



UNIVERSIDAD
TECNOLÓGICA
DEL PERÚ

Facultad de Ingeniería

Trabajo de Investigación

“Diseño del control servo visual de un robot de 3 GDL aplicando RNA para la identificación de elementos químicos Y manipulación mediante un rayo tractor sónico”

Autores:

Bozzeta De la Cruz, Fernando Francisco - 1531526

Bruno Villena, Ronny Anderson – 1531569

Para obtener el grado de Bachiller en:

Ingeniería Mecatrónica

Lima, diciembre 2018

RESUMEN

A lo largo de los años, la industria química se ha visto envuelta en múltiples accidentes por la incorrecta manipulación de elementos químicos ya sea por negligencia o falla de algún equipo de laboratorio. Por ello, la presente investigación se basa en la necesidad de manipular y reconocer eficientemente los elementos químicos que puedan ser nocivos para la salud. Como medida de solución, se ha desarrollado un algoritmo para el reconocimiento de la posición del elemento dentro del área de trabajo basado en el procesamiento digital de imágenes aplicado a un robot de 3 grados de libertad, a este método se le denomina control servo visual. Así mismo, se ha planteado asociar los colores a los distintos elementos químicos que se puedan presentar, posteriormente el encargado del reconocimiento de dichos elementos son las redes neuronales artificiales. Por último, en medida de realizar una manipulación limpia y sin dejar impurezas se ha diseñado un rayo tractor sónico para reemplazar el clásico griper como elemento manipulador.

Las simulaciones para el posicionamiento del robot en base a la adquisición de imagen del área de trabajo coincidieron con el ángulo y la distancia que forma el robot respecto al objeto, los datos fueron corroborados gracias al prototipo, como resultado el ángulo de desfase fue de un $\pm 10^\circ$ debido a factores de precisión del servomotor. Para comprobar el rendimiento de la red neuronal se realizaron múltiples pruebas, de las cuales se obtuvo un 85% de acierto, un resultado aceptable. En conclusión, gracias a los datos obtenidos se ha determinado que el control servo visual de un robot es viable al ser aplicado a un manipulador de 3GDL, además, es posible asociar los colores de los elementos químicos para determinar la clase ya que se obtuvo un eficiente resultado en pruebas reales.

Dedicatoria

Este trabajo está dedicado con todo cariño a nuestras familias; de manera especial a Hanko y Benny; quienes han depositado toda su confianza para lograr un objetivo más en nuestras vidas.

Agradecimiento

Un agradecimiento especial a las personas que estuvieron en todo momento apoyándonos en los buenos y malos momentos a lo largo de esta maravillosa etapa universitaria.

ÍNDICE

Declaración de Autores	2
Resumen	4
Dedicatoria	5
Agradecimiento	6
Índice	7
Índice de Tablas	9
Índice de Figuras	10
Introducción	13
Capítulo I: Antecedentes de la investigación	15
1.1 Planteamiento del Problema	15
1.2 Objetivos	16
1.3 Alcances de la Investigación	16
Capítulo II: Marco Teórico	19
2.1 Estado del Arte	19
2.2 Bases teóricas	21
2.3 Modelamiento	43
2.4 Tabla de Parámetros	52
Capítulo III: Diseño	54
3.1 Diseño del sistema Robótico	55
3.2 Diseño del algoritmo para la identificación de colores-RNA	79
3.3 Diseño del Algoritmo para el Control Servo visual	93
3.4 Diseño del Rayo Tractor Sónico	103
Capítulo IV: Análisis y Resultados	111
4.1. Simulaciones y Validaciones	111

Conclusiones	123
Recomendaciones	124
Bibliografía	125
Anexo1	129
Anexo2	132
Anexo3	134

ÍNDICE DE TABLAS

Tabla 2.1 Comparativa entre un ordenador secuencial y el sistema biológico neuronal.	32
Tabla 2.2 Parámetros D-H. para la figura 4.2.	47
Tabla 2.3 Parámetros generales.	52
Tabla 3.1 Parámetros Denavit-Hartenberg.	56
Tabla 3.2 Resultados obtenidos del modelamiento cinemático del Robot.	66
Tabla 3.3 Tabla de Torques para cada articulación.	76
Tabla 3.4 Datos de salida deseada de la red neuronal.	85
Tabla 3.5 Salidas obtenidas de las 12 muestras de pruebas.	92
Tabla 3.6 Elemento a manipular.	105
Tabla 3.7 Tabla de resultados del elemento.	105
Tabla 3.8 Parámetros para la elección de los dispositivos.	110
Tabla 4.1 Simulaciones y desfases de ángulos	116
Tabla 4.2 Resultados de las simulaciones para el reconocimiento de elementos.	120

ÍNDICE DE FIGURAS

Figura 2.1 Robot PUMA.	24
Figura 2.2 Bocetos de algunos tipos de robots.	25
Figura 2.3 Los 6 pares inferiores de Reuleaux.	26
Figura 2.4 Cadena cinemática a) abierta y b) cerrada.	27
Figura 2.5 Estructuras de robots más usadas en la industria.	28
Figura 2.6 Robots planares.	29
Figura 2.7 La neurona biológica.	31
Figura 2.8 Neurotransmisores en una sinapsis.	33
Figura 2.9 Modelo de una Neurona Artificial.	34
Figura 2.10 Funciones de activación.	35
Figura 2.11 RNA monocapa.	37
Figura 2.12 RNA multicapa.	38
Figura 2.13 Diagrama arquitectónico de una red neuronal tipo feedforward(RNAf).	38
Figura 2.14 RNA tipo recurrentes.	39
Figura 2.15 Vórtice de sonido encapsulando esfera de poliestireno.	40
Figura 2.16 Diagrama de bloques.	44
Figura 2.17 Diagrama pictórico.	44
Figura 2.18 Robot cilíndrico aplicando algoritmo D-H.	46
Figura 2.19 Diagrama de flujo.	50
Figura 3.1 Plano de los puntos de referencia del robot de 3 grados de libertad.	55
Figura 3.2 Contenidos en un plano y en configuración codo abajo (a) y configuración codo arriba (b).	60
Figura 3.3 Representación gráfica en 2D del Robot manipulador.	68
Figura 3.4 Código para determinar los pares para cada articulación.	73
Figura 3.5 Torque gráfico de la primera articulación.	74

Figura 3.6 Torque gráfico de la segunda articulación.	75
Figura 3.7 Torque gráfico de la segunda articulación.	75
Figura 3.8 Servomotor de 2. 5Kg.cm.	76
Figura 3.9 Servomotor de 21.5 Kg.cm.	77
Figura 3.10 Servomotor de 6. 5Kg.cm.	77
Figura 3.11 Simulación del robot manipulador en Matlab.	78
Figura 3.12 Diseño Preliminar del Prototipo.	78
Figura 3.13 Elementos de muestras.	80
Figura 3.14 Estudio de toma de datos.	80
Figura 3.15 Diagrama de flujo para recolección de datos.	81
Figura 3.16 Código de programación en Matlab para las capturas de muestras.	83
Figura 3.17 Diagrama de Flujo para obtención de características.	84
Figura 3.18 Código para la obtención de características.	85
Figura 3.19 Arquitectura de la red neuronal.	87
Figura 3.20 Validación del rendimiento de la Red entrenada.	87
Figura 3.21 Matriz de confusión de la red entrenada.	88
Figura 3.22 Código de la Red Neuronal.	89
Figura 3.23 Código para la extracción de características de imágenes de pruebas.	90
Figura 3.24 Código para testear la red neuronal.	91
Figura 3.25 Datos obtenidos de la prueba para la primera muestra.	91
Figura 3.26 Código para el procesamiento de imagen.	94
Figura 3.27 Imagen proceda de la muestra.	95
Figura 3.28 Código para determinar las propiedades de la figura.	95
Figura 3.29 Posición del objeto en el plano 2D.	96
Figura 3.30 Posición del robot respecto al objeto sobre el plano.	97

Figura 3.31 Geometría de la posición de los eslabones.	98
Figura 3.32 Posición del objeto en diferentes cuadrantes.	99
Figura 3.33 Algoritmo para la conversión y obtención de ángulos.	100
Figura 3.34 Código para la simulación del movimiento.	101
Figura 3.35 Posición final del robot manipulador.	102
Figura 3.36 Nueva muestra en coordenadas 2D.	102
Figura 3.37 Posición del robot manipulador en coordenadas de la nueva muestra.	103
Figura 3.38 Funcionamiento del rayo tractor sónico.	109
Figura 3.39 Diseñado por el fabricante.	110
Figura 4.1 Prototipo del robot manipulador de 3 GDL.	112
Figura 4.2 Vista general del prototipo para las pruebas.	112
Figura 4.3 Reconocimiento del objeto en el mapa, prueba 1.	113
Figura 4.4 Captura del desfase de la posición, prueba 1.	114
Figura 4.5 Reconocimiento del objeto en el mapa, prueba 2	115
Figura 4.6 Captura del desfase de la posición, prueba 2.	115

INTRODUCCIÓN

En la actualidad dentro de los laboratorios e industrias químicas, la gran mayoría de procesos de manipulación son exclusivamente manuales; por lo tanto, existe una gran exposición de los operadores al contacto con diferentes sustancias químicas, las cuales pueden ser nocivos para la salud. Por lo general, los operadores se rigen con ciertos parámetros previos para manipular determinados elementos dentro del entorno de trabajo. Sin embargo, muchos de ellos obvian algunos pasos de seguridad ya sea por negligencias, contratiempos, estrés, falta de equipos de protección, etc. Factores que ponen en riesgo constante al operador y a su entorno pudiendo ocasionar daños irreversibles.

En base a lo anterior se ha determinado que existe un problema en la manipulación de elementos químicos. Por ello, como solución se ha propuesto la implementación de un nuevo sistema para realizar este tipo de actividades aplicando las Redes Neuronales Artificiales(RNA) ya que, en la en los últimos años, es un tema de mucho interés, por la gran capacidad de procesar información y parámetros complejos. Este nuevo sistema se basa en un control de un robot manipulador de 3 grados de libertad, siendo capaz de reconocer las posiciones de los elementos químicos mediante el procesamiento digital de imágenes y a la vez la identificación del tipo de elemento a manipular gracias a las redes neuronales. Así mismo, cuenta con un rayo tractor sónico, el cual es un equipo que emite

ondas mecánicas de sonido con la capacidad de capturar el objeto haciendo que se suspenda en el aire, esto permite que al realizar las manipulaciones de los elementos químicos no se generen pérdidas o algún tipo de contaminación al tener contacto físico con otro agente externo.

CAPÍTULO I

ANTECEDENTES DE LA INVESTIGACIÓN

1.1. Planteamiento del Problema

La tecnología en la industria química se encuentra en avance continuo día a día, debido a que surgen nuevos equipos digitalizados especializados para satisfacer muchas necesidades dentro del área de trabajo químico. Sin embargo, detrás de las necesidades que posee la industria existe el personal capacitado quien asumen la labor diaria y se exponen a diferentes ambientes, donde muchos de ellos pueden ser tóxicos por los distintos elementos a manipular o por el solo hecho de interactuar continuamente en áreas dañinas para la salud, pudiendo generar males inmediatos o como a futuro contra su persona.

Por lo tanto, ¿Qué relación existe entre el aislamiento de trabajo de un operador con respecto al diseño de un servo mecanismo inteligente capaz de evitar contacto alguno con sustancias sumamente dañinas? Siendo el eje principal la seguridad del operador, en el proceso del desarrollo de esta tesis se dará resultado a este preciso cuestionamiento.

1.2. Objetivos

Para la presente investigación los objetivos han sido divididos en dos partes, generales y específicos, siendo este último aquellos que en conjunto cumplan con el objetivo general.

1.2.1. Objetivo General

Diseñar el control servo visual que permita identificar dentro del sistema de coordenadas el objeto a manipular. Además, diseñar una red neuronal que permita la identificación de los elementos químicos.

1.2.2. Objetivos Específicos

- Realizar el modelamiento cinemático directo e inverso del robot manipulador.
- Modelar la dinámica del robot de 3 GDL y realizar las primeras simulaciones.
- Desarrollar el algoritmo para el reconocimiento de los elementos químicos asociados a colores.
- Desarrollar el algoritmo para el procesamiento de imágenes y extracción de características del espacio de trabajo.
- Determinar la clase de transductor sónico para el rayo tractor.

