que permite almacenar en su memoria un nivel de voltaje de salida predeterminado, es decir, cada vez que el dispositivo se energice proporcione el mismo voltaje de salida, este integrado tiene una resolución de 12 bits, emplea el protocolo de comunicación I^2C para su configuración. En la figura 3.25 se muestra el dispositivo utilizado.

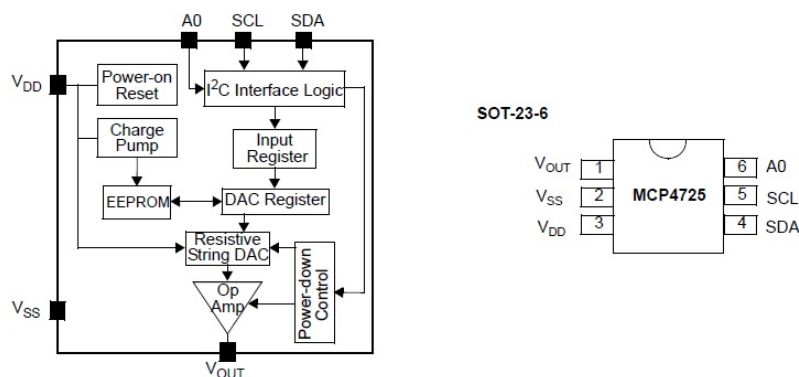


Figura 3.25: MCP4725 convertidor digital a analógico.

Para enviar datos al DAC se utiliza el protocolo I^2C al mismo tiempo respetando el comando de escritura de este dispositivo, tal como se muestra en la figura 3.26, donde en primer lugar se manda la dirección del dispositivo, la cual corresponde al número 0x46, posteriormente

se indica el modo de operación, el siguiente byte está compuesto de 4 bits de modo de trabajo y 4 bits más significativos del valor que se desea enviar, el siguiente byte corresponde a los bits menos significativos y por último el bit de parada. En caso de querer enviar varios datos de manera consecutiva, después del bit de parada se pueden enviar los últimos dos bytes sin necesidad de mandar la dirección del dispositivo.

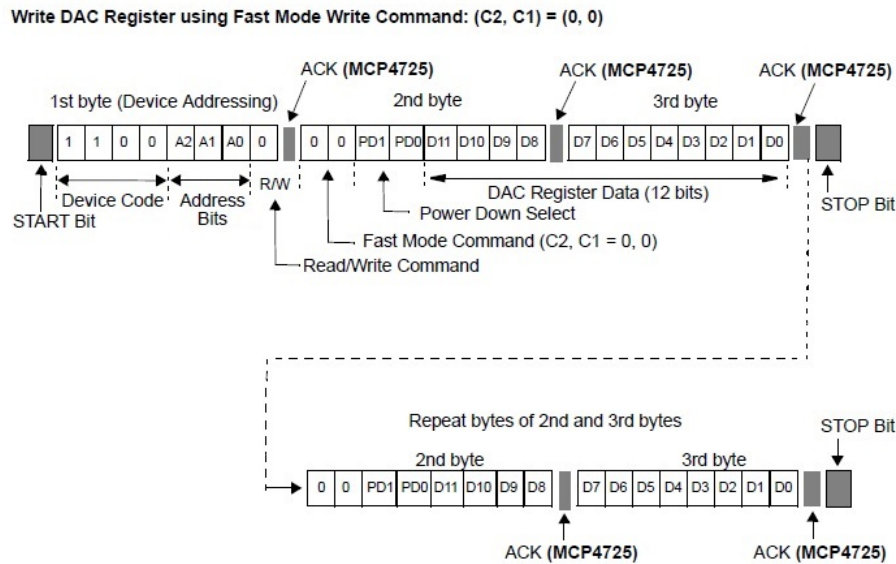


Figura 3.26: Protocolo de comunicación I²C para escritura de datos.

Se desarrolló una tarjeta electrónica para obtener un voltaje de salida de -7.5 Vcd a 7.5 Vcd. Para obtener este resultado se utilizó el amplificador operacional TL084, el cual contiene 4 operacionales internos. Considerando el diagrama eléctrico que se muestra en la figura 3.27 se describe el modo de operación de la tarjeta desarrollada. Para el uso del protocolo de comunicación I²C se emplearon dos resistencias *pull-up*, correspondientes a R1 y R2, la salida del DAC se conectó al operacional U3 configurado como seguidor de voltaje. El operacional U4 funciona como seguidor de voltaje para la referencia de 2.5 Vcd, dado que el voltaje es obtenido por un arreglo de resistencias en serie y es necesario colocar un seguidor de voltaje. El operacional U5 se encuentra diseñado como un restador-amplificador, restando el voltaje entrada de 2.5 Vcd del divisor de voltaje al voltaje de salida del DAC, el voltaje obtenido es amplificado 2 veces, de esta manera se obtienen valores positivos y negativos de voltaje. El último operacional funciona como seguidor de voltaje. Para ajustar la salida de voltaje se tienen dos resistencias variables, la resistencia R5 se encarga de ajustar el valor de 0 Vcd y la resistencia variable R8 ajusta la ganancia que debe de proporcionar 7.5 Vcd de salida.

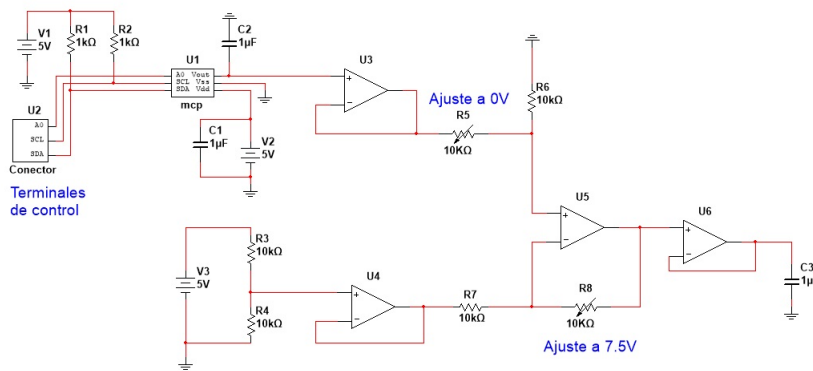


Figura 3.27: Esquemático del DAC.

En la figura 3.28 se muestra el esquemático desarrollado en el programa Altium Designer para el diseño de la tarjeta del convertido digital a analógico para proporcionar la señal de control hacia el driver del ROTRADI. En los conectores Con1 y Con2 se introducen los voltajes de alimentación, correspondientes a 12 Vcd, -12 Vcd y 5 Vcd. La terminal Con3 proporciona las señales I^2C de la tarjeta Arduino Due hacia el módulo MCP4725. La terminal Con6 proporciona el voltaje analógico de salida hacia el driver del ROTRADI. El conector Con4 selecciona el módulo MCP4725 para proporcionar la señal de salida a cada uno de los grados de libertad del ROTRADI. Los conectores C5, C7 y C8 proporcionan los voltajes de alimentación y señales de control a las tarjetas que se colocan en la parte superior, considerando que este diseño es en forma de torreta.

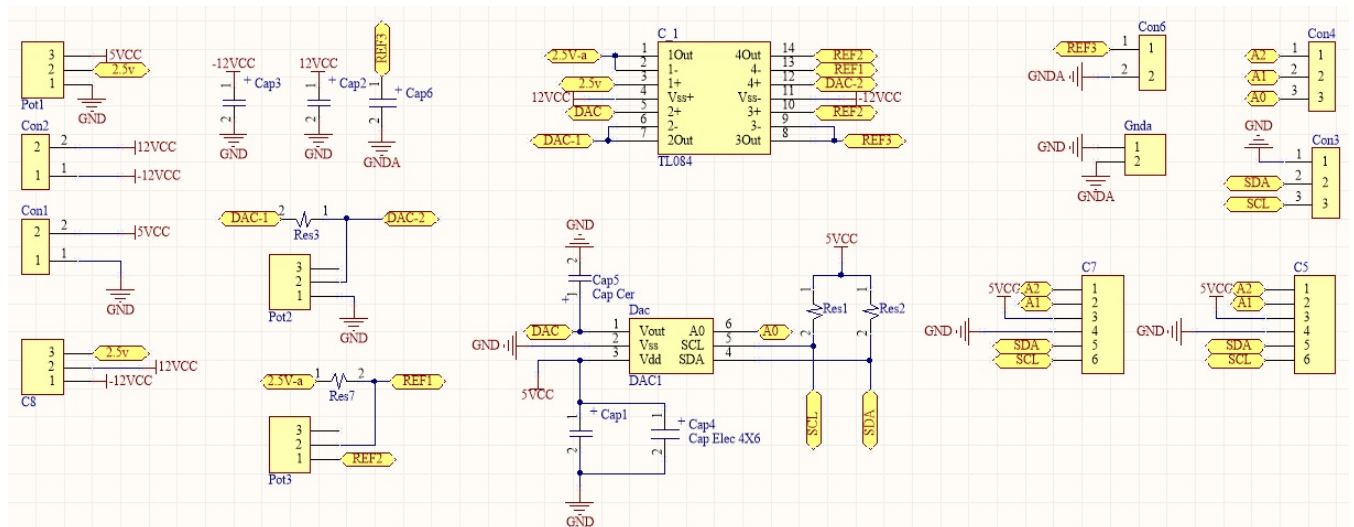


Figura 3.28: Diseño de la PCB del DAC.

En la figura 3.29 se muestra el diseño de la PCB del DAC. El diseño de estas tarjetas se realiza en forma de torreta para reducir el tamaño del driver de comunicación.

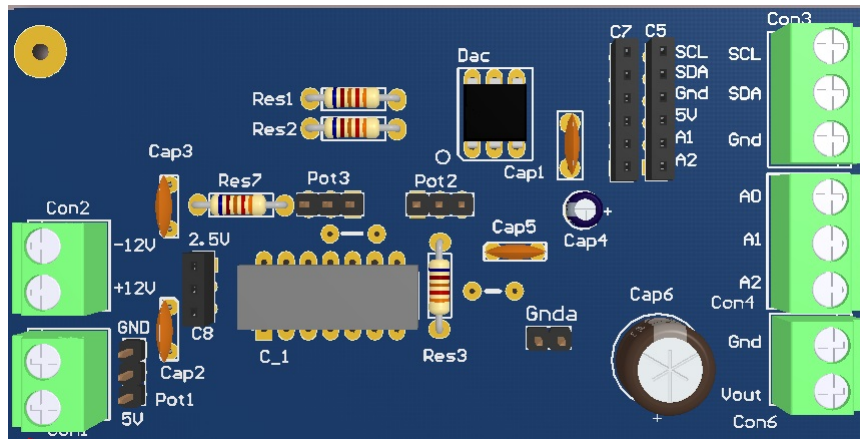


Figura 3.29: Diseño de la PCB del DAC para proporcionar la señal de control hacia el ROTRADI.

Considerando que el ROTRADI cuenta con tres servomotores se desarrollan tres placas de lectura de encoder y tres placas de DAC's. Para el caso de las tarjetas de DAC's es necesario realizar un barrido de voltaje para saber si las tarjetas desarrolladas son lineales. En la figuras 3.30, 3.31 y 3.32 se muestra la salida de voltaje de cada tarjeta diseñada, donde se observa que el voltaje de salida de cada tarjeta es lineal.

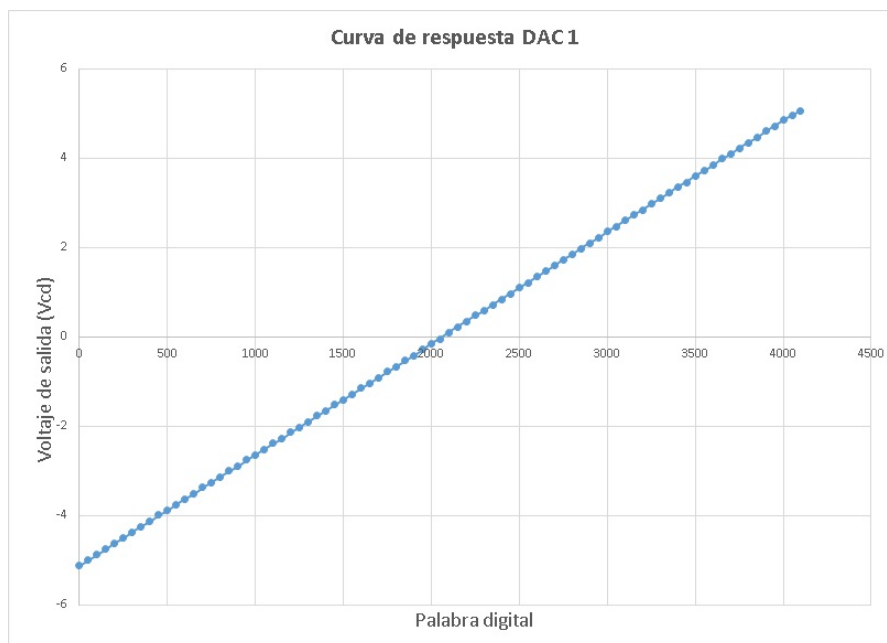


Figura 3.30: Respuesta de tarjeta DAC 1.

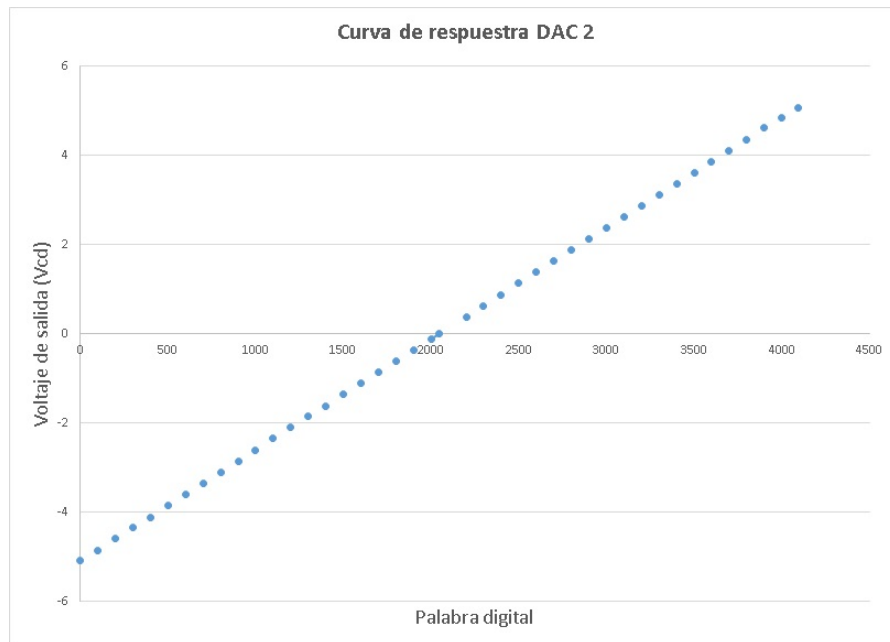


Figura 3.31: Respuesta de tarjeta DAC 2.

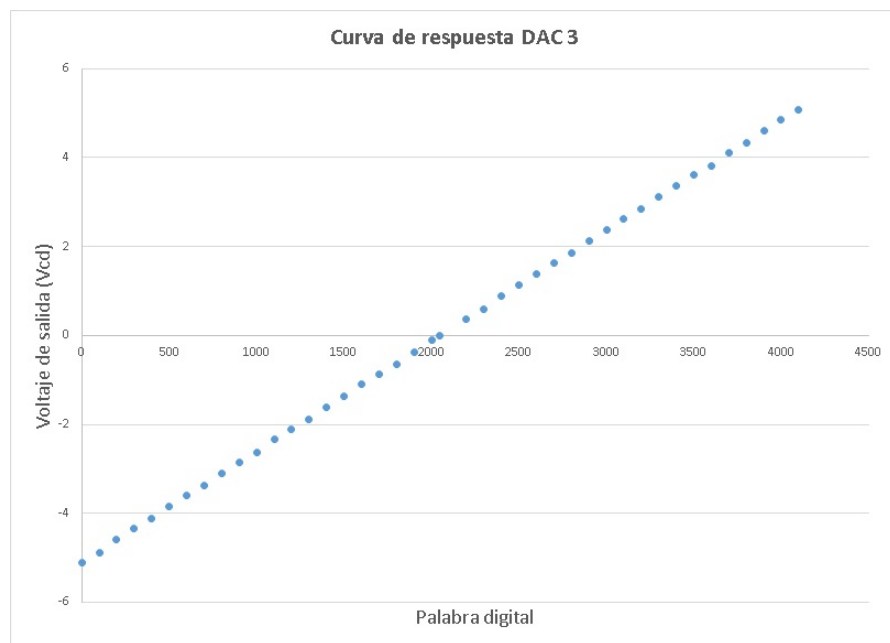


Figura 3.32: Respuesta de tarjeta DAC 3.

En este capítulo se desarrolló la electrónica necesaria para el control de la celda robótica, así como la instrumentación para el control de la banda transportadora y las tarjetas de control para el ROTRADI.

Capítulo 4

Diseño de algoritmos de detección de movimiento, reconocimiento de características y detección de componentes

En este capítulo se muestra los algoritmos desarrollados para realizar el reconocimiento de componentes en una tarjeta electrónica, el proceso empleado se muestra a continuación:

- Las tarjetas electrónicas serán colocadas en el banda transportadora, está permanecerá en movimiento hasta que la tarjeta electrónica sea identificada por la cámara web.
- Por medio del algoritmo de reconocimiento se identificará la tarjeta electrónica. Considerando la tarjeta electrónica que el operador ha seleccionado se procede a realizar una de las siguientes opciones: en caso de que la tarjeta identificada sea la tarjeta definida por el usuario se procederá a realizar el proceso de supervisión, en caso contrario la tarjeta deberá de ser descartada.
- El proceso de supervisión consiste en identificar que la tarjeta electrónica contenga todos los componentes electrónicos, tomando como referencia una base de datos exportada del programa Altium Designer, en caso de que se identifiquen todos los componentes se notificará el estado correcto de la tarjeta, en caso contrario se notificará cuáles son los componentes faltantes en dicha tarjeta.

Además, en este capítulo se muestran los programas desarrollados en MATLAB para el proceso de detección de movimiento, identificación de características críticas y reconocimiento de componentes, utilizando un cámara web, la cual se muestra en la figura 4.1.



Figura 4.1: Cámara web.

Se utiliza cámara web de 5 M pixeles a 30 cuadros por segundo, con resolución dinámica de 640x480 pixeles, por lo tanto es una cámara web normal de bajo costo. La implementación del sistema se realizó en el Laboratorio de Robótica en la FCE-BUAP, este presenta una luminiscencia en el intervalo de 90 lux a 380 lux , no se requiere de una fuente de luz adicional. La cámara web se coloca de manera perpendicular con respecto a la banda transportadora a una distancia de 28 a 50 cm. La separación entre cada tarjeta electrónica en la banda transportadora es de 15 cm. En la figura 4.2 se muestra la colocación de la cámara web con respecto a la banda transportadora.



Figura 4.2: Colocación de la cámara web y la banda transportadora.

Las especificaciones técnicas de la banda transportadora se muestran en la tabla 4.1, la velocidad de desplazamiento utilizada es de 0.4 m/s.

Descripción	Valor
Fabricante	Rexroth
Modelo	Variow Flow S
Ancho de la cinta transportadora	12 cm
Velocidad de desplazamiento	0.1 - 0.8 m/s

Tabla 4.1: Características físicas de la banda transportadora.

Las especificaciones técnicas de las tarjetas electrónicas se muestran en la tabla 4.2, las cuales deben de ser respetadas para el correcto funcionamiento del sistema.

Descripción	Valor
Ancho máximo de la PCB	12 cm
Ancho mínimo de la PCB	5 cm
Largo máximo de la PCB	16 cm
Largo mínimo de la PCB	6 cm
Tiempo de reconocimiento	4 s - 6 s

Tabla 4.2: Especificaciones técnicas de PCB.

4.1. Algoritmo de detección de movimiento

El programa implementado para la detección de movimiento consiste en configurar la cámara web a una resolución de 320x240 pixeles, es necesario tomar dos imágenes consecutivas, estas serán transformadas a imágenes en el formato de escalas de grises, posteriormente se realiza una diferenciación entre cada una de las imágenes. El proceso de identificación de movimiento consiste en comparar dos imágenes de un mismo lugar, considerando que la imagen se encuentra en el formato RGB, es necesario convertir la imagen en el formato de escalas de grises, resulta más simple trabajar con una matriz de una dimensión que una matriz de tres dimensiones, cuando no existe movimiento las dos imágenes adquiridas son iguales por lo tanto al realizar el proceso de diferenciación el resultado es una matriz de ceros, indicando que no existe diferencia. Caso contrario cuando existe movimiento, el resultado de la diferenciación entre ambas matrices resulta ser una matriz con valores diferentes a ceros. En la figura 4.3 se muestra el diagrama a bloques del proceso descrito.

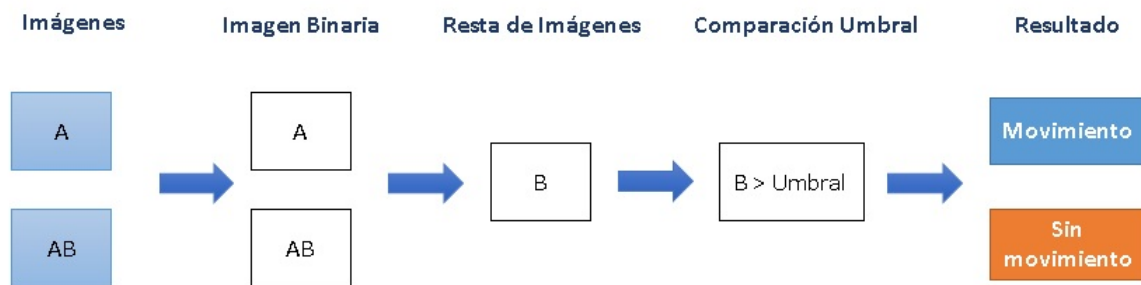


Figura 4.3: Diagrama de bloques de algoritmo de identificación de movimiento.

En este caso se desea detectar cambios en una sola región de la imagen, para ello es necesario dividir la imagen en dos secciones, posteriormente se evalúan los cambios solo en la sección seleccionada. El valor obtenido es comparado con un valor predeterminado, en caso de que el valor obtenido sobrepase al valor asignado se considera la existencia de movimiento y se genera un mensaje para indicar este evento, caso contrario cuando el valor obtenido no sobrepase el valor asignado, no se reporte presencia de movimiento.

En la figura 4.4 se muestra el resultado del proceso anteriormente descrito. Se ejecuta una prueba empleando una tarjeta electrónica, la cual al no ser desplazada el programa no detecta cambio de movimiento, el resultado es una matriz con valores iguales a ceros, la imagen del lado izquierdo muestra la imagen real de la tarjeta, la imagen del lado derecho muestra el resultado de la diferenciación entre dos imágenes, el título de esta última imagen menciona el estado del evento, para este ejemplo indica que no se ha detectado movimiento.

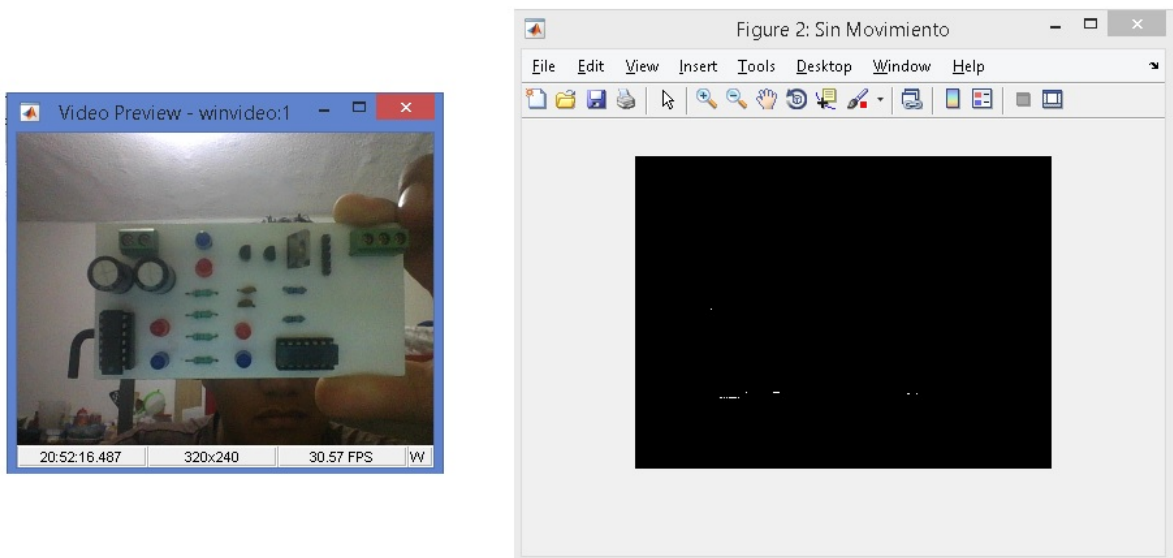


Figura 4.4: Algoritmo detector de movimiento 1.

En la figura 4.5 se muestra el resultado de desplazar la tarjeta electrónica, el algoritmo identifica que existe una diferencia entre la dos imágenes obtenidas por medio de la cámara web, tal como se muestra en la imagen del lado izquierdo, esta contiene una representación próxima a la imagen real en representación de una imagen en blanco y negro, dando como resultado la identificación del movimiento, el título de la imagen cambia, indicando que se detectado movimiento.

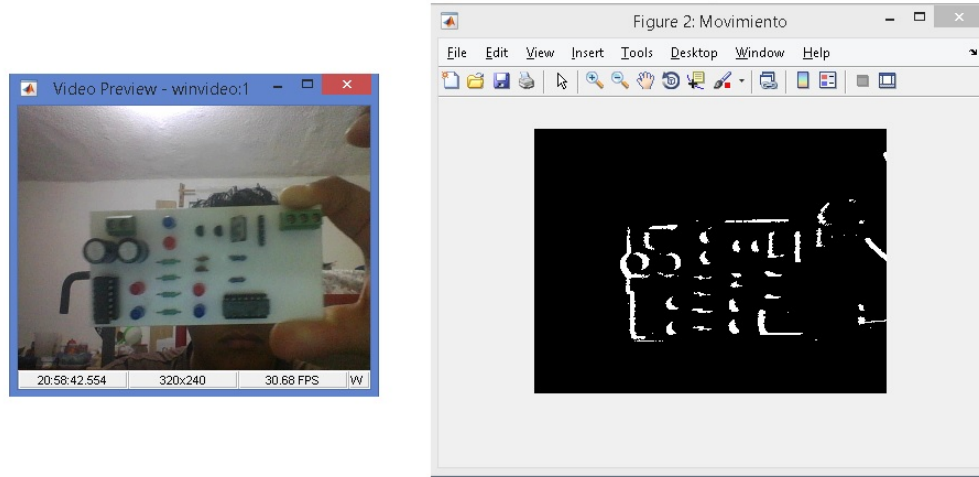


Figura 4.5: Algoritmo detector de movimiento 2.

4.2. Algoritmo de reconocimiento de características críticas

En esta sección consiste en realizar la identificación de las características críticas de una imagen a partir del algoritmo propuesto por *Paul Viola* en el año 2002 en su publicación *Rapid Object Detection using a Boosted Cascade of Simple Features* [30]. Este algoritmo consiste en identificar las características principales de un conjunto de imágenes de un mismo objeto y posteriormente poder identificarlo. Este algoritmo fue posteriormente empleado para la identificación de rostros tal como lo reporta *Ole Helvig Jensen* en su publicación *Implementing the Viola-Jones Face Detection Algorithm* [31].

El algoritmo propuesto por *Paul Viola* se encuentra constituido por tres aportaciones importantes, la primera aportación es la representación de una nueva imagen llamada *Integral image*, la cual consiste en realizar un análisis de un conjunto de imágenes representadas en el formato de escala de grises o también llamado formato de intensidades, se realiza una suma de todos los píxeles de la imagen para poder obtener un margen de referencia.

La segunda contribución de este trabajo es la construcción de un clasificador basado en un número pequeño de características importantes de una imagen utilizando un algoritmo de aprendizaje AdaBoost, de esta manera se realiza el reconocimiento de las características de una imagen. Para poder garantizar una rápida identificación, el proceso de aprendizaje debe excluir una gran cantidad de funciones disponibles, se centra solamente en un conjunto de características críticas.

La tercera contribución consiste en implementar un conjunto de clasificadores combinados sucesivamente utilizando la estructura cascada, de esta manera la identificación resulta ser más rápida y es posible realizar el análisis de imágenes más complejas. El análisis resulta ser demasiado rápido que se pueden generar falsos positivos, lo cual es un proceso a mejorar, en la figura 4.6 se muestra el diagrama de flujo del proceso descrito anteriormente.

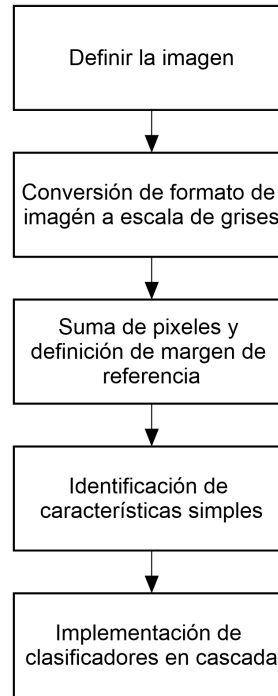


Figura 4.6: Algoritmo de Paul Viola.

4.2.1. Algoritmo de aprendizaje automático

El algoritmo de aprendizaje automático es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. Estos algoritmos son capaces de generalizar comportamientos y reconocer patrones a partir de una información suministrada en forma de ejemplos. El algoritmo de aprendizaje automático se relaciona con el área de la probabilidad y estadística, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático se centra más en el estudio de la complejidad computacional de los problemas.

El aprendizaje automático aplicado al procesamiento digital de imágenes se basa en extraer información sobre algunas propiedades observadas en un conjunto de imágenes repetitivas de un mismo objeto, generando un conjunto de etiquetas que contiene esta información. Las

etiquetas generadas son almacenadas en una matriz de información.

Las ventajas que tiene este algoritmo es su amplia variabilidad en el contenido visual de las imágenes, por ejemplo la iluminación, tamaño, posición y fondo de la imagen. Al mismo tiempo permite identificar más de un objeto en una misma imagen. En la figura 4.7 se muestra el esquema general del algoritmo de aprendizaje automático, el objetivo es extraer las características principales de una imagen, guardar esta información en una base de datos y poder identificar en un otra imagen con diferentes variantes la imagen original.

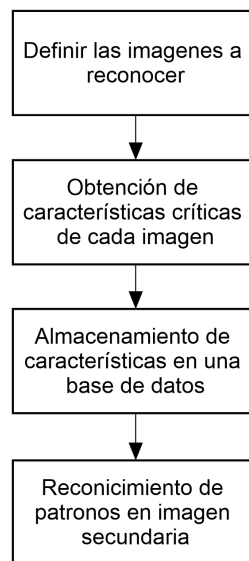


Figura 4.7: Algoritmo de aprendizaje automático.

La extracción de las características consiste en extraer las características visuales de una imagen, donde cada característica corresponde a un vector numérico que describe de manera matemática una región visual de la imagen. La extracción de características se conforma de dos metodologías. La primera es la detección de puntos de interés, los cuales corresponde a los cambios de contraste dentro de la imagen, generación de bordes y detección de esquinas. La segunda corresponde a la descripción local de los puntos de interés en una representación vectorial. En la figura 4.8 se muestran los puntos de interés de una tarjeta, para este ejemplo se detectan las 100 características más importantes, entre las cuales se destacan los contornos, los cambios de contraste y los cambios de blanco y negro.

Cada punto que se obtiene se encuentra representado dentro de una matriz, esta matriz se llama matriz de información y contiene las características más importante que definen a la tarjeta. Para poder obtener un mejor resultado es necesario tomar diferentes muestras del objeto que se desea identificar. El algoritmo de aprendizaje automático requiere de al me-



Figura 4.8: Extracción de características.

nos 100 muestras del mismo objeto, de cada muestra se obtiene su matriz de información, se implementa un algoritmo que compara las características encontradas en cada matriz y define cuáles son las características que predominan sobre el objeto, la información obtenida se almacena en una matriz diferente.

El segundo paso consiste en representar la matriz de información en un arreglo vectorial para su fácil interpretación. Este vector se crea cuando existe más de un objeto que se desea identificar, el arreglo vectorial de cada imagen se define como puntos locales. En la figura 4.9 se observa la tarjeta que se desea identificar en conjunto con la banda transportadora, la tarjeta se encuentra en el centro de la imagen.



Figura 4.9: Tarjeta colocada sobre la banda transportadora.

Cuando se implementa el algoritmo de aprendizaje se realiza una búsqueda en la imagen de las características del objeto que se desea identificar, en la figura 4.10 se observa el algoritmo que identifica estas características en el centro de la imagen correspondientes a la tarjeta electrónica.

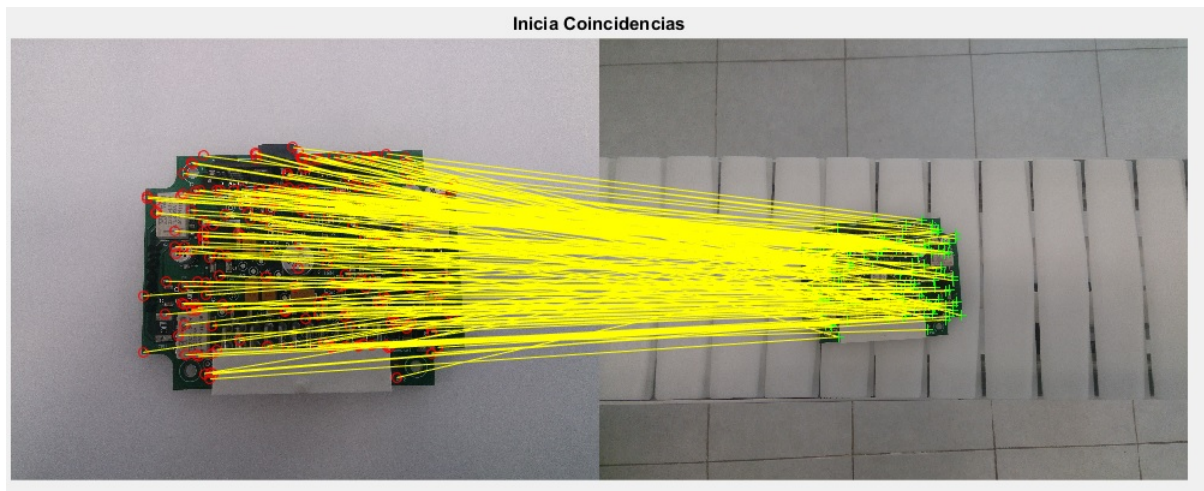


Figura 4.10: Búsqueda de características.

En el siguiente paso se remueven las características fuera de la transformación geométrica, tal como se observa en la figura 4.11, se reduce el número de las características identificadas. Este proceso es muy importante para eliminar características que no correspondan al objeto, por ejemplo, es posible que más de un objeto presente algunas características similares al objeto a identificar, por lo que el objeto resulta ser confundido, el propósito del algoritmo de filtrado es descartar estos eventos.

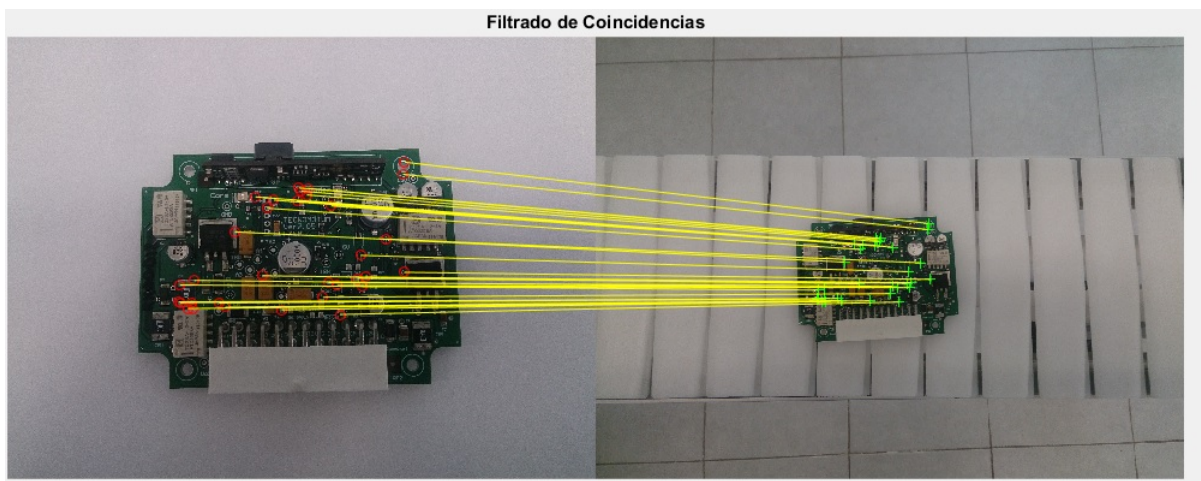


Figura 4.11: Filtrado de características críticas.

El último paso es clasificar el objeto, en la figura 4.12 se observa que el objeto ha sido identificado correctamente dentro de la imagen.