1.3. Alcances de la investigación

La investigación debe de estar respaldado por los alcances que este tendrá en diferentes ámbitos, para esta investigación se propuso, evaluar los beneficios dentro del ámbito tecnológico, económico, social y ambiental.

1.3.1. Alcance Tecnológico

La tecnología robótica ha sido ampliamente desarrollada para ser aplicada a la industria de la manufactura, teniendo como finalidad aumentar la cantidad y calidad de producción de un producto, sin embargo, pocos han desarrollado diseños y prototipos para ser aplicados al área de investigación de la industria química.

Hoy en día la labor dentro de esta área se realiza manualmente haciendo uso de cabinas presurizadas para aislar las manipulaciones y reacciones químicas que se puedan generar. Por lo tanto, el desarrollo de la tesis, presenta una integración de tecnologías como la robótica, las redes neuronales artificiales y un rayo tractor sónico, siendo esta última, una tecnología recientemente desarrollada, no obstante, presenta características positivas para ser aplicadas a esta área. La interacción y el trabajo en conjunto de estas tecnologías forman un sistema innovador que permite la eficiente manipulación de sustancias químicas haciendo posible que las investigaciones con diversos elementos puedan realizarse de manera segura y sin contacto directo con el operador.

1.3.2. Alcance Económico

El desarrollo de un prototipo que cumpla con los objetivos propuestos no demanda una gran inversión económica, ya que, los materiales y equipos electrónicos que lo conforman existen en gran cantidad en el mercado siendo totalmente accesibles. Sin embargo, de tratarse de un prototipo a escala superior se debe considerar que la inversión para su implementación será mayor debido a que, los equipos electrónicos y mecánicos del sistema tienen un elevado costo ya que la variación del precio se rige por las características propias de cada elemento.

1.3.3. Alcance Social

Gracias a la tesis desarrollada se espera un impacto positivo y favorable para la sociedad. La implementación de un prototipo que sea aplicado al área de investigación química y que gracias a ello desarrollen nuevos compuestos que permitan generar un producto a menor costo y más duradero estos factores aportan a la economía y desarrollo social. Por otro lado, presentar el prototipo desarrollado para fines educativos dentro de un área de estudio aporta para

que una nueva generación de estudiantes se pueda formar dentro del ámbito tecnológico.

1.3.4. Alcance Ambiental

El medio ambiente es un factor importante que se ha considerado para el diseño e implementación del prototipo, este sistema procura no dañar el ecosistema donde se desarrolla, esto gracias a que hace uso solo de energía renovable como es la energía eléctrica. Además, no ha de generar ningún otro tipo de contaminación, el ruido es mínimo, no existen fuertes vibraciones, no existe fuerza de impacto que pueda ocasionar algún tipo de molestia al operador. Por otro lado, se ha previsto que al realizar manipulaciones de elementos químicos siempre quedan residuos dentro de los recipientes, probetas, matraz o algún otro tipo de contenedor. Por este motivo, se asegura que la manipulación de estos elementos se realizará de manera limpia gracias al nuevo sistema de manipulación, empleando un rayo tractor sónico se evita contacto con algún agente externo que pueda contaminar la muestra o dejar residuos o partículas dentro de un recipiente.

CAPÍTULO II

MARCO TEÓRICO

2.1. Estado del arte

La tesis desarrollada por W. Carrera denominada “Diseño y construcción del subsistema electrónico para el control de un brazo robot de 5 grados de libertad”, desarrollada en la Universidad Católica del Perú (2015), diseña una interfaz PC usando el lenguaje de programación Visual Basic para el control de las articulaciones, donde se ingresaran los movimientos que se desean para ejecutar el movimiento del brazo, analizando la cinemática directa e inversa de dicho robot, como también el seguimiento de su trayectoria. Como resultado final se escoge diferentes dispositivos como sensores, excitadores, tarjetas de control y tipo de interfaz, con la finalidad de que cada elemento corresponda adecuadamente y tenga la funcionalidad exacta para las pruebas que se somete el sistema en general de cinemática directa, inversa y finalmente de generación de trayectorias que son corroboradas con el software Matlab. En el paper implementado por R. Rodríguez, M. Sánchez, M. Ravichagua y S. Castillo, denominado “Control del brazo robótico de seis grados de libertad, mediante retroalimentación visual, para la cosecha del ají paprika” (2011) de la Universidad

Nacional de Ingeniería UNI, el presente trabajo hace énfasis a tres materias de suma importancia en la carrera de ingeniería mecatrónica como visión artificial y control de robots.¹

En esta oportunidad, se implementa un robot de 6 grados de libertad, el cual desarrollará tareas automatizadas para la cosecha del ají paprika, debido a que este equipo brinda mayor versatilidad para el desarrollo de movimientos complejos y operación en terrenos agrestes. Para poder controlar dichos movimientos y las posiciones espaciales del robot se acopla una serie de dispositivos electrónicos, de los cuales cabe resaltar los siguientes: un sensor visual, encargado de reconocer el ambiente de trabajo, mientras que para la profundidad y recolección del ají es necesario un sensor de distancia.

La investigación desarrollada por C. Luyo denominada “Diseño e implementación de un sistema de control e interfaz para un brazo robótico de 5 GLD” de la Universidad Católica del Perú (2015), implementan un diseño de control para la manipulación y planificación de trayectorias capaz de ser utilizado para pruebas dinámicas y cinemáticas.²

El sistema está establecido para que sea comandado por un operador quien será el encargado de definir las coordenadas finales para que el robot pueda realizar su trayectoria.

Por otro lado, se realizó un diseño mecánico, el cual permitió al autor la fabricación de cada uno de los eslabones. En la parte electrónica, se determinó el uso de dispositivos electrónicos capaces de producir señales que permiten al sistema controlar las posiciones y movimientos de los eslabones, bajo criterios de un control discreto,

¹ RODRÍGUEZ R., SÁNCHEZ M., RAVICHAGUA M. y CASTILLO S. (2011). Control del brazo robótico de seis grados de libertad, mediante retroalimentación visual, para la cosecha del ají paprika. Lima, Perú: UNI

² LUYO Christian (2015). Diseño e implementación de un sistema de control e interfaz para un brazo robótico de 5 GLD. Lima, Perú: PUCP.

quienes se encargarán de seguir el modelamiento matemático planteado con anterioridad para el equipo según la función de transferencia.

El control desarrollado en este proyecto cumple con las expectativas a nivel de control angular, haciendo previamente pruebas individuales en cada eslabón para verificar su exactitud de trabajo. También, se define que el brazo robótico es controlado completamente por una interfaz gráfica, la cual no solo permitió movimientos aislados de los eslabones disponibles en la estructura, sino también, realizar trayectorias, al definir el punto final donde debe llegar.

Según D. Triana, S. Roa y C. Forero, realizan un trabajo de investigación denominado “Desarrollo y control de un brazo robótico mediante la adquisición de datos en tiempo real hacia un espacio no real”, de la Universidad Autónoma de Bucaramanga, Colombia (2015)³, donde se implementan sistemas de control más accesibles para el usuario y mando del manipulador. El proceso desarrollado facilita el reconocimiento de la dinámica del robot, partiendo desde la posición, velocidad y aceleración (datos adquiridos en tiempo real) para poder determinar la trayectoria que se ejecutará en este sistema robótico en un espacio no real (simulación). Por lo tanto, se obtiene como resultado un exoesqueleto maestro capaz de controlar al manipulador virtual, visualizando todos los actos físicos del brazo en la simulación.

Según S. Pandey, realiza una investigación del “Concepto para la clasificación automatizada por medio de un brazo robótico”, de la Universidad de Baroda, Vadodara, Gujarat, India (2018)⁴, el cual parte de una necesidad como las tareas repetitivas y precisión en distintos procesos de la industria. Brindando soluciones por medio de aplicaciones robóticas autónomas, quienes se encargarán de realizar trabajos repetitivos manteniendo las exigencias de precisión y velocidad. Por lo tanto, se diseña

³ TRIANA D., ROA S. y FORERO C. (2015). Desarrollo y control de un brazo robótico mediante la adquisición de datos en tiempo real hacia un espacio no real. Bucaramanga, Colombia: UAB

⁴ PANDEY, Shalini (2018) Concept for automated classification by means of a robotic arm. Gujarat, India: UBV

un brazo robótico versátil y de bajo costo que finiquite con las necesidades de la industria.

El proyecto desarrollado compite con otros basados en la misma funcionabilidad, destacando en la flexibilidad y versatilidad en términos de detección de objetos y selección según las características de estos, con la finalidad de aumentar la productividad de la industria o en otro entorno donde se instale.

El paper desarrollado por A. Jamali, I. Mat, P. Mohd y M. Tokhi, denominado “Modelado inteligente del manipulador robótico flexible de doble enlace utilizando una red neuronal artificial” de la Universidad de Tecnología de Malasia, Malasia (2017)⁵, describe el funcionamiento de una red neuronal artificial en el modelamiento de un manipulador robótico flexible, bajo un sistema de la RNA de multi-entrada y multi-salida.

Las señales recogidas para ser tratadas por la red son dadas por los codificadores, acoplados a los motores, acelerómetros, capaces de reconocer las vibraciones generadas en el brazo robótico, una tarjeta de adquisición de datos para la conversión analógica a digital. Por lo tanto, estos dispositivos en conjunto serán simulados bajo el entorno del software Matlab/Simulink.

2.2. Bases Teóricas

Para el desarrollo del proyecto planteado, previamente se debe de tener fundamentos teóricos esenciales para comprender la complejidad y el funcionamiento de todos los sub-sistemas empleados, en las siguientes líneas se pasa a detallar los principios teóricos que lo conforman.

2.2.1. La Robótica

Los autores Barrientos, Peñín, Balaguer y Aracil del libro denominado Fundamentos de Robótica (2007), definen el término como una disciplina en

⁵ JAMALI A., MAT I., MOHD P. y TOKHI M. (2017) Intelligent modeling of double link flexible robotic manipulator using artificial neural network. Skudai, Malasia: UTM

auge el cuál hace una inclusión de varias ramas de la ciencia y de las tecnologías, tales como la automatización industrial, mecánica, informática, electrónica, teorías de control, álgebra, entre otras. [6]

A lo largo de la historia, el hombre ha desarrollado una serie de mecanismos automáticos siendo uno de los más populares la creación del gallo de Estrasburgo en el año 1352. Por otro lado, la primera patente de un dispositivo robótico se presentó en el año 1954 gracias al inventor de procedencia británica C.W. Kenward. Años después, debido a la amplia experiencia en el desarrollo de mecanismo, autómatas y siendo el autor de varias patentes, George C. Devol fue el encargado de establecer las bases del robot industrial moderno.

En finales de los sesenta y principios de la década de los setenta las bases de la investigación de la robótica se proponen en universidades norteamericanas. Posteriormente se crearon departamentos de investigación en inteligencia artificial desarrollando exitosamente los primeros robots móviles con cierto porcentaje de autonomía. Gracias al desarrollo del *Stanford Cart* en la universidad de Standford, siendo este el primer manipulador controlado por computador y con un accionamiento eléctrico, dio paso a los diseños actuales de los robots manipulador. Posteriormente, esta idea dio inicio al diseño, fabricación y comercialización del robot PUMA mostrado en la figura 2.1, el robot más famoso y utilizado en la industria de la producción, como también en el área de investigación. [7]

⁶ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica (Segunda ed.)*. Madrid, España (pp.xii): McGraw-Hill/Interamericana de españa, S.A.U.

⁷ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica (Segunda ed.)*. Madrid, España(pp.2-10): McGraw-Hill/Interamericana de españa, S.A.U.

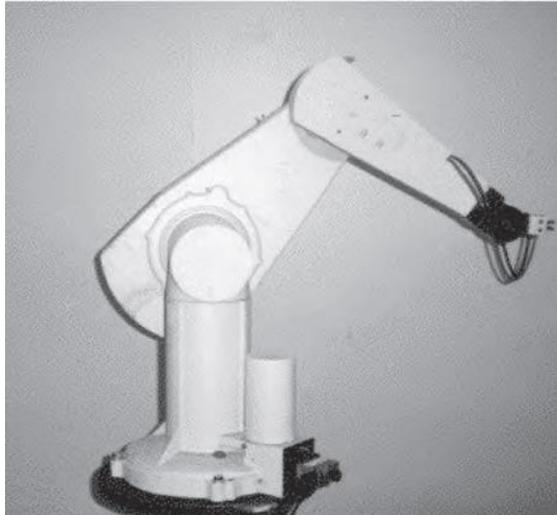


Figura 2.1. Robot PUMA (Barrientos, Peñin , Balaguer , & Aracil, 2007)

2.2.1.1. Clasificación de los robots

Si bien es cierto, los robots pueden clasificarse de diversas maneras ya sea por el tipo de configuración, el número de ejes que este pueda tener, de acuerdo al tipo de actuadores, el área de aplicación, etc. A continuación, se detalla la clasificación en base al tipo de control todas regidas por la norma ISO 837 mostradas en la figura 2.2.