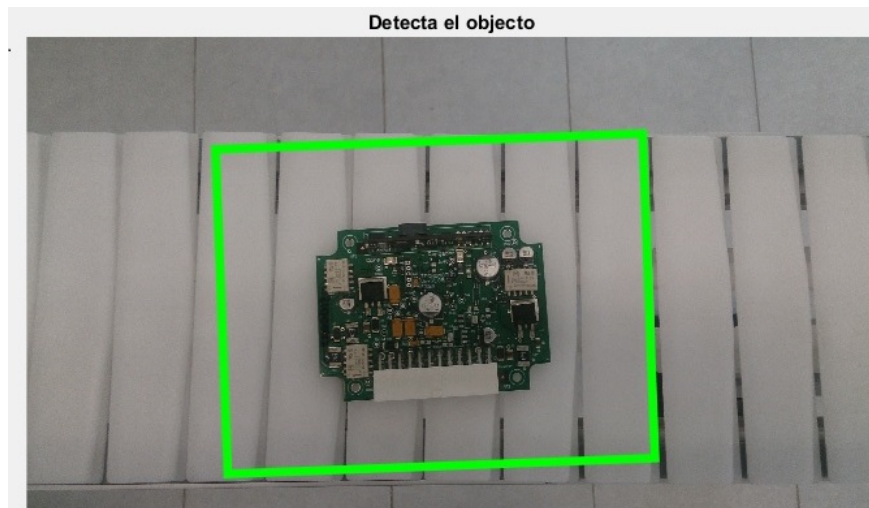


Figura 4.12: Tarjeta identificada dentro de la imagen.

La importancia de los algoritmos de aprendizaje automático radica en la estructura de los clasificadores, los cuales contiene la metodología de la obtención de los puntos de interés y la descripción local de los puntos. La plataforma de MATLAB contiene diferentes tipos de clasificadores, como por ejemplo *Simple Tree*, *linear SVM*, *Cubic SVM*, *Fine KNN*, entre otros [32].

En MATLAB se encuentra implementado el algoritmo propuesto por Paul-Viola dentro de sus librerías, el cual será empleado para realizar el reconocimiento de las tarjetas electrónicas; para utilizar este algoritmo se emplea el siguiente procedimiento:

- Realizar la captura de imágenes en diferentes perspectivas de cada una de las tarjetas electrónicas que se desean identificar.
- Agrupar las imágenes obtenidas en diferentes carpetas, el nombre de la carpeta es el identificador dentro del proceso de aprendizaje.
- Empleando el software de MATLAB se realiza la lectura de cada una de las carpetas.
- Identificar las características críticas de cada una de las tarjetas, como resultado se obtiene una matriz con las parámetros más importantes de cada tarjeta
- MATLAB contiene diferentes clasificadores, seleccionar el que ofrezca mayor estabilidad en reconocimiento.
- Realizar las pruebas físicas del sistema entrenado y comprobar su eficacia.

Para realizar la captura de las imágenes se desarrolla una Guide que nos permita realizar la adquisición de imágenes utilizando la cámara web y guardando estas en formato jpg [33]. La

Guide desarrollada se muestra en la figura 6.15, contiene tres botones y una sección donde se muestra el video adquirido. El botón *Encender Camara* enciende la cámara web, al presionar el botón *Adquiere Imagen* se realiza la captura de 500 imágenes consecutivas, las cuales son almacenadas bajo el nombre de tarjetax.jpg (este nombre se puede cambiar de acuerdo al nombre de la tarjeta), donde el valor de x representa el número de imágenes adquiridas. Si se desea suspender el proceso de adquisición de imágenes se cuenta con el botón *Parar*, este desactiva la cámara de video.

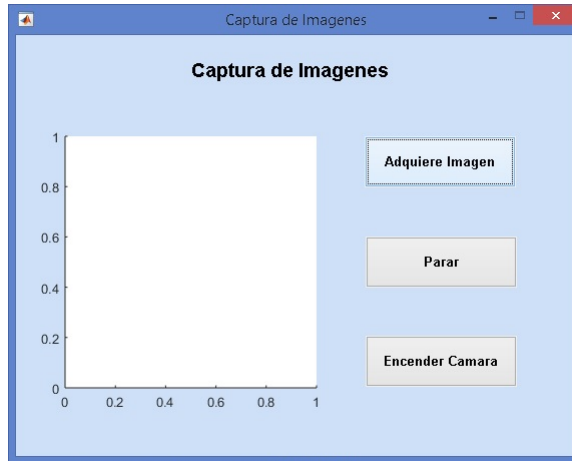


Figura 4.13: Guide de captura de imágenes.

Para la adquisición de imágenes de cada una de las tarjetas a identificar se utiliza la Guide anterior, la captura de las imágenes debe de ser en diferentes ángulos de vista. Empleando esta Guide se realiza la captura de las siguientes tarjetas.



Figura 4.14: Tarjeta Arduino Mega y Motum.

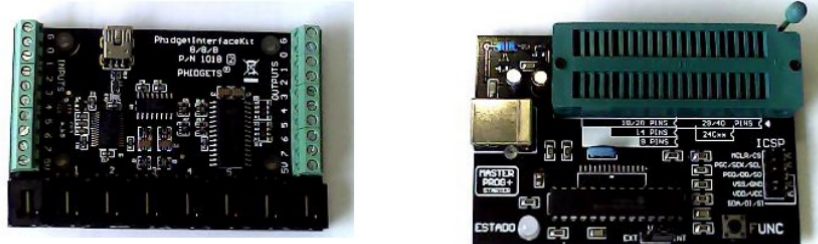


Figura 4.15: Tarjeta selector y programador.



Figura 4.16: Tarjeta controlador y programador.

Para realizar la implementación del algoritmo de Paul-Viola es necesario contar con las imágenes de los objetos a identificar, para el desarrollo de esta tesis se consideran 6 tipos de tarjetas electrónicas, se realiza la captura de imágenes de cada una de las tarjetas y se almacenan en carpetas diferentes, tal como se muestra en la figura 4.17.

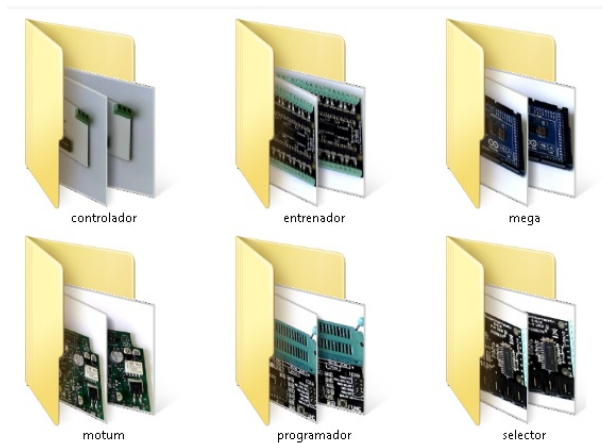


Figura 4.17: Carpetas de imágenes de tarjetas electrónicas.

Utilizando el software de MATLAB se realiza la lectura de la carpeta utilizando la instrucción `imset = imageSet('tarjetas', 'recursive')`, en este caso el nombre de la carpeta que contiene las imágenes se llama `'tarjetas'`.

Se obtienen las características críticas de las imágenes, mediante la instrucción `bag = bagOfFeatures(imset, 'VocabularySize', 200, 'PointSelection', 'Detector')`, esta instrucción genera un vector de características críticas, es decir, de cada imagen se extraen 200 características relevantes y se almacenan dentro un vector, por lo tanto el vector obtenido contiene 200 características de cada imagen, dado que tenemos 6 diferentes tarjetas se obtienen 6 diferentes vectores. En la figura 4.18 se muestra el resultado de este proceso.

```

Encoding 6 image sets using Bag-Of-Features.
-----

* Image set 1: controlador.
* Image set 2: entrenador.
* Image set 3: mega.
* Image set 4: motum.
* Image set 5: programador.
* Image set 6: selector.

* Encoding 501 images from image set 1...done.
* Encoding 501 images from image set 2...done.
* Encoding 501 images from image set 3...done.
* Encoding 501 images from image set 4...done.
* Encoding 501 images from image set 5...done.
* Encoding 501 images from image set 6...done.

```

Figura 4.18: Vector de características críticas.

El siguiente paso a implementar es realizar una matriz que contenga solo las características más importantes de cada una de las tarjetas, mediante la instrucción `imagefeatures = encode(bag, imset)`, se agrupan todas las características obtenidas y se seleccionan solo aquellas que sean más concurrentes. En la figura 4.19 se muestra el proceso realizado por MATLAB para la obtención de la matriz de concurrencia.

```

* Extracting SURF features using the Detector selection method.
** detectSURFfeatures is used to detect key points for feature extraction.

* Extracting features from 501 images in image set 1...done. Extracted 67809 features.
* Extracting features from 501 images in image set 2...done. Extracted 268210 features.
* Extracting features from 501 images in image set 3...done. Extracted 190484 features.
* Extracting features from 501 images in image set 4...done. Extracted 309067 features.
* Extracting features from 501 images in image set 5...done. Extracted 299511 features.
* Extracting features from 501 images in image set 6...done. Extracted 297973 features.

```

Figura 4.19: Matriz de características críticas.

Una vez obtenida la matriz de características críticas, se procede a realizar el entrenamiento del clasificador, el objetivo principal es reconocer de manera correcta cada una de las tarjetas con el mínimo error de confusión. La instrucción `classificationLearner` MATLAB permite realizar el entrenamiento de diferentes clasificadores, cuando se ejecuta este comando MATLAB solicita seleccionar la matriz de características críticas, tal como se muestra en la figura 4.20.

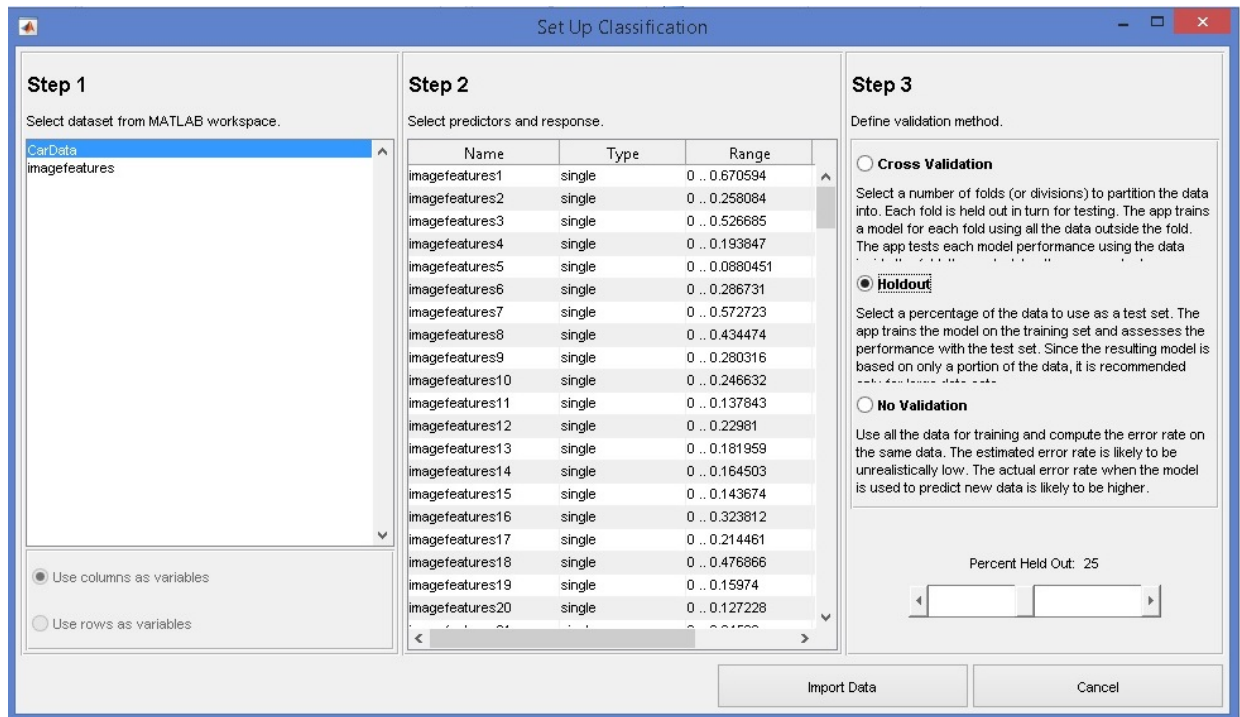


Figura 4.20: Selección de matriz de características críticas.

Una vez seleccionados los datos, se procede a entrenar el clasificador; MATLAB proporciona diferentes tipos de clasificadores basados en diferentes metodologías, resulta conveniente realizar el entrenamiento de diferentes clasificadores para observar el comportamiento de cada uno de ellos. En la figura 4.21 se muestran diferentes clasificadores entrenados y el porcentaje de efectividad que presentan, por ejemplo el clasificador con menor índice de efectividad es el *Simple Tree*, en cambio uno de los clasificadores con mayor eficiencia es el *Cubic KNN* [32].

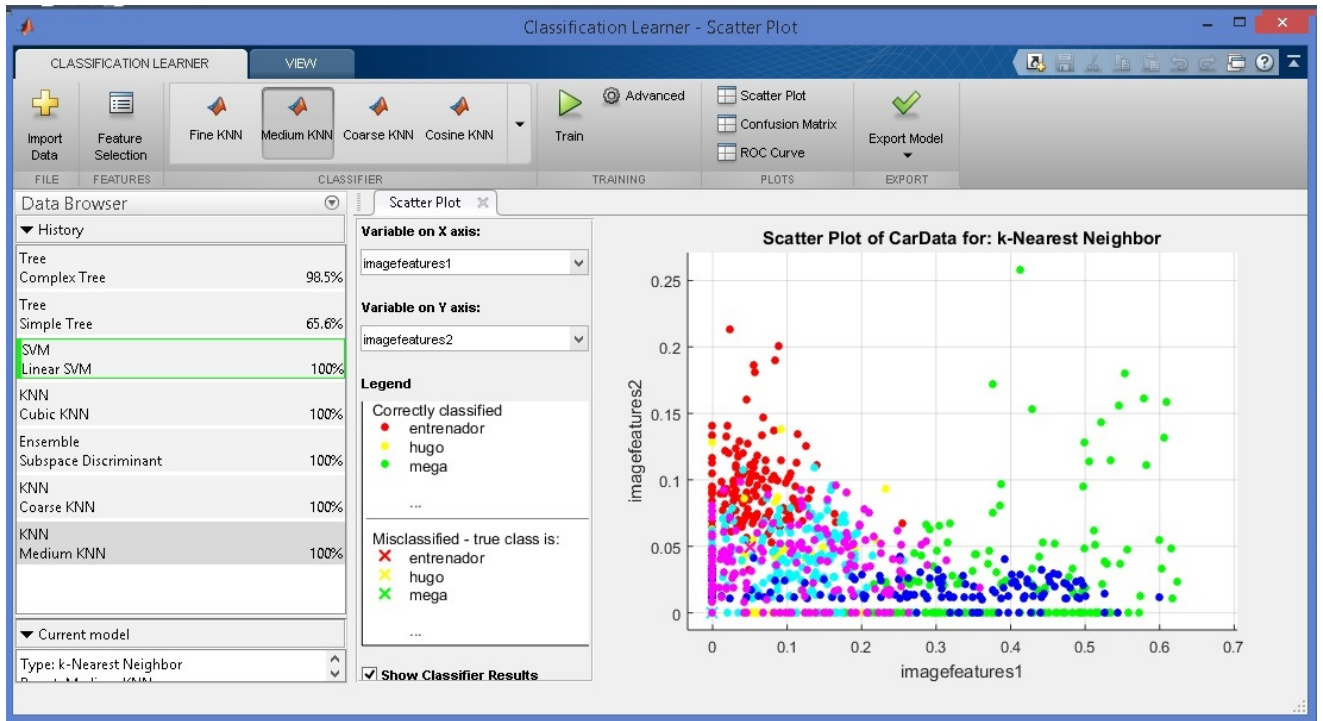


Figura 4.21: Entrenadores de Clasificador.

El algoritmo *Cubic KNN* se basa en clasificación de la clase más frecuente a la que pertenecen sus K vecinos.

		X_1	\dots	X_j	\dots	X_n	C
(x_1, c_1)	1	x_{11}	\dots	x_{1j}	\dots	x_{1n}	c_1
(x_i, c_i)	i	x_{i1}	\dots	x_{ij}	\dots	x_{in}	c_i
(x_N, c_N)	N	x_{N1}	\dots	x_{Nj}	\dots	x_{Nn}	c_j
(x_N, c_N)	N	x_{N1}	\dots	x_{Nj}	\dots	x_{Nn}	c_N
x	$N + 1$	$x_{N+1,1}$	\dots	$x_{N+1,j}$	\dots	$x_{N+1,n}$	

Tabla 4.3: Parámetros de DH del péndulo.

- X_N indica un fichero con N casos, cada uno de los cuales está caracterizado por n variables predictoras X_1, \dots, X_n y una variable a predecir la clase C .
- Los N casos se denotan por $(x_i, c_i), \dots, (x_N, c_N)$
- C_N denotan los m posibles valores de la variable clase C .

El funcionamiento de este algoritmo se basa en calcular las distancias de todos los casos ya clasificados al nuevo caso x , que se pretende clasificar. Una vez seleccionados los K casos ya clasificados más cercanos al nuevo caso, x , a este se le asigna la clase (valor de la variable C) más frecuente de entre los K objetos. En la figura 4.22 se muestra de manera gráfica un ejemplo: tenemos 24 casos ya clasificados en dos posibles valores ($m = 2$), las variables predictoras son X_1 y X_2 , se ha seleccionado $K=3$. De los 3 casos ya clasificados que se encuentran más cercanos al nuevo caso a clasificar, x (representado por un punto), dos de ellos pertenecen a la clase \circ , por tanto el clasificador 3-NN predice la clase \circ para el nuevo caso. Se puede observar que el caso más cercano a x pertenece a la clase $+$. Es decir, que si hubiésemos utilizado un clasificador 1-NN, x se hubiera asignado a $+$.

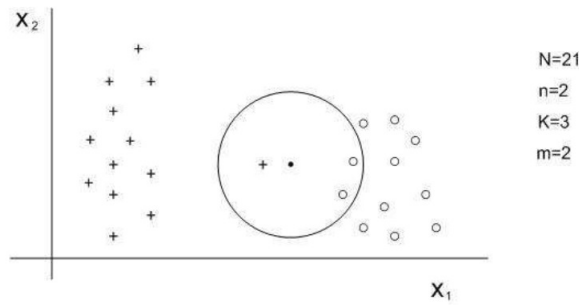


Figura 4.22: Ejemplo de aplicación del algoritmo K -NN.

Para visualizar el comportamiento del entrenador se utiliza la tabla de confusión, la cual se muestra en la figura 4.23. En esta tabla se muestran que las tarjetas pueden ser confundidas unas con otras, de esta manera se puede conocer la eficiencia de nuestro entrenador. La tarjeta Motum y la tarjeta Programador presentan características similares con las demás tarjetas, por lo que pueden ser confundidas. Sin embargo la tarjeta Entrenador y la tarjeta Hugo son tarjetas que presentan características únicas que evitan que se confundan con el resto de las tarjetas.

Una vez obtenida la matriz de características críticas los valores son exportados y almacenados como variables dentro de la plataforma de MATLAB. El siguiente paso consiste en colocar las tarjetas electrónicas sobre la banda transportadora y realizar el reconocimiento de cada una de estas. Para este proceso se utiliza la instrucción `CarFinderLive()`, la cual es un subprograma que activa la cámara web, realiza la adquisición de imágenes, obtiene las características críticas de cada imagen y las compara con la matriz obtenida anteriormente, cuando las características coinciden la imagen es identificada, en caso contrario se descarta, en la figura 4.24 se muestra el diagrama de flujo de este subprograma.

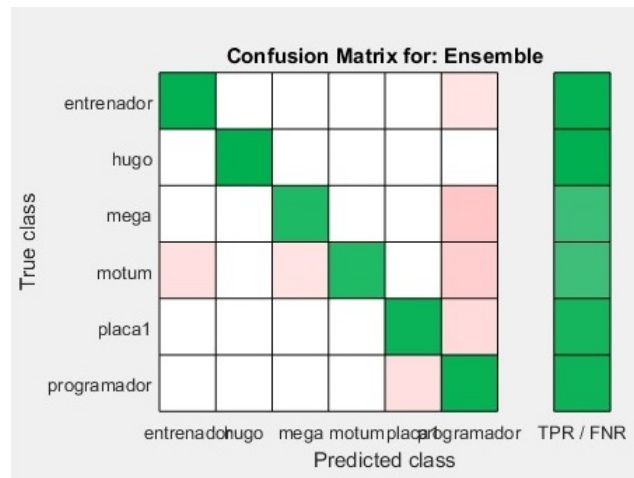


Figura 4.23: Tabla de confusión.

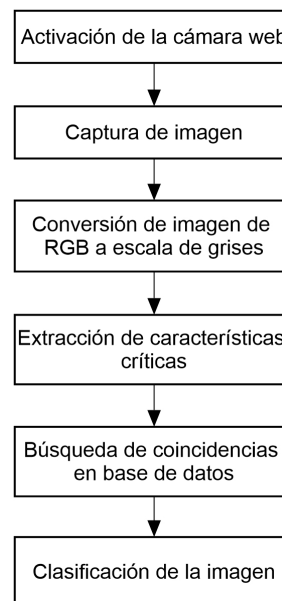


Figura 4.24: Diagrama de flujo del comando CarFinderLive.

En la figura 4.25 se muestra el resultado al identificar la tarjeta Controlador, se observa que el entrenador reconoce la tarjeta sin ningún tipo de problema. En la parte inferior de la figura se observa una gráfica de probabilidad de identificación, para este caso la probabilidad es del 100 %, por lo tanto en la parte superior de la figura se etiqueta la imagen identificada [32].

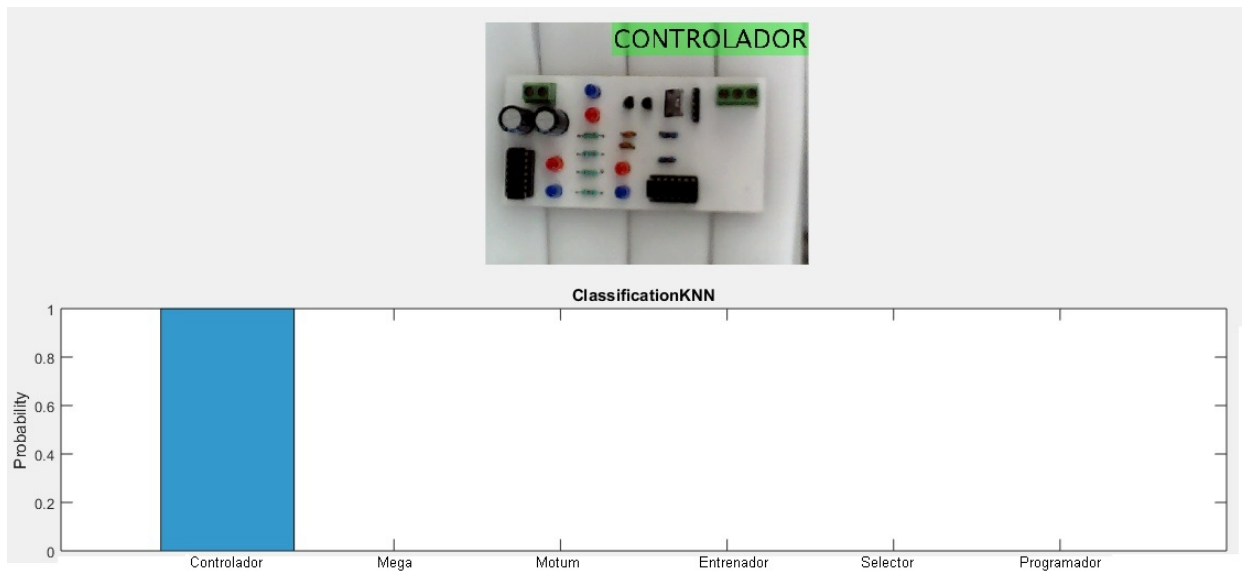


Figura 4.25: Reconocimiento de tarjeta Controlador.

La siguiente prueba a realizar es la identificación de la tarjeta Motum, se observa que se reconoce de manera inmediata con una probabilidad del 100 %, las tarjetas son colocadas sobre la banda transportadora.

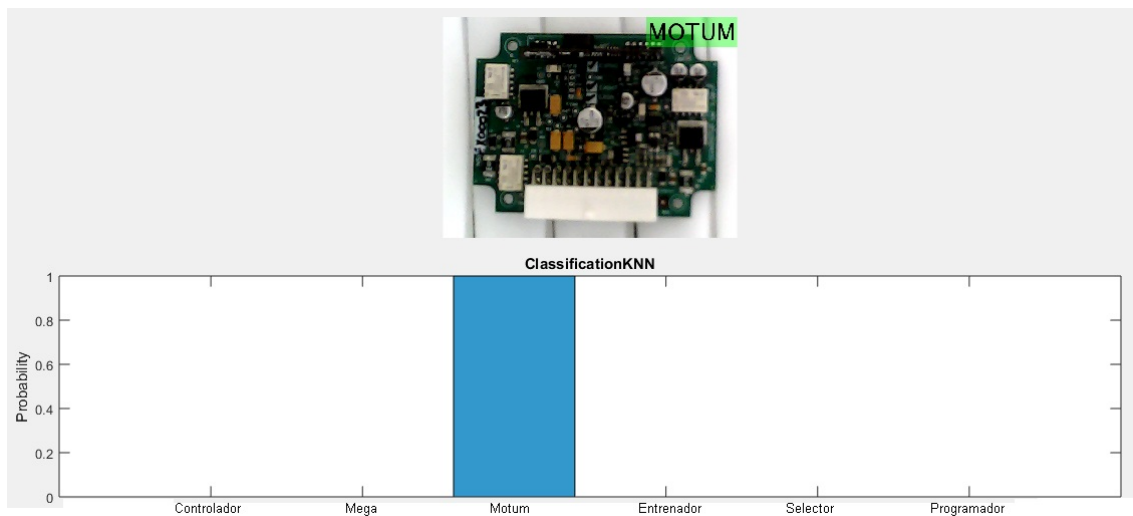


Figura 4.26: Reconocimiento de tarjeta Motum.

Se realiza la identificación de la tarjeta Programador, se observa que se reconoce de manera inmediata con una probabilidad del 100 %.

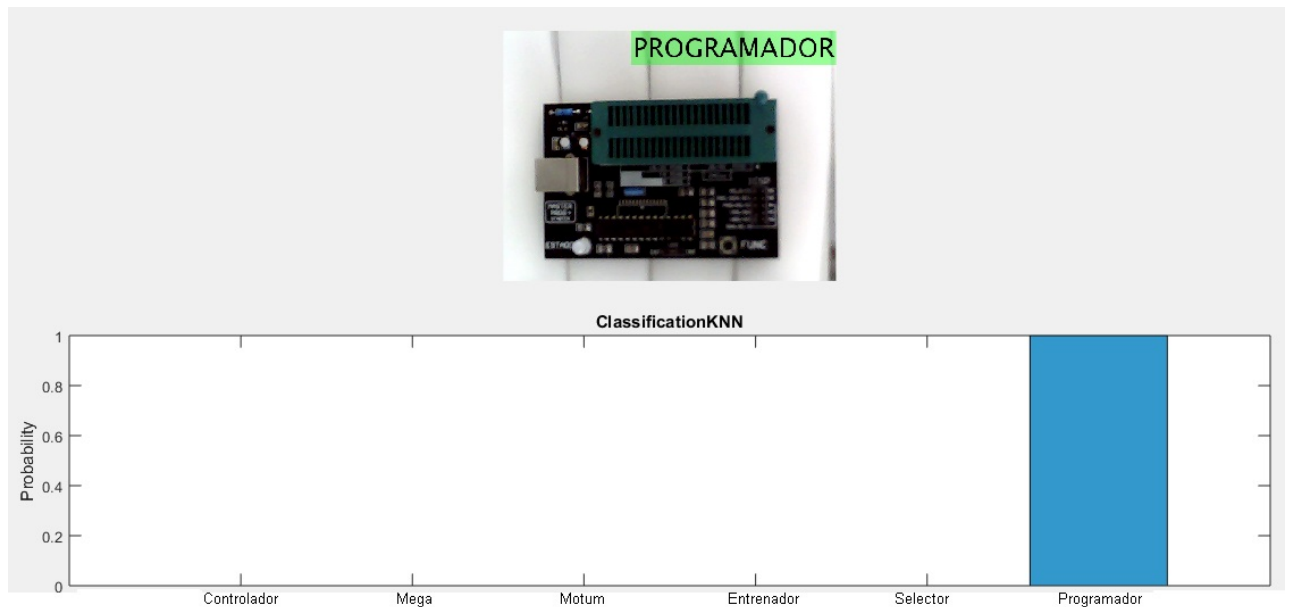


Figura 4.27: Reconocimiento de tarjeta Programador.

De esta manera se comprueba el funcionamiento del algoritmo de Paul-Viola para el reconocimiento de características críticas de un objeto, en nuestro caso el algoritmo es aplicado para la identificación de tarjetas electrónicas. Se puede obtener un mejor reconocimiento de las tarjetas incrementado el número de muestras (número de imágenes) de cada una de las tarjetas electrónicas, así como también la elección de un entrenador diferente y comprobar su índice de desempeño. El algoritmo general de aprendizaje automático se muestra en el apéndice D.1.

4.3. Algoritmo de búsqueda de componentes

Para la implementación del algoritmo de búsqueda de componentes físicos en una tarjeta electrónica es necesario realizar un mapeo de la imagen, con el fin de realizar una conversión de unidades de medida, es decir, interpretar la imagen en cm y no en pixeles. Posteriormente se realiza la búsqueda de cada componente físico dentro de la tarjeta electrónica, este proceso se realiza comparando una base de datos que contiene una lista de componentes y sus posiciones dadas en unidades de cm, de esta manera es posible realizar la búsqueda de cada uno de los componentes físicos de la tarjeta electrónica. La base de datos es obtenida de un archivo denominado GERBER el cual es generado por el software de diseño.

4.3.1. Altium Designer

Altium Designer es un software de diseño electrónico del más alto nivel, enfocado en el desarrollo de tarjetas electrónicas. Esta es una plataforma de diseño que surge en el año de 1985 con el nombre de *Protel* y en el año de 1998 se consolida como uno de los mejores software de diseño, dada la facilidad de incluir elementos en 3D.

Algunas de las características más importantes de Altium Designer son las siguientes:

- Gestión de biblioteca de componentes.
- Diseño de esquemáticos y footprints.
- Edición de librerías.
- Integración con varios distribuidores, permitiendo la búsqueda de componentes y acceso a los datos del fabricante.
- Simulación de circuitos con SPICE.
- Exportación de netlist.
- Exportación de informes y listas de materiales.
- Generación de archivos CAMtastic.
- Vista de PCB en 3D.
- Desarrollo de reglas de diseño.
- Identificación automática de errores de diseño.

Los archivos CAMtastic son generados después de haber culminado el diseño de una PCB y su objetivo principal es la construcción de la tarjeta electrónica. Estos archivos contienen toda la información cualitativa del PCB, entre las cuales se encuentran las vías de conducción, planos de tierra, planos de voltaje, drills, solder mask, solder paste, entre otros. Altium Designer también permite generar lista de materiales y reportes. Por medio de estos archivos se genera la base de datos para realizar la comparación con la imagen adquirida por la cámara web. En la figura 4.28 se muestra la lista de componentes de una PCB, generados en la plataforma de Altium Designer, la cual contiene el nombre del componente, la descripción, su valor numérico, su posición en coordenadas cartesianas y la ubicación del centro del componente. Utilizando estos valores se realiza la búsqueda de componentes.

Com...	De...	Foot...	Cent...	Center-Y(mm)
R 1K	R13	3216[1206]	70.246	31.384
R 100K	R14	3216[1206]	85.232	27.828
R 100K	R15	3216[1206]	83.962	24.018
R 100K	R16	3216[1206]	82.692	27.828
R 4.7K	R17	3216[1206]	23.51	45.1
R 47K	R18	2012[0805]	48.656	31.638
R 47K	R19	2012[0805]	35.448	28.844
R 470	R20	3216[1206]	51.45	37.734
R 10K	R21	3216[1206]	44.592	32.908
R 1K	R22	3216[1206]	35.702	44.846
R 1K	R23	3216[1206]	33.416	44.973
R 5.6K	R24	2012[0805]	39.766	45.992
R 14.7K	R25	3216[1206]	40.528	42.306
R 10	R26	3216[1206]	45.608	44.725
R 100	R27	3216[1206]	41.544	12.08
R 220	R28	3216[1206]	41.544	14.366
R 10K	R29	3216[1206]	33.949	21.173
R 10K	R30	3216[1206]	33.274	14.366
L5973D st	U1	SO-G8_E3	46.624	18.176
TCMT1600	U2	SOP-4	26.558	51.196
PIC18F4685T -	U5	TQFP-44_M	24.018	35.448
MAX232DR	U46	SOIC-16	61.102	35.194
TLV2375IPWR	U51	TSSOP-16	35.423	35.956
10MHZ	Y3	ABMM2	22.494	25.21

Export Options
 File Format: Microsoft Excel Worksheet (*.xls;*.xlsx*.xl)
 Add to Project
 Open Exported

Excel Options
 Template: [Dropdown]
 Relative Path to Template File

Supplier Options
 <none> Production Quantity: 1
 Round up Supplier Order Qty to cheaper price bi
 Use cached pricing data in parameters if offline

Include Not Fitted Components Include Parameters From Database
 Include Parameters From PCB Include Parameters From Vault

Figura 4.28: Reporte de componentes.

4.3.2. Implementación de algoritmo

El algoritmo que permite la identificación de componentes electrónicos sobre una PCB se muestra en la figura 4.29. El proceso general consiste en adquirir una imagen por medio de la cámara web. Posteriormente convertir esta imagen de formato RGB a una imagen binaria. Encontrar las esquinas superior izquierda e inferior derecha, para conocer la dimensión de la tarjeta, dado que las unidades de medida en una imagen es en pixeles se realiza la conversión a centímetros. Se realiza un filtrado sobre la imagen para eliminar el ruido y regiones no deseadas. Se obtienen los centroides de cada componente identificado, así como también su valor en coordenadas cartesianas. Se realiza la lectura del archivo GERBER de la tarjeta, donde se obtienen el número de componentes electrónicos a identificar y sus coordenadas. Se compara la posición del centroide obtenido con respecto al componente del archivo GERBER realizando la conversión de unidades de pixeles a centímetros. Cuando el componente no se detecta se envía un aviso de ausencia, en caso contrario se realiza la búsqueda del siguiente componente electrónico. En la figura 4.30 se presenta de manera visual el proceso de filtrado de la imagen.

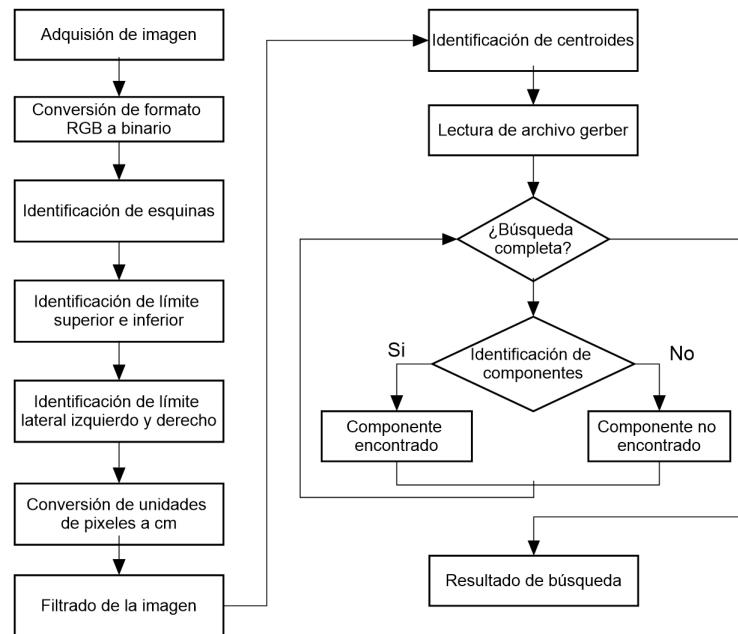


Figura 4.29: Algoritmo de reconocimiento de componentes electrónicos.

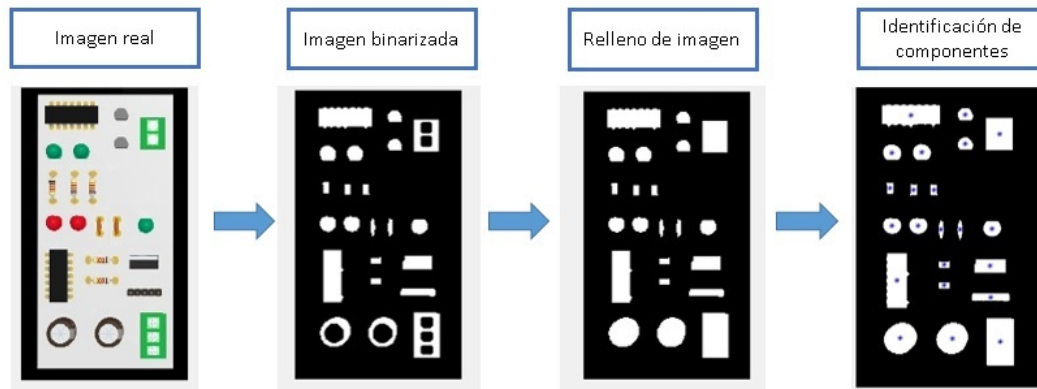


Figura 4.30: Proceso de filtrado para reconocimiento de componentes electrónicos.

Tomando como referencia la tarjeta electrónica que se muestra en la figura 4.31 se realizan las pruebas de reconocimiento de componentes. Esta placa está compuesta por 6 leds de 5mm, 2 transistores, 2 capacitores cerámicos, una tira de pines de 2.54 mm, 2 circuitos electrónicos de 14 pines, 1 transistor TIP-42, una clema de dos terminales y una clema de 3 terminales. Como se puede observar, la tarjeta contiene diferentes tipos de componentes electrónicos de diferentes tamaños, formas y colores.

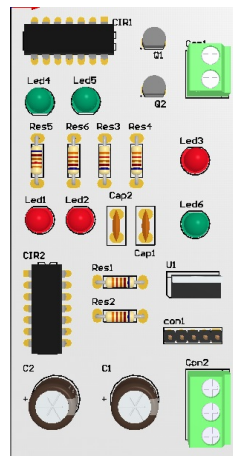


Figura 4.31: PCB de referencia.