Robot teleoperado: el robot es controlado de forma manual por un operador, gracias a ello el control se puede realizar desde diferentes partes del mundo de forma remota en tiempo real.

- a) Robot Secuencial: En este tipo de robots solo se puede controlar una serie de movimientos en base a dos puntos establecidos (*Point to Point: PTP*). Los manipuladores de tipo neumático son un claro ejemplo de este tipo de control.
- b) Robot controlado por trayectoria: Robots controlados en base a una programación de trayectoria previa, lo cual permite que el robot siga una trayectoria continua dentro de su área de trabajo, usado en la industria para trabajos cíclicos.

c) Robot adaptativo: el robot se adapta al entorno de trabajo para cumplir su tarea programada, esto se logra gracias a sensores, visión artificial, redes neuronales, entre otros. Actualmente aplicados a la industria de la manufactura. [8]

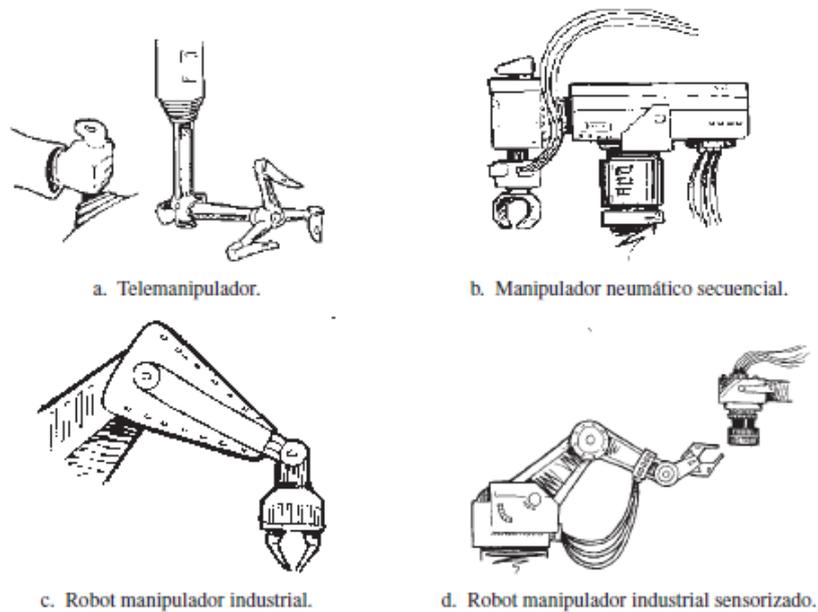


Figura 2.2. Bocetos de algunos tipos de robots. (Barrientos, Peñin , Balaguer , & Aracil, 2007)

2.2.1.2. Grados de libertad de un robot

Un robot está constituido por una serie de elementos mecánicos o también denominados eslabones y unidos mediante articulaciones que permiten un movimiento relativo entre dos eslabones consecutivos. Se usan términos como cuerpo, brazo, codo y muñeca para hacer referencia a los distintos elementos que conforman un robot.

El movimiento que puede realizar un robot puede ser de desplazamiento, giro, o una combinación de ambos. Franz Reuleaux en el año 1876 ideó los movimientos relativos posibles entre dos elementos en contacto. A los

⁸ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica (Segunda ed.)*. Madrid, España (pp. 26-28): McGraw-Hill/Interamericana de España, S.A.U.

contactos que se producen entre superficies lo denominó pares inferiores (*low pairs*), por otro lado, si el contacto es puntual o lineal lo denominó pares superiores. En la figura 2.3 se muestran los movimientos posibles definidos por Reuleaux.^[9]

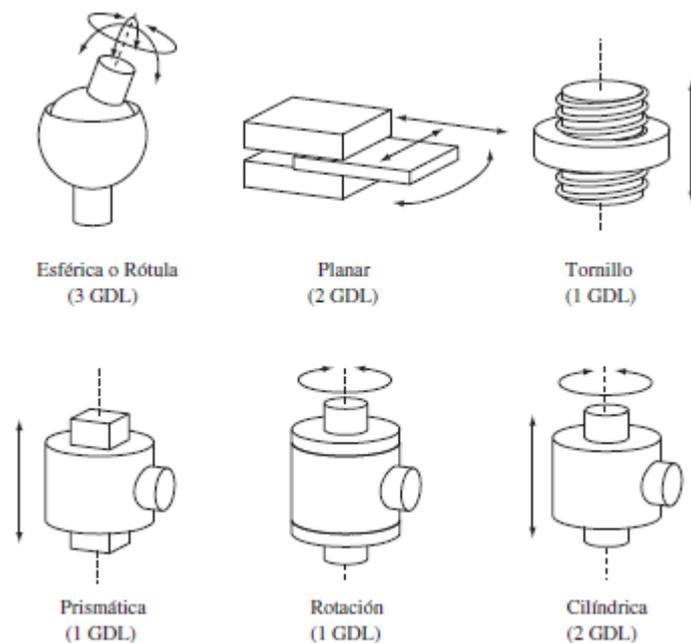


Figura 2.3. Los 6 pares inferiores de Reuleaux. (Barrientos, Peñin , Balaguer , & Aracil, 2007)

Una cadena cinemática es un camino por el cual se puede llegar a un eslabón, se denomina cadena cinemática abierta cuando solo existe un camino, mientras que, cuando existan dos caminos para llegar a cualquier otro eslabón, tomará el nombre de cadena cinemática cerrada, ambos ejemplos mostrados en la figura 2.4.

⁹ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica* (Segunda ed.). Madrid, España (pp.31-34): McGraw-Hill/Interamericana de españa, S.A.U.

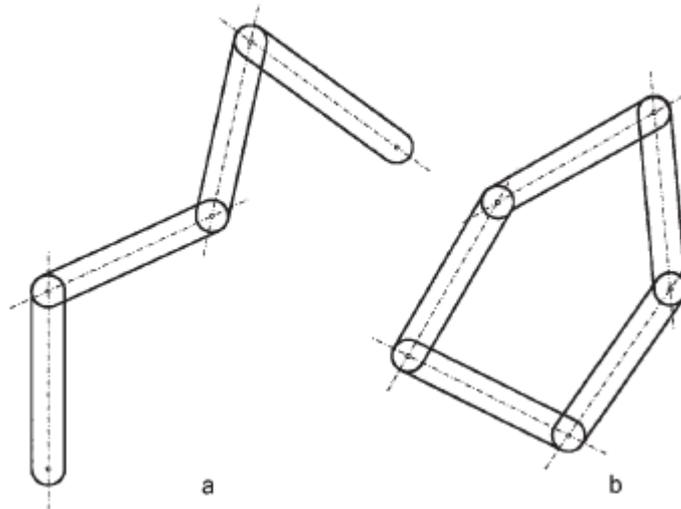


Figura 2.4. Cadena cinemática a) abierta y b) cerrada. (Barrientos, Peñín , Balaguer , & Aracil, 2007)

Para la obtención de los grados de libertad de un robot se aplica la fórmula de Grubler mostrada en la ecuación 2.1.

$$NGDL = \lambda \cdot (n - j - 1) + \sum_{i=1}^j f_i \quad (2.1)$$

Donde:

λ = GDL del espacio de trabajo (Típicamente tres en el plano, seis en el espacio).

n = Número de eslabones (debe de incluirse el eslabón fijo o base).

j = número de articulaciones.

f_i = Grados de libertad permitidos a la articulación i .

La mayoría de robots manipuladores modernos son cadenas cinemáticas abiertas y las articulaciones de tipo rotación o prismática. Debido a esto, es sencillo obtener el número de grados de libertad ya que, al desarrollar la ecuación anterior, el valor coincide con el número de articulaciones que se propuso. Así mismo, dichos robots poseen una estructura angular, conocida como articular, dentro de la industria también se pueden encontrar robots de

tipo cartesiana y SCARA, siendo estos 3 los más usados. Sin embargo, en la figura 2.5 se aprecia las diferentes estructuras de robots que se han diseñado a lo largo de la historia.^[10]

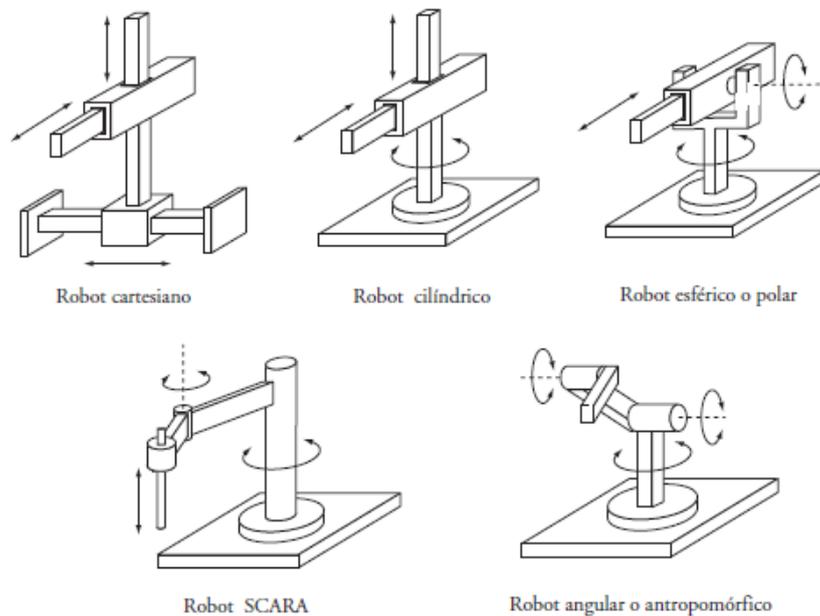


Figura 2.5. Estructuras de robots más usadas en la industria. (Barrientos, Peñin , Balaguer , & Aracil, 2007)

Los grados de libertad dependerán de la complejidad de la tarea programada al robot y del alcance que este debe de tener para cumplirla, dentro de la industria se hace uso de 6 GDL para aquellas tareas que impliquen demasiados obstáculos en la trayectoria. Sin embargo, muchos de ellos solo cuentan con 5 o 4 GDL ya que son actividades cíclicas donde el área de trabajo está despejada para que el robot pueda desenvolverse sin problemas.

En la figura 2.6. se muestra a dos robots que presentan diferentes tipos de configuraciones mecánicas para lograr una tarea específica, una de ellas es un robot con 3GDL el cual debe de surcar un obstáculo para cumplir con su labor.

¹⁰ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica* (Segunda ed.). Madrid, España(pp: 34) McGraw-Hill/Interamericana de españa, S.A.U.

Por otro lado, se tiene un robot de 3 GDL que realiza un desplazamiento lineal y dos movimientos articulares. [11]

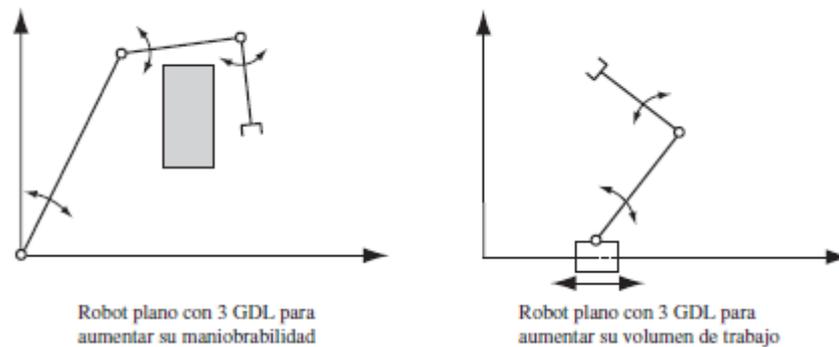


Figura 2.6. Robots planares. (Barrientos, Peñin , Balaguer , & Aracil, 2007)

2.2.2. Redes Neuronales Artificiales:

Las redes neuronales artificiales (RNA) surgieron de la necesidad del ser humano para modelar y comprender diversas funciones que son realizadas por los seres vivos, una de ellas, replicar el comportamiento de las neuronas de nuestro cerebro.

En el científico Ramón y Cajal de procedencia española, logró describir los distintos tipos de neuronas en el siglo XIX, así mismo, plantea que nuestro sistema nervioso está constituido por neuronas individuales y que la comunicación entre ellas es posible gracias a la sinapsis, es decir, por contacto entre neuronas.

Posteriormente se desarrollaron modelos matemáticos para describir el comportamiento de una neurona, se realizaron algunos intentos para su implementación, dicho desarrollo no tuvo un impacto histórico debido a la limitada tecnología. En aquel entonces, aún no se contaba con los sistemas computarizados. Años después, gracias a las computadoras, fue factible poder

¹¹ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica* (Segunda ed.). Madrid, España(pp.35-37): McGraw-Hill/Interamericana de españa, S.A.U.

desarrollar simulaciones de redes neuronales artificiales debido a la gran capacidad de procesamiento. Sin embargo, Este periodo de investigación y desarrollo de las RNA no fue muy prolongado, las RNA perdieron importancia y prestigio debido a que se encontraron diferentes limitaciones en la lógica para solucionar problemas complejos con el Perceptrón, esencialmente aquellos problemas en las que sus variables no son linealmente separables, todas estas observaciones fueron presentadas por Minsky y Papert en el año 1969 y como resultado los científicos perdieron interés en profundizar en el tema. No obstante, varios investigadores continuaron con el desarrollo de las RNA mediante métodos inspirados en el sistema computacional neuromorfológico, aplicados a solucionar problemas de identificación de patrones y clasificaciones de diferentes clases. Dentro de este periodo se obtuvo grandes resultados, Steve Grossberg y Gail Carpenter propusieron la teoría de la resonancia adaptativa, ART (Adaptative Resonance Theory) (1976). Por otro lado, Anderson y Kohonen en 1982 desarrollaron diferentes técnicas para un aprendizaje asociativo, el popular método de aprendizaje denominado Back-Propagation fue desarrollado por Paul Werbos en 1982 convirtiéndose en el método más utilizado actualmente en la arquitectura de Perceptrón multicapa. El resurgimiento del interés por las RNA fue durante el final de la década de los setenta y principios de los ochenta, algunos de los factores que influenciaron este movimiento fue la aparición de libros y conferencias que muestran las capacidades de solución a problemas con esta técnica dentro de distintas áreas.

En la actualidad, las RNA han tenido un impacto positivo dentro del área comercial e industrial, las investigaciones continúan para ampliar el campo de

aplicación. Incluso, los nuevos circuitos integrados son basados en RNA los cuales pueden resolver problemas aún más complejos. [12]

2.2.2.1. La neurona

Una neurona es una célula principal del sistema nervioso y por su morfología son semejantes a las células biológicas, el tamaño del cuerpo celular de una neurona es de alrededor de 5 a 10 micras de diámetro, de ella se desprende el axón, principal rama de la neurona, las ramas de menor dimensión son denominadas dendritas. La función principal de una neurona es la de recibir, procesar y enviar información hacia otras neuronas vecinas gracias a la reacción química que se produce dentro de su membrana plasmática. [13]

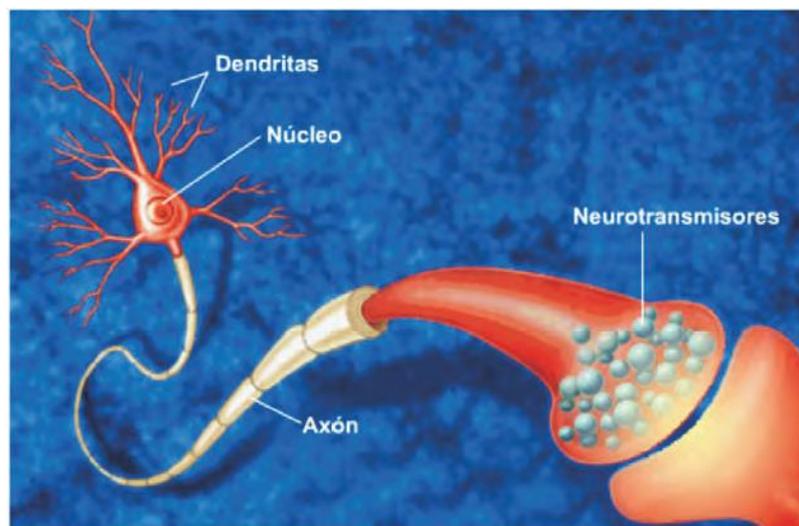


Figura 2.7. La neurona biológica. (Caicedo & López, 2009)

Las neuronas se diferencian del resto de células vivas por su capacidad de comunicación. Tomando como referencia la Figura 2.7, los encargados de recibir la información de entrada son las dendritas, mientras que el cuerpo

¹² Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 13-17): Programa Editorial Universidad del Valle.

¹³ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 17): Programa Editorial Universidad del Valle.

celular combina y emite las señales de salida. El axón es el cuerpo de la neurona, encargado de distribuir la información a las siguientes neuronas.

Una neurona, por lo general, recibe una gran cantidad de información provenientes de miles de otras neuronas y, a su vez, envía información a otros miles de neuronas más, formando una gran red de comunicación. Según investigaciones, los especialistas estiman que en el cerebro humano se generan 10^{15} conexiones. [14]

En la tabla 2.1 se realiza una comparativa entre un ordenador clásico secuencial con una arquitectura Von Neumann y un sistema biológico neurona, teniendo como finalidad, dar a conocer las diferentes características de cada una de ellas como la capacidad para poder procesar datos, retener información, y el poder de su unidad de procesamiento.

Tabla 2.1 Comparativa entre un ordenador secuencial y el sistema biológico neuronal

Característica	Computador Secuencial	Sistema biológico neuronal
Unidad de Procesamiento	Compleja Alta velocidad Una sola unidad	Simple Baja velocidad Muchas unidades
Memoria	Separada del procesador Localizada Direccionable aleatoriamente	Integrada dentro del procesador Distribuida Direccionable por contenido
Procesamiento de Datos	Centralizado Secuencial Instrucciones almacenadas en Programas	Distribuido Paralelo Capacidad de aprendizaje
Confiabilidad	Muy vulnerable ante fallos	Robusto ante fallos
Ambiente de operación	Bien definido	Puede ser ambiguo

Fuente (Caicedo & López, 2009)

¹⁴ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 18): Programa Editorial Universidad del Valle.

2.2.2.2. Neurona biológica

En una neurona biológica se puede encontrar dos tipos de señales, eléctricas y químicas. La señal que se genera en la neurona y que recorre lo largo del axón es eléctrica, por otro lado, la señal que transmite el axón hacia otras neuronas mediante neurotransmisores que se encuentran en las uniones de las neuronas denominada sinapsis, son de origen químico. En la figura 2.8 se observa la sinapsis entre dos neuronas y la reacción química que se genera para poder transmitir la información.[¹⁵]

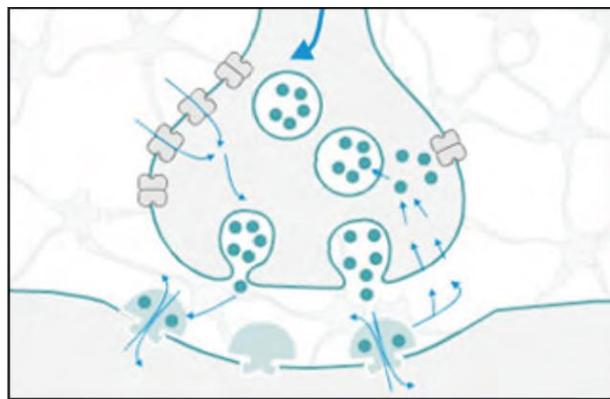


Figura 2.8. Neurotransmisores en una sinapsis (Caicedo & López, 2009)

2.2.2.3. Neurona artificial

Teniendo como referencia la fisiología de una neurona biológica, en las siguientes líneas se propone un modelo de una neurona artificial, para una mejor representación, la figura 2.9 muestra el modelo de una N.A. en forma de bloques.

¹⁵ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 19): Programa Editorial Universidad del Valle.

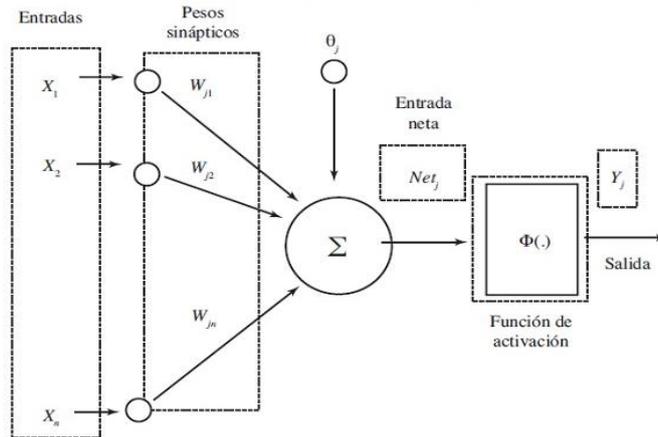


Figura 2.9. Modelo de una Neurona Artificial (Caicedo & López, 2009)

La neurona artificial recibe las entradas que pueden provenir de un sistema sensorial externo, o de otra conexión de neuronas y se define con el vector de entradas $X = [X_1 \ X_2 \ \dots \ X_n]$.

Los pesos sinápticos cumplen el rol de emular la sinapsis entre neuronas biológicas, de igual manera, los pesos se definen por un vector W . El parámetro θ representa a una configuración neuronal cuyo valor es independiente de las entradas, una de sus aplicaciones es para resolver problemas con Perceptrón con valores que sean linealmente no separables.

La entrada neta se produce por la sumatoria de la multiplicación de entradas por los pesos sinápticos tal como se muestra en la ecuación 2.2.

$$Net_j = \sum_{i=1}^N x_i w_j + \theta_j \quad (2.2)$$

$$Net_j = x_1 w_1 + x_2 w_2 + x_3 w_3 + \dots + x_N w_N + \theta_j \quad (2.3)$$

$$Net_j = w^T X_j + \theta_j \quad (2.4)$$

La neurona dependerá de la función de activación para ser activada, por lo general se utiliza la función conocida como *heaviside*, una función de escalón unitario donde la salida será 1 cuando los valores de Net sean mayores que

zero, in case contrary the value of the output will be 0. To represent in the equation 2.5 the activation function will use the expression $Fact_j$.

$$y_j = Fact_j(Net_j) \quad (2.5)$$

Existen diferentes tipos de funciones de activación que ayudan a resolver distintos problemas, en la figura 2.10 se muestran alguna de ellas, como el del tipo escalón, lineal, etc. [16]

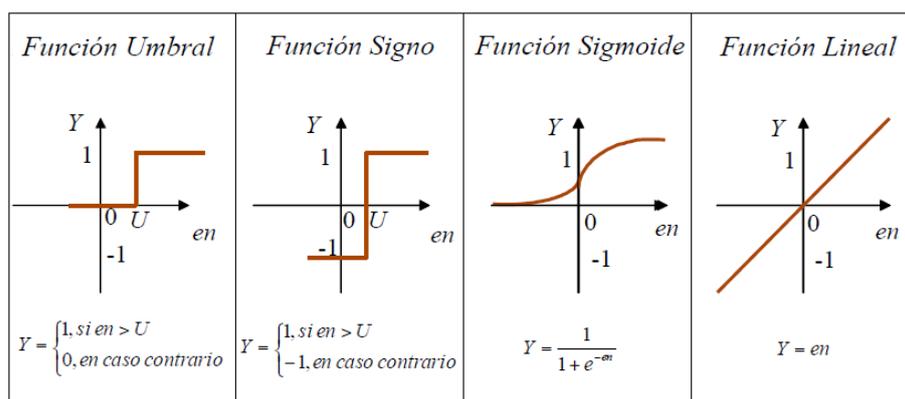


Figura 2.10. Funciones de activación (Caicedo & López, 2009)

2.2.2.4. Red neuronal Artificial

Una neurona artificial posee una baja capacidad de procesamiento y carece de aplicaciones, el verdadero potencial radica en la interconexión de las mismas formando una gran red de procesamiento, tal como sucede en el cerebro humano. [17].