Se desarrolló una GUIDE [33] para implementar el algoritmo de búsqueda de componentes electrónicos sobre la tarjeta que se muestra en la figura 4.31. La figura 4.32 corresponde a una tarjeta con todos los componentes electrónicos, por lo tanto la GUIDE indica que la tarjeta esta completa.

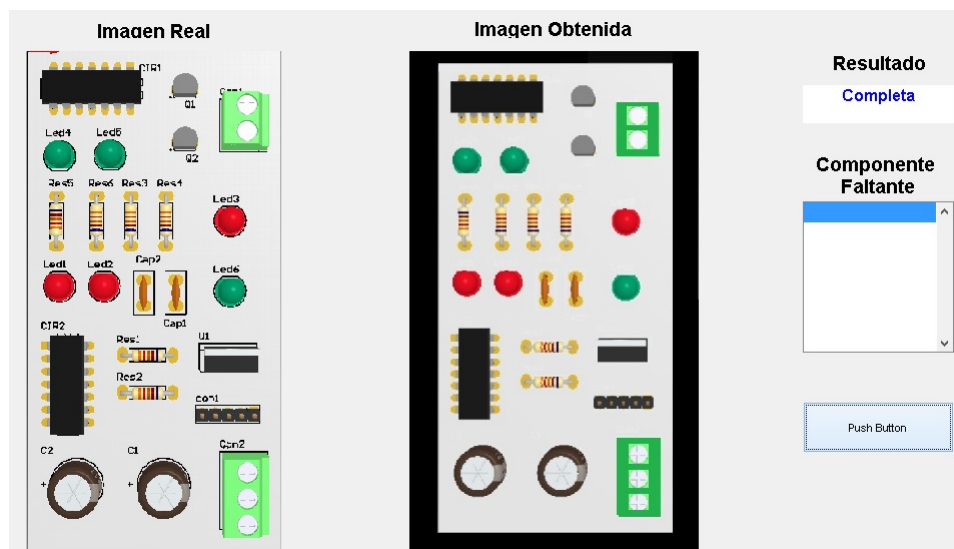


Figura 4.32: PCB completa.

En la figura 4.33 se muestra una tarjeta que no contiene los componentes, el *Led3* y *Res4*, corresponde a un led de color rojo ubicado en la parte superior derecha y corresponde a la última resistencia a un costado del *Led3*. La GUIDE indica que la tarjeta se encuentra incompleta y al mismo tiempo menciona la ausencia de estos dos componentes.

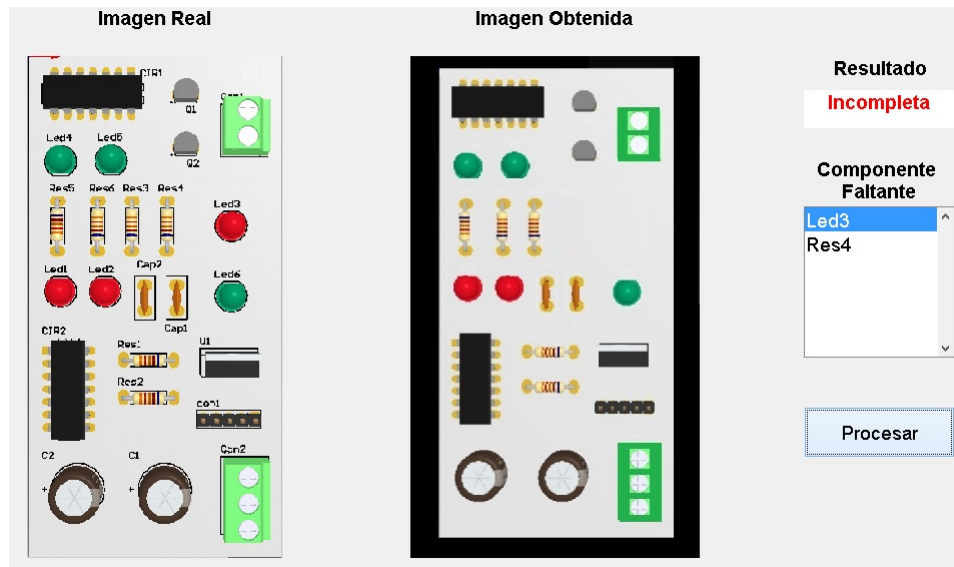


Figura 4.33: PCB incompleta.

Este algoritmo tiene la capacidad de reconocer de manera correcta los componentes electrónicos, algunos de los problemas que se pueden presentar es cuando se generan proyecciones de sombra sobre los componentes cercanos, para solucionar este problema es necesario contar con una fuente externa de luz. Este algoritmo funciona de manera correcta en tarjetas electrónicas que estén formadas por componentes de montaje superficial.

Capítulo 5

Internet de las cosas

En los últimos años ha surgido una área de interés denominada el *Internet de las cosas* (*Internet of things, IoT*) [34]. Este es un concepto que refiere a la interconexión de objetos inteligentes que trabajan en conjuntos conectados a la Internet. Involucra sensores, circuitos, sistemas embebidos, comunicaciones, interfaces inteligentes, gestión de energía, gestión de datos, fusión de datos, gestión de conocimiento, sistemas en tiempo real, procesamiento distribuido, diseño de sistemas y técnicas sofisticadas de software que se relacionan con los grandes datos. El *Internet de las cosas* también incluye desde la Internet industrial, la cual podría cambiar profundamente como los productos son producidos, hasta la salud-electrónica, que ofrece nuevas formas para el cuidado de humanos y el tratamiento de dolencias.

Se implementa la tecnología del *Internet de las cosas* dentro de la automatización de la banda transportadora, el objetivo es mandar información a una página web que indique la cantidad de tarjetas electrónicas identificadas sobre la banda transportadora. Para el desarrollo de este sistema se utiliza la tarjeta Arduino Mega y el módulo ESP8266, este último corresponde a un módulo WiFi (en el apéndice B se muestra la hoja de datos de este dispositivo).

5.1. Internet de las cosas

El *Internet de las cosas* corresponde a una red que interconecta objetos físicos por medio del Internet. Este concepto fue propuesto por Kevin Ashton en el Auto-ID Center del MIT en 1999. El concepto radica en realizar la conexión de diferentes dispositivos que cuenten con un sistema embebido y la capacidad de conectarse a Internet, de esta manera los dispositivos conectados pueden ser consultados o controlados desde la Internet, sin necesidad de estar cerca del dispositivo físico.

Se implementa la metodología del *Internet de las cosas* en la banda transportadora para contar con un sistema automatizado, el cual envíe información a una página de Internet y

pueda ser consultado en una página web, sin la necesidad de tener contacto directamente con el proceso. Para implementar este sistema se utilizan el módulo ESP8266, el cual se muestra en la figura 5.1.

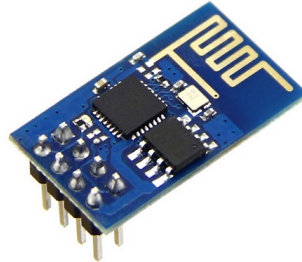


Figura 5.1: Módulo ESP8266.

Es un módulo WiFi que utiliza el estándar de comunicación 802.11, este estándar establece los niveles inferiores del modelo OSI para las conexiones inalámbricas que utilizan ondas electromagnéticas. Este dispositivo se considera un módulo de desarrollo de alto desempeño, considerando las prestaciones que ofrece y el bajo costo. Para establecer la comunicación entre la banda transportadora y el módulo WiFi se utiliza la tarjeta Arduino Mega, la comunicación entre ambos dispositivos es por medio de comandos AT (lenguaje desarrollado por la compañía *Hayes Communications*), utilizando la comunicación serial a una velocidad de 115200 baudios.

Es necesario configurar el módulo ESP8266 para que realice la conexión hacia una red inalámbrica, para esto, se utiliza la tarjeta Arduino Mega; empleando el puerto serie se envían los comandos AT específicos hacia el módulo ESP8266. Cuando no se cuenta con una señal inalámbrica, el modo ofrece la oportunidad de crear un punto de acceso, creando una red local. En el apéndice E se muestran los comandos AT permitidos para este módulo. Una de las principales ventajas de este módulo es el alcance de comunicación, el cual se encuentra alrededor de 600 m al aire libre o 90 m en espacios cerrados, considerando el espacio físico donde se encuentra la celda robótica se puede utilizar este sensor.

Se realizó el diseño de una tarjeta electrónica que permita facilitar la comunicación entre la tarjeta Arduino Mega, el módulo ESP8266, el control de la banda transportadora y el sensor de temperatura. En la figura 5.2 se muestra el esquemático de la tarjeta, donde el conector *C1* es la entrada de voltaje, el conector *C3* y *C4* son los conectores para la tarjeta Arduino Mega, el módulo WiFi Esp8266 está representado por *U1*, el cual se conectada a las terminales Rx y Tx del Arduino Mega, las terminales de control para el driver de la banda transportadora se encuentran en las terminales *M1* y *M2*. Esta tarjeta tiene la capacidad de soportar 4 sensores de temperatura DS19b20, los cuales se conectan en las terminales *T1*, *T2*, *T3* y *T4*. Las terminales *Acel* y *Acel1* se utilizan para conectar sensores adicionales que

utilicen el protocolo de comunicación I^2C .

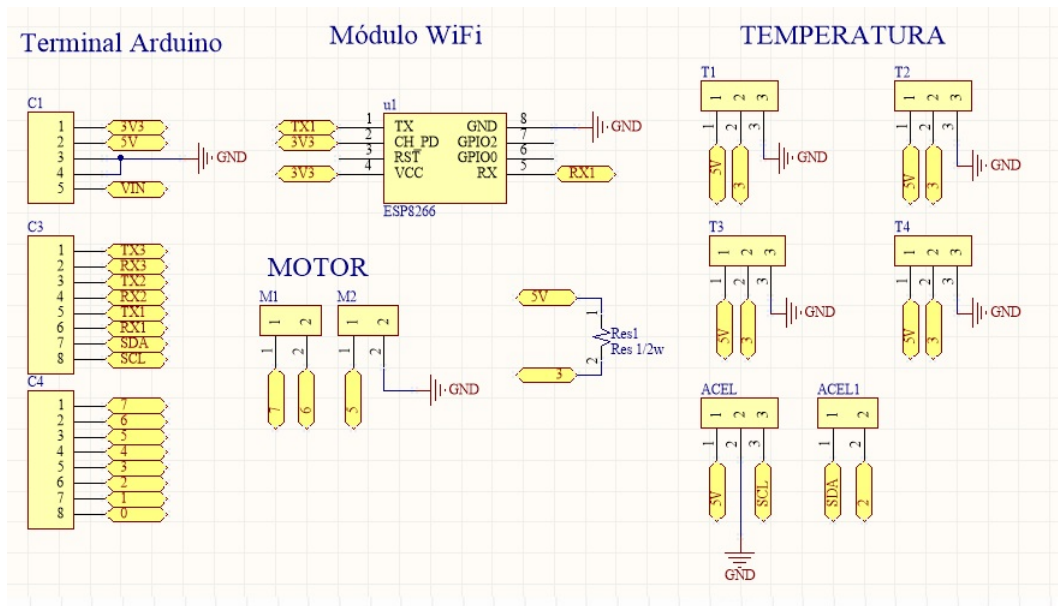


Figura 5.2: Esquemático de tarjeta de comunicación Arduino Mega - ESP8266.

En la figura 5.3 se muestra la tarjeta diseñada con el software Altium Designer, presenta la forma física de un shield para la tarjeta Arduino Mega para su fácil interconexión. Esta tarjeta se encarga de enviar información hacia una página web, enviar las señales de control hacia el controlador de la banda transportadora y la conexión de los sensores de temperatura DS19B20.

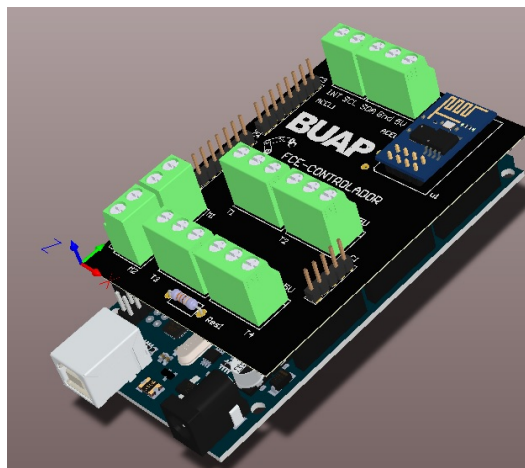
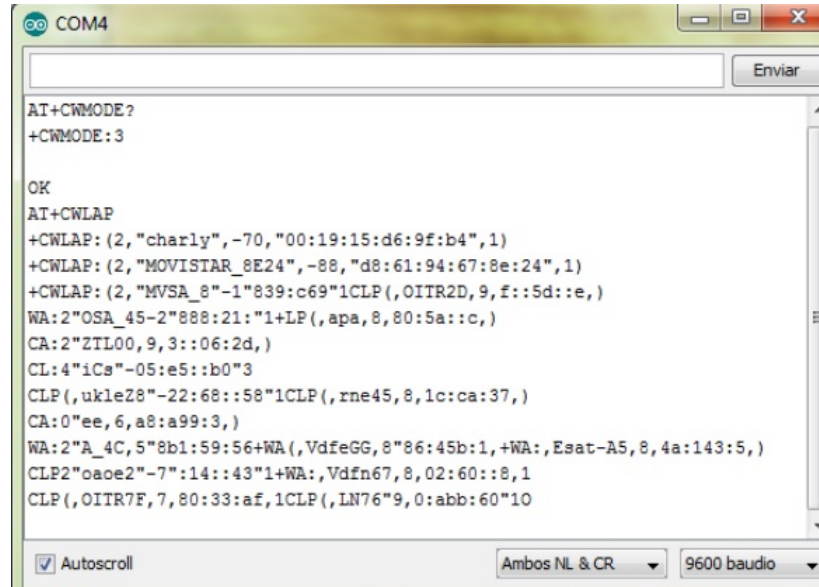


Figura 5.3: Shield Arduino Mega - ESP8266.

Para establecer la comunicación entre el módulo Esp8266 y la página web se utilizan los

comandos AT. El primer paso consiste en identificar las redes inalámbricas que se encuentren disponibles, utilizando el comando `AT+CWLAP` se solicita al módulo Esp8266 que realice la búsqueda de las redes inalámbricas disponibles. En la figura 5.4 se muestra que se han encontradas 3 redes inalámbricas CWLAP, la primera corresponde a la red *charly* la cual una señal de -70 dB, la segunda red corresponde a *Movistar8E24* con una señal de -88 dB y la última red encontrada corresponde a *MVSA8* con una intensidad de -1 dB.



```

COM4
[Enviar]
AT+CWMODE?
+CWMODE:3

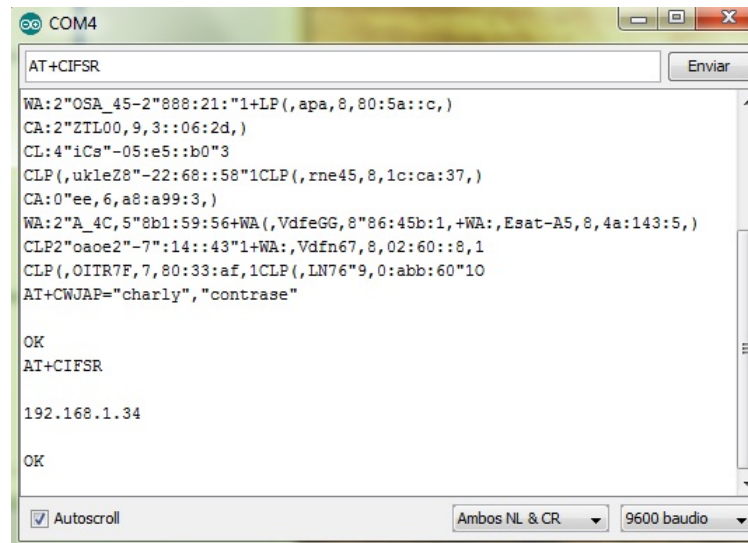
OK
AT+CWLAP
+CWLAP: (2, "charly", -70, "00:19:15:d6:9f:b4", 1)
+CWLAP: (2, "MOVISTAR_8E24", -88, "d8:61:94:67:8e:24", 1)
+CWLAP: (2, "MVSA_8"-1"839:c69"1CLP(, OITR2D, 9, f::5d::e, )
WA:2"OSA_45-2"888:21:"1+LP(, apa, 8, 80:5a::c, )
CA:2"ZTL00, 9, 3::06:2d, )
CL:4"iCs"-05:e5::b0"3
CLP(, ukleZ8"-22:68::58"1CLP(, rne45, 8, 1c:ca:37, )
CA:0"ee, 6, a8:a99:3, )
WA:2"A_4C, 5"8b1:59:56+WA(, VdfeGG, 8"86:45b:1, +WA:, Esat-A5, 8, 4a:143:5, )
CLP2"oaoe2"-7":14::43"1+WA:, Vdfn67, 8, 02:60::8, 1
CLP(, OITR7F, 7, 80:33:af, 1CLP(, LN76"9, 0:abb:60"10
Autoscroll Ambos NL & CR 9600 baudio

```

Figura 5.4: ESP8266 búsqueda de redes inalámbricas.

Una vez que se encuentran las redes inalámbricas se realiza la conexión a una red en específico, es necesario saber si la red es libre o es necesario proporcionar una contraseña. Para acceder a una red inalámbrica se utiliza el comando `AT+CWJAP="Red a conectar", "Contraseña"`, en este caso se realiza la conexión a la red *charly* proporcionando la contraseña proporcionando la instrucción `AT+CWJAP="charly", "contrase"`, cuando el modulo se ha conectado correctamente a la red deseada se obtiene un mensaje de *ok*. Cuando se requiere saber cual es la dirección IP que se nos proporciona al conectarnos a una red se utiliza el comando `AT+CIFSR`, en este caso la dirección IP proporcionada es la 192.168.1.34.

Cuando la configuración se ha establecido es necesario mandar la información a un servidor que se encuentre disponible en Internet. Actualmente existen diferentes servidores gratuitos que permiten establecer conexión con diferentes dispositivos, utilizando la metodología del *Internet de las cosas*, se encuentran *Thingspeak*, *Adafruit* y *Ubidots*. Para el desarrollo de esta tesis se utiliza la plataforma *Thingspeak*, en la figura 5.6 se observa la página web de este servidor, es un servidor de uso gratuito, su plataforma de desarrollo está basado en MATLAB.



```

COM4
AT+CIFSR
WA:2"OSA_45-2"888:21:"1+LP(,apa,8,80:5a::c,)
CA:2"ZTL00,9,3::06:2d,)
CL:4"iC@"-05:e5::b0"3
CLP(,ukleZ8"-22:68::58"1CLP(,rne45,8,1c:ca:37,)
CA:0"ee,6,a8:a99:3,)
WA:2"A_4C,5"8b1:59:56+WA(,VdfeGG,8"86:45b:1,+WA:,Esat-A5,8,4a:143:5,)
CLP2"oaoe2"-7":14::43"1+WA:,Vdfn67,8,02:60::8,1
CLP(,OITR7F,7,80:33:af,1CLP(,LN76"9,0:abb:60"10
AT+CWJAP="charly","contrase"
OK
AT+CIFSR
192.168.1.34
OK
Autoscroll Ambos NL & CR 9600 baudio

```

Figura 5.5: ESP8266 conexión a red inalámbrica.

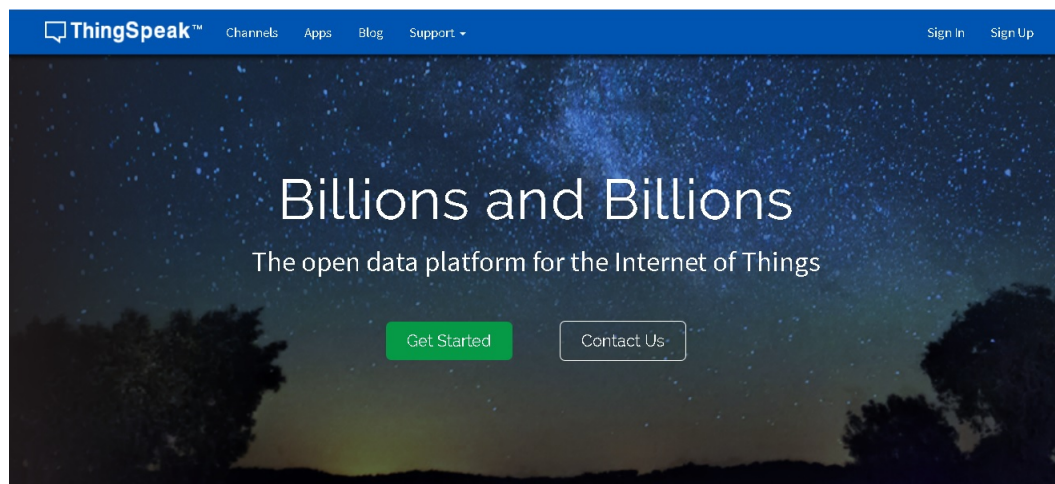


Figura 5.6: Página web de Thingspeak.

Para el uso de esta página se realiza el registro en su servidor. Para realizar la conexión entre el módulo ESP8266 y este servidor es necesario contar con un canal para la transmisión de datos y un identificador por el cual nuestro dispositivo va ser reconocido. El canal de comunicación para el envío de datos es el puerto 80, el identificador que nos proporciona el servidor es el número 169758. Para la escritura y lectura de datos es necesario conocer el valor del APIKeys, para realizar la escritura en el servidor se nos proporciona el APIKey *NB0RGXF37BPJV0J7* y para realizar la lectura corresponde al APIKey *BLG4HKGC2XDVBWID*, tal como se muestra en la figura 5.7.

The screenshot shows the ThingSpeak interface for channel 'BANDA'. At the top, there are navigation links for Channels, Apps, Blog, and Support, along with Account and Sign Out options. The channel details section includes Channel ID: 169758, Author: edgarhugo130590, and Access: Public. Below this, there are tabs for Private View, Public View, Channel Settings, API Keys (selected), and Data Import / Export. The 'API Keys' section is divided into 'Write API Key' and 'Read API Keys'. Under 'Write API Key', there is a text box containing the key 'NB0RGXF37BPJV0J7' and a button labeled 'Generate New Write API Key'. Under 'Read API Keys', there is a text box containing the key 'BLG4HKGC2XDVBWID'.

Figura 5.7: APIKeys de conexión.

Se desarrolla un algoritmo en la tarjeta Arduino Mega para establecer la comunicación entre el módulo ESP8266 y la servidor *ThingSpeak*. En primera instancia se realiza la conexión hacia la red inalámbrica con acceso a Internet, se abre el puerto de comunicación, se establece el enlace con el servidor *ThingSpeak*, para enviar un dato se coloca el APIKey de escritura y el dato a enviar, el servidor nos responde si el dato ha sido enviado ha sido el correcto. El envío de datos se realiza cada 2 minutos. En la figura 5.8 se muestra el diagrama de flujo del algoritmo.

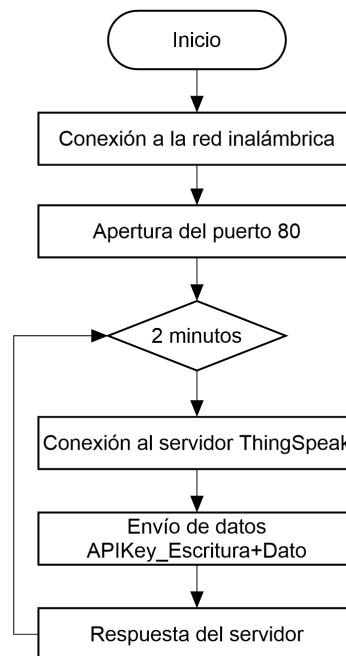


Figura 5.8: Diagrama de flujo del algoritmo de comunicación entre el módulo ESP82266 y el servidor *ThingSpeak*.

Se realiza un ejemplo de comunicación entre el modulo ESP82266 y el servidor de ThingSpeak, donde se envía un valor numérico constante, el envío de datos se realiza cada minuto. Dentro de la servidor se puede configurar la posición del lugar donde el dato es recibido. En la figura 5.9 se observa que el dato corresponde al valor 10 y se indica la posición de la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla, el algoritmo completo se encuentra en el apéndice C.3

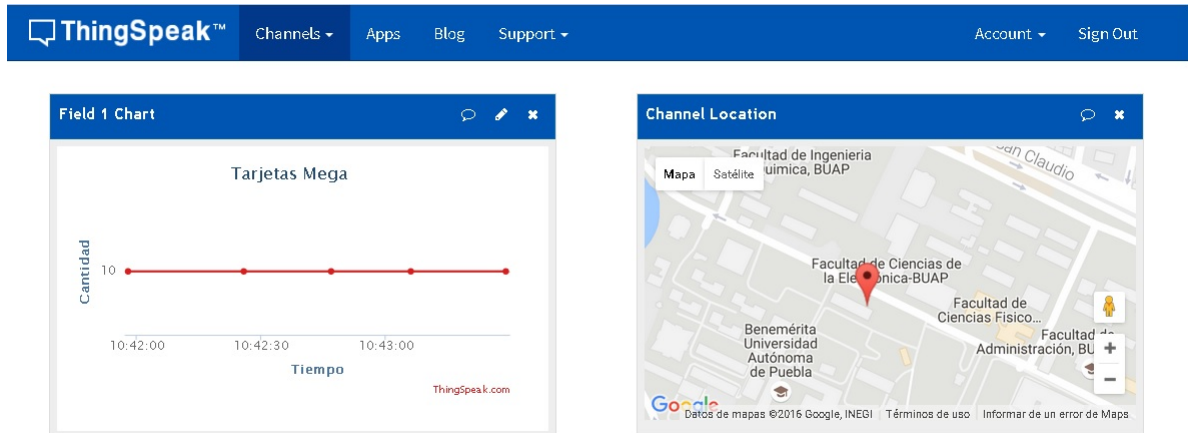


Figura 5.9: Envío de datos a ThingSpeak.

En este capítulo se desarrollo la conexión a Internet, utilizando el servidor ThingSpeak y el módulo Esp8266. Se explico el funcionamiento de este dispositivos y los pasos a desarrollar para el envío de datos.

Capítulo 6

Resultados

En este capítulo se presentan los resultados obtenidos de la integración de la celda robótica con la instrumentación electrónica, los algoritmos de procesamiento digital de imágenes, las tarjetas de control para el ROTRADI de 1 GDL y 3 GDL, los algoritmos de control implementados para el ROTRADI de 1 GDL y 3 GDL, la tarjeta conexión WiFi y la GUIDE desarrollada que permite la interacción de todo el sistema en conjunto. En la primera sección abarca la parte electrónica y las tablas de conexión. En la siguiente sección se presentan las gráficas de comportamiento utilizando diferentes estructuras de control al ROTRADI de 1 GDL y 3 GDL. En la última sección se muestra el uso de la GUIDE junto con los resultados obtenidos. En la figura 6.1 se observa la banda transportadora, el ROTRADI de 3 GDL, el ROTRADI de 1 GDL y la cámara web que integran la celda robótica.

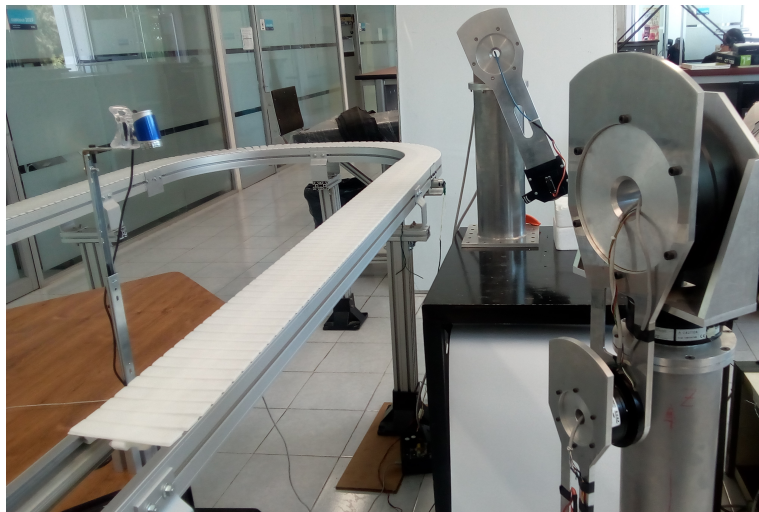


Figura 6.1: Integración de celda robótica.

6.1. Resultados de la instrumentación electrónica

Se desarrollaron las tarjetas electrónicas para realizar la lectura del encoder y proporcionar la señal analógica al ROTRADI. Cada una de estas se comunican con una adquisición de datos Arduino Due. El sistema implementado se muestra en la figura 6.2. Se realizaron 6 tarjetas, 3 tarjetas especializadas para la lectura del encoder y 3 tarjetas para proporcionar la salida analógica. Cada conjunto de tarjetas corresponde a un grado de libertad del ROTRADI (este posee tres grados de libertad). Las tarjetas son colocadas en forma de torreta para reducir el espacio físico. Las tarjetas que se muestran en la parte superior corresponde a la lectura del encoder y las que se muestran en la parte inferior corresponde a las salidas analógicas.

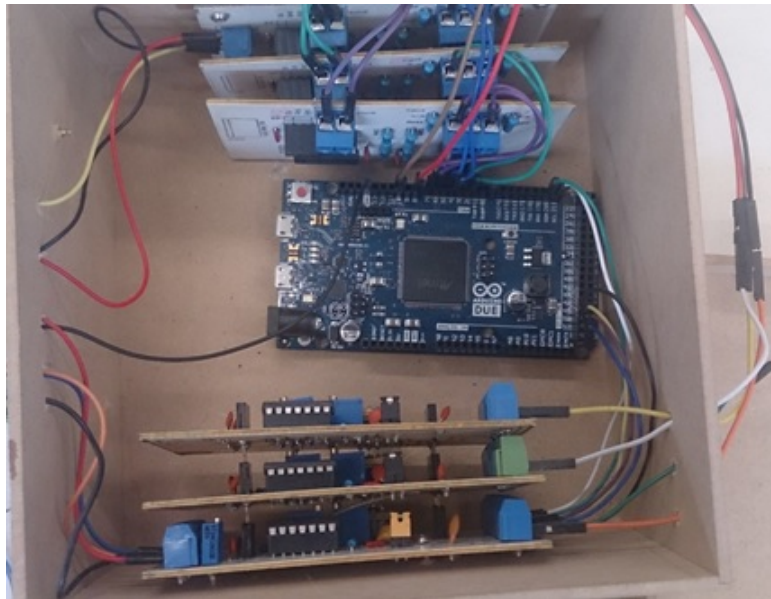


Figura 6.2: Instrumentación electrónica para el control del ROTRADI.

Una de las principales ventajas del desarrollo de esta tarjetas de control, es el uso de los circuitos integrados MCP4725 como convertidores digital a analógicos, utilizando el protocolo de comunicación I^2C , de esta manera se reduce el tamaño de las tarjetas electrónicas y su distribución. En el diseño de la tarjeta electrónica del DAC es necesario no colocar planos de tierras, al mismo tiempo, la unión de la señal de tierra de la parte digital con la parte analógica debe de ser en un solo punto, para evitar que la señal analógica que se introducirá al ROTRADI no presente ruido. La tarjeta que realiza la lectura del encoder debe estar diseñada de tal forma que capte la menor cantidad de ruido, dado que la señal de salida se encuentra en el rango de 0 Vcd a 3.3 Vcd y es suministrada a la tarjeta Arduino Due es mas susceptible al ruido. En la tabla 6.1 se muestran las conexiones entre las tarjetas del encoder y el DAC hacia la tarjeta Arduino Due.

Pin Arduino Due	Encoder	Pin Arduino Due	Analógica
Encoder A1	2	20 (SDA)	SDA
Encoder A2	3	21 (SCL)	SCL
Encoder B1	4	52	A1
Encoder B2	5	50	A2
Encoder C1	6	48	A3
Encoder C2	7	Gnd	Gnd

Tabla 6.1: Tabla de conexión con Arduino Due.

Es necesario que la fuente de alimentación para la tarjetas electrónicas sea estable, es decir, que el voltaje de entrada no varíe y que exista la mínima cantidad de ruido sobre la señal. Considerando que la fuente de voltaje corresponde a una fuente conmutada se colocan reguladores de voltaje lineales. En la figura 6.3 se muestra el esquemático de la fuente lineal.

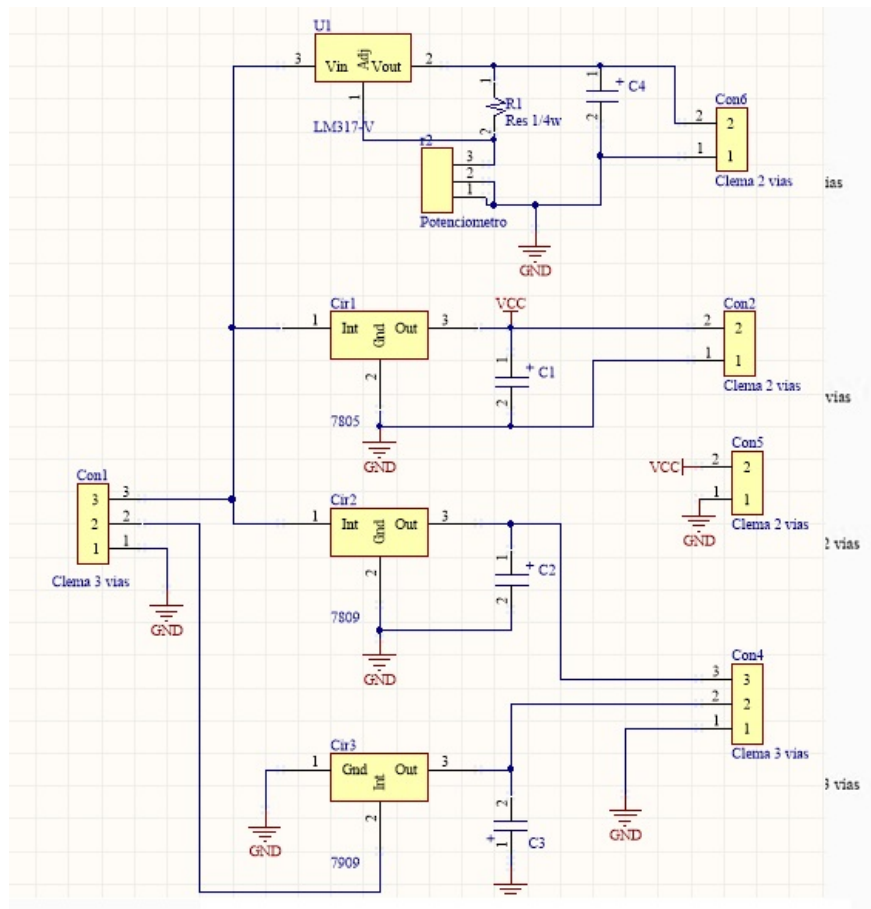


Figura 6.3: Esquemático de fuente de voltaje lineal.

En la figura 6.3 el circuito U1 proporciona un voltaje de salida de 0.7 Vcd a 9 Vcd, que sirve como alimentación para la tarjeta del DAC. El circuito Cir1 proporciona un voltaje de salida de 5 Vcd, el circuito Cir2 proporciona un voltaje de 9 Vcd y el circuito Cir3 otorga un voltaje de -9 Vcd, los cuales son utilizadas para alimentar las tarjetas del Encoder y del DAC para cada ROTRADI. En la figura 6.4 se muestra la tarjeta desarrollada para la fuente de voltaje lineal.



Figura 6.4: Fuente de voltaje lineal.

El primer conector proporciona un voltaje de 3.3 Vcd, el segundo conector 9 Vcd, el tercer conector -9 Vcd y el último conector otorga 5 Vcd. De esta manera se garantiza el correcto funcionamiento del sistema electrónico.

6.2. Estructura de control

Considerando el ROTRADI de un grado de libertad, se evalúan diferentes estructuras de control. La primera corresponde a la estructura de control Proporcional-Derivativo, el cual se encuentra representado por la siguiente ecuación:

$$\boldsymbol{\tau} = K_p(\tilde{\mathbf{q}}) - K_v(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \quad (6.1)$$

donde $\tilde{\mathbf{q}} \in \mathbb{R}^n$ es el vector de error de posicionamiento que se define como la diferencia entre la posición deseada \mathbf{q}_d y la posición actual del robot \mathbf{q} . $K_p \in \mathbb{R}^{n \times n}$ es una matriz definida positiva, $K_v \in \mathbb{R}^{n \times n}$ es la ganancia derivativa la cual es una matriz definida positiva.

La ecuación en lazo cerrado en variables de estado que definen el problema de control de posición es:

$$\frac{d}{dt} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} -\dot{\mathbf{q}} \\ M(\mathbf{q})^{-1} [K_p \tilde{\mathbf{q}} - K_v \dot{\mathbf{q}} - C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}] \end{bmatrix} \quad (6.2)$$

Para llevar a cabo el análisis de la existencia y unicidad del punto de equilibrio $[\tilde{\mathbf{q}}^T \ \dot{\mathbf{q}}^T]^T = [0^T \ 0^T]^T$ se toman en cuenta las siguientes consideraciones:

- La primera componente de la ecuación en lazo cerrado 6.2 $-\dot{\mathbf{q}} = -I\dot{\mathbf{q}} = 0 \iff \dot{\mathbf{q}} = 0$, ya que la matriz identidad $I \in \mathbb{R}^{n \times n}$ es una matriz definida positiva.
- Para la segunda componente de la ecuación 6.2 la matriz de inercia $M(\mathbf{q})$ es una matriz definida positiva, y su matriz inversa existe $M(\mathbf{q})^{-1}$ y es definida positiva.
- La ganancia proporcional K_p es una matriz diagonal definida positiva y la ganancia derivativa K_v es una matriz definida positiva.
- Debido a que se ha demostrado para la primera componente de la ecuación en lazo cerrado que $\dot{\mathbf{q}} = 0$, entonces la matriz de fuerzas centrípetas y de Coriolis satisface $C(\mathbf{q}, \dot{\mathbf{q}}) = 0 \in \mathbb{R}^{n \times n}$.
- Por otro lado $K_p \tilde{\mathbf{q}} = 0 \iff \tilde{\mathbf{q}} = 0$, debido a que la ganancia proporcional es una matriz definida positiva, es decir $K_p > 0$.