El científico finlandés T. Kohonen las define como redes de elementos simples que usualmente son adaptativos, conectados de forma masiva que intentan comunicarse con el mundo real tal como lo hace un sistema nervioso biológico.

F. Caicedo y J. López consideran que las redes neuronales se basaron en la biología natural teniendo como referencia de desarrollo los siguientes aspectos.

¹⁶ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 21-23): Programa Editorial Universidad del Valle.

¹⁷ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 23-24): Programa Editorial Universidad del Valle.

1. La unidad de procesamiento mínima del cerebro está a cargo de la neurona.
2. La información se transmite entre neuronas gracias a las conexiones entre ellas.
3. Los pesos sinápticos son los encargados de multiplicar la señal transmitida, este proceso se encuentra en cada enlace de conexión.
4. El proceso para determinar la señal de salida se realiza a través de una función de activación que por lo general no es una función lineal, se realiza una sumatoria de las entradas multiplicadas por los pesos sinápticos de cada rama. [18]

2.2.2.5. Arquitectura de una RNA

La arquitectura de una RNA es la forma de organizar las neuronas artificiales en su interior y está relacionado con el algoritmo empleado para entrenar la red.

Existen diferentes tipos de arquitectura, entre ellas las redes monocapa, multicapa, redes feedforward y redes recurrentes. Los cuales se detallan a continuación.

a) Red Monocapa

En la figura 2.11 se observa dos niveles de neuronas, siendo el primero asociado al vector de entrada X ; sin embargo, no se considera como una capa debido a que es directamente una interfaz de comunicación de la neurona con los datos de entrada.

La siguiente columna corresponde a la capa de salida, como se aprecia, en esta arquitectura solo se dispone de una capa de procesamiento. Así mismo,

¹⁸ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 23-24): Programa Editorial Universidad del Valle.

se observa la conexión total entre la capa de entrada y la capa de salida. La capa de salida posee m neuronas cada una con una función de activación los cuales permitirá obtener las salidas Y . [19]

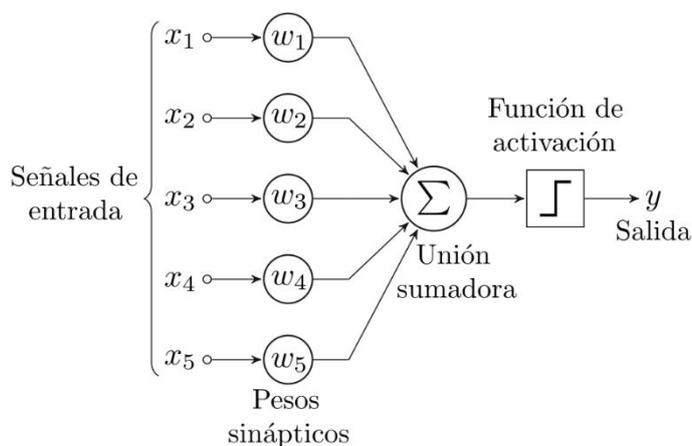


Figura 2.11. RNA monocapa. (Caicedo & López, 2009)

b) Rede Multicapa

En esta arquitectura la red posee un nivel de entrada con n neuronas y una capa de salida con m neuronas, el comportamiento es similar a una red monocapa. Sin embargo, aquí surgen las capas ocultas compuestas por h neuronas. Se denomina capa oculta debido a que estas neuronas no tienen contacto directo con los datos de entrada ni con los datos de salida. El número de capas ocultas dependerá de la complejidad del problema a solucionar, el diseño de una RNA multicapa se aprecia en la figura 2.12, donde se tiene una capa de entrada, una capa oculta formada por h neuronas y una capa de salida de m neuronas.

¹⁹ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 25-26): Programa Editorial Universidad del Valle.

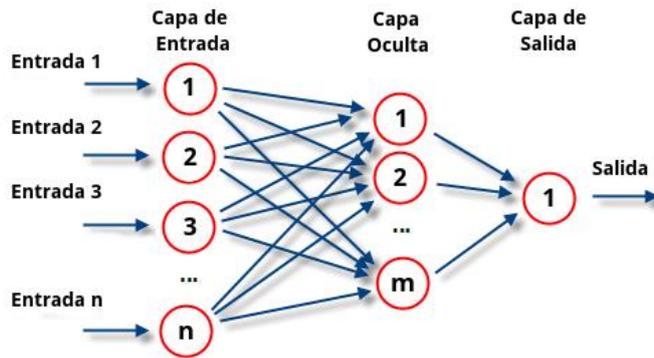


Figura 2.12. RNA multicapa.
(Caicedo & López, 2009)

c) Red feedforward

Una red neuronal de tipo feedforward se caracteriza por el conjunto de neuronas que reciben información multivariable, la salida procesada puede ser también multivariable. En la arquitectura mostrada en la figura 2.13, las neuronas hacen fluir la información en un solo sentido haciendo posible restringir el paso hasta que se genere la respuesta de salida. [20]

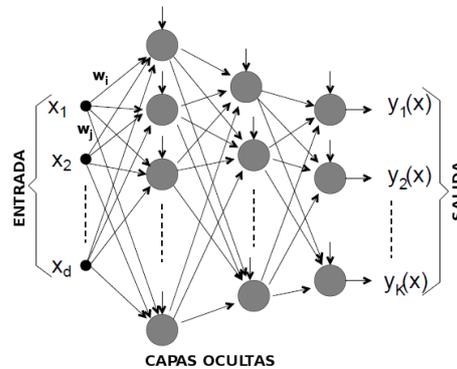


Figura 2.13. Diagrama arquitectónico de una red neuronal tipo feedforward(RNAf)
(Vásquez López, 2014)

d) Redes recurrentes

La arquitectura de las redes recurrentes hace posible que la información no siempre fluya en un sentido, esto debido a que tienen la propiedad de poder

²⁰ Vásquez López, J. P. (2014). Red Neuronal Feedforward como Estimador de Patrones de Corrientes en el interior del Puerto de Manzanillo sujeto a la acción de Tsunamis. *Publicación técnica No. 406*.(pp. 5-7).

realimentarse hacia capas anteriores a través de conexiones sinápticas. En la figura 2.14. se muestra dos configuraciones de redes neuronales recurrentes monocapa, donde la primera figura 2.14. a, muestra en la salida pasa por una unidad de retardo para posteriormente ser llevado a otras neuronas, excepto a sí mismas. En la figura 2.14. b, se aprecia en las entradas la información que proviene de unidades de retardos, los cuales transportan sus señales de estímulos a las neuronas de salida. A este conjunto de neuronas se denomina neuronas de contexto. [21]

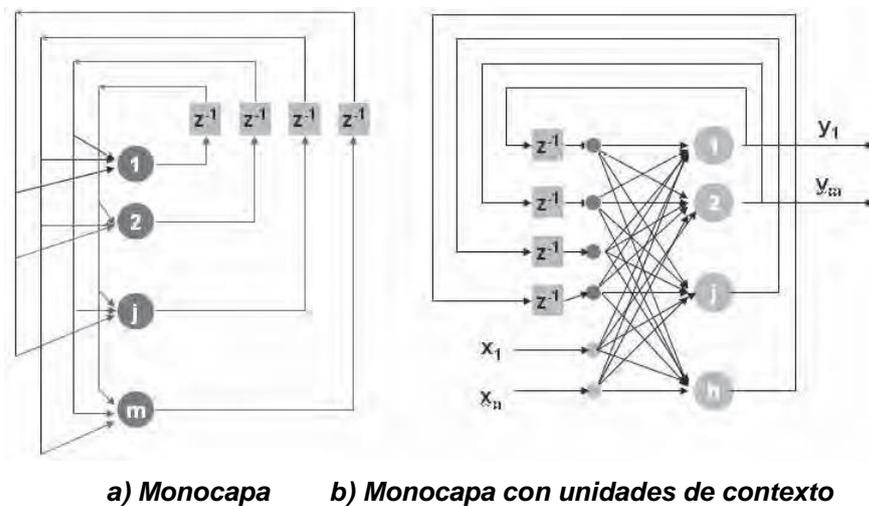


Figura 2.14. RNA tipo recurrentes. (Caicedo & López, 2009)

2.2.3. Rayo tractor sónico

El rayo tractor sónico utiliza el poder de las ondas mecánicas de sonido para mantener las partículas suspendidas en el aire. Estas ondas emiten radiación acústica, cuando estas fuerzas son suficientemente fuertes y convergen de todas direcciones encapsula la partícula y hace posible atrapar de forma estable objetos pequeños. Fue desarrollado por primera vez en la Universidad de Bristol, este descubrimiento da paso a múltiples aplicaciones, como la

²¹ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 27-29): Programa Editorial Universidad del Valle.

manipulación de cápsula de medicamentos, instrumentos microquirúrgicos, el transporte sin contenedor, etc.

Por muchos años, los investigadores se han visto frustrados por no poder mantener en suspensión partículas de orden superior, sin embargo, los ingenieros de la Universidad de Bristol utilizaron un nuevo enfoque, empleando los denominados *tornados de sonido*: vórtices que fluctúan rápidamente simulando a un tornado con un sonido fuerte que rodea un núcleo silencioso. Para llegar a estos logros se trabajó con ondas ultrasónicas de aproximadamente 40KHz, dando como resultados la suspensión de una esfera de poliestireno de un diámetro de dos centímetros y como resultado en la figura 2.15. muestra la simulación de cómo es encapsulada la esfera dentro del vórtice de sonido. [22]

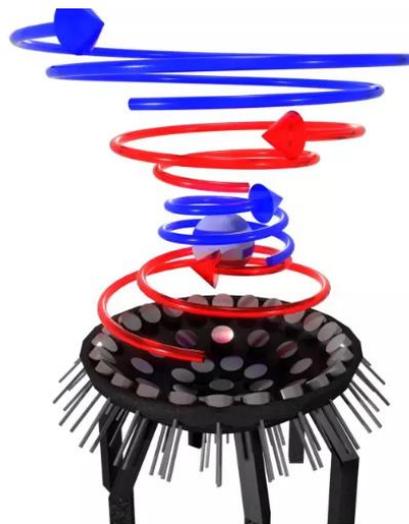


Figura 2.15. Vórtice de sonido encapsulando esfera de poliestireno.

Para los cálculos de los elementos acústicos holográficos para su manipulación de objetos levitados se ha considerado las investigaciones realizadas por

²² Universidad de Bristol. (2018, 21 de enero). El haz de tractor acústico más poderoso del mundo podría allanar el camino para la levitación de humanos. ScienceDaily . Obtenido el 15 de septiembre de 2018 en www.sciencedaily.com/releases/2018/01/180121221627.htm

Nature Communications, con el fin de obtener la información clave para el modelamiento matemático requerido

El primer análisis a realizar se basa en la caracterización de un punto de levitación, teniendo en cuenta la fuerza de radiación acústica (F) ejercida sobre una pequeña partícula esférica se puede calcular a partir del gradiente del potencial de Gor'kov.

$$F = -\nabla U \quad (2.6)$$

Donde:

F= Radiación Acústica

∇U = Gradiente del potencial de Gor'kov

Se caracteriza un punto de levitación como un máximo del operador Laplaciano (convergencia del gradiente) aplicado al potencial de Gor'kov, es decir, un punto hacia el cual convergen todas las fuerzas.

$$\nabla^2 U = U_{xx} + U_{yy} + U_{zz} \quad (2.7)$$

Dónde:

$$U_x = \frac{\partial U}{\partial x}, U_{xx} = \frac{\partial^2 U}{\partial x^2}$$

$$U_y = \frac{\partial U}{\partial y}, U_{yy} = \frac{\partial^2 U}{\partial y^2}$$

$$U_z = \frac{\partial U}{\partial z}, U_{zz} = \frac{\partial^2 U}{\partial z^2}$$

El potencial de Gor'kov (U), en términos de la presión acústica compleja (p) y sus derivadas espaciales, es dado por:

$$U = K_1(|p|^2) - K_1 \left(|p_x|^2 + (|p_y|^2) + (|p_z|^2) \right) \quad (2.8)$$

$$U = K_1(|p|^2) - K_1 \left(|p_x|^2 + (|p_y|^2) + (|p_z|^2) \right) \quad (2.9)$$

$$K_1 = \frac{1}{4}V \left(\frac{1}{c_0^2 \rho_0} - \frac{1}{c_p^2 \rho_p} \right) \quad (2.10)$$

Dónde:

V : Volumen de la partícula esférica.

ω : Frecuencia de emisión de ondas.

ρ : Densidad de la partícula.

c : Velocidad del sonido.

Siendo los subíndices '0' y 'p' referencia del medio en que se encuentra la partícula y el material en sí.

Si un transductor acústico emite con una frecuencia y amplitud constantes, entonces la presión compleja que el transductor crea en un punto puede expresarse como:

$$p^j = e^{i\phi^j} M^j \quad (2.11)$$

Dónde:

ϕ^j : Retardo de fase del transductor.

M^j : Número complejo constante para un transductor dado y punto en el espacio.

Con las ecuaciones planteadas en este modelamiento para los hologramas acústicos, se resumirán las ecuaciones matemáticas en la expresión 2.12.

$$U = 2\pi R^3 \left(\frac{\rho^2}{3\rho_m c_m^2} - \frac{\rho_m(u)^2}{2} \right) \quad (2.12)$$

Dónde:

R: Radio de la esfera

ρ_m : Densidad del medio

c_m : Velocidad del sonido

$\rho^2, (u)^2$: Amplitudes cuadradas medias de la presión y la velocidad en el centro de la esfera.

Por lo tanto, el gradiente del potencial representa las fuerzas que actúan sobre una pequeña esfera. Entonces, un punto de levitación es una posición a la que convergen todas las fuerzas; Es decir, un mínimo del potencial.

Estas ondas estacionarias se crean cuando dos ondas de la misma frecuencia que se encuentran en direcciones opuestas se intersectan.

La capacidad de controlar las ondas estacionarias permite la traslación de los objetos levitando que están contenidos dentro, en este caso los insumos químicos a tratar.

2.3. Modelamiento

Para llevar a cabo el correcto modelamiento de la tesis presentada, se ha propuesto realizar la siguiente metodología la cual está descrito por los siguientes aspectos:

- Diagrama de bloques.
- Diagrama pictórico.
- Modelamiento matemático.
- Diagramas de Flujo.
- Desarrollo de Algoritmos.
- Modelamiento por parámetros.

2.3.1. Diagrama de bloques

El diagrama de bloques muestra la secuencia que debe de tener el sistema para lograr el objetivo, dentro de cada bloque, las acciones que se van a

realizar. En la figura 2.16 se muestra el diagrama de bloques diseñada por los autores de la tesis.



Figura 2.16. Diagrama de bloques (diseñada por: Autores de la tesis)

2.3.2. Diagrama pictórico

En la figura 2.17 se tiene el diagrama pictórico, el cual, es una vista a profundidad de cada subsistema definidas en el diagrama de bloques, donde se puede apreciar gráficamente la parte fundamental de cada bloque.



Figura. 2.17. Diagrama pictórico (diseñado por: Autores de la tesis)

2.3.3. Modelamiento matemático

2.2.3.1. Cinemática de un robot

En el capítulo 4 del libro Fundamentos de robótica, los autores definen la cinemática como el estudio del movimiento del robot respecto a un sistema de referencia sin considerar las fuerzas que intervienen. Así mismo, esta ciencia se interesa por analizar el movimiento espacial en función del tiempo, como también, la relación entre la posición y la orientación.

En el desarrollo de la cinemática del robot se tiene dos parámetros importantes, el primero la cinemática directa. El cual consiste en hallar la posición y orientación del extremo final del robot teniendo como referencia un sistema de coordenadas. Por otro lado, se encuentra la cinemática inversa, aquella que determina la configuración de un robot para una posición y orientación conocidas.[²³]

Denavit-Hartenberg en 1955 propusieron un método matricial para obtener el modelo cinemático directo de un robot. En la ecuación 2.13. se establece los parámetros que se deben de considerar para su solución.

$${}^i - 1_{A_i} = Rotz(\theta_i)T(0,0,d_i)T(a_i,0,0)Rotx(\alpha_i) \quad (2.13)$$

Donde:

- θ : Es el ángulo de rotación alrededor del eje Z_{i-1} .
- $d_i(0,0,d_i)$: vector correspondiente a la distancia de traslación a lo largo del eje Z_{i-1} .
- a_i ; vector $a_i(a_i,0,0)$: Vector de traslación a lo largo del eje x_i con una distancia a_i .
- α_i : Ángulo de rotación alrededor del eje x_i .

²³ Barrientos, A., Peñin, L., Balaguer, C., & Aracil, R. (2007). *Fundamentos de Robótica* (Segunda ed.). Madrid, España (pp. 119-120) : McGraw-Hill/Interamericana de España, S.A.U.

Luego de realizar el producto entre matrices se obtiene la siguiente expresión 2.14.

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

$$= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

Donde las variables θ_i , a_i , α_i son los parámetros D-H del eslabón i que relacionan cada uno de los eslabones que conforman el robot. Finalmente, Denavit-Hartenberg establecen un algoritmo para la solución al problema cinemático directo. Todos los pasos para resolver la cinemática de un robot aplicando D-H se aprecian de manera gráfica en la figura 2.18.[24]

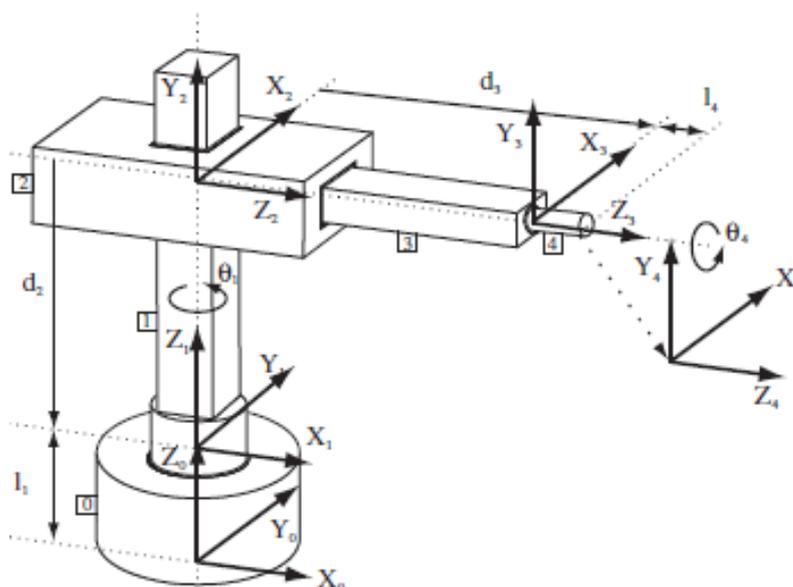


Figura 2.18. Robot cilíndrico aplicando algoritmo D-H. (Barrientos, Peñin , Balaguer , & Aracil, 2007)

²⁴ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica* (Segunda ed.). Madrid, España (pp. 125-126) : McGraw-Hill/Interamericana de españa, S.A.U.

Al seguir los pasos establecidos por D-H. se procede a realizar la tabla 2.2 para determinar sus parámetros y posteriormente efectuar las matrices correspondientes.

Tabla 2.2. Parámetros D-H. para la figura 4.2.

Articulación	θ	d	a	α
1	q_1	l_1	0	0
2	90	d_2	0	90
3	0	d_3	0	0
4	q_4	l_4	0	0

Fuente (Barrientos, Peñin , Balaguer , & Aracil, 2007)

2.2.3.2. Dinámica del Robot

La obtención del modelo dinámico mediante LaGrange muestra ventaja sobre la formulación Newtoniana al aumentar el número de grados de libertad del robot.

A continuación, se presenta la ecuación 2.15 que rige la dinámica de un robot.

$$\tau = D\ddot{q} + H + C \quad (2.15)$$

donde:

\ddot{q} : vector de coordenadas articulares

τ : vector de fuerza aplicada a cada articulación

Dq : matriz de inercias cuyos elementos son en función de q.

$H(q, \dot{q})$: matriz ($n \times 1$) de fuerza de Coriolis, dependientes de q y \dot{q} .

$C(q)$: matriz ($n \times 1$) de fuerzas de gravedad, dependiente de q

n: número de grados de libertad del robot. [25]

²⁵ Barrientos, A., Peñin , L., Balaguer , C., & Aracil, R. (2007). *Fundamentos de Robótica* (Segunda ed.). Madrid, España (pp. 218-219) : McGraw-Hill/Interamericana de españa, S.A.U.

2.2.3.3. Salida de una red neuronal

Al tener una red neuronal es importante determinar el valor numérico de la salida. La ecuación 2.16 muestra cómo se puede obtener dicho valor.

$$Neta_{pj} = \sum_{i=1}^N w_{ij}x_{pi} + \theta_j \quad (2.16)$$

Posteriormente el valor obtenido es evaluado en la ecuación 2.17 por la función de activación para determinar la salida y de la red neuronal.

$$y_{pj} = f_j(Neta_{pj}) \quad (2.17)$$

2.2.3.4. Aprendizaje de una red neuronal

El modelamiento matemático para el aprendizaje de una red neuronal se describe matemáticamente por la ecuación 2.18.

$$w(t + 1) = w(t) + \Delta w(t) \quad (2.18)$$

Donde:

$w(t + 1)$: Valor actualizado del peso sináptico.

$w(t)$: Valor actual del peso sináptico.

$\Delta w(t)$: Variación del peso sináptico.^[26]

2.2.3.5. Error del p-ésimo patrón de la red neuronal

Determinar el error procedente de las actualizaciones de los pesos $w(t)$ de la red neuronal es importante para poder obtener un correcto aprendizaje. Si al desarrollar la ecuación 2.19 el error es diferente de cero, los pesos se actualizan para poder llegar a obtener un error de cero.

$$e_p = \sum_{j=1}^M (d_{pj} - y_{pj})^2 \quad (2.19)$$

²⁶ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 28-29): Programa Editorial Universidad del Valle.

Donde:

p : P-ésimo patrón del conjunto de entrenamiento

d_p : Valor deseado de salida para el p-ésimo patrón

y_p : Valor de salida de la red para el p-ésimo patrón [²⁷]

2.3.4. Diagrama de flujo

Para poder desarrollar el diagrama de flujo se ha detallado la secuencia que deberá de tener el sistema. Los cuales son:

- a) Solicitar cierre de cabina.
- b) El sistema mapea el área de trabajo.
- c) Determina la posición del objeto.
- d) Genera la trayectoria.
- e) El brazo robótico inicia el movimiento generado.
- f) Activación del rayo tractor sónico para atrapar el objeto.
- g) Graduación de la intensidad para mantenerlo estable en suspensión.
- h) Acercamiento del objeto hacia la cámara fija.
- i) Determinar el elemento químico.
- j) Colocarlo en su respectivo envase.
- k) Volver a la posición segura.

Gracias a la lista secuenciada previa, se obtiene una vista panorámica del sistema desarrollado. En la figura 2.19 se muestra un bosquejo del diagrama de flujo, el cual permitirá poder desarrollar un algoritmo de programación para dar solución a lo planteado.

²⁷ Caicedo, E., & López, J. A. (2009). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Santiago de Cali, Colombia (pp. 44-46): Programa Editorial Universidad del Valle.