Por lo tanto, el punto de equilibrio de la ecuación en lazo cerrado 6.2 existe y es único.

La demostración de estabilidad del punto de equilibrio considerando la estructura general del modelo dinámico de un Robot y la estructura de control anterior en el sentido de Lyapunov queda de la siguiente manera. Se considera la siguiente función candidata de Lyapunov:

$$V(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2} \tilde{\mathbf{q}}^T K_p \tilde{\mathbf{q}} \quad (6.3)$$

El primer término es una función definida positiva debido a que en la energía cinética, la matriz de inercia $M(\mathbf{q})$ es definida positiva. El segundo término es la energía potencial artificial la cual es una función cuadrática del error de posición $\tilde{\mathbf{q}}$ y por diseño la ganancia proporcional K_p es una matriz definida positiva. Por lo tanto, la función $V(\dot{\mathbf{q}}, \tilde{\mathbf{q}})$ es definida positiva.

La derivada temporal de la función de energía queda de la siguiente forma:

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \dot{\mathbf{q}}^T M(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{M}(\mathbf{q}) \dot{\mathbf{q}} + \tilde{\mathbf{q}}^T K_p \dot{\tilde{\mathbf{q}}} \quad (6.4)$$

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \dot{\mathbf{q}}^T M(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{M}(\mathbf{q}) \dot{\mathbf{q}} + \tilde{\mathbf{q}}^T K_p \dot{\mathbf{q}} \quad (6.5)$$

La derivada del error es $\dot{\tilde{\mathbf{q}}} = -\dot{\mathbf{q}}$ debido a que la posición deseada \mathbf{q}_d es una constante. Sustituyendo la aceleración contenida en la segunda componente de la ecuación 6.2 se obtiene:

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \dot{\mathbf{q}}^T K_p \tilde{\mathbf{q}} - \dot{\mathbf{q}}^T K_v \dot{\mathbf{q}} - \dot{\mathbf{q}}^T C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{M}(\mathbf{q}) \dot{\mathbf{q}} + \tilde{\mathbf{q}}^T K_p \mathbf{q} \quad (6.6)$$

El primer elemento se elimina en conjunto con último elemento. Se considera la propiedad de antisimetría, agrupando los términos tercero y cuarto de la ecuación 6.6 se obtiene:

$$\frac{1}{2} \dot{\mathbf{q}}^T \dot{M}(\mathbf{q}) \dot{\mathbf{q}} - \dot{\mathbf{q}}^T C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T [\dot{M}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}})] \dot{\mathbf{q}} = 0 \quad (6.7)$$

Tomando en cuenta los anteriores argumentos, la derivada temporal de la función de energía 6.3 obtiene la siguiente forma:

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = -\dot{\mathbf{q}}^T K_v \dot{\mathbf{q}} \leq 0 \quad (6.8)$$

Por lo tanto se cumple con el teorema de estabilidad de Lyapunov, por lo que queda demostrada la estabilidad del punto de equilibrio de la ecuación en lazo cerrado [2].

Se desea que el péndulo se posicione a 90° , se utiliza una ganancia proporcional de 6.2 y una ganancia derivativa de 0.5, en la figura 6.5 se muestra el comportamiento del ROTRADI. Se observa que el ROTRADI se posiciona a 91° , se obtiene un error en estado estable de 1° . La respuesta del ROTRADI es estable y se estabiliza en 0.6 segundos.

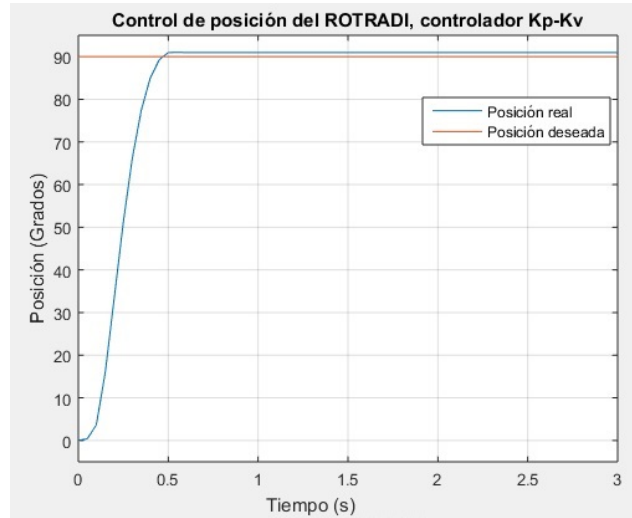


Figura 6.5: Control de posición PD aplicado al ROTRADI de 1 GDL.

En la figura 6.6 se muestra el comportamiento del torque aplicado al péndulo utilizando el controlador proporcional-derivativo. Se observa que se aplica un torque máximo de 9 Nm en 0.05 segundos. Considerando que el ROTRADI se estabiliza en 0.6 segundos, en este instante, se suministra un torque de 0.9 Nm.

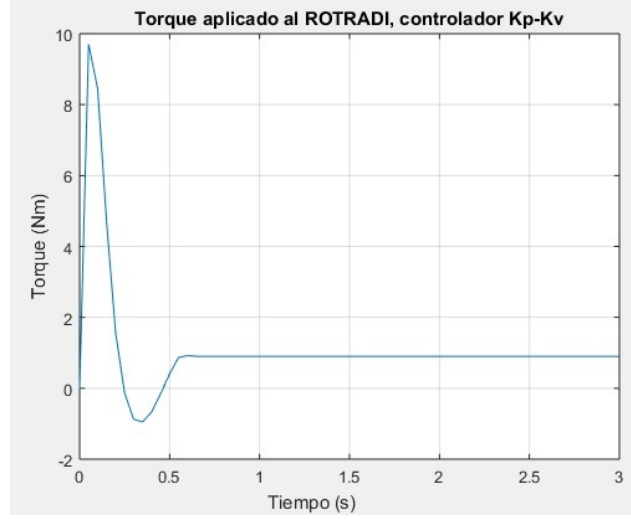


Figura 6.6: Torque aplicado al ROTRADI de 1 GDL, utilizando el controlador PD.

El segundo controlador propuesto corresponde a un controlador acotado. El control con acciones acotadas es un tipo de particular de algoritmos que se relaciona directamente con el problema de saturación de robots. La saturación es un fenómeno no lineal que es muy frecuente en control de robots manipuladores; causa que los servomotores operen en sus límites físicos, conocido como región de saturación. Este problema se presenta cuando los algoritmos de control demandan señales grandes de control, como son los casos del control PD y PID, dando como resultado que se exija un mayor torque del que se pueda suministrar. El controlador acotado propuesto se encuentra representado por la siguiente ecuación:

$$\boldsymbol{\tau} = K_p \tanh(\tilde{\mathbf{q}}) - K_v \tanh(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \quad (6.9)$$

La ecuación en lazo cerrado formada por el modelo dinámico del robot y el control tangente hiperbólica puede ser expresada en variables de estado $[\tilde{\mathbf{q}}^T \ \dot{\mathbf{q}}^T]^T$ de la siguiente forma

$$\frac{d}{dt} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} -\dot{\mathbf{q}} \\ M(\mathbf{q})^{-1} [K_p \tanh(\tilde{\mathbf{q}}) - K_v \tanh(\dot{\mathbf{q}}) - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - B\dot{\mathbf{q}}] \end{bmatrix} \quad (6.10)$$

Para llevar a cabo el análisis de la existencia y unicidad del punto de equilibrio $[\tilde{\mathbf{q}}^T \ \dot{\mathbf{q}}^T]^T = [0^T \ 0^T]^T$ se toman en cuenta las siguientes consideraciones:

- La primera componente de la ecuación en lazo cerrado 6.10 $-\dot{\mathbf{q}} = -I\dot{\mathbf{q}} = 0 \iff \dot{\mathbf{q}} = 0$, ya que la matriz identidad $I \in \mathbb{R}^{n \times n}$ es una matriz definida positiva.
- Para la segunda componente de la ecuación 6.10 la matriz de inercia $M(\mathbf{q})$ es una matriz definida positiva, y su matriz inversa existe $M(\mathbf{q})^{-1}$ y es definida positiva.

- La ganancia proporcional K_p es una matriz diagonal definida positiva y la ganancia derivativa K_v es una matriz definida positiva.
- La matriz de fuerzas centrípetas y de Coriolis $C(\mathbf{q}, \dot{\mathbf{q}}) = 0 \in \mathbb{R}^{n \times n}$ si $\dot{\mathbf{q}} = 0 \in \mathbb{R}^n$.
- Puesto que $\dot{\mathbf{q}} = 0 \in \mathbb{R}^n$, entonces la función disipativa $f_v(K_v, \dot{\mathbf{q}}) = 0 \iff \dot{\mathbf{q}} = 0$.
- Por otro lado, $K_p \tanh(\tilde{\mathbf{q}}) = 0 \iff \tilde{\mathbf{q}} = 0$.

Por lo tanto, el punto de equilibrio de la ecuación en lazo cerrado 6.10 existe y es único.

La demostración de estabilidad del punto de equilibrio considerando la estructura general del modelo dinámico de un Robot y la estructura de control anterior en el sentido de Lyapunov queda de la siguiente manera. Se considera la siguiente función candidata de Lyapunov:

$$V(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} + \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))} K_p \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))} \quad (6.11)$$

El primer término es una función definida positiva debido a que en la energía cinética, la matriz de inercia $M(\mathbf{q})$ es definida positiva. El segundo término es la energía potencial artificial, también es una función definida positiva con respecto al vector de error de posición $\tilde{\mathbf{q}}$, puesto que $K_p > 0$. Note que el $\cosh(0) = 1$ y el $\ln(1) = 0$. Por lo que la función 6.11 evaluada en $\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = V(0, 0) = 0$ y $\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) > 0$ si $\tilde{\mathbf{q}} \neq 0$ y/o $\dot{\mathbf{q}} \neq 0$.

La derivada temporal de la función de energía queda de la siguiente forma:

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \dot{\mathbf{q}}^T M(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{M}(\mathbf{q}) \dot{\mathbf{q}} - \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))} K_p \left[\frac{\sinh(\tilde{\mathbf{q}}) \dot{\mathbf{q}}}{\cosh(\tilde{\mathbf{q}}) \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))}} \right] \quad (6.12)$$

La derivada del error es $\dot{\tilde{\mathbf{q}}} = -\dot{\mathbf{q}}$ debido a que la posición deseada \mathbf{q}_d es una constante. Sustituyendo la aceleración contenida en la segunda componente de la ecuación 6.10 se obtiene:

$$\begin{aligned} \dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = & \dot{\mathbf{q}}^T K_p \tanh(\tilde{\mathbf{q}}) - \dot{\mathbf{q}}^T K_v \tanh(\dot{\mathbf{q}}) - \dot{\mathbf{q}}^T C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{M}(\mathbf{q}) \dot{\mathbf{q}} \\ & - \dot{\mathbf{q}}^T B \dot{\mathbf{q}} - \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))} K_p \left[\frac{\sinh(\tilde{\mathbf{q}}) \dot{\mathbf{q}}}{\cosh(\tilde{\mathbf{q}}) \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))}} \right] \end{aligned} \quad (6.13)$$

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \dot{\mathbf{q}}^T K_p \tanh(\tilde{\mathbf{q}}) - \dot{\mathbf{q}}^T K_v \tanh(\dot{\mathbf{q}}) - \dot{\mathbf{q}}^T B \dot{\mathbf{q}} - \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))} K_p \left[\frac{\sinh(\tilde{\mathbf{q}}) \dot{\mathbf{q}}}{\cosh(\tilde{\mathbf{q}}) \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))}} \right] \quad (6.14)$$

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \dot{\mathbf{q}}^T K_p \tanh(\tilde{\mathbf{q}}) - \dot{\mathbf{q}}^T K_v \tanh(\dot{\mathbf{q}}) - \dot{\mathbf{q}}^T B \dot{\mathbf{q}} - \sqrt{\ln(\cosh(\tilde{\mathbf{q}}))} K_p \left[\frac{\tanh(\tilde{\mathbf{q}}) \dot{\mathbf{q}}}{\sqrt{\ln(\cosh(\tilde{\mathbf{q}}))}} \right] \quad (6.15)$$

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = \dot{\mathbf{q}}^T K_p \tanh(\tilde{\mathbf{q}}) - \dot{\mathbf{q}}^T K_v \tanh(\dot{\mathbf{q}}) - \dot{\mathbf{q}}^T B \dot{\mathbf{q}} - \tanh(\tilde{\mathbf{q}}) K_p \dot{\mathbf{q}} \quad (6.16)$$

$$\dot{V}(\dot{\mathbf{q}}, \tilde{\mathbf{q}}) = -\dot{\mathbf{q}}^T K_v \tanh(\dot{\mathbf{q}}) - \dot{\mathbf{q}}^T B \dot{\mathbf{q}} \leq 0 \quad (6.17)$$

Observese que $\dot{\mathbf{q}}^T K_v \tanh(\dot{\mathbf{q}}) > 0$ es una función disipativa, por lo que se demuestra estabilidad global del punto de equilibrio. La estabilidad asintótica se obtiene aplicando el teorema de Barbashin Krasovskii LaSalle [2].

Se desea que el ROTRADI se posicione a 90° . El valor de la ganancia proporcional utilizada es de 6 y la ganancia derivativa de 1, en la figura 6.7 se muestra el comportamiento del ROTRADI. Se observa que el ROTRADI se posiciona a 89° , se obtiene un error en estado estable de 1° . La respuesta del ROTRADI presenta un sobre impulso de 94° en 0.65 segundos. El sistema se estabiliza en un tiempo de 0.95 segundos.

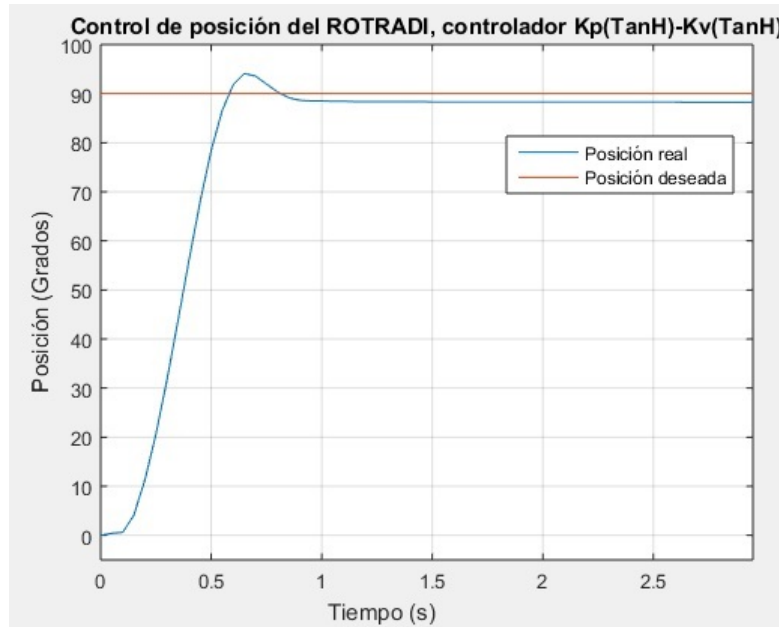


Figura 6.7: Control de posición \tanh aplicado al ROTRADI de 1 GDL.

En la figura 6.8 se muestra el comportamiento del torque aplicado al péndulo utilizando el controlador tangente hiperbólico. El torque máximo aplicado corresponde a 5.5 Nm en 0.05 segundos. Considerando que el ROTRADI se estabiliza en 0.95 segundos, en este instante, se suministra un torque de 0.2 Nm.

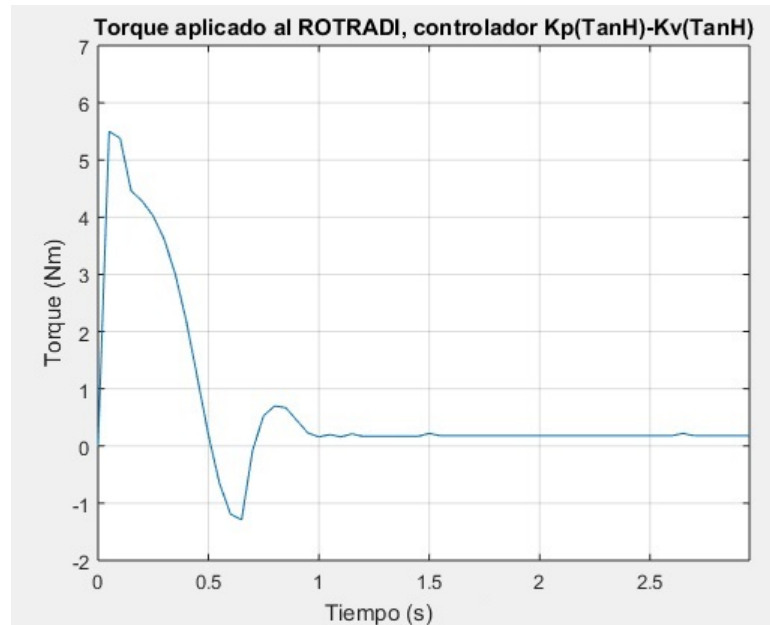


Figura 6.8: Torque aplicado al ROTRADI de 1 GDL, utilizando el controlador \tanh .

Se considera el ROTRADI de tres grados de libertad y se evalúa el comportamiento de dos diferentes estructuras de control. La primera estructura de control corresponde al controlador proporcional-derivativo. Los valores de las ganancias se muestran en la tabla 6.2.

Descripción	Valor
K_p Base	8
K_v Base	2.8
K_p Hombro	35
K_v Hombro	8
K_p Codo	2.1
K_v Codo	0.5

Tabla 6.2: Ganancias utilizadas en el ROTRADI de tres grados de libertad, utilizando el controlador Proporcional-Derivativo.

Se implementa el controlador Proporcional-Derivativo en la articulación del codo del ROTRADI. La posición deseada es de 90° , se observa en la figura 6.9 que el eslabón no llega a la posición deseada, presenta un error en estado estable de 1° , la respuesta del sistema es suave y se estabiliza en 0.75 segundos.

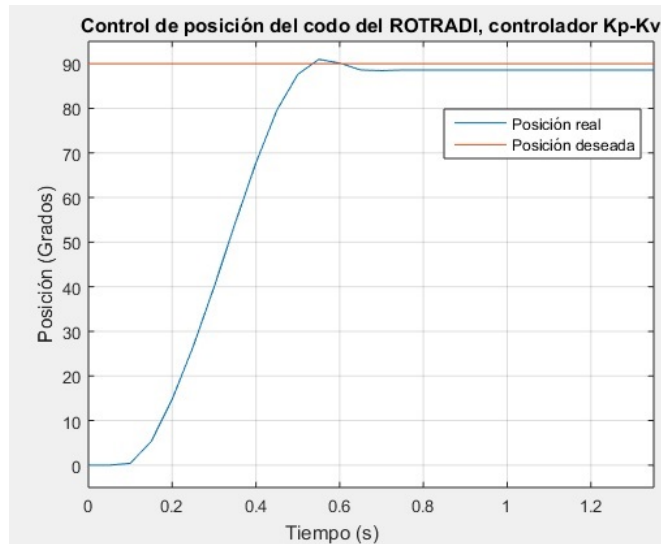


Figura 6.9: Control de posición PD aplicado al codo del ROTRADI.

En la figura 6.10 se muestra el torque aplicado a la articulación del codo del ROTRADI, el torque máximo aplicado es de 1.84 Nm en 0.1 segundos. Cuando el eslabón se estabiliza se requiere de un torque de 0.5 Nm.

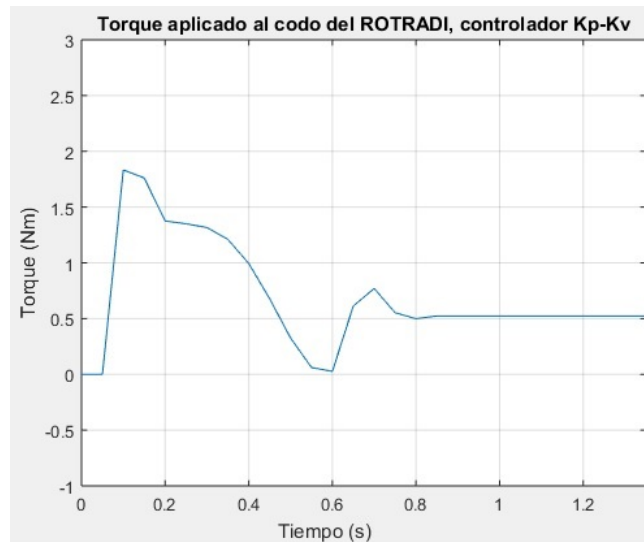


Figura 6.10: Controlador PD, torque aplicado al codo del ROTRADI.

Utilizando el controlador Proporcional-Derivativo en la articulación del hombro del ROTRADI. Con una posición deseada de 90° , se observa en la figura 6.11 que el eslabón no llega a la posición deseada, presenta un error en estado estable de 2° , la respuesta del sistema es suave y se estabiliza en 0.8 segundos.

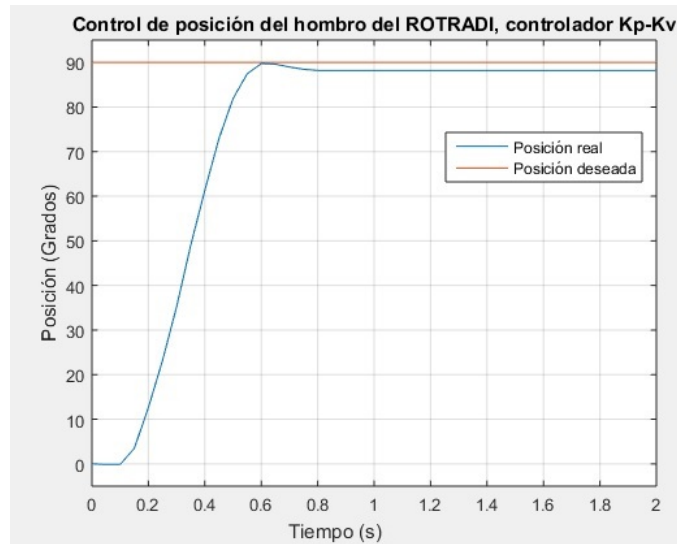


Figura 6.11: Control de posición PD aplicado al hombro del ROTRADI.

En la figura 6.12 se muestra el torque aplicado a la articulación del hombro del ROTRADI, el torque máximo aplicado es de 32 Nm en 0.1 segundos. Cuando el eslabón se encuentra en el estado estacionario se requiere de un torque de 10.5 Nm.

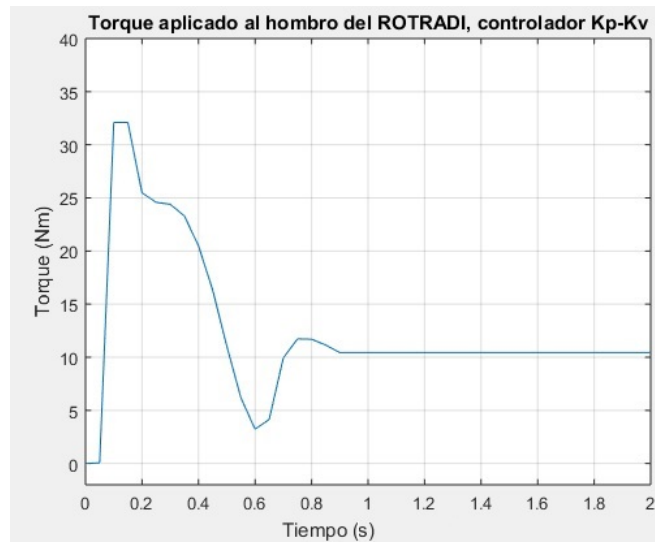


Figura 6.12: Controlador PD, torque aplicado al hombro del ROTRADI.

Utilizando el controlador Proporcional-Derivativo en la articulación de la base del ROTRADI. Con una posición deseada de 90° , se observa en la figura 6.13 que el eslabón presenta un error en estado estable de 0.15° , la respuesta del sistema es suave y se estabiliza en 0.95 segundos.

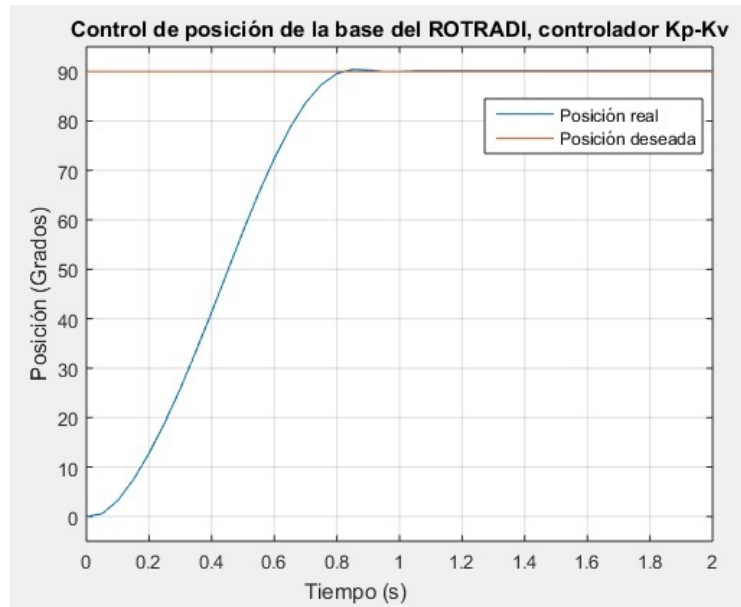


Figura 6.13: Control de posición PD aplicado a la base del ROTRADI.

En la figura 6.14 se muestra el torque aplicado a la articulación de la base del ROTRADI, el torque máximo es de 7.3 Nm en 0.05 segundos. Cuando el eslabón se encuentra en el estado estacionario se requiere de un torque de 0.1 Nm.

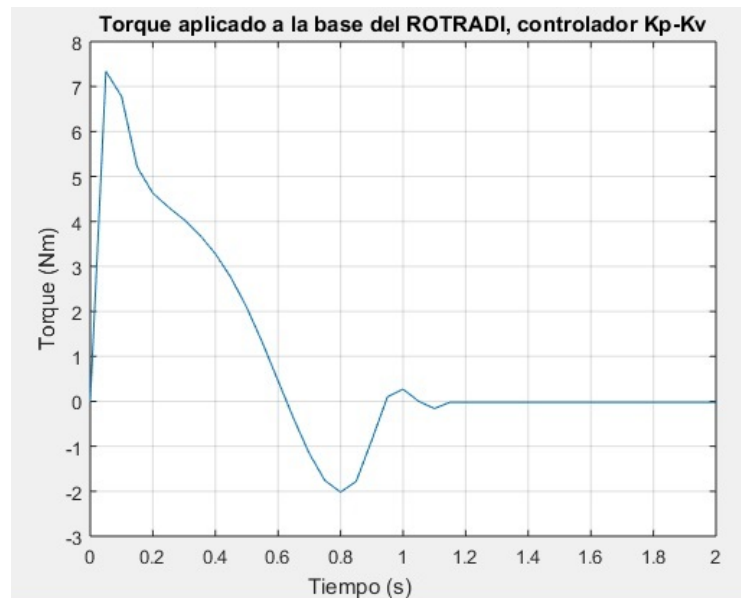


Figura 6.14: Controlador PD, torque aplicado a la base del ROTRADI.

6.2.1. Implementación de Guide

Se desarrolla una interfaz en MATLAB utilizando la herramienta Guide. El objetivo es establecer un sistema de control para la identificación de tarjetas electrónicas que son colocadas en la banda transportadora y con el ROTRADI son retirados en caso de que no cumplan con las especificaciones dadas por el usuario. Para implementar este sistema se establece la comunicación con el controlador de la banda transportadora y el controlador del ROTRADI. En la figura 6.15 se muestra la interfaz desarrollada.

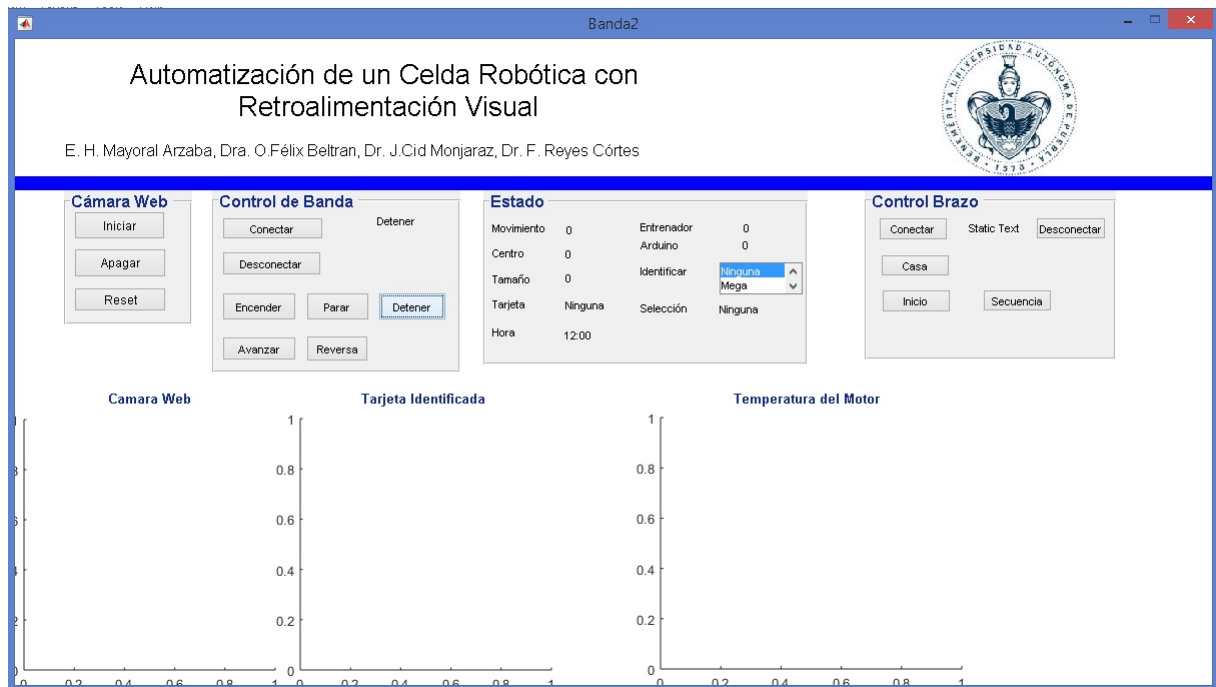


Figura 6.15: Guide implementada.

El panel de control con el nombre *Cámara Web*, activa la cámara y se ejecuta el algoritmo de detección de movimiento, este contiene el filtro digital para descartar regiones pequeñas, cuando se identifica una tarjeta electrónica se envía un comando al controlador de la banda transportadora para detenerla. Posteriormente se activa el algoritmo de aprendizaje automático para identificar la tarjeta electrónica, una vez que es identificada se compara con la tarjeta seleccionada por el usuario, en caso de que sea la correcta se revisa si contiene todos sus componentes electrónicos, cuando la tarjeta no presenta ningún problema el proceso continúa, en caso contrario, se manda un comando al controlador del ROTRADI para retirar la tarjeta de la banda transportadora. La información que se genera se envía a la página web de ThingSpeak, en la cual se visualiza la cantidad y el tipo de tarjetas electrónicas que se encuentran en la banda transportadora. Por último se manda un comando al controlador de la banda transportadora para que esta continúe en movimiento.

El panel de control con el nombre de *Control de Banda* contiene los controles para activar y desactivar la banda de manera manual, al mismo tiempo se controla el sentido de giro del motor, ya sea para desplazar la banda hacia adelante o viceversa.

El panel de control con el nombre de *Estado* contiene la información de la banda transportadora, muestra las dimensiones físicas del objeto identificado, el centro del objeto, el tipo de tarjeta identificada, la cantidad de tarjetas identificadas e incluye la fecha del proceso.

El panel de control con el nombre *Control Brazo* posee los mandos para activar y desactivar ROTRADI, colocar en su posición de casa y visualizar la rutina que se encuentra cargada dentro del controlador.

La Guide muestra tres gráficas diferentes, en la primera se visualiza la imagen de la cámara web en tiempo real, la segunda gráfica muestra cuando una tarjeta es identificada y la última gráfica se muestra la temperatura del motor de la banda transportadora. Cuando la temperatura es mayor a 40°C se desactiva la banda transportadora, para evitar un daño en el motor.

Para el uso correcto de la interfaz se realiza el siguiente procedimiento.

- El controlador de la banda transportadora debe de estar energizado.
- El botón de paro de emergencia del controlador debe del ROTRADI debe de estar activado.
- Energizar el sistema del ROTRADI.
- Realizar la conexión USB del controlador de la banda transportadora y revisar el número del puerto COM que es asignado.
- Realizar la conexión USB del controlador del ROTRADI y revisar el número del puerto COM que es asignado.
- Activar la GUIDE en MATLAB revisando en el código fuente que los puertos son los correctos.
- En el panel *Control Brazo* se oprime el botón de conectar.
- Desactivar el botón de paro de emergencia del controlador del ROTRADI, el brazo se posiciona junto a la banda transportadora.
- En el panel *Control de Banda* seleccionar el botón de conectar, esperar 5 segundos y seleccionar el botón de encender. La banda transportadora comienza a desplazarse.

- En el panel *Estado* seleccionar el tipo de tarjeta que se desea reconocer.
- En el panel *Cámara Web* seleccionar el botón iniciar para activar la cámara web e inicializar el proceso de supervisión de las tarjetas electrónicas.

Para detener el proceso de la celda robótica se aplica el siguiente procedimiento:

- Del panel *Cámara Web* seleccionar el botón apagar.
- En el panel *Control Brazo* seleccionar el botón desconectar.
- En el panel *Control de Banda* seleccionar el botón apagar, esperar 5 segundos, hasta que la banda transportadora se encuentre detenida.
- En el panel *Control de Banda* seleccionar el botón desconectar.
- Retirar todas las fuentes de alimentación de la celda robótica.

6.2.2. Resultados de ejecución

Considerando el procedimiento anteriormente descrito se realiza la activación de la celda robótica. En la figura 6.16 se muestra que el ROTRADI de 3 GDL se posiciona a un lado de la banda transportadora.



Figura 6.16: Posición de trabajo del ROTRADI de 3 GDL.

El siguiente paso es colocar una tarjeta electrónica sobre la banda transportadora utilizando el ROTRADI de 1 GDL, se selecciona el botón *Inicio P* en el cuadro de control *Control Brazo* de la interfaz, tal como se muestra en la figura 6.17.

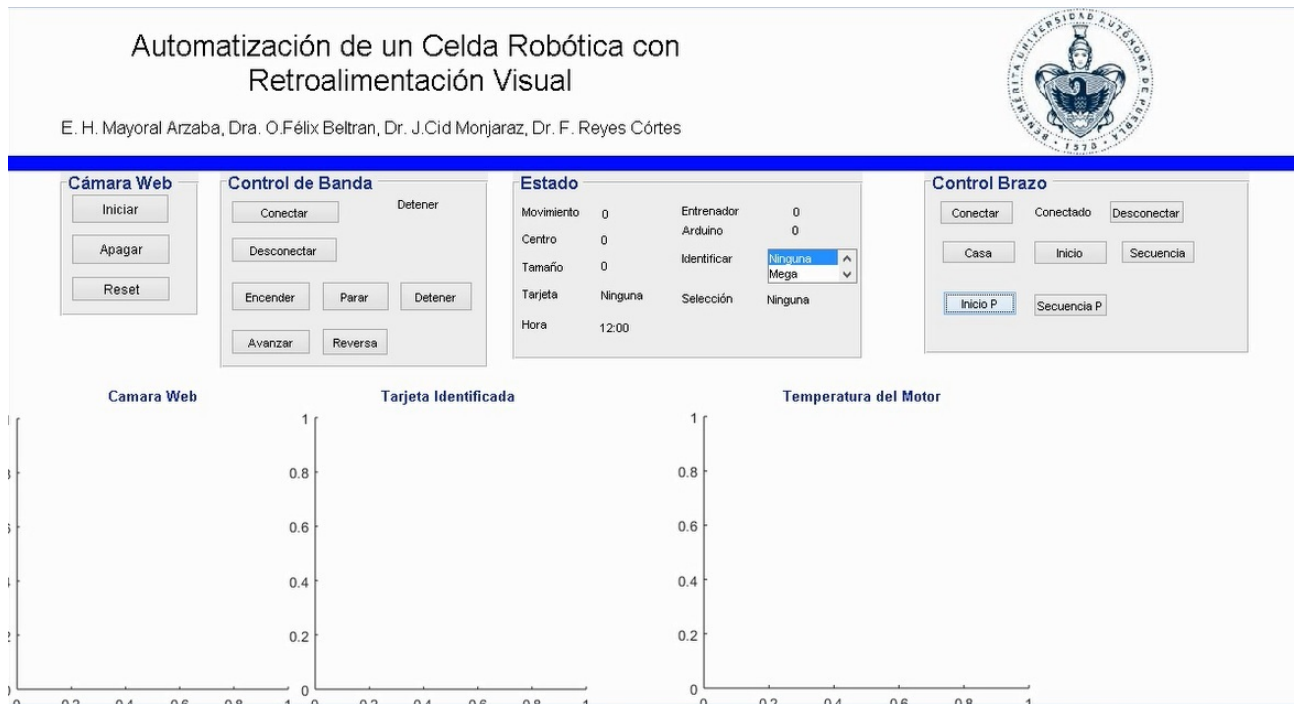


Figura 6.17: Activación del ROTRADI de 1 GDL.