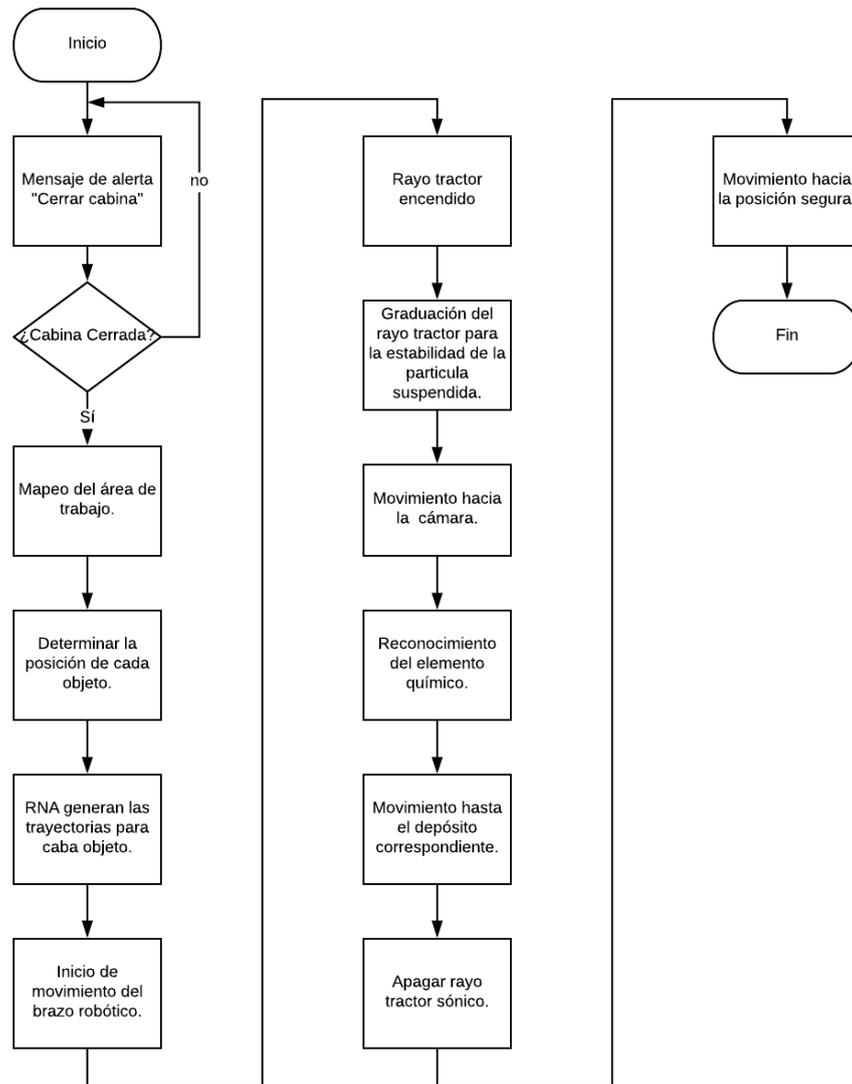


Figura 2.19. Diagrama de flujo (Elaborado por: Autores de la tesis.)

2.3.5. Modelamiento por parámetros

Del ítem desarrollado *Modelamiento matemático*, se ha seleccionado las ecuaciones de los sistemas a emplear.

Para empezar, las ecuaciones referentes al modelamiento del robot son esenciales para poder realizar las simulaciones respectivas. La ecuación 2.20 corresponde a la lógica desarrollada por Denavit-Hartenberg, así mismo se empleará el algoritmo desarrollado por estos autores ya que facilitan el desarrollo de sistemas con varias articulaciones, los parámetros D-H. son la base de este algoritmo.

$$i - 1_{A_i} = Rotz(\theta_i)T(0,0,d_i)T(a_i, 0, 0)Rotx(\alpha_i) \quad (2.20)$$

Por otra parte, desarrollar el modelo dinámico del robot de 4GDL es importante para obtener parámetros de fuerzas y torques, para ello se hará uso de la metodología de desarrollo Lagrangiano, una ecuación relacionada al torque del sistema, relacionando la fuerza de gravedad, la inercia del robot y la pérdida por Coriolis.

$$\tau = D\ddot{q} + H + C \quad (2.21)$$

Pasando a otro punto de desarrollo, para el desarrollo de las redes neuronales artificiales y la lógica a emplear, se harán uso de las siguientes ecuaciones. La primera de ellas determina la salida de una red neuronal (Ec. 2.22).

$$Neta_{pj} = \sum_{i=1}^N w_{ij}x_{pi} + \theta_j \quad (2.22)$$

Es necesaria contar con la función de activación que evalúe el valor obtenido de la ecuación 2.23.

$$y_{pj} = f_j(Neta_{pj}) \quad (2.23)$$

Además, para el entrenamiento de la RNA se necesita emplear el algoritmo matemático 2.24, en donde existe una actualización constante de pesos sinápticos para llegar a obtener un error nulo.

$$w(t + 1) = w(t) + \Delta w(t) \quad (2.24)$$

Por último, determinar el error de la red neuronal en el aprendizaje es de suma importancia ya que permitirá saber si los pesos sinápticos deben de actualizarse, este error es definido la ecuación 2.25:

$$e_p = \sum_{j=1}^M (d_{pj} - y_{pj})^2 \quad (2.25)$$

2.4. Tabla de parámetros

En base al modelamiento matemático, la tabla de parámetros indica las variables que en el próximo capítulo se deben de determinar para dar cumplimiento con lo propuesto en la tesis.

A continuación, se muestra la tabla 2.3 de parámetros donde la primera columna indica la simbología de las variables, la segunda columna indica la procedencia de la variable y por último se escriben las unidades de las variables, en muchos casos de ellos, por tratarse de matrices y vectores.

Tabla 2.3. Parámetros generales

Símbolos	Descripción	Unidades
θ	Ángulo de rotación	rad
d_i	Vector de traslación, eje Z	-
a_i	Vector de traslación, eje X	-
α_i	Vector de rotación, eje X	-
\ddot{q}	Vector de coordenadas articulares	-
τ	Fuerza aplicada en la articulación	-
Dq	Matriz de inercias	-
$H(q, \dot{q})$	Matriz fuerza Coriolis	-
$C(q)$	Matriz fuerza de gravedad	-
n	GDL	-
$w(t)$	Valor actual del peso sináptico	-
$w(t+1)$	Valor actualizado, peso sináptico	-

$\Delta w(t)$	Variación del peso sináptico	-
p	P-ésimo patrón del conjunto de entrenamiento	-
d_p	Valor deseado de salida para el p-ésimo patrón	-
y_p	Valor de salida de la red para el p-ésimo patrón	-

(Elaborado por: Autores de la tesis)

CAPÍTULO III

DISEÑO DEL SISTEMA

En el presente capítulo se abordará el desarrollo de la cinemática directa e inversa del robot manipulador de 3 GDL, así como el cálculo de la dinámica directa e inversa, con la finalidad de poder realizar las simulaciones pertinentes del sistema en conjunto, además de poder determinar los parámetros que rigen el movimiento del robot, sin embargo, no se abordará los cálculos para el diseño mecánico del robot manipulador ya que el propósito de la presente tesis se basa en el control mediante procesamiento de imágenes e identificación de colores por medio de RNA. Por otro lado, el desarrollo de las redes neuronales no implica un modelamiento matemático en específico, se hace uso de una herramienta integrada en el programa Matlab denominado *nprtool*, donde toma como datos los valores de una matriz de entrada cuyas filas corresponden al número de muestras para el entrenamiento, mientras que las columnas hacen referencia al número de características que posee cada muestra, al culminar con el entrenamiento, como resultado se obtiene los pesos sinápticos y el algoritmo generado tal como lo veremos en el subcapítulo correspondiente a la identificación de colores. El capítulo también aborda el diseño electrónico y programación del rayo tractor sónico, el cual permitirá la manipulación de los objetos dentro del área de trabajo, ya que, es una tecnología en desarrollo aún no existe una ciencia concreta en cuanto a fórmulas matemáticas que rijan el modelo estándar de un

rayo tractor sónico, en general se basa en la manipulación de la frecuencia de sonido. Finalmente, se diseña el algoritmo computacional para el procesamiento de imágenes con los cuales bajo ciertos parámetros, identificar las coordenadas de los objetos dispersos en la mesa de trabajo para que posteriormente el robot se dirija directamente a cada uno de ellos para su manipulación.

3.1. Diseño del sistema robótico

3.1.1. Cinemática Directa

Para el desarrollo de la cinemática directa del robot de tres grados de libertad, es necesario utilizar el método de *transformación homogénea*, donde se aplicarán nociones fundamentalmente de álgebra vectorial y álgebra matricial para representar y describir la localización de un objeto en un espacio tridimensional con respecto a un sistema de referencia fijo.

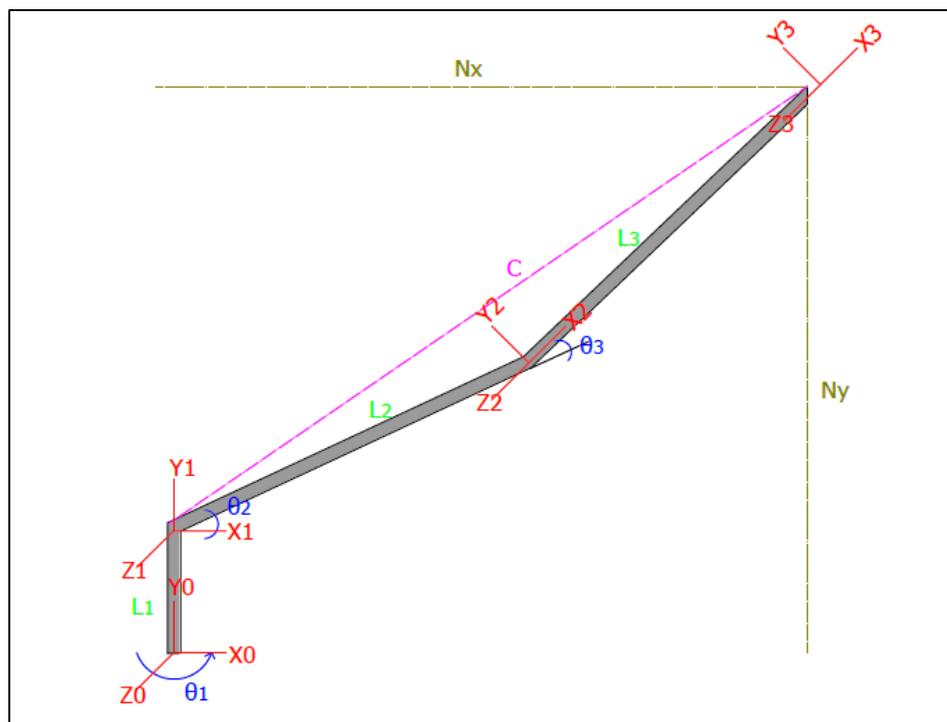


Figura 3.1. Plano de los puntos de referencia del robot de 3 grados de libertad.
(Elaborado por: Autores de la tesis)

Como bien se muestra en la figura 3.1 el robot de tres grados de libertad se puede considerar como una cadena formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones. Por lo tanto, se ha establecido un punto de referencia fijo en la base principal del robot que será de guía para poder describir la localización de cada uno de los eslabones con respecto a este dicho punto.

En el siguiente análisis se detalla cómo se obtiene la matriz de transformación homogénea en función a las coordenadas articulares que relacionará la posición y orientación del extremo del robot respecto al sistema de referencia fijado en la base de este mismo.

En primer lugar, se deben definir los parámetros de Denavit-Hartenberg que se muestran en la tabla 3.1.

Tabla 3.1 Parámetros Denavit-Hartenberg

Articulación	θ	d	a	α
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
3	θ_3	0	l_3	0

(Elaborado por autores de la tesis)

Para obtener las matrices de transformación se debe tener en cuenta el punto de referencia posterior respecto al punto de referencia fijo en la base, y así continuar hasta el último eslabón, definiendo los parámetros de acuerdo a la variación de cada ángulo.

Desarrollo de matrices:

- **Primer punto de referencia:**

$${}^0H_1(\theta_1) = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0R_1(\theta_1) & {}^0r_{\theta_0\theta_1} \\ 000 & 1 \end{bmatrix} \quad (3.1)$$

- **Segundo punto de referencia:**

$${}^1H_2(\theta_2) = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^1R_2(\theta_2) & {}^1r_{\theta_1\theta_2} \\ 000 & 1 \end{bmatrix} \quad (3.2)$$

- **Tercer punto de referencia:**

$${}^2H_3(\theta_3) = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^2R_3(\theta_3) & {}^2r_{\theta_2\theta_3} \\ 000 & 1 \end{bmatrix} \quad (3.3)$$

- **Posición de orientación respecto al sistema de referencia:**

$${}^0H_3(\theta_1, \theta_2, \theta_3) = {}^0H_1(\theta_1) * {}^1H_2(\theta_2) * {}^2H_3(\theta_3)$$

$${}^0H_3(\theta_1, \theta_2, \theta_3) = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Resolución paso a paso:

Parte 1: ${}^0H_1(\theta_1) * {}^1H_2(\theta_2)$

$$= \begin{bmatrix} \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 & -\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 & 0 & l_2 \cos \theta_1 \cos \theta_2 - l_2 \sin \theta_1 \sin \theta_2 + l_1 \cos \theta_1 \\ \sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2 & -\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 & 0 & l_2 \sin \theta_1 \cos \theta_2 + l_2 \cos \theta_1 \sin \theta_2 + l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0H_1(\theta_1) * {}^1H_2(\theta_2)$$

$$= \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Parte 2: } {}^0H_1(\theta_1) * {}^1H_2(\theta_2) * {}^2H_3(\theta_3)$$

$$= \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$* \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & l_3 \cos(\theta_1 + \theta_2 + \theta_3) + l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_3 \sin(\theta_1 + \theta_2 + \theta_3) + l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0H_3(\theta_1, \theta_2, \theta_3) = \begin{bmatrix} {}^0R_3(\theta_1, \theta_2, \theta_3) & {}^0r_{\theta_0\theta_3} \\ 000 & 1 \end{bmatrix} \quad (3.5)$$

El resultado hallado describe la posición de la pinza (rayo tractor sónico) al terminal, esto quiere decir, la descripción de la posición de orientación respecto al sistema de referencia cero que depende de los tres grados de libertad que posee el robot.