Cuando se activa este comando se le indica al ROTRADI de 1 GDL que sujete la tarjeta electrónica de prueba y la coloque sobre la banda, el proceso de sujeción se realiza por un medio de un electroimán. En la figura 6.18 se observa el ROTRADI sujetando el tarjeta electrónica y en la figura 6.19 el ROTRADI coloca la tarjeta sobre la banda transportadora.

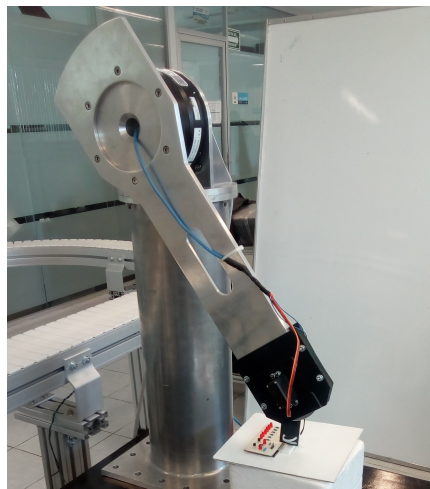


Figura 6.18: ROTRADI de 1 GDL sujetando la tarjeta electrónica de prueba.

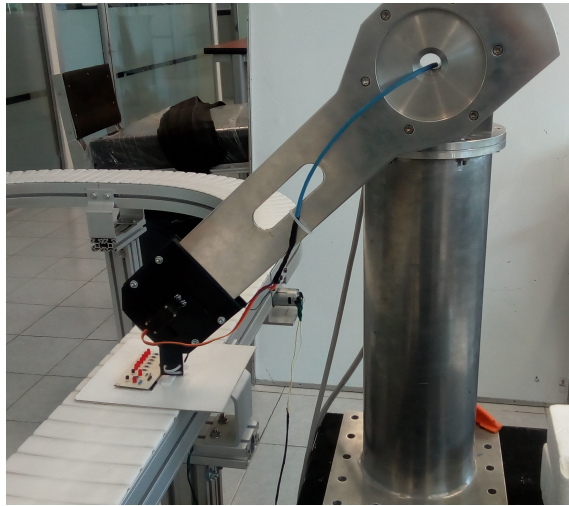


Figura 6.19: ROTRADI de 1 GDL colocando tarjeta de prueba sobre banda transportadora.

Cuando la tarjeta se coloca sobre la banda transportadora, por medio del botón *Encender* del cuadro de control *Control Banda*, se acciona la banda transportadora y esta comienza a desplazarse. Posteriormente se activa la cámara web por medio del botón *Iniciar* en el cuadro de control *Cámara Web*. Cuando la cámara web se enciende se visualiza la banda transportadora, tal como se muestra en la figura 6.20.

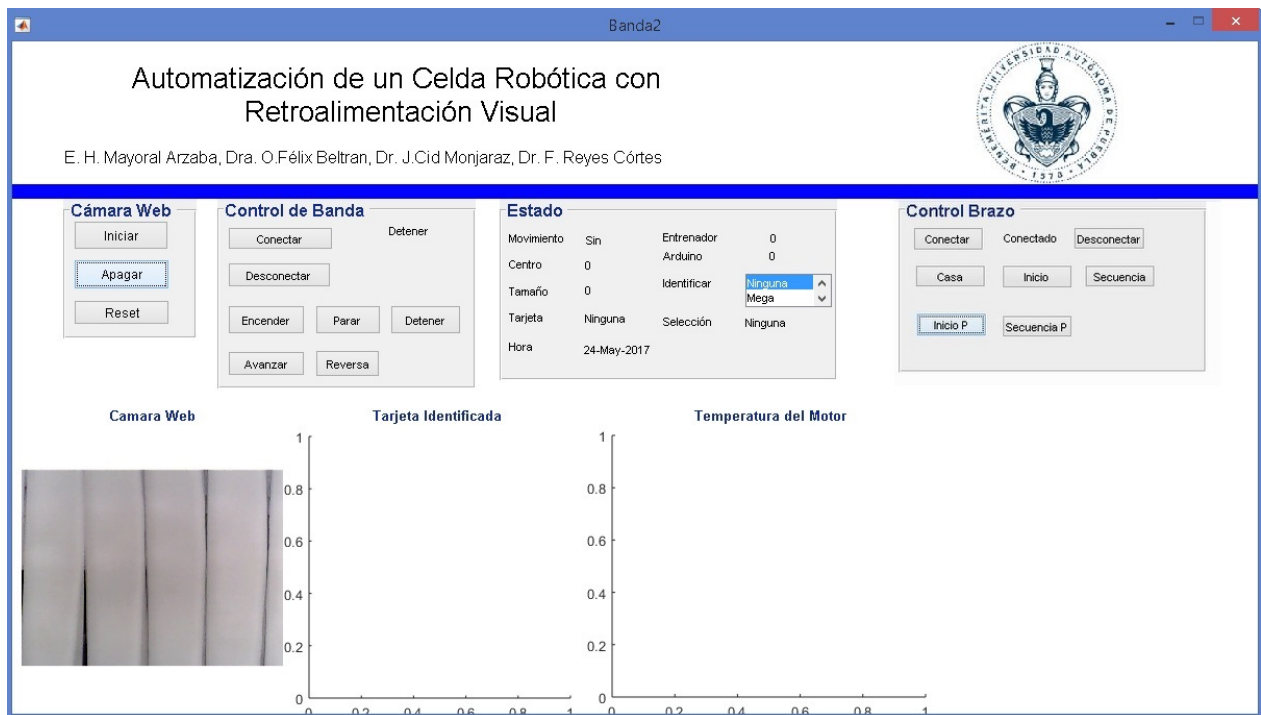


Figura 6.20: Reconocimiento de banda transportadora.

En el panel de control *Estado* se selecciona la tarjeta que se desea identificar, en este caso se desea reconocer la tarjeta Entrenador. Se disponen de dos tarjetas similares, tal como se observa en la figura 6.21. La tarjeta de la parte superior es una tarjeta incompleta, es decir, no contiene todos sus componentes físicos. La tarjeta que se muestra en la parte inferior es una tarjeta completa.

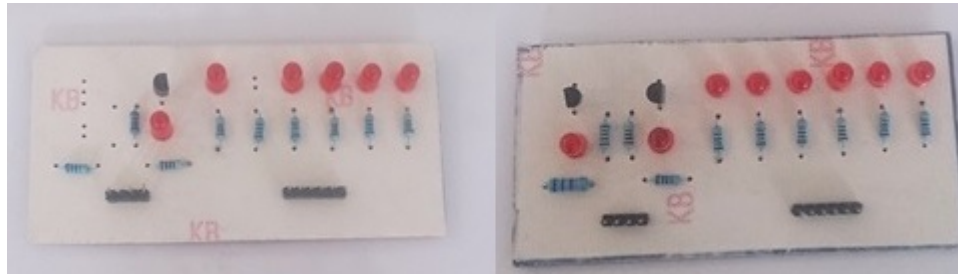


Figura 6.21: Tarjeta Entrenador, tarjeta lateral izquierda incompleta, tarjeta lateral derecha completa.

Cuando la tarjeta se encuentra en el área visible de la cámara web se detiene la banda transportadora y se ejecuta el algoritmo de reconocimiento automáticamente, este identifica que la tarjeta que se encuentra sobre la banda corresponde a la tarjeta Entrenador, tal como se observa en la figura 6.22.



Figura 6.22: Reconocimiento de tarjeta Entrenador incompleta.

Cuando la tarjeta es identificada se ejecuta el algoritmo de búsqueda de componentes, para realizar este proceso es necesario contar con los archivos Gerber de la tarjeta Entrenador para conocer las coordenadas cartesianas de cada componente, en la figura 6.23 se muestra la lista de componentes de esta tarjeta.

Designator	Center-X(mm)	Center-Y(mm)
C1	32.258	37.846
C2	76.581	37.846
Led1	8.89	9.271
Led2	18.161	9.271
Led3	27.432	9.271
Led4	36.703	9.271
Led5	45.974	9.271
Led6	55.245	9.271
Led7	68.58	21.209
Led8	87.503	21.209
Res1	8.967	21.441
Res2	18.238	21.441
Res3	27.509	21.441
Res4	36.78	21.441
Res5	46.051	21.441
Res6	55.322	21.441
Res7	74.853	19.917
Res8	67.288	30.022
Res9	89.049	30.022
Res10	80.341	19.917
q1	68.58	11.049

Figura 6.23: Tarjeta Entrenador incompleta.

Al implementar el algoritmo de búsqueda de componentes se identifica que la tarjeta se encuentra incompleta, tal y como se muestra en la figura 6.24, se puede observar que falta el led *Led8*, el led *Led5*, el transistor *q2* y la resistencia *Res10*.

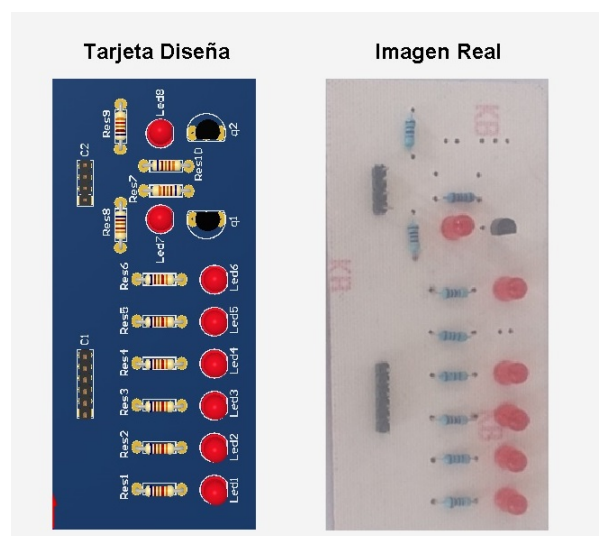


Figura 6.24: Tarjeta Entrenador incompleta.

El ROTRADI retira de la banda transportadora la tarjeta electrónica por medio del gripper, tal como se muestra en la figura 6.25. La estructura de control que se encuentra implementada en el ROTRADI permite que este realice un movimiento suave. Cuando la tarjeta es retirada, la banda transportadora se activa nuevamente para continuar con el proceso. El ROTRADI se coloca nuevamente a un costado de la banda transportadora para retirar la siguiente tarjeta que no cumpla con las especificaciones del usuario.

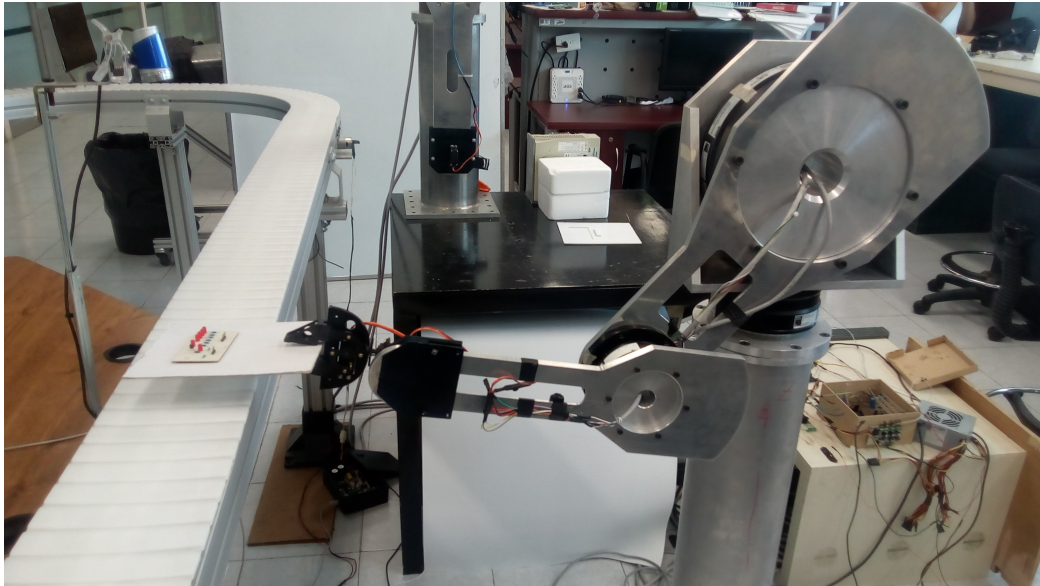


Figura 6.25: ROTRADI retirando la tarjeta Entrenador incompleta.

A continuación por medio del ROTRADI de 1 GDL, se coloca la tarjeta Entrenador completa sobre la banda transportadora, esta acción es controlada por el usuario utilizando la GUIDE en el panel de control *Control Brazo* activando el botón *InicioP*.

Cuando la tarjeta se encuentra al alcance de la cámara web se detiene la banda transportadora y se ejecuta el algoritmo de reconocimiento, el cual da como resultado que la tarjeta corresponde a la familia Entrenador. Posteriormente se ejecuta el algoritmo de búsqueda de componentes, en la figura 6.26 se observa que la tarjeta real contiene todos sus componentes electrónicos tal como se observa en la tarjeta diseñada, en este caso el algoritmo realiza una notificación de que la tarjeta se encuentra completa. Considerando que la tarjeta identificada es la correcta y contiene todos sus componentes electrónicos la celda robótica permite que la tarjeta se quede sobre la banda transportadora y el ROTRADI de tres grados de libertad no se activa.

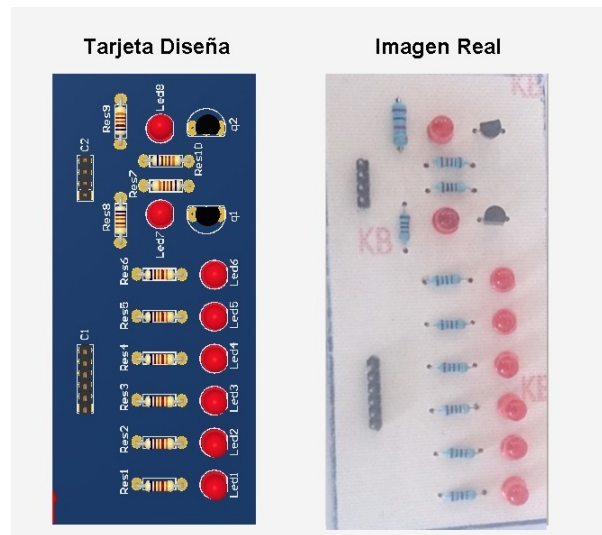


Figura 6.26: Tarjeta Entrenador incompleta.

En la figura 6.27 se muestra la interfaz reconociendo de manera correcta la tarjeta Entrenador. En el panel *Estado* indica el número de tarjetas que han sido reconocidas, indicando 2 tarjetas Entrenador y 0 tarjetas Arduino. Se indica el registro de temperatura del motor, el cual corresponde a 26° C. En este caso la banda se acciona nuevamente y el proceso continúa, el ROTRADI se mantiene en la misma posición.

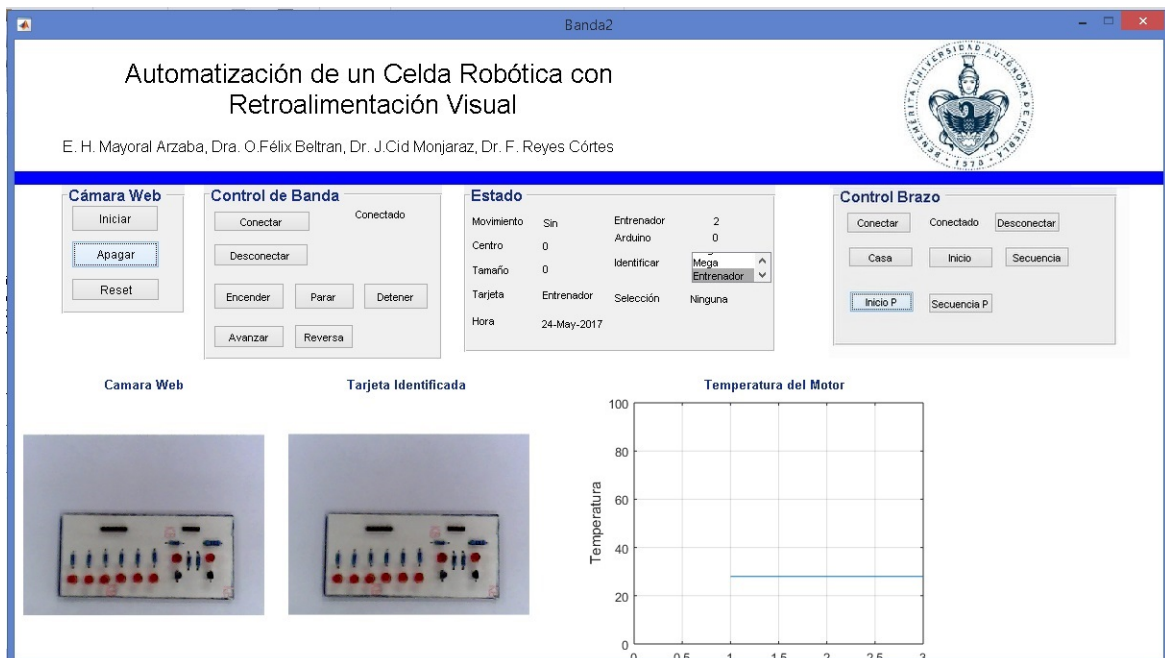


Figura 6.27: Reconocimiento de tarjeta Entrenador completa.

La siguiente tarjeta a identificar corresponde a la tarjeta Arduino Mega. La cual es colocada por el ROTRADI de 1 GDL tal como se observa en la figura 6.28.



Figura 6.28: ROTRADI colocando tarjeta Arduino sobre la banda transportador.

El algoritmo de reconocimiento identifica la tarjeta electrónica dentro de la familia de Arduino. El algoritmo de búsqueda de componentes no se implementa para esta tarjeta, por que no se cuentan con los archivos Gerber. En la figura 6.29 en el panel *Estado* se observa el nombre de la tarjeta reconocida el cual corresponde a Mega, y se tiene un registro de reconocimiento de 2 tarjetas Entrenador y 1 tarjeta Arduino. La temperatura del motor se mantiene a 26° C.



Figura 6.29: Reconocimiento de tarjeta Arduino Mega.

Considerando que la tarjeta seleccionada anteriormente corresponde a la tarjeta Entrenador, la interfaz manda un comando al controlador del ROTRADI para retirar la tarjeta identificada. En la figura 6.30 se muestra el ROTRADI retirando la tarjeta Arduino de la banda transportadora.

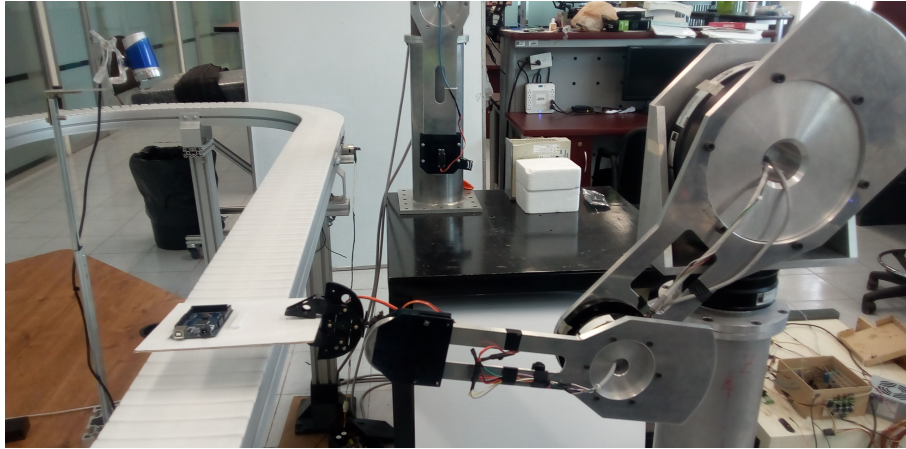


Figura 6.30: ROTRADI retirando la tarjeta Arduino de la banda transportadora.

Como resulta final se desea que la información obtenida por la Guide sea enviada a la plataforma de ThingSpeak utilizando el módulo ESP8266. Se colocan diferentes tarjetas en la banda transportadora y se visualiza la información en la página de ThingSpeak, en la figura 6.31 muestra que se han reportado 4 tarjetas Entrenador y 3 tarjetas Arduino. La temperatura registrada por motor se encuentra entre 26°C y 28°C .

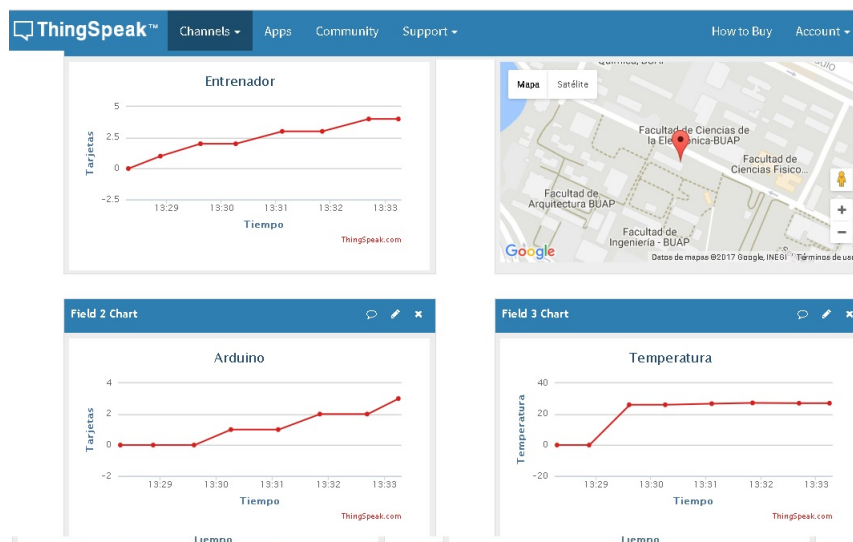


Figura 6.31: Guide implementada.

En este capítulo se mostraron los resultados obtenidos de la integración de la celda robótica, la cual es controlada por medio de una Guide. Se mostró el comportamiento de los algoritmos de control de posición para los ROTRADI's de 3 GDL y 1 GDL, para esto se utilizaron las tarjetas electrónicas diseñadas anteriormente para realizar la lectura del Encoder y introducir por medio de una señal de voltaje la representación del torque hacia el controlador del ROTRADI. Se observó el funcionamiento de los algoritmos de reconocimiento de tarjetas utilizando los algoritmos de aprendizaje automático, al mismo tiempo se mostró el algoritmo de búsqueda de componentes para reconocer si una tarjeta contienen todos sus elementos. Al final del capítulo se muestra que la información obtenida por la celda robótica es enviada a la plataforma de ThingSpeak utilizando el módulo ESP8266.

Capítulo 7

Conclusiones

La integración de la celda robótica en conjunto con las interfaces de control, los algoritmos de visión y la electrónica desarrollada en este trabajo de tesis, permitió implementar un proceso complejo como es el caso de reconocimiento de tarjetas electrónicas, lo cual tiene un alto grado de dificultad, dado el tamaño de los componentes electrónicos. Este resultado fue posible gracias a la implementación de nuevos algoritmos de procesamiento de imágenes como el algoritmo de aprendizaje automático.

Es necesario tener los fundamentos matemáticos para desarrollar la cinemática y el modelo dinámico de cada uno de los ROTRADI's, para este caso el uso de la convención de Denavit-Hartenberg para obtener la cinemática es de gran utilidad y a partir de esta es posible comenzar a obtener el modelo dinámico.

Un aspecto muy importante en esta tesis corresponde a las tarjetas de adquisición de datos para realizar la lectura de los encoders de los motores de transmisión directa de cada uno de los ROTRADI's. Es importante considerar dentro del diseño de las tarjetas electrónicas filtros que permitan eliminar el ruido, pero al mismo tiempo no afectar la señal de lectura de los encoders de los ROTRADI's. Para la lectura del encoder se utilizó un microcontrolador de alta velocidad de procesamiento, en caso contrario no hubiera posible conocer la posición del eslabón de manera correcta. En el diseño de la tarjeta electrónica se utilizaron DAC's para suministrar la señal de control hacia los drivers de los ROTRADI's, estos componentes utilizan un protocolo de comunicación I^2C y el DAC necesita un tiempo para realizar el procesamiento del dato y proporcionar el voltaje de salida, el cual es aproximadamente de 2 ms, por lo tanto el tiempo de muestreo debe ser mayor a este valor.

Para detener la banda transportadora se implementó un algoritmo de detección de movimiento, se quiso que la banda transportadora se detenga cuando detecte una tarjeta. Considerando el diseño de la cinta transportadora, la cual está conformada por una serie de eslabones, estos al desplazarse generan aberturas y causan problemas en el algoritmo de reconocimiento

de tarjetas electrónicas. Para esto fue necesario implementar filtros que eliminaran estas regiones, dando como resultado que la banda transportadora solo se detuviera cuando se detectara una tarjeta electrónica.

Los algoritmos de aprendizaje automático fueron una herramienta importante para el desarrollo de esta tesis, para garantizar que las tarjetas electrónicas fueran identificadas correctamente, fue necesario contar una gran cantidad de muestras de cada tarjeta en diversas posiciones. Al mismo tiempo se realizó el muestreo de la banda transportadora para que fuera un elemento adicional en la base de datos, de esta manera cuando se realizara la banda transportadora no se confundiera con una tarjeta.

Para el diseño del algoritmo de búsqueda de componentes electrónicos fue necesario tener cuidado al implementar algoritmos para la reducción del ruido, dado que es posible eliminar algunos componentes considerando su tamaño. Es conveniente realizar el muestreo de la misma tarjeta en diferentes posiciones.

Considerando que fue necesario implementar algoritmos de control en los ROTRADI, tener el control de la banda transportadora, el procesamiento de imágenes y el envío de información hacia una página web, resultó conveniente separar cada uno de estos procesos. Para ello se utilizaron microcontroladores para el control de la banda transportadora y los ROTRADI's, en este caso la computadora funcionó como una estación central, permitiendo realizar actividades más complejas y establecer la comunicación con los microcontroladores.

Como trabajo a futuro se pueden desarrollar nuevas tarjetas electrónicas con calidad industrial, el DAC utilizado se puede cambiar por un DAC con mayor resolución para aumentar la precisión de la señal de control. Se puede aumentar el número de tarjetas electrónicas a identificar utilizando el algoritmo de aprendizaje automático. Mejorando la calidad de la cámara web se puede implementar el algoritmo de búsqueda de componentes electrónicas en tarjetas con mayor densidad.

Apéndice A

Hojas de diseño del soporte del motor

En este apéndice se muestran los dibujos técnicos de las piezas mecánicas elaboradas para la sujeción del motor con la banda transportadora.

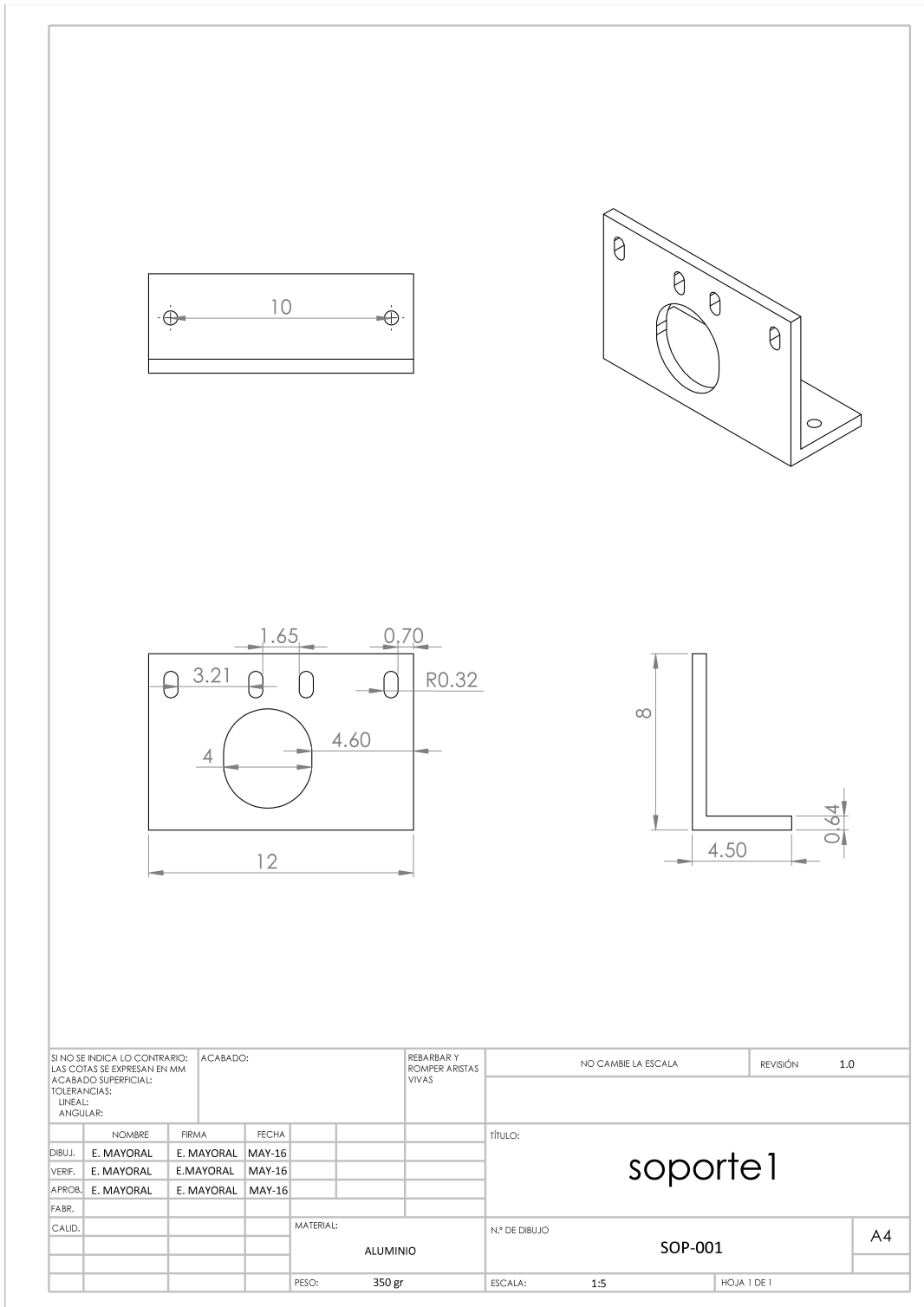


Figura A.1: Soporte 1 del motor Crouzet.

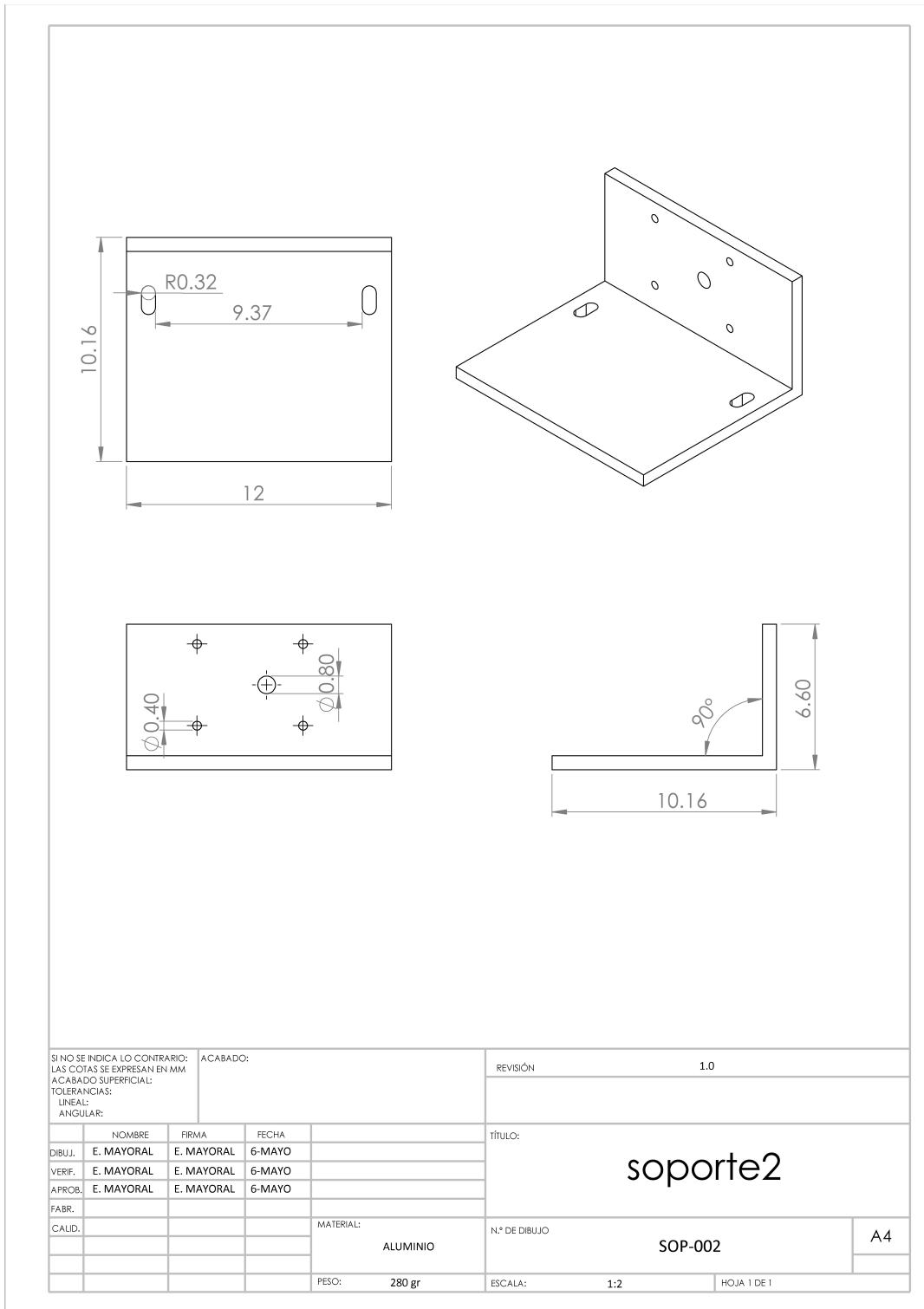


Figura A.2: Soporte 2 del motor Crouzet.

Apéndice B

Hoja de datos

En este apéndice se muestra las hojas de datos de los componentes utilizados para el desarrollo de las tarjetas electrónicas del driver del motor, la tarjeta de lectura del encoder del ROTRADI, la tarjeta que proporciona la señal de salida para el control del ROTRADI y la tarjeta de comunicación WiFi.

12-Bit Digital-to-Analog Converter with EEPROM Memory in SOT-23-6

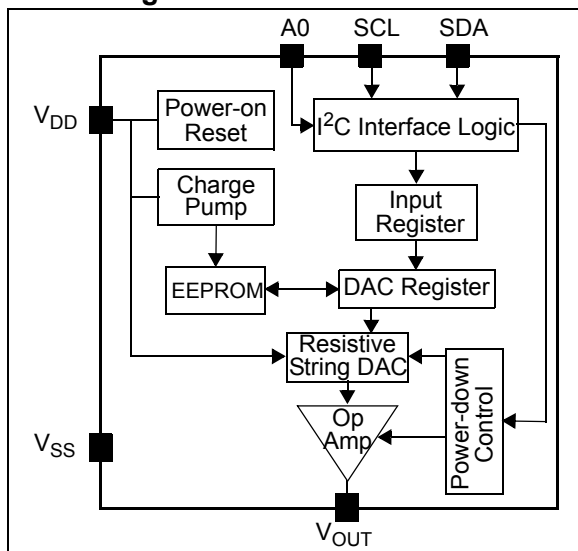
Features

- 12-Bit Resolution
- On-Board Non-Volatile Memory (EEPROM)
- ± 0.2 LSB DNL (typical)
- External A0 Address Pin
- Normal or Power-Down Mode
- Fast Settling Time: 6 μ s (typical)
- External Voltage Reference (V_{DD})
- Rail-to-Rail Output
- Low Power Consumption
- Single-Supply Operation: 2.7V to 5.5V
- I²C™ Interface:
 - Eight Available Addresses
 - Standard (100 kbps), Fast (400 kbps), and High-Speed (3.4 Mbps) Modes
- Small 6-lead SOT-23 Package
- Extended Temperature Range: -40°C to +125°C

Applications

- Set Point or Offset Trimming
- Sensor Calibration
- Closed-Loop Servo Control
- Low Power Portable Instrumentation
- PC Peripherals
- Data Acquisition Systems

Block Diagram



DESCRIPTION

The MCP4725 is a low-power, high accuracy, single channel, 12-bit buffered voltage output Digital-to-Analog Converter (DAC) with non-volatile memory (EEPROM). Its on-board precision output amplifier allows it to achieve rail-to-rail analog output swing.

The DAC input and configuration data can be programmed to the non-volatile memory (EEPROM) by the user using I²C interface command. The non-volatile memory feature enables the DAC device to hold the DAC input code during power-off time, and the DAC output is available immediately after power-up. This feature is very useful when the DAC device is used as a supporting device for other devices in the network.

The device includes a Power-On-Reset (POR) circuit to ensure reliable power-up and an on-board charge pump for the EEPROM programming voltage. The DAC reference is driven from V_{DD} directly. In power-down mode, the output amplifier can be configured to present a known low, medium, or high resistance output load.

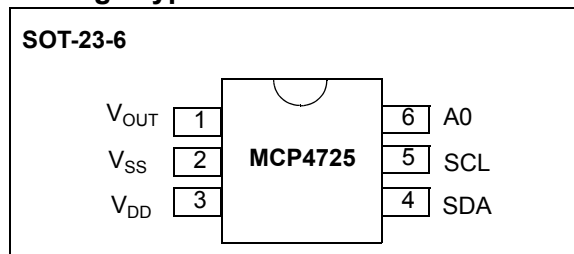
The MCP4725 has an external A0 address bit selection pin. This A0 pin can be tied to V_{DD} or V_{SS} of the user's application board.

The MCP4725 has a two-wire I²C™ compatible serial interface for standard (100 kHz), fast (400 kHz), or high speed (3.4 MHz) mode.

The MCP4725 is an ideal DAC device where design simplicity and small footprint is desired, and for applications requiring the DAC device settings to be saved during power-off time.

The device is available in a small 6-pin SOT-23 package.

Package Type





DS1920 Temperature iButton

SPECIAL FEATURES

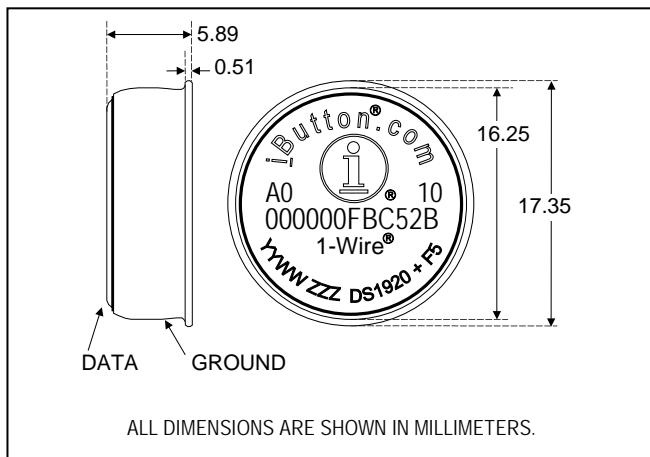
- Digital Thermometer Measures Temperatures from -55°C to $+100^{\circ}\text{C}$ in Typically 0.2s
- Zero Standby Power
- 0.5°C Resolution, Digital Temperature Reading is Two's Complement of $^{\circ}\text{C}$ Value
- Access to Internal Counters Allows Increased Resolution Through Interpolation
- Reduces control, Address, Data, and Power to a Single Data Contact
- 8-Bit Device-Generated CRC for Data Integrity
- 8-Bit Family Code Specifies DS1920 Communications Requirements to Reader
- Special Command Set Allows User to Skip ROM Section and Perform Temperature Measurements Simultaneously for All Devices on the Bus
- 2 Bytes of EEPROM to Be Used Either as Alarm Triggers or User Memory
- Alarm Search Directly Indicates Which Device Senses Alarming Temperatures

ORDERING INFORMATION

PART	PIN-PACKAGE
DS1920-F5+	F5 MicroCAN

+Denotes a lead(Pb)-free/RoHS-compliant package.

F5 MicroCAN



COMMON iButton FEATURES

- Unique, Factory-Lasered and Tested 64-Bit Registration Number (8-Bit Family Code + 48-Bit Serial Number + 8-Bit CRC Tester) Assures Absolute Traceability Because No Two Parts Are Alike
- Multidrop Controller for MicroLAN
- Digital Identification and Information by Momentary Contact
- Chip-Based Data Carrier Compactly Stores Information
- Data Can Be Accessed While Affixed To Object
- Economically Communicates to Bus Master with a Single Digital Signal at 16.3kbps
- Standard 16mm Diameter and 1-Wire[®] Protocol Ensure Compatibility with iButton[®] Family
- Button Shape is Self-Aligning with Cup-Shaped Probes
- Durable Stainless Steel Case Engraved with Registration Number Withstands Harsh Environments
- Easily Affixed with Self-Stick Adhesive Backing, Latched by Its Flange, or Locked with a Ring Pressed Onto Its Rim
- Presence Detector Acknowledges When Reader First Applies Voltage

EXAMPLES OF ACCESSORIES

DS9096P	Self-Stick Adhesive Pad
DS9101	Multipurpose Clip
DS9093RA	Mounting Lock Ring
DS9093F	Snap-In Fob
DS9092	iButton Probe

iButton and *1-Wire* are registered trademarks of Maxim Integrated Products, Inc.

2. Hardware Overview

2.1. Pin Definitions

The pin assignments for 32-pin QFN package is illustrated in Fig.2.

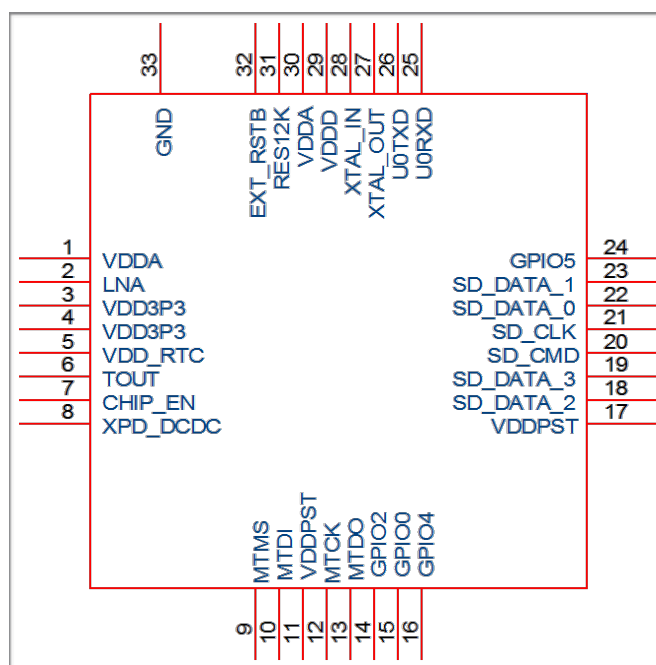


Figure 2 Pin Assignments

Table 2 below presents an overview on the general pin attributes and the functions of each pin.

Table 2 Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC (1.1V)

Apéndice C

Firmware para tarjetas Arduinos

En este apéndice se muestran los códigos desarrollados para las tarjetas Arduino. El primer código genera una señal PWM de frecuencia variable y ancho de pulso variable para el control de velocidad de la banda transportadora. El segundo código realiza la lectura del sensor de temperatura DS1920, este sensor utiliza el protocolo de comunicación 1-Wire. El tercer código establece la comunicación con el modulo WiFi, configurando el modulo como servidor y con punto de conexión hacia la red de Internet, estableciendo la comunicación con el servidor de ThingSpeak.

C.1. Firmware PWM variable para control de la banda transportadora

```
//Variables para comunicación serial
char lecturaS[4],cambio=0, pPWM=22;;
int posicion=0, centenas=0, frecuencia=0, pw2=500, fr2=500;;
short unidades=0,decenas=0, porcentaje=0;
unsigned long pw1=500,pwm=0;
float frecuencia2=0;

//Función de interrupción
void velocidad2()
{
  if (cambio==0) // Flanco descendente
    { digitalWrite(pPWM,0); digitalWrite(13,0);
      Timer1.initialize(pw2); cambio=1;    }
  else
    { digitalWrite(pPWM,1); digitalWrite(13,1);
      Timer1.initialize(pw1); cambio=0; }
}
```

```
short conv_numero(char caracter)
{
    short valor=0;
    switch (caracter)
        { case '1': valor=1; break; case '2': valor=2; break;
          case '3': valor=3; break; case '4': valor=4; break;
          case '5': valor=5; break; case '6': valor=6; break;
          case '7': valor=7; break; case '8': valor=8; break;
          case '9': valor=9; break; case '0': valor=0; break;
          default: valor=0; break; }
    return valor;
}

void limpiar()
{
    lecturaS[0]=0; lecturaS[1]=0; lecturaS[2]=0;
    unidades=0; decenas=0; centenas=0;
    pwm=0; frecuencia=0;
}

void setup() {
    Serial.begin(9600);
    //Sentencia de interrupciones
    Timer1.initialize(fr2);
    Timer1.attachInterrupt(velocidad2);
    //Declaracion de pines de salida
    pinMode(pPWM,OUTPUT); pinMode(13,OUTPUT);
}

void loop() {
    if (Serial.available())
    { Serial.readBytesUntil('\n',lecturaS,4);
      Serial.println(lecturaS); }
    if (lecturaS[0]=='f')
        { unidades=conv_numero(lecturaS[3]);
          decenas=conv_numero(lecturaS[2])*10;
          centenas=conv_numero(lecturaS[1])*100;
          frecuencia=unidades+decenas+centenas;
          fr2=frecuencia*100; limpiar(); }
    else if (lecturaS[0]=='p')
        {
```

```

    unidades=conv_numero(lecturaS[2]);
    decenas=conv_numero(lecturaS[1])*10;
    pwm=unidades+decenas; pw1=0;
    pw1=pwm*fr2; delay(20);
    pw1=pw1/100; pw2=fr2-pw1;
    limpiar(); }
}

```

C.2. Firmware para DS1920

A continuación se muestra el algoritmo implementado en la plataforma Arduino para realizar la lectura de temperatura del sensor DS1920.

```

#include <OneWire.h>
#include <DallasTemperature.h>

unsigned char imp[10], v1=0;

//--Configuración del sensor 1-Wire--
OneWire ourWire(2);
DallasTemperature sensors(&ourWire);
//--Dirección física del sensor--
DeviceAddress address1 = {0x10, 0x15, 0x24, 0xA3, 0x1, 0x8, 0x0, 0x50};

//--Función para realizar impresión de temperatura--
void impresion(unsigned char valor)
{
    unsigned char d1=0,d2=0,d3=0,c1;
    if (valor>=100)
    {
        d3=valor/100; d2=valor-(d3*100);
        if(d2>=10)
            { valor=d2; d2=valor/10; d1=valor-(d2*10); }
    }
    else if((valor<100)&&(valor>=10))
    { d3=0; d2=valor/10; d1=valor-(d2*10); }
    else if(valor<10)
    { d3=0; d2=0; d1=valor; }
}

void setup() {

```

```
delay(1000);
Serial.begin(9600);
sensors.begin(); //Se inicia el sensor
}

void loop() {
  //--Obtención de temperatura--
  sensors.requestTemperatures();
  float temp1= sensors.getTempC(address1);

  //--Impresión de Temperatura--
  v1=temp1;
  Serial.print(1);
  Serial.print(" ");
  impresion(v1);
  Serial.println(" ");
  delay(100);
}
```

C.3. Firmware conexión ThingSpeak

A continuación se muestra el algoritmo implementado en la plataforma Arduino para establecer la comunicación con el servidor ThingSpeak

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <WiFiServer.h>
#include <WiFiUdp.h>
#include <stdlib.h>

float temp1=10;
char buf[16];
String strTemp = dtostrf(temp1, 4, 1, buf);

String API = "XXXXXXXXXXXX";
String apiKey = "XXXXXXXXXXXX";
// enter your ssid and password
const char* ssid = "XXXXXX";
const char* password = "XXXXXX";
// your message
String tweet="PRUEBA_WIFI_V4\r\n";
String tw="XXX";
```

```
// open client
WiFiClient client;
void setup() {
  pinMode(16,OUTPUT); Serial.begin(115200);
  delay(3000);

  WiFi.mode(WIFI_STA); WiFi.disconnect();
  delay(100);

  // connect to wifi
  WiFi.begin(ssid, password);

  // allow time to make connection
  while (WiFi.status() != WL_CONNECTED)
  { delay(500); Serial.print("."); }

// if connection to thingspeak.com is successful, send your tweet!
if (client.connect("184.106.153.149", 80))
{
  client.print("GET /apps/thingtweet/1/statuses/update?key=" + API + "&status="
+ tweet + tw + " HTTP/1.1\r\n");
  client.print("Host: api.thingspeak.com\r\n"); client.print("Accept: */*\r\n");
  client.print("User-Agent: Mozilla/4.0 (compatible; esp8266 Lua; Windows NT 5.1)\r\n");
  client.print("\r\n");
}}

void loop()
{
  String getStr = "GET /update?api_key=";
  getStr += apiKey; getStr += "&field1=";
  getStr += String(strTemp); getStr += "\r\n\r\n";
  client.print("POST /update HTTP/1.1\n");
  client.print("Host: api.thingspeak.com\n");
  client.print("Connection: close\n");
  client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
  client.print("Content-Type: application/x-www-form-urlencoded\n");
  client.print("Content-Length: ");
  client.print(getStr.length());
  client.print("\n\n");
  client.print(getStr);
}
```

Apéndice D

Códigos de MATLAB

En esta sección se muestra el algoritmo de aprendizaje automático implementado en MATLAB para el reconocimiento de características críticas de una imagen.

D.1. Algoritmo de aprendizaje automatico

```
%% Load image data
imshow = imageSet('tarjetas','recursive');

%% Pre-process Training Data: *Feature Extraction*
% Requires: Computer Vision System Toolbox

% Create a bag-of-features from the Car image database
bag = bagOfFeatures(imset,'VocabularySize',200,...
    'PointSelection','Detector');

% Encode the images as new features
imagefeatures = encode(bag,imshow);

%% Create a Table using the encoded features
CarData      = array2table(imagefeatures);
CarData.carType = getImageLabels(imset);

%% Use the new features to train a model and assess its performance
classificationLearner

%% Test Trained Model
CarFinderLive(trainedClassifier,bag)
```

D.2. Algoritmo de la instrucción CarFinderLive

```

function CarFinderLive(trainedClassifier,bag)
[fig, ax1, ax2] = figureSetup(trainedClassifier)

% Start webcam
wcam = webcam(2);

% Run live car detection
while ishandle(fig)
    % Step 1: Get Next Frame
    img = snapshot(wcam);
    graying = rgb2gray(img);

    % Step 2: Extract Features
    imagefeatures = double(encode(bag,grayimg));

    % Step 3: Predict car using extracted features
    [imagepred, probabilities] = predict(trainedClassifier,imagefeatures);

    % Step 4: Plot Results
    try
        imshow(insertText(img,[640,1],upper(cellstr(imagepred)),...
            'AnchorPoint','RightTop','FontSize',50,'BoxColor','Green',...
            'BoxOpacity',0.4),'Parent',ax1);
        ax2.Children.YData = probabilities;
        ax2.YLim = [0 1];
    catch err
    end
    drawnow
end

function cname = getClassifierName(trainedClassifier)
cname = class(trainedClassifier);
if isa(trainedClassifier,'ClassificationECOC')
    cname = 'SVM';
end
pos = strfind(cname,'.');
if ~isempty(pos)
    cname = cname(pos(end)+1:end);
end

```

```
function [fig, ax1, ax2] = figureSetup(trainedClassifier)
warning('off','images:imshow:magnificationMustBeFitForDockedFigure')
set(0,'defaultfigurewindowstyle','docked')
fig = figure('Name','Car Finder Go!','NumberTitle','off');
ax1 = subplot(2,1,1);
ax2 = subplot(2,1,2);
bar(ax2,zeros(1,numel(trainedClassifier.ClassNames)),'FaceColor',[0.2 0.6 0.8])
set(ax2,'XTickLabel',cellstr(trainedClassifier.ClassNames));
title(getClassifierName(trainedClassifier)), ylabel('Probability')
set(0,'defaultfigurewindowstyle','normal')
```

Apéndice E

Comandos AT-ESP8266

En este apéndice se muestran los comandos AT para establecer comunicación entre el Arduino Mega con el modulo ESP8266, para la transmisión de datos por internet al servidor de ThingSpeak.

ESP8266 AT Command Set

Function	AT Command	Response
Working	AT	OK
Restart	AT+RST	OK [System Ready, Vendor:www.ai-thinker.com]
Firmware version	AT+GMR	AT+GMR 0018000902 OK
List Access Points	AT+CWLAP	AT+CWLAP +CWLAP:(4,"RocheFortSurLac",-38,"70:62:b8:6f:6d:58",1) +CWLAP:(4,"LiliPad2.4",-83,"f8:7b:8c:1e:7c:6d",1) OK
Join Access Point	AT+CWJAP? AT+CWJAP="SSID","Password"	Query AT+CWJAP? +CWJAP:"RocheFortSurLac" OK
Quit Access Point	AT+CWQAP=? AT+CWQAP	Query OK
Get IP Address	AT+CIFSR	AT+CIFSR 192.168.0.105 OK
Set Parameters of Access Point	AT+ CWSAP? AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn>	Query ssid, pwd chl = channel, ecn = encryption
WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Set up TCP or UDP connection	AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	Query id = 0-4, type = TCP/UDP, addr = IP address, port= port
TCP/UDP Connections	AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1	Query Single Multiple
Check join devices' IP	AT+CWLIF	
TCP/IP Connection Status	AT+CIPSTATUS	AT+CIPSTATUS? no this fun
Send TCP/IP data	(CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length>	
Close TCP / UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	
Set as server	AT+ CIPSERVER= <mode>[,<port>]	mode 0 to close server mode; mode 1 to open; port = port
Set the server timeout	AT+CIPSTO? AT+CIPSTO=<time>	Query <time>0~28800 in seconds
Baud Rate*	AT+CI0BAUD? Supported: 9600, 19200, 38400, 74880, 115200, 230400, 460800, 921600	Query AT+CI0BAUD? +CI0BAUD:9600 OK
Check IP address	AT+CIFSR	AT+CIFSR 192.168.0.106 OK
Firmware Upgrade (from Cloud)	AT+CIUPDATE	1. +CIPUPDATE:1 found server 2. +CIPUPDATE:2 connect server 3. +CIPUPDATE:3 got edition 4. +CIPUPDATE:4 start update
Received data	+IPD	(CIPMUX=0): + IPD, <len>; (CIPMUX=1): + IPD, <id>, <len>; <data>
Watchdog Enable*	AT+CSYSWDTENABLE	Watchdog, auto restart when program errors occur: enable
Watchdog Disable*	AT+CSYSWDTDISABLE	Watchdog, auto restart when program errors occur: disable

* New in V0.9.2.2 (from <http://www.electrodragon.com/w/Wi07c>)

Apéndice F

Congresos y publicaciones

En este apéndice se muestran las evidencias de la participación en los congresos nacionales (2) e internacionales (2). Se participó en el *VII ENCUENTRO NACIONAL DE CIENCIAS Luis Rivera Terrazas 2016*, realizado en Puebla, Pue., México, del 20 al 23 de septiembre de 2013; se presentó el poster titulado “Implementación de un algoritmo de aprendizaje automático para identificación de tarjetas electrónicas”. Se participó en el evento internacional *VII Congreso Internacional de Ingeniería Electromecánica y de Sistemas (CIESS)*, realizado en la ciudad de México, del 17 al 21 de octubre de 2016; se presentó el artículo “Implementation of machine learning algorithm for identifying electronic boards” (ponencia). Se participó en el *VII Congreso Nacional de Ingenierías* realizado en la ciudad de Orizaba, del 17 al 21 de octubre de 2016 con el curso denominado “Diseño PCB con ALTIUM”, por invitación. Se participó en el *IEEE Mexican Humanitarian Technology Conference (MHTC) 2016* realizado en la ciudad de Puebla, el 31 de marzo de 2017, se presentó el artículo “Automatización de una Celda Robótica usando algoritmos de Aprendizaje Automático e Internet de las cosas”. Finalmente se publica en la revista *THEOREMA* en su sexta edición el trabajo “IMPLEMENTACIÓN DE ALGORITMO DE APRENDIZAJE AUTOMÁTICO PARA IDENTIFICACIÓN DE TARJETAS ELECTRÓNICAS”, el trabajo fue enviado el día 11 de noviembre de 2016 y aceptado el 22 de mayo de 2017.



VII ENCUENTRO NACIONAL DE CIENCIAS LUIS RIVERA TERRAZAS

Puebla Pue, 13 de septiembre de 2016

Apreciable

Mayoral Arzaba Edgar Hugo

Me es grato comunicarle que su trabajo cuyo titulo es

*Implementación de un algoritmo de aprendizaje
automático para identificación de tarjetas electrónicas*

Cuyos autores son

Félix Beltrán Olga

Cid Monjaraz Jaime

Reyes Cortés Fernando

Ha sido aceptado para su presentación en el VII Encuentro Nacional de Ciencias Luis Rivera Terrazas, que se celebrará del 20 al 23 de septiembre de 2016. En la página del Encuentro se publicará la fecha y hora de su presentación.

Atentamente



Dra. Guadalupe Quintero Téllez
Comité organizador LRT 2016





Implementación de un algoritmo de aprendizaje automático para identificación de tarjetas electrónicas

E. H. Mayoral Arzaba, J. Cid Monjaraz, F. Reyes Cortés, O. Félix Beltrán
Maestría en Ciencias de la Electrónica opción Automatización, BUAP, edgarhugo130590@hotmail.com

Resumen— El procesamiento de imágenes es un tema en evolución en los últimos años, permite una gran variedad de aplicaciones. En este trabajo se describe un algoritmo de aprendizaje automático que es capaz de reconocer de manera inmediata cada diferente tipo de tarjeta electrónica y posteriormente verifica que contiene todos sus componentes electrónicos en la posición correcta.

INTRODUCCIÓN

Una parte importante de la investigación en el sector de las ciencias computacionales es el desarrollo de nuevas técnicas que permitan realizar el procesamiento de imágenes con una mayor eficacia. Dada la gran cantidad de información que se obtiene al procesar un conjunto de imágenes, es necesario contar con una herramienta que permita manipular esta cantidad de información. El aprendizaje automático tiene como enfoque obtener la información más relevante a partir de una gran cantidad de datos almacenados en una instancia [1].

DESARROLLO

Se implementa un sistema que está constituido por una banda transportadora, un brazo robótico, una cámara web y un sistema de adquisición de datos, tal como se muestra en la figura 1.



Figura 1.-Sistema banda transportadora-brazo robótico y cámara web.

Las tarjetas electrónicas se encuentran colocadas en la banda transportadora. Por medio de la cámara web, en conjunto con los algoritmos de reconocimiento, se identifica la tarjeta electrónica y posteriormente se realiza el proceso de verificación de los componentes electrónicos.

Utilizando el software de Matlab y el algoritmo propuesto por Paul Viola [2], en conjunto con el algoritmo de identificación de rostros en tiempo real [3], se realiza la identificación de seis tarjetas electrónicas de diferente modelo.

Para implementar el algoritmo de aprendizaje, se adquieren 500 imágenes de cada una de las tarjetas, de las cuales se extraen las 200 características más importantes, obteniendo una matriz con la información más relevante de cada tarjeta. Una vez obtenida la matriz de información se entrenan diferentes algoritmos y se comparan los resultados obtenidos de cada algoritmo entrenado. Se implementa el algoritmo que presenta el mayor índice de eficacia. En la figura 2 se muestra los índices de efectividad de cada uno de los algoritmos entrenados, entre los cuales se encuentran el algoritmo Complex Tree, Medium Gaussian SVM, Quadratic SVM, entre otros [4].

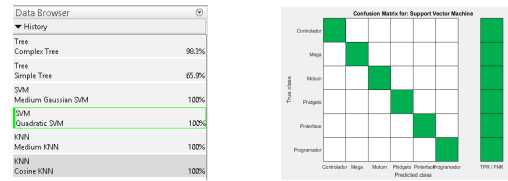


Figura 2.-Índice de efectividad de algoritmos entrenados.

Las pruebas de reconocimiento de cada una de las tarjetas electrónicas entrenadas se muestran en la figura 3.

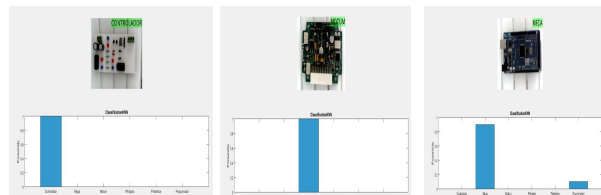


Figura 3.- Resultado del algoritmo de identificación de tarjetas.

El algoritmo de reconocimiento de componentes electrónicos consiste en identificar los centroides de una imagen, por medio de una serie de algoritmos y filtros digitales.

Se realizan varias pruebas para validar el funcionamiento del algoritmo entrenado, desarrollando una interfaz en Matlab. Esta interfaz indica si la tarjeta identifica se encuentra completa o incompleta, en el segundo caso se enlistan los componentes faltantes.

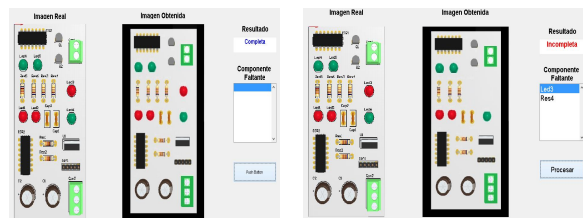


Figura 4.- Proceso de identificación de componentes electrónicos.

CONCLUSIONES

Se implemento un algoritmo de aprendizaje automático capaz de reconocer diferentes tipos de tarjetas electrónicas, así como sus componentes electrónicos. Este algoritmo tiene la capacidad de realizar la automatización de procesos de manufactura de tarjetas electrónicas, reduciendo el costo de ensamble y al mismo tiempo el costo de producción. Por lo tanto se genera una nueva aplicación de los algoritmos con sistema de aprendizaje automático.

REFERENCIAS

[1] Avrim L. B., Part L., Selection of relevant features and examples in machine learning, *Elsevier Artificial Intelligence*, School of Computer Science, Carnegie Mellon University Pittsburgh, pp 245-271, 1997.
[2] Viola P. y Jones M., Rapid object detection using a boosted cascade of simple features, *Conference on Computer Vision and Pattern Recognition*, pp 1-9, 2001.
[3] Viola P. y Jones M., Robust real-time face detection, *International Journal of Computer Vision*, pp 1-17, 2004.
[4] Smola A. y Vishwanathan S., *Introduction to Machine Learning*, pp 3-219, 2008.



VIII Congreso Internacional de Ingeniería Electromecánica y de Sistemas

Otorga el Presente

DIPLOMA

A:

Edgar Hugo Mayoral Arzaba,
Jaime Cid Monjaraz,
Jose Fernando Reyes Cortés y
Olga Félix Beltrán

Por su participación con el trabajo:

Implementation of machine learning
algorithm for identifying electronic boards

Ciudad de México, 17 al 21 de octubre de 2016



DR. MIGUEL TOLEDO VELÁZQUEZ
Jefe de la Sección de Estudios de Posgrado
e Investigación de la ESIME, Unidad Zacatenco



DR. RICARDO OCTAVIO MOTA PALOMINO
Director de la ESIME, Unidad Zacatenco

Implementation of machine learning algorithm for identifying electronic boards (Implementación de algoritmo de aprendizaje automático para identificación de tarjetas electrónicas)

E.H. Mayoral Arzaba, J. Cid Monjaraz, J.F. Reyes Cortés, O. Félix Beltrán

Resumen— El procesamiento de imágenes es un tema en evolución en los últimos años, permite una gran variedad de aplicaciones. En este artículo se describe un algoritmo de aprendizaje automático que es capaz de reconocer de manera inmediata cada diferente tipo de tarjeta electrónica y posteriormente verifica que contiene todos sus componentes electrónicos en la posición correcta. Este proceso depende de la tarjeta electrónica identificada, con la finalidad de poder analizar diferentes tipos de tarjetas sin la necesidad de realizar modificaciones de parámetros o instrucciones en el algoritmo, obteniendo como resultado un proceso automático e inteligente. Para el procesamiento de imágenes se utiliza una cámara web.

Palabras Clave—Algoritmo de aprendizaje automático, tarjeta electrónica y procesamiento de imágenes.

Abstract— In recent years the imagen processing is an envolving issue, allows a great variety of applications. This article describes an machine learning algorithm that is able to immediately recognize each different type of electronic board, and verify subsequently that contains all electronic components in the correct position. This process depends on the electronic board identified, order to be able to analyze different types of board without the need to make modifications of parameters or instructions in the algorithm, getting as result an automatic and intelligent process. For image processing a webcam is used.

Keywords— Machine learning algorithm, electronic board and image processing.

I. INTRODUCCIÓN

En el proceso de manufactura de tarjetas electrónicas se emplean máquinas especializadas denominadas AOI (*Automated Optical Inspection*), se utilizan para detectar defectos en una PCB (*Printed Circuit Board*) [1]. Por medio de una cámara de alta definición se realiza de manera automática el escaneo del dispositivo bajo prueba con el propósito de validar que no posea ningún defecto, como componentes faltantes, componentes mal orientados, componentes incorrectos, entre otros. Para realizar el proceso de supervisión utilizando este tipo de máquina es necesario contar con los archivos de fabricación y de ensamble de la PCB denominados archivos *gerber's*. Estos contienen información de la posición, orientación y dimensión de cada componente utilizado sobre la tarjeta electrónica. Es necesario configurar la máquina AOI para cada tarjeta electrónica diferente que se desea supervisar. En los últimos se han desarrollado una gran cantidad de algoritmos para las máquinas AOI. En el trabajo presentado por Huibin Zhao [2] presenta un dispositivo y un algoritmo basado en visión para el reconocimiento de componentes electrónicos en una PCB de densidad media, utilizando el software de Labview en conjunto con la herramienta de NI Vision (National Instruments Vision), el sistema consiste en una cámara IEEE1394, una interfaz y una plataforma móvil de dos grados de libertad. Este sistema presenta dos desventajas notables, el uso de una de cámara especial y la necesidad de un hardware externo que permita controlar la posición de la cámara. Una parte importante de la investigación en el sector de las ciencias computacionales es el desarrollo de nuevas técnicas que permitan realizar el procesamiento de imágenes con una mayor eficacia, por lo que es necesario involucrar otras áreas del conocimiento para obtener mejores resultados. Dada la gran cantidad de información que se obtiene al procesar un conjunto de imágenes, es necesario contar con una herramienta que permita manipular esta cantidad de información y poder obtener los datos más relevantes.

E.H Mayoral, Benemérita Universidad Autónoma de Puebla, México, edgarhugo130590@hotmail.com.

El aprendizaje automático tiene como enfoque obtener la información más relevante a partir de una gran cantidad de datos almacenados en una instancia [3]. Se encuentra constituido por una serie de algoritmos computacionales que permiten identificar las características más importantes sin la necesidad de contar con un modelo matemático o una ecuación predeterminada, la obtención de los datos depende de la cantidad de información disponible [4]. Existe una amplia variedad de aplicaciones donde se implementa este tipo de algoritmo, por ejemplo para la estimación de movimiento de un exoesqueleto asistido [5], modelado de un brazo robótico por aprendizaje automático [6], algoritmo de soporte de decisión para el proceso de control de manufactura de circuitos impresos [7]. Este último ejemplo realiza la identificación de fallas en las líneas de conducción de la PCB, como por ejemplo la ruptura de una línea de conducción, emplean una máquina AOI para realizar la inspección de la tarjeta electrónica.

En este trabajo se implementa un sistema de aprendizaje automático en conjunto con el procesamiento digital de imágenes. Se describe un algoritmo que permite identificar diferentes tipos de tarjetas electrónicas de manera automática, posteriormente se realiza el proceso de supervisión de los componentes electrónicos de cada tarjeta, sin la necesidad de modificar algún parámetro en el algoritmo, de esta manera se evita que la línea de producción se detenga.

Una de las principales ventajas de este trabajo es que no requiere de una máquina de inspección AOI para realizar el proceso de supervisión de las tarjetas electrónicas. Se sustituye la máquina de inspección AOI por una cámara web, reduciendo el costo de la infraestructura.

El método empleado se basa en la obtención de las características críticas de cada una de las tarjetas electrónicas y una base de datos que contiene los archivos *gerber* de cada tarjeta. Por medio de una cámara web en conjunto con el algoritmo de aprendizaje automático se identifica el tipo de tarjeta electrónica y automáticamente se selecciona de la base de datos el archivo *gerber* correspondiente a la tarjeta, posteriormente se implementa un algoritmo que permite identificar los componentes físicos que contiene la tarjeta electrónica.

El artículo está organizado de la siguiente forma: en la sección II.A se implementa el algoritmo de aprendizaje automático para el reconocimiento de las características críticas de cada tarjeta; en la sección II.B se muestra el algoritmo de búsqueda de componentes físicos a partir de los archivos *gerber*; en la sección III se muestran los resultados experimentales, finalmente en la sección IV se presentan las conclusiones de este trabajo.

II. DESARROLLO Y CAMPO TEÓRICO

Se implementa un sistema que está constituido por una banda transportadora, un brazo robótico [8], una cámara web y un sistema de adquisición de datos, tal como se muestra en la figura 1.



Figura 1.- Sistema banda transportadora-brazo robótico y cámara web.

Las tarjetas electrónicas se encuentran colocadas en la banda transportadora, la cual se encuentran en movimiento con una velocidad constante tal como se muestra en la figura 2. Por medio de la cámara web, en conjunto con los algoritmos de reconocimiento, se identifica la tarjeta electrónica y posteriormente se realiza el proceso de verificación de los componentes electrónicos de la tarjeta. Las tarjetas que no presenten ningún error son retiradas de la banda transportadora por medio del brazo robot.



Figura 2.-Reconocimiento de tarjetas electrónicas, Laboratorio de Robótica, FCE-BUAP.

En la tabla 1 se muestran las especificaciones técnicas de la cámara web que se utilizó para el desarrollo de este proyecto, resulta ser una cámara de características básicas.

Tabla 1.- Especificaciones de la cámara web.

Descripción	Valor
Resolución de cámara web	5 Mpx
Velocidad de fotograma	30 Fps
Resolución en pixeles	640 x 480
Sensor de imagen	CMOS

La implementación del sistema se realizó en el Laboratorio de Robótica en la FCE-BUAP. Este presenta una luminiscencia en el intervalo de 90 lux a 380 lux, no se requirió de una fuente de luz adicional. La cámara web se colocó de manera perpendicular con respecto a la banda transportadora, las pruebas realizadas se llevaron a cabo en el intervalo 28 cm a 50 cm de distancia entre la cámara web y la banda transportadora. En este rango se identifica de manera correcta cada tarjeta electrónica.

El formato de las imágenes adquiridas por medio de la cámara web se muestra en la tabla 2. La separación entre cada tarjeta electrónica es de 10 cm.

Tabla 2.- Especificaciones de la imagen adquirida por la cámara web.

Descripción	Valor
Formato de la imagen	RGB
Resolución en pixeles	320 x 240

Las especificaciones técnicas de la banda transportadora se muestran en la tabla 3, la velocidad de desplazamiento utilizada es de 0.4 m/s.

Tabla 3.- Especificaciones técnicas de la banda transportadora.

Descripción	Valor
Fabricante	Rexroth
Modelo	Variow Flow S
Ancho de la cinta transportadora	12 cm
Velocidad de desplazamiento	0.1-0.8 m/s

II.A. Algoritmo de aprendizaje automático para reconocimiento de características críticas

En esta sección se aborda la identificación de las características críticas de una imagen utilizando los algoritmos de aprendizaje automático [3].

Por medio del software de Matlab se realiza la identificación de seis tarjetas electrónicas de diferente modelo, tal como se muestran en la figura 3.

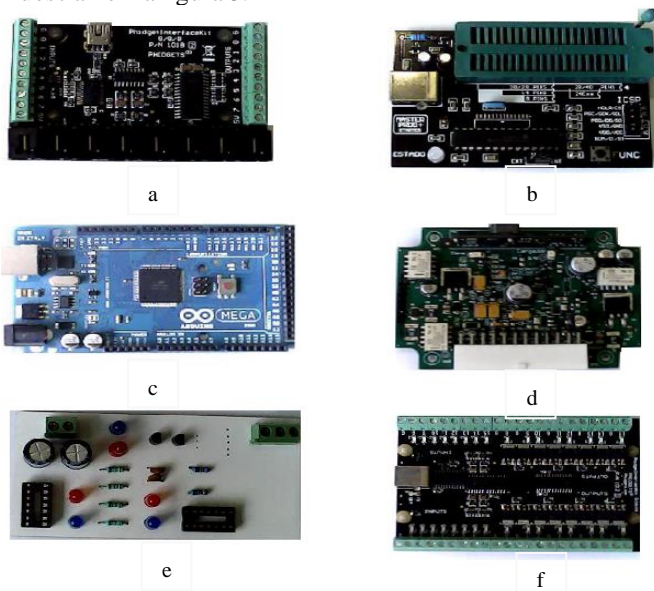


Figura 3.-Tarjeta a) Pinterface, b) Programador, c) Mega, d) Motum e) Controlador, f) Phidgets.

Para implementar el algoritmo de aprendizaje, se adquieren 500 imágenes en diferente posición de cada tarjeta por medio de un

video, posteriormente se aplica el algoritmo y extrae las 200 características más importantes de las imágenes obtenidas, de esta manera se genera una matriz que contiene la información más relevante de cada tarjeta, en la figura 4 se observa la matriz de información obtenida.

```
* Image set 1: Controlador.
* Image set 2: Mega.
* Image set 3: Motum.
* Image set 4: Phidgets.
* Image set 5: Pinterface.
* Image set 6: Programador.

* Encoding 501 images from image set 1...done.
* Encoding 501 images from image set 2...done.
* Encoding 501 images from image set 3...done.
* Encoding 501 images from image set 4...done.
* Encoding 501 images from image set 5...done.
* Encoding 501 images from image set 6...done.

* Finished encoding images.
```

Figura 4.- Matriz de información de características críticas.

Una vez obtenida la matriz de información se entrenan diferentes algoritmos con la finalidad de identificar correctamente cada una de las tarjetas. Se comparan los resultados obtenidos de cada algoritmo entrenado y se implementa el algoritmo que presenta el mayor índice de eficacia. En la figura 5 se muestra los índices de efectividad de cada uno de los algoritmos entrenados, entre los cuales se encuentran el algoritmo Complex Tree, Medium Gaussian SVM, Quadratic SVM, entre otros. Cada algoritmo cuenta con un principio de estimación y optimización diferente, una descripción más amplia de este tipo de algoritmos se puede observar en la referencia [9].

Data Browser	
▼ History	
Tree	
Complex Tree	98.3%
Tree	
Simple Tree	65.9%
SVM	
Medium Gaussian SVM	100%
SVM	
Quadratic SVM	100%
KNN	
Medium KNN	100%
KNN	
Cosine KNN	100%

Figura 5.- Eficacia de algoritmos entrenados.

Utilizando la matriz de confusión que se muestra en la figura 6 se observa el porcentaje de efectividad del algoritmo entrenado, se muestra en el valor de porcentaje la posibilidad de poder identificar correctamente cada una de las tarjetas anteriormente mencionadas.

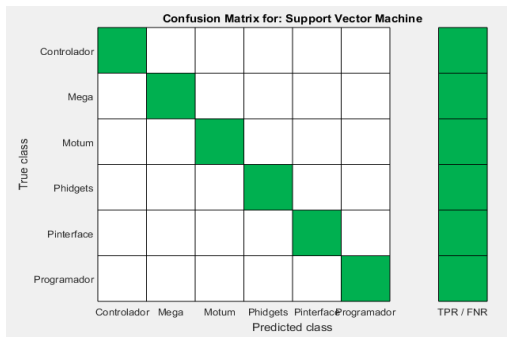


Figura 6.- Matriz de confusión.

II.B Algoritmo de búsqueda de componentes físicos

Para la implementación del algoritmo de búsqueda de componentes físicos dentro de la tarjeta electrónica es necesario realizar un mapeo de la imagen, con la finalidad de realizar una conversión de unidades de medida, es decir, interpretar la imagen en cm y no en píxeles. Posteriormente se realiza la búsqueda de cada componente físico dentro de la tarjeta electrónica, empleando para ello los archivos *gerber* correspondientes a cada tarjeta.

La metodología para la implementación del algoritmo de búsqueda se muestra en la figura 7. Se realiza la conversión de una imagen RGB en su representación binarizada, al obtener esta última imagen se implementa la lectura del archivo *gerber* para obtener las coordenadas de cada uno de los componentes que integran la tarjeta, posteriormente se analiza si existe un componente en la ubicación mencionada.

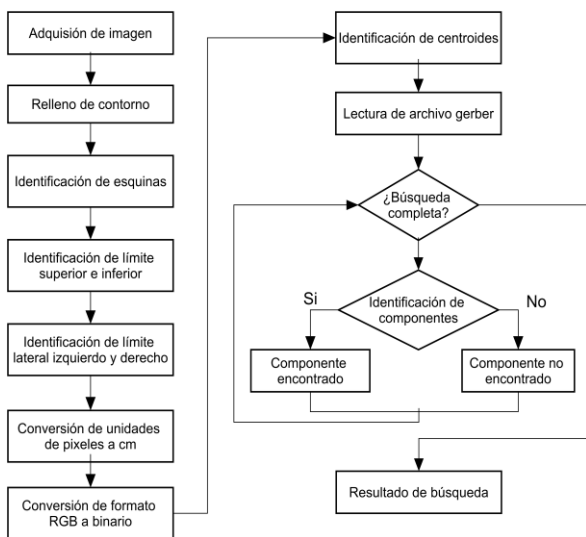


Figura 7.- Diagrama de flujo de algoritmo de búsqueda de componentes electrónicos.

Una vez que se realiza la lectura completa del archivo *gerber*, el algoritmo indica la cantidad y el nombre de cada componente faltante, el nombre que se muestra es de acuerdo a la descripción del componente electrónico.

En la siguiente sección se muestran los resultados obtenidos al implementar los dos algoritmos citados anteriormente.

III. RESULTADOS

Las pruebas de reconocimiento de cada una de las tarjetas electrónicas son mostradas a continuación. Se debe observar el comportamiento del algoritmo entrenado en conjunto con la cámara web. La primera prueba que se realiza es observar la tarjeta “Controlador” por medio de la cámara web, en la figura 8 se muestra que el algoritmo reconoce la tarjeta de manera inmediata, en la parte inferior de la figura se muestra una gráfica de probabilidad de identificación, para este caso la probabilidad es del 100%, por la tanto en la parte superior derecha se inserta una etiqueta con el nombre de la tarjeta identificada.

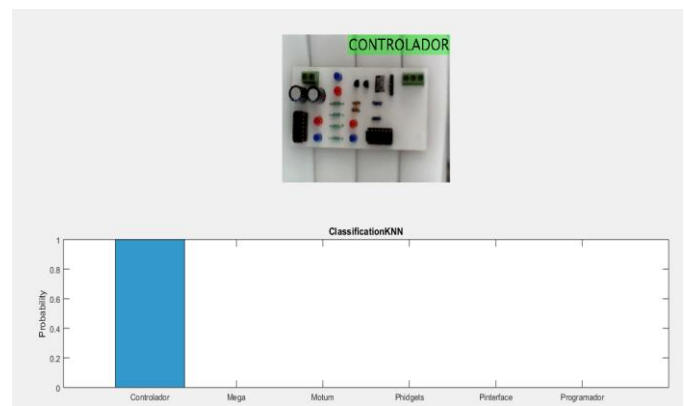


Figura 8.- Reconocimiento de tarjeta Controlador.

En la figura 9 se implementa la tarjeta Arduino Mega como objeto de prueba, el algoritmo reconoce la tarjeta de manera inmediata, en este caso la probabilidad de eficacia es del 90%.

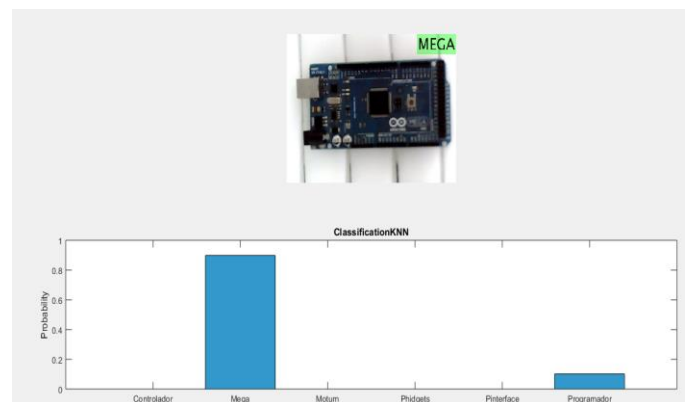


Figura 9- Reconocimiento de tarjeta Arduino Mega.

En la figura 10 se implementa la tarjeta Programador como objeto de prueba, el algoritmo identifica la tarjeta de manera eficaz, con un índice de probabilidad del 100%.

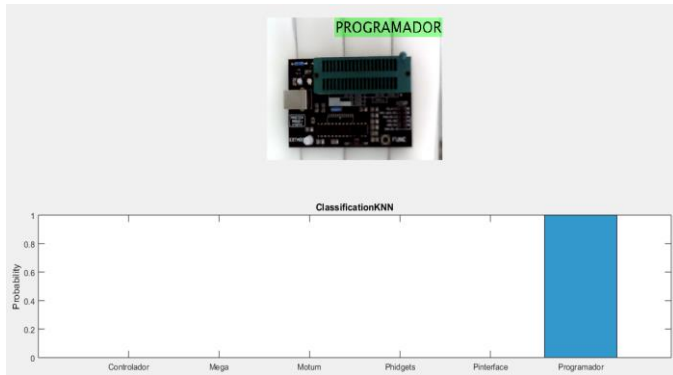


Figura 10- Reconocimiento de tarjeta Programador.

En la figura 11 se implementa la tarjeta Motum como objeto bajo prueba, el algoritmo indica una probabilidad del 100%, que corresponda a la tarjeta Motum.

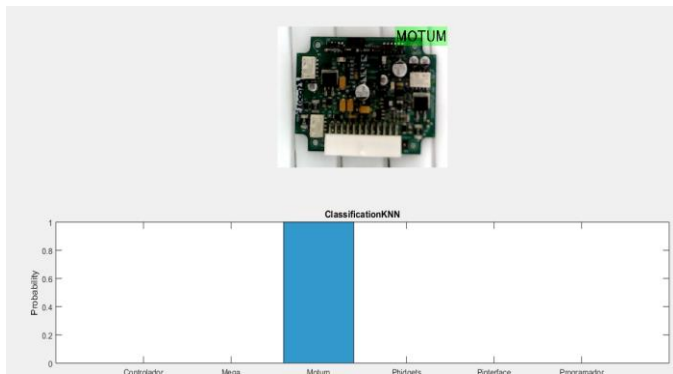


Figura 11- Reconocimiento de tarjeta motum.

El reconocimiento de las tarjetas electrónicas restantes se realiza de manera correcta.

Cuando la distancia entre la cámara web y la banda transportadora es menor a 28 cm, el algoritmo puede llegar a identificar de manera incorrecta la tarjeta electrónica.

Considerando como referencia la tarjeta electrónica que se muestra en la figura 12, se implementa el algoritmo de reconocimiento de componentes. En la figura 12 se muestran los nombres de cada uno de las componentes electrónicas que integran a la tarjeta.

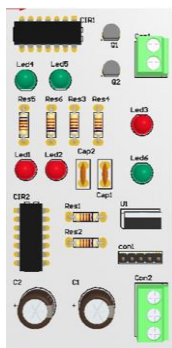


Figura 12- Tarjeta de referencia.

El proceso de identificación de centroides del algoritmo de búsqueda que se muestra en la figura 7 consiste en identificar cada uno de los centroides de los componentes electrónicos por medio de diferentes filtros digitales [10]. El resultado de esta sección del algoritmo se muestra en la figura 13.

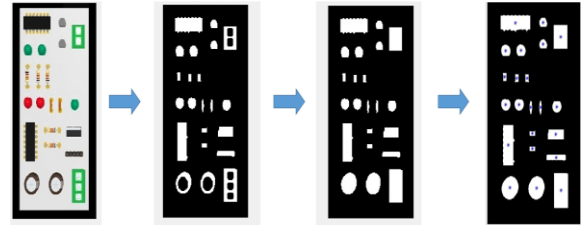


Figura 13- Implementación del algoritmo de búsqueda de centroides en una tarjeta electrónica.

Posteriormente se realiza la lectura del archivo *gerber* para obtener las coordenadas de cada uno de los componentes, el archivo *gerber* es obtenido utilizando el software de diseño Altium Designer. En la figura 14 se muestra de manera representativa la interpretación de este archivo.

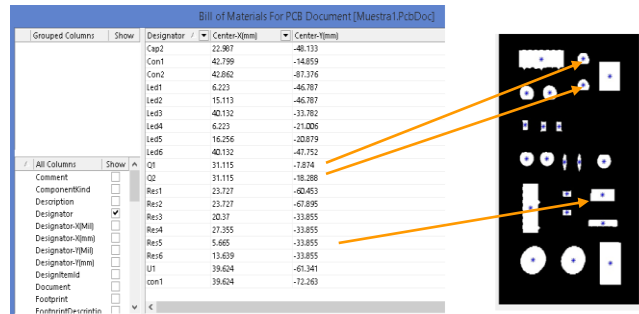


Figura 14- Interpretación de archivos *gerber*.

Se realizan diferentes pruebas para validar el correcto funcionamiento del algoritmo de búsqueda de componentes electrónicos por medio de una interfaz, en primera instancia se implementa una tarjeta electrónica que contiene todos sus componentes. En la figura 15 se muestra la tarjeta real de lado izquierdo y la imagen de la tarjeta obtenida del lado derecho, el algoritmo indica que la tarjeta electrónica se encuentra completa, tal como se observa en el mensaje de la parte superior derecha.

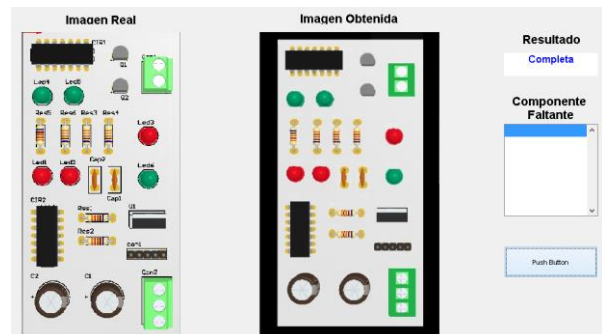


Figura 15- Algoritmo de búsqueda con tarjeta completa.

En la segunda prueba se retira el transistor Q2, el cual se encuentra localizado en la parte superior derecha de la tarjeta; el algoritmo indica que la tarjeta se encuentra incompleta e identifica el componente faltante, el cual corresponde al transistor Q2, tal como se muestra en la figura 16.

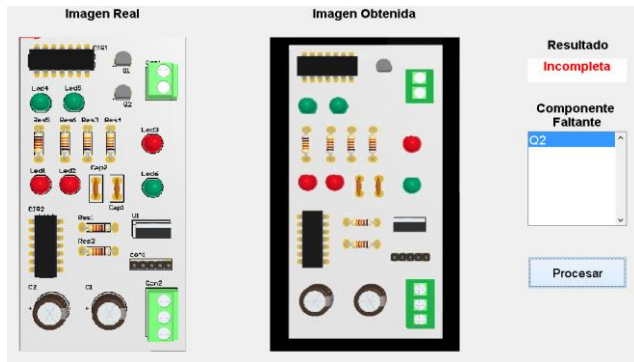


Figura 16- Ausencia de componente Q2.

En la siguiente prueba se retira el Led3, corresponde a un led de color rojo, ubicado en la parte superior derecha, en conjunto con el componente Res4, corresponde a la última resistencia a un costado del Led3. Al implementar el algoritmo, este indica que la tarjeta se encuentra incompleta y al mismo tiempo menciona la ausencia de estos dos componentes. En la figura 17 se observa el listado de los nombres de los componentes ausentes.

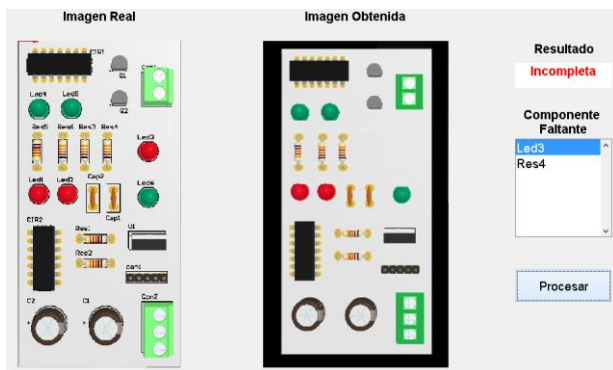


Figura 17- Ausencia de componente Led3 y Res4.

Este algoritmo tiene la capacidad de reconocer de manera correcta los componentes electrónicos, algunos de los problemas que se pueden presentar es cuando se generan proyecciones de sombra sobre los componentes cercanos, para solucionar este problema es necesario contar con una fuente externa de luz. Este algoritmo funciona de manera correcta en tarjetas electrónicas que estén formadas por componentes de montaje superficial.

IV. CONCLUSIONES

Se describió un algoritmo de aprendizaje automático capaz de reconocer diferentes tipos de tarjetas electrónicas, así como

sus componentes electrónicos. De esta manera se observan que los dos algoritmos desarrollados funcionan de manera correcta. El primer algoritmo cuenta con la capacidad de reconocer de manera automática cualquier tipo de tarjeta electrónica sin importar el color, posición, ángulo de la cámara para la captura del fotograma y el nivel de luminosidad, solo es necesario entrenar al algoritmo con los objetos que se desean identificar. El trabajo en conjunto de los dos algoritmos tienen la capacidad de realizar la automatización de procesos de manufactura de tarjetas electrónicas, reduciendo el costo de ensamble, costo de producción y costo de equipo especializado. Por lo tanto se genera una nueva aplicación de los algoritmos con sistema de aprendizaje automático. Esta metodología permite la implementación de otros tipos de aplicaciones muy variadas; entrenando el algoritmo correctamente se puede implementar en el sector de salud para el reconocimiento de patologías presentes en los electrocardiogramas.

V. AGRADECIMIENTOS

Los autores agradecen al CONACYT, a la Benemérita Universidad Autónoma de Puebla-Facultad de Ciencias de la Electrónica, al Instituto Politécnico Nacional, a la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco, por el apoyo brindado.

VI. REFERENCIAS

- [1] Ziyin L. y Qi Y., System design for PCB defects detection based on AOI technology, *4th International Congress on Image and Signal Processing, China*, pp 1- 4, 2011.
- [2] Huibin Z., Jun C. y Jianxun J., NI Vision Based Automatic Optical Inspection (AOI) for Surface Mount Devices: Devices and Method, *Proceedings of 2009 IEEE International Conference on Applied Superconductivity and Electromagnetic Devices China*, pp 25-27, 2009.
- [3] Berral J., A Quick View on Current Techniques and Machine Learning Algorithms for Big Data Analytics, *International Conference on Transparent Optical Networks (ITCON)*, Barcelona, pp 1-4, 2016.
- [4] *Ebook MathWorks Introducing Machine Learning*, Matlab, pp 2-11, 2016.
- [5] Manan A., Khan F. y Changsoo H., Estimation of Desired Motion Intention using Extreme Learning Machine for Upper Limb Assist Exoskeleton, *IEEE International Conference on Advanced Intelligent Mechatronics*, Canada, pp 1-5, 2016.
- [6] Faruk O., Ferhat U. y Korkmaz D., Extreme Learning Machine Based Robotic Arm Modeling, *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp 1-4, 2016.
- [7] Rau H., Wu C., Shiang W. y Huang P., A Decision Support System of Statistical Process Control for Printed Circuit

Board Manufacturing, *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, Qingdao, pp 1-5, 2010.

- [8] Reyes F., Cid J., Limón M. A. y Cervantes M., Square Root Type-Control for Robot Manipulators, *International Journal of Advanced Robotic Systems*, pp 1-7, 2012.
- [9] *Ebook Statistics and Machine Learning Toolbox User's Guide*, Matlab, pp 5-8708, 2016.
- [10] *Ebook Computer Vision System Toolbox User's Guide*, Matlab, pp 25-454, 2015.



Olga Félix Beltrán. Doctora en Ciencias (física). Profesora investigadora de la Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica. Desde el año 2005 ha impartido cursos de físico-matemáticas a nivel licenciatura y posgrado. Miembro del Sistema Nacional de Investigadores (Nivel I), cultiva las líneas de investigación y generación del conocimiento Física de Altas Energías y sus Aplicaciones y Física-matemáticas: modelado dinámico y sistemas no lineales. Es autora de 15 artículos indizados y más de 40 artículos arbitrados productos de sus investigaciones, ha dirigido 26 tesis de maestría y licenciatura, cuenta con 3 solicitudes de patente en trámite.

VII. BIOGRAFÍA



Edgar Hugo Mayoral Arzaba. Realizo su licenciatura en Electrónica en el Instituto Tecnológico de Orizaba en 2013, estudiante de maestría de la Facultad de Ciencias de la Electrónica de la Universidad Autónoma de Puebla, México.

Experiencia laboral en centros de investigación y desarrollo de tecnología, líder de proyectos de automatización de procesos industriales, programación de sistemas embebidos y sistemas de visión.



Jaime Cid Monjaraz. Doctorado en Ingeniería Mecatrónica. El actualmente labora en la Benemérita Universidad Autónoma de Puebla, es Investigador en la Facultad de Ciencias de la Electrónica. Profesor en la BUAP desde 1981 a la fecha impartiendo los cursos en las áreas de Matemáticas, Ciencias Computacionales, Electrónica, Robótica, Control, Control Digital y

Control Moderno. Es autor de 3 libros, con más de 100 publicaciones científicas, en congresos nacionales e internacionales y revistas, asesor en más de 50 trabajos de tesis profesionales de licenciatura y maestría, tiene 9 solicitudes de patente, su línea de investigación es en Visión y Robótica.

Dr. Cid es senior member del IEEE y miembro de distintas asociaciones académicas y científicas. Ha sido Director de la Facultad de Ciencias de la Electrónica de la BUAP, Director de Divulgación Científica de la Vicerrectoría de Investigación y Estudios de Posgrado de la BUAP, Coordinador de Transferencia de Tecnología, miembro del Padrón de Investigadores de la VIEP y del Padrón de Consultores de la BUAP.



Fernando Reyes Cortés. En 1984 obtiene el grado de Licenciado en Electrónica, en la Facultad de Ciencias Físico-Matemático de la BUAP, en 1990 realiza la Maestría en Ciencias con Especialidad en Electrónica en el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) y el grado de Doctor en Ciencias en Electrónica y Telecomunicaciones lo obtiene en 1997 en el Centro

de Investigación Científica y de Educación Superior de Ensenada (CICESE).

Pertenece al sistema nacional de investigadores desde 1993, desde 1980 se encuentra laborando en la BUAP. Es titular de los cursos de robótica de las carreras de Ingeniería Mecatrónica y Licenciatura en Electrónica, así como en el Posgrado de Automatización de la Facultad de Ciencias de la Electrónica. Autor de 4 libros y múltiples artículos nacionales e internacionales, ha titulado a más de 150 alumnos de los niveles de ingeniería/licenciatura, maestría y doctorado.

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO



Congreso Nacional
INGENIERÍAS^{sp}
Formando entendimiento entre sociedad
ingeniería y medio ambiente

EL INSTITUTO TECNOLÓGICO DE ORIZABA
OTORGA EL PRESENTE

RECONOCIMIENTO

A:

ING. EDGAR MAYORAL ARZABA

por su destacada participación como

INSTRUCTOR

En el curso denominada:

"DISEÑO PCB CON ALTIUM"

En el marco del 7° Congreso Nacional de Ingenierías
Octubre 2016



Ing. Rogelio García Camacho
DIRECTOR



EXTENDS ITS APPRECIATION

Edgar Hugo Mayoral Arzaba

For the lecture:

Automatización de una Celda Robótica usando algoritmos de Aprendizaje Automático e Internet de las cosas

in the "Mexican Humanitarian Technology Conference (MHTC) 2017"

Puebla, Mexico, March 31, 2017



Dr. German A. Muñoz-Hernandez
IEEE Puebla Section Chair

Handwritten signature of German A. Muñoz-Hernandez in blue ink.

Dr. Jaime J. Cid-Monjaraz
Organizing Committee MHTC 2017 Chair

Handwritten signature of Jaime J. Cid-Monjaraz in blue ink.

Automation of a robotic cell using machine learning algorithms and Internet of things

E.H. Mayoral Arzaba*, O. Félix Beltrán, J. Cid Monjaraz, F. Reyes Cortés
 Benemerita Universidad Autonoma de Puebla
 Facultad de Ciencias de la Electrónica
 Email:edgarhugo130590@hotmail.com*

Abstract—In recent years the imagen processing is an involving issue, allows a great variety of applications. This article describe the process of automation of a conveyor belt using a machine learning algorithms that is able to immediately recognize each type different electronic board. For image processing a webcam is used. The obtained information is sent a free server of a web page by a WiFi module. Is obtained an automatic and intelligent process of low cost of robotic cell.

Keywords- *Automatation, conveyor belt, learning machie algorithms, WiFi.*

I. INTRODUCCIÓN

En el proceso de manufactura de tarjetas electrónicas se emplean bandas transportadoras, sobre estas son colocadas las tarjetas electrónicas y máquinas especializadas denominadas AOI (*Automated Optical Inspection*, por sus siglas en inglés), que se utilizan para detectar defectos en una PCB (*Printed Circuit Board*, por sus siglas en inglés) [1]. Por medio de una cámara de alta definición se realiza de manera automática el escaneo del dispositivo bajo prueba con el propósito de validar que no posea ningún defecto, como componentes faltantes, mal orientados, incorrectos, entre otros. Es necesario configurar la máquina AOI para cada tipo de tarjeta electrónica. El costo de elaboración de tarjetas electrónicas resulta ser demasiado elevado. El objetivo es reducir el costo de producción de las tarjetas electrónicas empleando tecnologías actuales y de bajo costo.

Una parte importante de la investigación en el sector de las ciencias computacionales es el desarrollo de nuevas técnicas que permitan realizar el procesamiento de imágenes con una mayor eficacia.

El aprendizaje automático tiene como enfoque obtener la información más relevante a partir de una gran cantidad de datos obtenidos de un objeto [2]. Se encuentra constituido por una serie de algoritmos computacionales que permiten identificar las características más importantes sin la necesidad de contar con un modelo matemático o una ecuación predeterminada [3]. Existe una amplia variedad de aplicaciones donde se implementan estos tipos de algoritmos, por ejemplo, para la estimación de movimiento de un exoesqueleto asistido [4], modelado de un brazo robótico por aprendizaje automático [5], algoritmo de soporte de decisión para el proceso de control de manufactura de circuitos impresos [6]. Este último ejemplo realiza la identificación de fallas en las líneas de conducción de la PCB, utilizando una máquina AOI.

Por otro lado, en los últimos años ha surgido un área de interés denominada el IoT (*Internet of things*), es un concepto que refiere a la interconexión de objetos inteligentes que trabajan en conjunto conectados a la Internet [7]; algunas de las aplicaciones más utilizadas se encuentran en el área de control [8].

En este trabajo se implementa la automatización de un banda transportadora, empleando algoritmos de procesamiento digital de imágenes, algoritmos de aprendizaje automático y la metodología del IoT. Se describe un algoritmo que permite identificar diferentes tipos de tarjetas electrónicas de manera automática sin la necesidad de modificar parámetros en el algoritmo. La información que se genera es enviada a una pagina Web por medio de un módulo WiFi.

Los puntos relevantes de este trabajo son la sustitución de una máquina AOI por una cámara web y el desarrollo electrónico para la automatización

de la banda transportadora, utilizando tecnologías actuales y de bajo costo.

II. DISEÑO Y DESARROLLO

Se implementa una celda robótica que está constituida por una banda transportadora, un brazo robótico, una cámara web y un sistema de adquisición de datos [9], tal como se muestra en la figura 1.

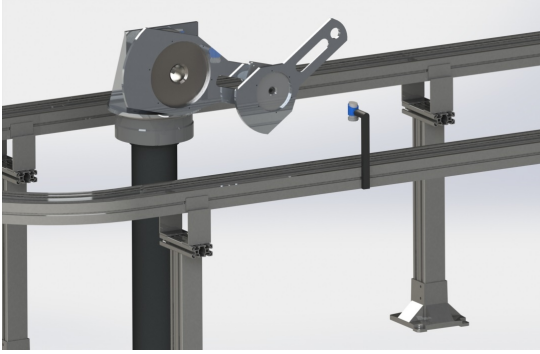


Fig. 1: Celda robótica.

Las tarjetas electrónicas se encuentran colocadas sobre la banda transportadora, la cual se mueve a una velocidad constante. Se desarrolla una interfaz que interactúa con la cámara Web y los algoritmos de aprendizaje automático para identificar una tarjeta electrónica en específico. Cuando la tarjeta es reconocida, la banda transportadora se detiene y el ROTRADI retira la tarjeta. La interfaz contiene un sistema de control de temperatura para el motor de la banda transportadora, cuando la temperatura del motor es mayor de 40°C la banda transportadora se detiene. La información generada por la celda robótica es enviada al servidor de *ThingSpeak* cada cinco minutos por medio de un dispositivo de comunicación WiFi.

En la tabla 1 se muestran las especificaciones técnicas de la banda transportadora y la cámara web que se utilizó para el desarrollo de este proyecto.

Descripción	Valor
Fabricante de la banda transportadora.	Rexroth
Banda transportadora.	Variow Flow S
Ancho de la banda transportador.	12 cm
Ancho de la banda transportador.	12 cm
Velocidad de desplazamiento	0.1-0.8 m/s
Resolución de la cámara web	5 Mpx
Velocidad del fotograma	30 Fps
Formato de la imagen adquirida	RGB
Resolución en pixeles	320 x 240
Robot manipulador	ROTRADI 6
Grados de libertad	3

Tabla I: Especificaciones técnicas de la celda robótica.

A. Modelo dinámico de un robot manipulador de tres grados de libertad

El modelo dinámico de un robot manipulador de n grados de libertad se presenta en la ecuación 1, en su forma compacta y con la notación más ampliamente utilizada en el área de la robótica [10].

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f_f(\dot{q}, f_e) \quad (1)$$

donde $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ son vectores de posición, velocidad y aceleración, respectivamente; $M(q) \in \mathbb{R}^{n \times n}$ corresponde a la matriz de inercia y es simétrica y definida positiva; $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ es la matriz de fuerzas centrípétas y de Coriolis; $g(q) \in \mathbb{R}^n$ es el vector fuerzas gravitacionales obtenido de la energía potencial y $f_f(\dot{q}, f_e) \in \mathbb{R}^n$ es el vector de pares de fricción que incluye la fricción viscosa, de Coulomb y estática de cada articulación [10].

Se utilizó un ROTRADI (robot de transmisión directa) de tres grados de libertad, el cual se encuentra en el Laboratorio de Robótica de la Facultad de Electrónica de la BUAP.

Para el control de posición del brazo robótico se proponen estructuras de control para el moldeo de energía. Se evalúan dos diferentes tipos de estructuras de control. La primera corresponde al controlador proporcional derivativo más compensación de gravedad [11]:

$$\tau = K_p \tilde{q} - K_v \dot{q} + g(q) + f(\tau, \dot{q}) \quad (2)$$

donde $\tilde{q} \in \mathbb{R}^n$ es el vector de error de posición que se define como la diferencia entre la posición deseada q_d y la posición actual del robot q ; $K_p \in \mathbb{R}^{n \times n}$ es una matriz definida positiva; $K_v \in \mathbb{R}^{n \times n}$ es la ganancia derivativa la cual es una matriz definida positiva; $g(q)$ corresponde a la compensación de gravedad.

La segunda estructura de control corresponde a un controlador con acciones acotadas, tal que el error de la posición \tilde{q} tienda asintóticamente a cero, independientemente de las condiciones iniciales del robot. El control con acciones acotadas es un algoritmo particular que se relaciona con el problema de saturación de robots manipuladores. En este caso la energía aplicada al robot se encuentra dentro de los límites de diseño, los cuales se dan a continuación:

$$\tau = \begin{cases} \tau_{max} & \text{si } \tau \geq \tau_{max} \\ \tau & \text{si } -\tau_{max} < \tau < \tau_{max} \\ -\tau_{max} & \text{si } \tau \leq -\tau_{max} \end{cases} \quad (3)$$

La estructura de control implementada corresponde al algoritmo tangente hiperbólico más compensación de gravedad [10], dado como:

$$\tau = k_p \tanh(\epsilon \tilde{q}) - k_v \tanh(\gamma \dot{q}) + g(q) \quad (4)$$

donde las ganancias proporcional y derivativa se encuentran representadas por $k_p, k_v \in \mathbb{R}_+$, respectivamente; $\epsilon, \gamma \in \mathbb{R}_+$ son las pendientes del control tangente hiperbólica [10].

B. Reconocimiento de tarjetas electrónicas

Se desarrolla una interfaz de usuario en Matlab que permita realizar la integración de los algoritmos de detección de movimiento y el algoritmo de aprendizaje automático.

Las tarjetas electrónicas son colocadas sobre la banda transportadora, posteriormente se activa la banda transportadora y la cámara web por medio de la interfaz. Por medio de la cámara web y un conjunto de algoritmos se detectan las tarjetas electrónicas. El primer algoritmo corresponde al procesamiento digital de imágenes utilizando la cámara web, que tiene como objetivo localizar una imagen semejante a una tarjeta electrónica. Cuando se detecta una área corresponde a una tarjeta electrónica, se detiene la banda transportadora y se ejecuta el algoritmo de aprendizaje automático, este algoritmo contiene un clasificador, el cual contiene información de cada una de las tarjetas electrónicas que se desean identificar. La información que se genera es enviada a una página web utilizando el módulo ESP8266 en conjunto con la tarjeta Arduino Mega, empleando el concepto del *Internet de las cosas*.

1) *Algoritmo de detección de movimiento:* El algoritmo de detección de movimiento consiste en configurar la cámara Web para realizar la adquisición de dos imágenes consecutivas de un mismo lugar sobre la banda transportadora. Estas imágenes son procesadas a imágenes binarias, posteriormente se realiza una diferenciación entre ambas y se obtiene una imagen que contiene solo los pixeles diferentes entre cada una de ellas, se realiza una suma de pixeles de esta imagen; en caso de que el valor obtenido sea mayor al valor de referencia se indica la presencia de movimiento.

2) *Algoritmo de aprendizaje automático:* En esta sección se aborda la identificación de las características críticas de una imagen utilizando los algoritmos de aprendizaje automático [12]. Utilizando el software de Matlab y el algoritmo propuesto por Paul Viola [13], en conjunto con el algoritmo de identificación de rostros en tiempo real [14], se realiza la identificación de cada tarjeta electrónica. Se extraen las características más importantes de cada tarjeta por medio del algoritmo de aprendizaje automático [15].

El algoritmo de aprendizaje automático consiste en extraer las características más importantes de una imagen. Las características que se buscan dentro de la imagen son: cambios de contrastes, cambio de contornos y cambios de intensidad, cada característica encontrada se denomina *punto local*. Los puntos locales son almacenados dentro de un vector. Cuando existe más de un objeto que se desea identificar se obtiene un vector de cada objeto y se forma una matriz. Una vez obtenida esta matriz se ejecuta el algoritmo de aprendizaje automático en una segunda imagen diferente la cual contiene el objeto a buscar. El algoritmo intenta identificar los puntos locales dentro de la imagen, en caso de ser identificados, se implementa un filtro digital, este tiene como finalidad solo considerar aquellos puntos locales que corresponda a la imagen deseada. Se realiza un promedio de los puntos locales filtrados y en caso de que el promedio sea mayor del 60% del total de puntos se considera que la imagen ha sido identificada. En la figura 2 se muestra el proceso descrito anteriormente.

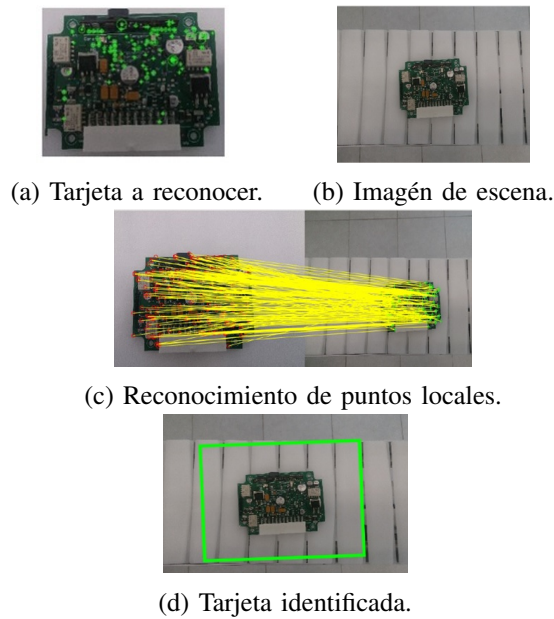


Fig. 2: Proceso de reconocimiento aplicando el algoritmo de aprendizaje automático.

Una vez obtenida la matriz de información, se entrenan diferentes algoritmos con la finalidad de identificar correctamente cada una de las tarjetas. Se comparan los resultados obtenidos de cada algoritmo entrenado y se implementa el algoritmo que presenta el mayor índice de eficacia. En la figura 3 se muestra los índices de efectividad de cada uno de los algoritmos entrenados, entre los cuales se encuentran los algoritmos *Complex Tree*, *Medium Gaussian SVM*, *Quadratic SVM*, entre otros. Cada algoritmo cuenta con un principio de estimación y optimización diferente, una descripción más amplia de este tipo de algoritmos se puede observar en la referencia [16].

Data Browser	
▼ History	
Tree	
Complex Tree	98.3%
Tree	
Simple Tree	65.9%
SVM	
Medium Gaussian SVM	100%
SVM	
Quadratic SVM	100%
KNN	
Medium KNN	100%
KNN	
Cosine KNN	100%

Fig. 3: Evaluación de desempeño de algoritmos de aprendizaje automático.

C. Internet de las cosas

El *Internet de las cosas* corresponde a una red que interconecta objetos físicos por medio del Internet. Este concepto fue propuesto por Kevin Ashton en el Auto-ID Center del MIT en 1999. El concepto radica en realizar la conexión de diferentes dispositivos que cuenten con un sistema embebido y la capacidad de conectarse a Internet. Se implementó la metodología del *Internet de las cosas* en la banda transportadora para contar con un sistema automatizado, con la finalidad de enviar información a una página de Internet y pueda ser consultado desde cualquier sitio, sin la necesidad de tener contacto directo con el proceso. Para el desarrollo de este sistema se utilizó el módulo ESP8266, este corresponde a un módulo WiFi que utiliza el estándar de comunicación 802.11. Para establecer la comunicación entre la interfaz (ver figura ??) y el módulo WiFi, se utiliza la tarjeta Arduino Mega y se desarrolla una tarjeta de interconexión (ver figura 4). El módulo ESP8266 se configura utilizando comandos AT (lenguaje desarrollado por la compañía *Hayes Communications*).

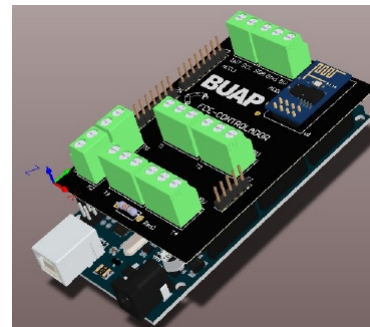


Fig. 4: Tarjeta de comunicación entre módulo ESP8366 y Arduino Mega.

Para mandar la información obtenida por el dispositivo, que se muestra en la figura 4, hacia una página web, es necesario contar con un servidor que se encuentre disponible en Internet. Actualmente existen diferentes servidores gratuitos, como por ejemplo *ThingSpeak*, *Adafruit* y *Ubidots*. Estos permiten establecer conexión con diferentes dispositivos, utilizando la metodología del *Internet de las cosas*. Para el desarrollo de este trabajo se utiliza la plataforma *Thingspeak*, que es un servidor gratuito, su plataforma de desarrollo está basado en MATLAB, tal como se muestra en la figura 5.

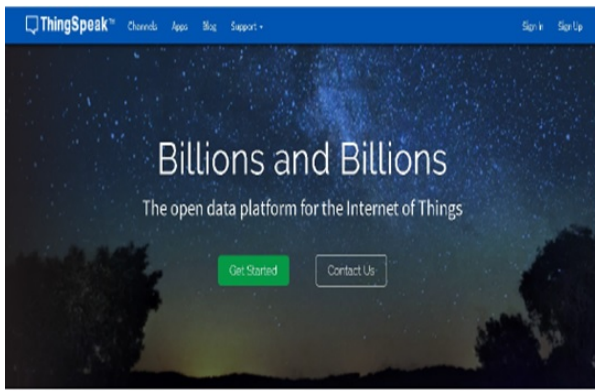


Fig. 5: Servidor Thingspeak.

Para enviar información hacia este servidor es necesario contar con API Keys de escritura y de lectura, corresponde a números de series para poder identificar el dispositivo que realiza la conexión, estos comandos son enviados utilizando el módulo ESP8266, en la figura 6 se muestra los API Keys proporcionados por el servidor. En caso de no enviar correctamente el API Key el servidor no reconoce la información recibida.

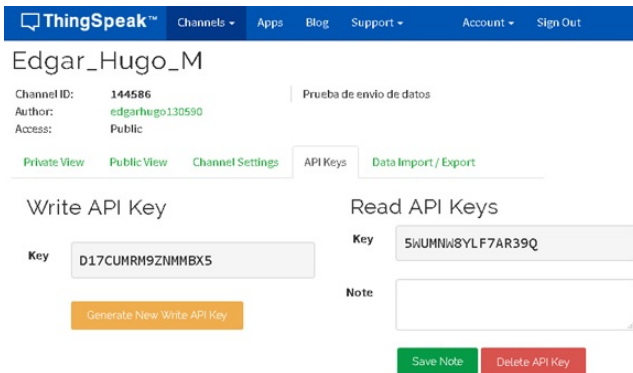


Fig. 6: API Keys.

III. RESULTADOS

En esta sección se muestran los resultados obtenidos del trabajo desarrollado. Se implementa la interfaz que contiene el algoritmo de detección de movimiento, el algoritmo de aprendizaje automático realiza la comunicación con la cámara web, establece el sistema de control de la banda transportadora, se implementa con el ESP8266 para establecer la comunicación con el servidor web y por último establece el algoritmo de control de posición del ROTRADI.

Cuando se inicia la interfaz, se activa la cámara y en automático se ejecuta el algoritmo de detección

de movimiento. Este contiene el filtro digital para descartar regiones pequeñas, cuando se identifica una tarjeta electrónica se envía una señal a la tarjeta Arduino Mega para detener la banda transportadora. Posteriormente se activa el algoritmo de aprendizaje automático para identificar la tarjeta electrónica. Una vez que es identificada la tarjeta se manda esta información a la página web de *ThingSpeak*, en la cual se visualiza la cantidad y el tipo de tarjetas electrónicas que se encuentran en la banda transportadora. Cuando este proceso ha finalizado se manda un comando a la tarjeta Arduino Mega para activar la banda transportadora y el algoritmo de detección de movimiento se vuelve a ejecutar.

La interfaz muestra tres gráficas diferentes: en la primera se visualiza la imagen de la cámara web en tiempo real, la segunda gráfica muestra cuando una tarjeta es identificada, en la última gráfica se da la temperatura del motor de la banda transportadora. Cuando la temperatura es mayor a 40°C se desactiva la banda transportadora debido a un sobrecalentamiento del motor.

En la figura 7 se muestra la interfaz gráfica, detectando la tarjeta Arduino Mega. En el cuadro *Panel* se indica el nombre de la tarjeta identificada y el número de tarjetas que han sido identificadas, en este caso corresponde a la tarjeta Arduino Mega y se han identificado dos piezas en el proceso. Al mismo tiempo se indica la fecha del proceso y la temperatura del motor, la cual corresponde a 23°C .



Fig. 7: Reconocimiento de tarjeta Arduino Mega.

En la figura 8 se muestra la interfaz gráfica identificando la tarjeta Programador. En el cuadro *Panel* muestra el nombre de la tarjeta identificada y el número de tarjetas que han sido identificadas, en

este caso corresponde a la tarjeta Programador y se ha identificado dos tarjetas del tipo Programador y dos tarjetas Arduino Mega, la temperatura del motor es aproximadamente de 23°C.

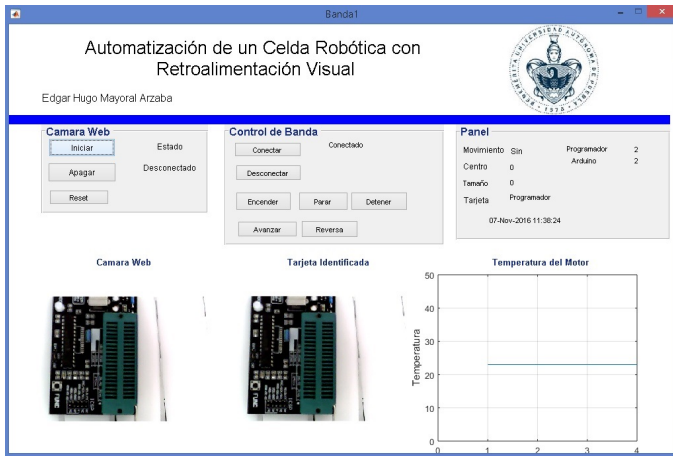


Fig. 8: Detección de tarjeta Programador.

Se colocan diferentes tarjetas en la banda transportadora y se visualiza esta información en la página de ThingSpeak, en la figura 9 muestra que se han reportado 4 tarjetas Arduino Mega y 4 tarjetas Programador, al mismo tiempo que la temperatura registrada del motor se encuentra entre 24 °C y 25 °C.

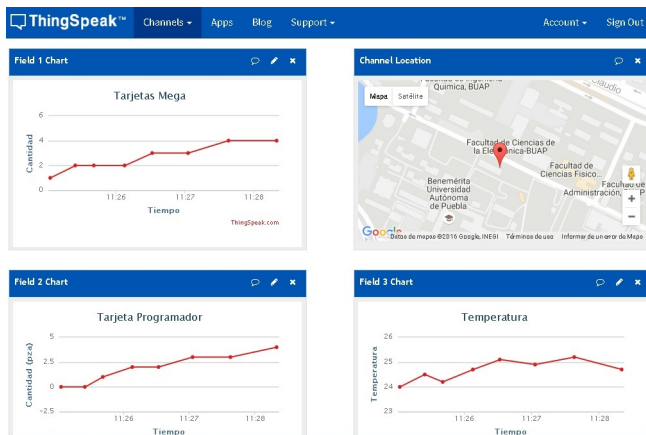


Fig. 9: Interfaz de resultados en página Web.

IV. CONCLUSIONES

Se desarrolló un sistema capaz de poder identificar diferentes tarjetas electrónicas sobre una banda transportadora, empleando nuevas tecnologías como aprendizaje automático e IoT. Se sustituyó la máquina AOI por una cámara Web, para poder identificar las tarjetas electrónicas, ya que este algoritmo

de reconocimiento no se encuentra incluido dentro de la AOI. Así, se obtiene una nueva aplicación de los algoritmos de aprendizaje automático y como resultado se reducen los costos de fabricación de las tarjetas electrónicas, de esta manera se cumple con el objetivo de desarrollar nuevas tecnologías a bajo costo.

El trabajo a futuro de este proyecto es incluir el algoritmo de búsqueda de componentes electrónicos [17] con el desarrollo de este proyecto, obteniendo como resultado un proceso completo de supervisión de tarjetas electrónicas por medio de una cámara web.

REFERENCES

- [1] Ziyin L. y Qi Y., *System design for PCB defects detection based on AOI technology*, 4th International Congress on Image and Signal Processing, China, pp 1- 4, 2011.
- [2] Berral J., *A Quick View on Current Techniques and Machine Learning Algorithms for Big Data Analytics*, International Conference on Transparent Optical Networks (ITCON), Barcelona, pp 1-4, 2016.
- [3] *Ebook MathWorks Introducing Machine Learning, Matlab*, pp 2-11, 2016.
- [4] Manan A., Khan F. y Changsoo H., *Estimation of Desired Motion Intention using Extreme Learning Machine for Upper Limb Assist Exoskeleton*, IEEE, International Conference on Advanced Intelligent Mechatronics, Canada, pp 1-5, 2016.
- [5] Faruk O., Ferhat U. y Korkmaz D., *Extreme Learning Machine Based Robotic Arm Modeling*, 21st International Conference on Methods and Models in Automation and Robotics (MMAR), pp 1-4, 2016.
- [6] Rau H., Wu C., Shiang W. y Huang P., *A Decision Support System of Statistical Process Control for Printed Circuit Board Manufacturing*, Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, pp 1-5, 2010.
- [7] Tongary K., *Sensor Data Computing as a Service in Internet of Things*, Symposium on Colossal Data Analysis and Networking, pp 1-4, 2016.
- [8] Rakesh K., Vijay S. y Pooja H., *Internet of Thing Based Home Appliances Control*, International Conference on Computational Intelligence and Communication Networks, pp 1-5, 2015.
- [9] Reyes Cortés F., Cid J., Limón M. A. y Cervantes M., *Square Root Type-Control for Robot Manipulators*, International Journal of Advanced Robotic Systems, pp 1-7, 2012.
- [10] Reyes Cortés F., *Robótica Control de Robots Manipuladores*. Alfaomega, 2013.
- [11] Cebada Reyes J., Sánchez Sánchez P., Michua Camarillo A. y Reyes Cortés F., *Creation of a simulator to evaluate the performance index of control structures applied to a robot of two degrees of freedom*, pp 1-6, 2008.
- [12] Berral J., *A Quick View on Current Techniques and Machine Learning Algorithms for Big Data Analytics*, International Conference on Transparent Optical Networks (ITCON), Barcelona, pp 1-4, 2016.
- [13] Viola P. y Jones M., *Rapid object detection using a boosted cascade of simple features*, Conference on Computer Vision and Pattern Recognition, pp 1-9, 2001.
- [14] Viola P. y Jones M., *Robust real-time face detection*, International Journal of Computer Vision, pp 1-17, 2004.

- [15] Smola A. y Vishwanthan S., *Introduction to Machine Learning* pp 3-219, 2012.
- [16] Ebook *Statistics and Machine Learning Toolbox User's Guide, Matlab*, pp 5-8708, 2016.
- [17] Mayoral Arzaba E. H., Cid Monjaraz J., Reyes Cortés F. y Felix Beltran O., *Implementation of machine learning algorithm for identifying electronic boards*, 8° Congreso Internacional de Ingeniería Electromecánica y de Sistemas, pp 1-7, 2016.

UTG THEOREMA

REVISTA CIENTÍFICA



ANÁLISIS DE ENGRANES RECTOS DE
PERFIL INVOLUTA CALCULADOS POR EL
MÉTODO DEL ELEMENTO FINITO
Y ESTÁNDARES AGMA <<

MEJORA CONTINUA CON APOYO DE HERRAMIENTAS
LEAN MANUFACTURING. EN LA PRODUCCIÓN
DE VOLANTES AUTOMOTRICES <<

DOS SALTOS CUÁNTICOS CLAVE: LA
CONSTANTE DE PLANCK Y EL ÁTOMO DE BOHR <<

NO. 6

+OCTUBRE 2017
NÚMERO ESPECIAL



IMPLEMENTACIÓN DE ALGORITMO DE APRENDIZAJE AUTOMÁTICO PARA IDENTIFICACIÓN DE TARJETAS ELECTRÓNICAS

Édgar Hugo Mayoral Arzaba¹, Olga Félix Beltrán², Jaime Cid Monjaraz³ y Fernando Reyes Cortés⁴

^{1,2,3,4}Facultad de Ciencias de la
Electrónica, Benemérita
Universidad Autónoma de Puebla,
Edif. 1FCE6, C.U., 18 Sur y Avenida
San Claudio,
Puebla, Puebla, México, C.P. 72570.



Edgarhugo130590@hotmail.com

Recibido: Noviembre 11, 2016.

Recibido en forma revisada: Marzo 18, 2017.

Aceptado: Mayo 22, 2017.

Resumen: El procesamiento de imágenes es un tema en evolución en los últimos años, permite una gran variedad de aplicaciones. En este artículo se describe un algoritmo de aprendizaje automático que es capaz de reconocer de manera inmediata cada tipo de tarjeta electrónica y posteriormente verifica que contiene todos sus componentes electrónicos en la posición correcta. Este proceso depende de la tarjeta electrónica identificada, con la finalidad de poder analizar diferentes tipos de tarjetas sin la necesidad de realizar modificaciones de parámetros o instrucciones en el algoritmo, obteniendo como resultado un proceso automático e inteligente. Para el procesamiento de imágenes se utiliza una cámara web.

+ **Palabras clave:** Algoritmo de aprendizaje automático, tarjeta electrónica y procesamiento de imágenes.

Abstract: In recent years the image processing is an evolving issue, allows a great variety of applications. This article describes a machine learning algorithm that is able to immediately recognize each different type of electronic board, and verify subsequently that contains all electronic components in the correct position. This process depends on the electronic board identified, in order to be able to analyze different types of board without the need to make modifications of parameters or instructions in the algorithm, getting as result an automatic and intelligent process. For image processing a webcam is used.

+ **Keywords:** Machine learning algorithm, electronic board and image processing.

Introducción

En el proceso de manufactura de tarjetas electrónicas se emplean bandas transportadoras, sobre estas son colocadas las tarjetas electrónicas y máquinas especializadas denominadas AOI (Automated Optical Inspection), se utilizan para detectar defectos en una Printed Circuit Board (Ziyin y Qi, 2011), Por medio de una cámara de alta definición se realiza de manera automática el escaneo del dispositivo bajo prueba, con el propósito de validar que no posea ningún defecto, como componentes faltantes, componentes mal orientados y componentes soldados de manera incorrecta. Para realizar el proceso de supervisión utilizando una máquina AOI, es necesario contar con los archivos de fabricación y de ensamble de la PCB denominados archivos Gerber's. Estos contienen información de la posición y orientación de cada componente utilizado sobre la tarjeta electrónica. Es necesario configurar la máquina AOI para cada tarjeta electrónica diferente que se desea supervisar. El costo de elaboración de tarjetas electrónicas resulta ser demasiado elevado. El objetivo es reducir el costo de producción de las tarjetas electrónicas empleando tecnologías actuales y de bajo costo.

En los últimos años se han desarrollado una gran cantidad de algoritmos para las máquinas AOI. En el trabajo presentado por Huibin Zhao (2009) se muestra un dispositivo y un algoritmo basado en visión para el reconocimiento de componentes electrónicos en una PCB de densidad media, utilizando el software Labview en conjunto con la herramienta de NI Vision (National Instruments Vision). El sistema consiste en una cámara IEEE1394, una interfaz y una plataforma móvil de dos grados de libertad. Este sistema presenta dos desventajas notables, el uso de una cámara especial y la necesidad de un hardware externo que permita controlar la posición de la cámara.

Una parte importante de la investigación en el sector de las ciencias computacionales es el desarrollo de nuevas técnicas que permitan realizar el procesamiento de imágenes con una mayor eficacia, por lo que es necesario involucrar otras áreas del conocimiento para obtener mejores resultados. Dada la gran cantidad de información que se obtiene al procesar un conjunto de imágenes, es necesario contar con una herramienta que permita manipular esta cantidad de información y poder obtener los datos más relevantes.

El aprendizaje automático tiene como enfoque obtener la información más relevante a partir de una gran cantidad de datos almacenados en una instancia (Berral García, 2016). Se encuentra constituido por una serie de algoritmos computacionales que permiten identificar las características más importantes sin la necesidad de contar con un modelo matemático o una ecuación predeterminada: la obtención de los datos depende de la cantidad de información disponible (Mathworks, 2016). Existe una amplia variedad de aplicaciones donde se implementa este tipo de algoritmo, por ejemplo para la estimación de movimiento de un exoesqueleto asistido (Manan Khan, Khan y Changsoo, 2016), modelado de un brazo robótico por aprendizaje automático (Faruk Alcin, Ferhat y Korkmaz, 2016) y algoritmo de soporte de decisión para el proceso de control de manufactura de circuitos impresos (Rau, Shiang y Huang, 2010). Este último ejemplo realiza la identificación de fallas en las líneas de conducción de la PCB, como por ejemplo la ruptura de una línea de conducción; emplean una máquina AOI para realizar la inspección de

la tarjeta electrónica.

En este trabajo se implementa un sistema de aprendizaje automático en conjunto con el procesamiento digital de imágenes. Se describe un algoritmo que permite identificar diferentes tipos de tarjetas electrónicas de manera automática, posteriormente se realiza el proceso de supervisión de los componentes electrónicos de cada tarjeta, sin la necesidad de modificar algún parámetro en el algoritmo; de esta manera, se evita que la línea de producción se detenga.

Una de las principales ventajas de este trabajo es que no requiere de una máquina de inspección AOI para realizar el proceso de supervisión de las tarjetas electrónicas. Se sustituye la máquina de inspección AOI por una cámara web, reduciendo el costo de la infraestructura.

Desarrollo

Se implementa una celda robótica que está constituida por una banda transportadora, una cámara web, un sistema de adquisición de datos y las tarjetas electrónicas a supervisar, tal como se muestra en la Figura 1. Las tarjetas electrónicas se colocan sobre la banda transportadora, la cual se encuentra en movimiento con una velocidad constante. Por medio de una cámara web, en conjunto con los algoritmos de aprendizaje automático, se identifica la tarjeta electrónica. Automáticamente se selecciona de una base de datos el archivo Gerber correspondiente a la tarjeta. Posteriormente se realiza el proceso de verificación de los componentes electrónicos de la tarjeta. Las tarjetas que no presenten ningún error son retiradas de la banda transportadora.



Figura 1. Celda robótica, Laboratorio de Robótica, FCE-BUAP.

La implementación del sistema se realizó en el Laboratorio de Robótica en la FCE-BUAP. Este presenta una luminiscencia en el intervalo de 90 lux a 380 lux, no se requirió de una fuente de luz adicional. La resolución de la cámara web es de 320x240 píxeles, se colocó de manera perpendicular con respecto a la banda transportadora, las pruebas realizadas se llevaron a cabo en el intervalo 28 cm a 50 cm de distancia entre la cámara web y la banda transportadora. La separación entre cada tarjeta electrónica es de 12 cm. La banda transportadora es el modelo Variow Flow S de la compañía Rextoh y se utilizó con una velocidad de desplazamiento de 0.4 m/s. En este rango se identifica de manera correcta cada tarjeta electrónica. Para obtener un control de velocidad

sobre la banda transportadora, se realiza la caracterización del motor, para encontrar la frecuencia donde el motor funcione como un sistema lineal y proporcione su máximo torque. En la Figura 2 se muestran las gráficas de la curva de caracterización, donde se observa que con una frecuencia 260 Hz, la respuesta del motor es lineal.

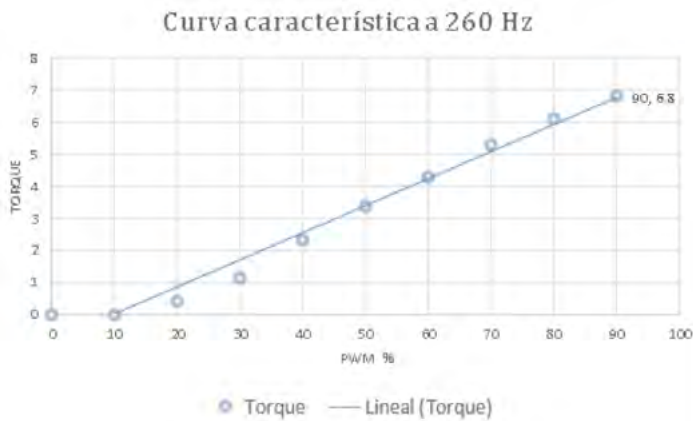


Figura 2. Respuesta de comportamiento del motor a una frecuencia de 260 Hz.

En esta sección se aborda la identificación de las características críticas de una imagen, utilizando los algoritmos de aprendizaje automático (Berral García, 2016). Utilizando el software Matlab se realiza la identificación de cuatro tarjetas electrónicas de diferente modelo, tal como se muestran en la Figura 3.

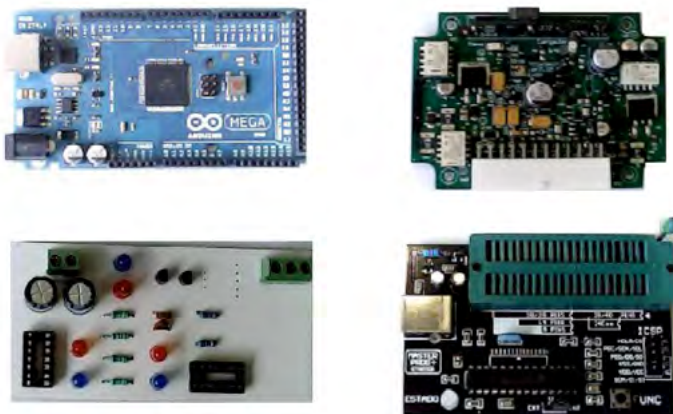


Figura 3. Tarjeta Arduino Mega, Tarjeta Motum, Tarjeta Controlador y Tarjeta programador.

Para implementar el algoritmo de aprendizaje, se adquieren 500 imágenes en diferente posición de cada tarjeta por medio de un video. Aplicando el algoritmo de aprendizaje automático se extraen las 200 características más importantes de cada una de las tarjetas. Una vez obtenida la matriz de información se entrenan diferentes algoritmos, con la finalidad de identificar correctamente cada una de las tarjetas. Se comparan los resultados obtenidos de cada algoritmo entrenado y se implementa el algoritmo que presenta el mayor índice de eficacia. En la Figura 4 se muestran los índices de efectividad de los algoritmos entrenados, entre los cuales se encuentran Complex Tree, Medium Gaussian SVM y Quadratic SVM, entre otros. Cada algoritmo cuenta con un principio de estimación y optimización diferente, una descripción más amplia de este tipo de algoritmos se puede observar en la referencia (Mathworks, 2016).

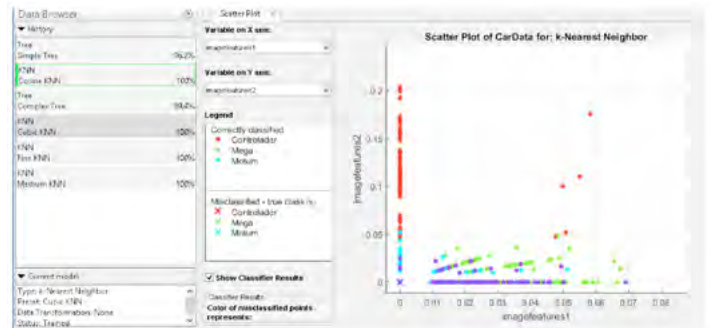
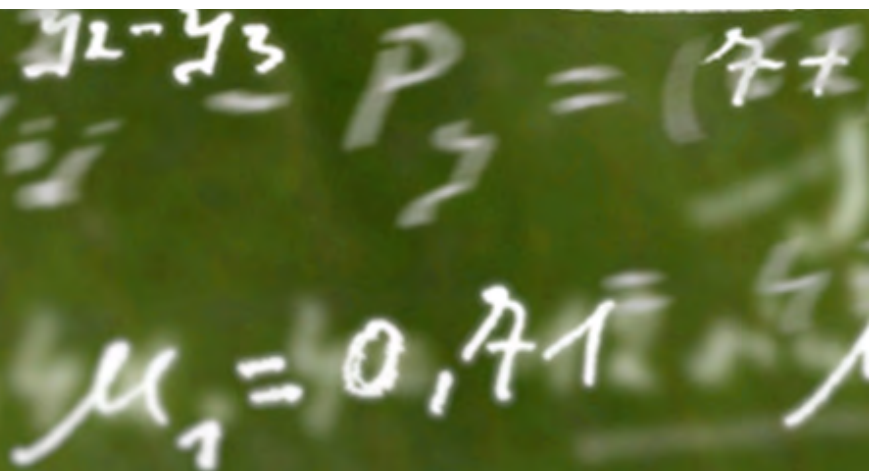


Figura 4. Respuesta de comportamiento del motor a una frecuencia de 260 Hz.

Para implementar el algoritmo de búsqueda de componentes físicos dentro de la tarjeta electrónica se realiza un mapeo de la imagen, para realizar la conversión de unidades de medida, es decir, interpretar la imagen en centímetros y no en píxeles. Posteriormente se realiza la búsqueda de cada componente físico dentro de la tarjeta electrónica, empleando para ello los archivos Gerber correspondientes a cada tarjeta.

La implementación del algoritmo de búsqueda se muestra en la Figura 5. Se obtiene una imagen en formato RGB utilizando la cámara web, la imagen es convertida en formato binario. Se realiza la lectura del archivo Gerber para obtener las coordenadas de cada uno de los componentes que integran la tarjeta. Posteriormente se realiza el mapeo y se analiza si existe un componente en la posición deseada.



Se desarrolla una interfaz gráfica en Matlab para validar el funcionamiento del algoritmo de búsqueda de componentes electrónicos por medio de una interfaz. En la Figura 9 se muestran dos tarjetas bajo prueba, la Figura 9(a) corresponde a una tarjeta completa, en este caso el algoritmo indica que la tarjeta se encuentra completa, caso contrario a la Figura 9(b), en esta se retiran dos componentes correspondientes al Led3 y Res4, los cuales son identificados por el algoritmo, indicando su ausencia y, por consiguiente, que la tarjeta se encuentra incompleta.



Figura 9. Algoritmo de reconocimiento, a) tarjeta completa y b) tarjeta incompleta.

Conclusiones

Se describió un algoritmo de aprendizaje automático capaz de reconocer diferentes tipos de tarjetas electrónicas, así como sus componentes electrónicos. De esta manera se observan que los dos algoritmos desarrollados funcionan de manera correcta. El primer algoritmo cuenta con la capacidad de reconocer de manera automática cualquier tipo de tarjeta electrónica, sin importar el color, posición, ángulo de la cámara para la captura del fotograma y el nivel de luminosidad, solo es necesario entrenar al algoritmo con los objetos que se desean identificar. El trabajo en conjunto de los dos algoritmos tiene la capacidad de realizar la automatización de procesos de manufactura de tarjetas electrónicas, reduciendo el costo de ensamble, costo de producción y costo de equipo especializado; por lo tanto, se genera una nueva aplicación de los algoritmos con sistema de aprendizaje automático.

Símbolos

l	longitud
(cm)	
v	velocidad
(m/s)	
lx	lux
px	pixel
	(lm/m^2)

Bibliografía

+ Ziyin, L. y Qi, Y. (2011). *System design for PCB defects detection based on AOI technology*, *International Congress on Image and Signal Processing*, 4, 1- 4.

+ Zhao, H.; Cheng, J. y Jin, J. (2009). *NI Vision Based Automatic Optical Inspection (AOI) for Surface Mount Devices: Devices and Method*, *Proceedings of 2009 IEEE International Conference on Applied Superconductivity and Electromagnetic Devices*, 25-27.

+ Berral García, J. (2016). *A Quick View on Current Techniques and Machine Learning Algorithms for Big Data Analytics*, *International Conference on Transparent Optical Networks (ITCON)*, 18, 1-4.

+ MathWors (2016). *Introducing Machine Learning*. Disponible en https://www.mathworks.com/tag-team/89703_92991v00_machine_learning_section1_ebook_v12.pdf?s_tid=solmain_mlr_cta2

+ Manan Khan, A.; Khan, F. y Changsoo, H. (2016). *Estimation of Desired Motion Intention using Extreme Learning Machine for Upper Limb Assist Exoskeleton*, *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 1-5.

+ Faruk Alcin, O; Ferhat, U. y Korkmaz, D. (2016). *Extreme Learning Machine Based Robotic Arm Modeling*, *International Conference on Methods and Models in Automation and Robotics (MMAR)*, 21, 1-4.

+ Rau, H.; Wu, C.; Shiang, W. y Huang, P. (2010). *A Decision Support System of Statistical Process Control for Printed Circuit Board Manufacturing*, *International Conference on Machine Learning and Cybernetics*, pp 1-5.

+ MathWors (2016). *Statistics and Machine Learning Toolbox User's Guide*. Disponible en https://www.mathworks.com/help/pdf_doc/stats/stats.pdf

Información Legal

UTCJ Theorema Revista Científica, es una publicación semestral, editada por la Universidad Tecnológica de Ciudad Juárez, Avenida Universidad Tecnológica # 3051, Col. Lote Bravo II, C.P. 32695, Tel. (656) 6490604, www.utej.edu.mx. Editor responsable: Idali Meléndez Domínguez. Reserva de Derechos al Uso Exclusivo No. 04-2016-061713532700-203, ISSN: 2448-7007, ambos otorgados por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número, Departamento de Prensa y Difusión, L.D.G. Pamela Hinostrza Espinoza, Avenida Universidad Tecnológica # 3051, Col. Lote Bravo II, C.P. 32695, fecha de la última modificación 10 de octubre de 2017.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación. Queda prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Nacional del Derecho de Autor.

Apéndice G

Solicitud de patente

En este apéndice se muestra un trámite de solicitud de patente, con el nombre *Sistema supervisor de motores eléctricos en tiempo real*, en el estado de Puebla, el día 8 de Diciembre del 2016, solicitada en el Instituto Mexicano de la Propiedad Industrial.



Solicitud de Patente
 Solicitud de Registro de Modelo de Utilidad

Solicitud de Registro de Diseño Industrial, especifique cuál:
 Modelo Industrial Dibujo Industrial

Uso exclusivo Delegaciones y Subdelegaciones de la Secretaría de Economía y Oficinas Regionales del IMPI

Sello

Folio de entrada

Fecha y hora de recepción

INSTITUTO MEXICANO DE LA PROPIEDAD INDUSTRIAL
 Dirección Divisinal de Patentes

OFICINA REGIONAL CENTRO

Solicitud: **MX/a/2016/016261**
 Expediente: **MX/a/2016/016261**
 Fecha: **8/DIC/2016** Hora: **14:19:49**
 Folio: **MX/E/2016/060053** 212328



Antes de llenar la forma lea las consideraciones generales al reverso

DATOS DEL (DE LOS) SOLICITANTE(S)

El solicitante es el inventor El solicitante es el causahabiente

1) Nombre (s): **BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

2) Nacionalidad (es): **MEXICANA**

3) Domicilio; calle, número, colonia y código postal: **4 SUR 104, COL. CENTRO, C.P. 72000**

Población, Estado y País: **PUEBLA, PUEBLA, MÉXICO;**

4) Teléfono (clave): (01 222) 229 55 00 Ext: 3058 5) Fax (clave): (01 222) 229 55 00 Ext: 5058

DATOS DEL (DE LOS) INVENTOR(ES)

6) NOMBRE (S): **C. JAIME JULIÁN CID MONJARAZ; C. JOSÉ FERNANDO REYES CORTES; C. MIGUEL ÁNGEL LEÓN CHÁVEZ; C. EDGAR HUGO MAYORAL ARZABA; C. JOSÉ MANUEL REYES RODRÍGUEZ**

7) Nacionalidad (es): **MEXICANA**

8) Domicilio; calle, número, colonia y código postal: **FACULTAD DE CIENCIAS DE LA ELECTRÓNICA, CIUDAD UNIVERSITARIA, COL. SAN MANUEL, C.P. 72570, FACULTAD DE CIENCIAS DE LA COMPUTACIÓN, CIUDAD UNIVERSITARIA, COL. SAN MANUEL, C.P. 72570**

Población, Estado y País: **PUEBLA, PUEBLA, MÉXICO**

9) Teléfono (clave): 10) Fax (clave):

DATOS DEL (DE LOS) APODERADO(S)

11) Nombre (s): **MTRA. ROSA ISELA ÁVALOS MÉNDEZ** 12) R.G.P.: **DDAJ-26065**

13) Domicilio; calle, número, colonia y código postal: **4 SUR 104, COL. CENTRO, C.P. 72000**

Población, Estado y País: **PUEBLA, PUEBLA, MÉXICO** 14) Teléfono (clave): (01 222) 229 55 00 Ext. 3058 15) Fax (clave): (01 222) 229 55 00 Ext. 3058

16) Personas Autorizadas para oír y recibir notificaciones:
LIC. JAIR ERIC VÁZQUEZ TORRES; LIC. GABRIELA SÁNCHEZ ESGUA

17) Denominación o Título de la Invención:

SISTEMA SUPERVISOR DE MOTORES ELÉCTRICOS EN TIEMPO REAL

18) Fecha de divulgación previa
 Día Mes Año

19) Clasificación Internacional uso exclusivo del IMPI

20) Divisional de la solicitud
 Número

21) Fecha de presentación
 Día Mes Año

22) Prioridad Reclamada:
 País Fecha de presentación No. de serie
 Día Mes Año

Lista de verificación (uso interno)

No. Hojas	Descripción	No. Hojas	Descripción
	Comprobante de pago de la tarifa		Documento de cesión de derechos
	Descripción y reivindicación (es) de la invención		Constancia de depósito de material biológico
	Dibujo (s) en su caso		Documento (s) comprobatorio(s) de divulgación previa
	Resumen de la descripción de la invención		Documento (s) de prioridad
	Documento que acredita la personalidad del apoderado		Traducción
			TOTAL DE HOJAS

Observaciones:

Bajo protesta de decir verdad, manifiesto que los datos asentados en esta solicitud son ciertos.

Mtra. Rosa Isela Ávalos Méndez
 Nombre y firma del solicitante o su apoderado

PUEBLA, PUEBLA, A 22 DE NOVIEMBRE DE 2016
 Lugar y fecha

Bibliografía

- [1] T. B. Sheridan, *Telerobotics, Automation and Human Supervisory Control*, Massachusetts Institute of Technology, 1992.
- [2] F. Reyes, *Robótica. Control de Robots Manipuladores*, Alfaomega, 2011.
- [3] M. López & M. Arteaga, *Control Servovisual de robots manipuladores: un enfoque hacia la industria*, Universidad de LaSalle, Bogotá-Colombia, 2012.
- [4] J. Hill & W.T. Park, *Real Time Control of a Robot with a Mobile Camera*, Proc. 9th ISIR Washinton, D.C., 1979.
- [5] P. Corke & S. A. Hutchinson, *Real-Time Vision, Tracking and Control*, International Conference on Robotics and Automation, 2000.
- [6] F. Reyes & L. E. Chiang, *Location and Classification of Moving Fruits in Real Time with a Single Color Camera*, Chilean Journal of Agricultural Research 69, 2009.
- [7] S. Hutchinson, G. Hager & Peter Corke, *A Tutorial on Visual Servo Control*, IEEE Transactions on Robotics and Automation, 1996.
- [8] D. M. Auslander, *What is Mechatronics?*, IEEE/ASME Transactions on Mechatronics Vol. 1 No. 1, pp. 5-9, 1996.
- [9] F. Reyes, J. Cid & E. Vargas., *Mecatrónica Control y Automatización*, Alfaomega, 2015.
- [10] L. Scavicco & B. Siciliano, *Modeling and Control of Robot Manipulator*, McGraw Hill, 2011.
- [11] R. Iñigo & H. Asada, *Robots industriales manipuladores*, Alfaomega, 2004, Barcelona, España.
- [12] M. Spong & M. Vidyasagar, *Robot Dynamics and Control*, John Wiley and sons, Inc, 1989, United States of America.
- [13] K. C. Jain & L. N. Aggarwa, *Robotics-Principles and Practice*, Khanna Publishers, 2002, Delhi, 3ra. Edición.
- [14] S. Hutchinson, G.D. Hager & P.I. Corke, *A Tutorial on Visual Servo Control*, IEEE Transactions on Robotics and Automation, 1996.
- [15] L. E. Weiss, A.C. Sanderson & C.P. Neuman, *Dynamic sensor-based control of robots with visual feedback*, IEEE Journal of Robot Automation, Oct. 1987.

- [16] C. Williams, W. J. Wilson & S. B. Graham, *Relative End-Effector Control Using Cartesian Position Based Visual Servoing*, IEEE Transactions on Robotics and Automation, 1996.
- [17] P.I. Corke, *Visual Control of Robot Manipulators. A review Visual Servoing*, K. Hashimoto, Ed. Singapore: World Scientific, 1993.
- [18] M. Takegaki & S. Arimoto, *A new feedback method for dynamic control of manipulators*, J. Dyn. Syst. Meas. Control Trans, 1981.
- [19] F. Reyes & R. Kelly, *Experimental Evaluation of FixedCamera Direct Visual Controllers on Robots*, 1998.
- [20] J. Park & J. Lee, *Robust Visual Servoing for Motion Control of the Ball on a Plate*, Mechatronics, 2003.
- [21] R. Kelly, P. Shirkley & M. Spong, *Fixed-Camera Visual Servo Control for Planar Robots*, IEEE International Conference on Robotics and Automation, 1996.
- [22] E. Malis & P. Rives, *Robustness of Image-based Visual Servoing whit Respect to Depth Dristribution Errors*, IEEE Intrnational Confrence on Robotics and Automation, 2003.
- [23] MATLAB, *Image Processing Toolbox, User Guide*, MathWorks, 2017.
- [24] E. Cuevas, D. Zaldivar & Marco Pérez, *Procesamiento digital de imágenes con Matlab y Simulink*, Ra Ma, 2015.
- [25] D.A. Wells, *Lagrangian dynamics*, Schaum's outlines, 1967.
- [26] W. Stadler, *Analytical robotics and mechatronics*, Graw Hill International Editions, 1995.
- [27] M. Takegaki & S. Arimoto, *A new feedback method for dynamic control of manipulators*, ASME j. Dyn. Syst. Meas. Control. 103: pp. 119-125, 1981.
- [28] R. Kelly & V.Santibáñez, *Control de movimiento de robots manipuladores*, Printice Hall, Pearson, 2003.
- [29] F. Reyes & J. Cid, *Arduino, aplicaciones en Robótica, Mecatrónica e Ingenierías*, ALfaomega, 2015.
- [30] P. Viola & M. Jones., *Rapid Object Detection using a Boosted Cascade of Simple Features*, 2001.
- [31] O. H. Jensen, *Implementing the Viola-Jones Face Detection Algorithm*, 2008.
- [32] *Ebook Statistics and Machine Learning Toolbox User´s Guide, Matlab*, pp 5-8708, 2016
- [33] MATLAB, *MATLAB, App Building*, 2016.
- [34] S. Vashi, J. Ram & J. Modi, *Internet of Things (IoT): A visión, architectural elements, and security issues*, International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017.