3.1.2. Cinemática Inversa

La cinemática inversa de un robot tiene como objetivo encontrar aquellos valores que deben adoptar las coordenadas articulares del robot $(\theta_1, \theta_2, \theta_3)$, para que la última articulación que por lo general es el griper pueda posicionarse y orientarse según una determinada localización espacial.

Para el desarrollo de esta sección se utiliza el *método de geométrico*, debido a que este procedimiento es adecuado para robots de pocos grados de libertad en este caso son solo tres GDL

El procedimiento se basa en encontrar suficiente número de relaciones geométricas en las que intervendrán las coordenadas del extremo el robot, sus coordenadas articulares y las dimensiones físicas de sus elementos.

Debido a la posición que se tiene de los eslabones respecto al sistema de referencia fijo se aplicará teoremas básicos de trigonometría para obtener los resultados esperados.

La partida de inicio se encuentra conformada por las coordenadas (n_x, n_y, n_z) mostradas en la figura 3.1.

- Primera sección: Teorema de Coseno

$$c^2 = n_x^2 + n_y^2 = l_3^2 + l_2^2 - 2l_3 * l_2 * \cos(\theta_3) \quad (3.6)$$

- Analizando la posición del ángulo $\Phi = \pi - \theta_2$, se obtiene para el grado de libertad n°3:

$$c^2 = l_3^2 + l_2^2 - 2l_3 * l_2 * \cos(\pi - \theta_3)$$

$$n_x^2 + n_y^2 = l_3^2 + l_2^2 + 2l_3 * l_2 * \cos(\theta_3)$$

$$\cos(\theta_3) = \frac{n_x^2 + n_y^2 - l_3^2 - l_2^2}{2l_3 * l_2}$$

$$\sin(\theta_3) = \pm \sqrt{1 - \left(\frac{n_x^2 + n_y^2 - l_3^2 - l_2^2}{2l_3 * l_2} \right)^2}$$

Por lo tanto:

$$\theta_3 = \tan^{-1}\left(\frac{-\sqrt{1 - \left(\frac{n_x^2 + n_y^2 - l_3^2 - l_2^2}{2l_3 l_2}\right)^2}}{\frac{n_x^2 + n_y^2 - l_3^2 - l_2^2}{2l_3 l_2}}\right) \quad (3.7)$$

Como se visualiza, existen 2 posibles soluciones para que θ_3 una con el signo positivo

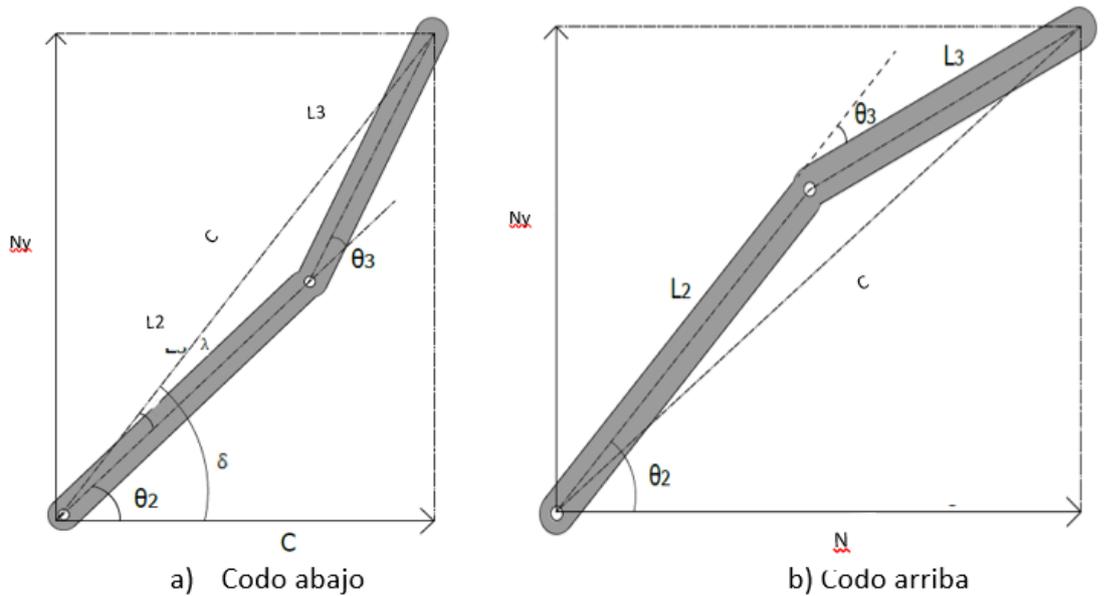


Figura 3.2: Contenidos en un plano y en configuración codo abajo (a) y configuración codo arriba (b). (Elaborado por: Autores de la tesis)

- Ecuación de restricción de existencia de soluciones:

$$l_3 + l_2 \geq \sqrt{n_x^2 + n_y^2} \quad (3.8)$$

- Luego, para la segunda ecuación Grado de libertad n°2:

$$\delta = \lambda + \theta_2$$

$$\theta_2 = \delta - \lambda$$

Entonces:

- Para δ :

$$\begin{aligned} \operatorname{tg}(\delta) &= \frac{n_y}{n_x} \\ \delta &= \tan^{-1}\left(\frac{n_y}{n_x}\right) \end{aligned} \quad (3.9)$$

- Para λ :

$$\begin{aligned} \operatorname{tg}(\lambda) &= \frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)} \\ \lambda &= \tan^{-1}\left(\frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)}\right) \end{aligned} \quad (3.10)$$

Del enunciado:

$$\begin{aligned} \theta_2 &= \delta - \lambda \\ \theta_2 &= \tan^{-1}\left(\frac{n_y}{n_x}\right) - \tan^{-1}\left(\frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)}\right) \end{aligned} \quad (3.11)$$

- Luego para la segunda ecuación Grado de libertad n°1:

$$\theta_1 = \tan^{-1}\left(\frac{n_z}{n_x}\right) \quad (3.12)$$

Con las soluciones obtenidas de manera matemática, al contrario de lo que sucedía en el análisis desarrollado en la cinemática directa, la solución del problema cinemático inverso no es única, debido a que existen diferentes duplas que posicionan y orientan al extremo del mismo robot.

3.1.3. Resolución matemática de Cinemática Inversa y Directa

En primer lugar, se desarrollará la Cinemática Inversa para poder obtener los ángulos de cada articulación y poder ser planteados en la Cinemática Directa.

- Datos fijos:

$$n_x = 0.360 \text{ m}, n_y = 0.320 \text{ m}, l_2 = 0.30 \text{ m}, l_3 = 0.20 \text{ m}, l_1 = 0.10 \text{ m}$$

- De la Ec. (3.6) para hallar C:

$$c^2 = n_x^2 + n_y^2$$

$$c^2 = 0.360^2 + 0.320^2$$

$$c = 0.4817 \text{ m}$$

- De la Ec. (3.7) para hallar θ_3 :

$$\theta_3 = \tan^{-1} \left(\frac{-\sqrt{1 - \left(\frac{n_x^2 + n_y^2 - l_3^2 - l_2^2}{2l_3 * l_2} \right)^2}}{\frac{n_x^2 + n_y^2 - l_3^2 - l_2^2}{2l_3 * l_2}} \right)$$

$$\theta_3 = \tan^{-1} \left(\frac{-\sqrt{1 - \left(\frac{0.360^2 + 0.320^2 - 0.20^2 - 0.30^2}{2 * 0.2 * 0.3} \right)^2}}{\frac{0.360^2 + 0.320^2 - 0.20^2 - 0.30^2}{2 * 0.2 * 0.3}} \right)$$

$$\theta_3 = \tan^{-1} \left(\frac{\sqrt{111}}{17} \right)$$

$$\theta_3 = 31.7883^\circ$$

- De la Ec. (3.8) para hallar restricción de existencia de soluciones:

$$l_3 + l_2 \geq \sqrt{n_x^2 + n_y^2}$$

$$0.20 + 0.30 \geq \sqrt{0.360^2 + 0.320^2}$$

$$0.50 \geq 0.4817$$

- De la Ec. (3.9) para hallar δ :

$$\delta = \tan^{-1} \left(\frac{n_y}{n_x} \right)$$

$$\delta = \tan^{-1} \left(\frac{0.320}{0.360} \right)$$

$$\delta = 40,6335^\circ$$

- De la Ec. (3.10) para hallar λ :

$$\lambda = \tan^{-1} \left(\frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)} \right)$$

$$\lambda = \tan^{-1} \left(\frac{0.20 \sin(31.7883)}{0.30 + 0.20 \cos(31.7883)} \right)$$

$$\lambda = \tan^{-1}(0.24416)$$

$$\lambda = 12.6347^\circ$$

- De la Ec. (3.11) para hallar θ_2 :

$$\theta_2 = \tan^{-1} \left(\frac{n_y}{n_x} \right) - \tan^{-1} \left(\frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)} \right)$$

$$\theta_2 = 40,6335^\circ - 12.6347^\circ$$

$$\theta_2 = 28,99^\circ \cong 29.00^\circ$$

- De la Ec. (3.12) para hallar θ_1 :

$$\theta_1 = \tan^{-1} \left(\frac{n_z}{n_x} \right)$$

$$\theta_1 = \tan^{-1} \left(\frac{0.10}{0.360} \right)$$

$$\theta_1 = 15.52^\circ$$

Ahora se procederá a desarrollar la Cinemática Directa con los valores de los ángulos obtenidos.

- Datos obtenidos: $\theta_1 = 15.52^\circ$, $\theta_2 = 29.00^\circ$, $\theta_3 = 31.7883^\circ$

- De la Ec. (3.1) se obtiene el Primer punto de referencia:

$${}^0H_1(\theta_1) = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0H_1(15.52) = \begin{bmatrix} \cos(15.52) & -\sin(15.52) & 0 & 0.10 \cos(15.52) \\ \sin(15.52) & \cos(15.52) & 0 & 0.10 \sin(15.52) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- De la Ec. (3.2) se obtiene el Segundo punto de referencia:

$${}^1H_2(\theta_2) = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1H_2(29) = \begin{bmatrix} \cos(29) & -\sin(29) & 0 & 0.30 \cos(29) \\ \sin(29) & \cos(29) & 0 & 0.30 \sin(29) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- De la Ec. (3.3) se obtiene el Tercer punto de referencia:

$${}^2H_3(\theta_3) = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2H_3(31.78) = \begin{bmatrix} \cos(31.78) & -\sin(31.78) & 0 & 0.20 \cos(31.78) \\ \sin(31.78) & \cos(31.78) & 0 & 0.20 \sin(31.78) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- De la Ec. (3.4) se obtiene la posición de orientación respecto al sistema de referencia:

$${}^0H_3(\theta_1, \theta_2, \theta_3) = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$* \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$* \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0H_3(15.52, 29, 31.78)$$

$$= \begin{bmatrix} \cos(15.52) & -\sin(15.52) & 0 & 0.10 \cos(15.52) \\ \sin(15.52) & \cos(15.52) & 0 & 0.10 \sin(15.52) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$* \begin{bmatrix} \cos(29) & -\sin(29) & 0 & 0.30 \cos(29) \\ \sin(29) & \cos(29) & 0 & 0.30 \sin(29) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$* \begin{bmatrix} \cos(31.78) & -\sin(31.78) & 0 & 0.20 \cos(31.78) \\ \sin(31.78) & \cos(31.78) & 0 & 0.20 \sin(31.78) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Resolviendo Ec. (3.4) para obtener la posición de orientación respecto al sistema de referencia:

$${}^0H_3(\theta_1, \theta_2, \theta_3) =$$

$$\begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & l_3 \cos(\theta_1 + \theta_2 + \theta_3) + l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_3 \sin(\theta_1 + \theta_2 + \theta_3) + l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0H_3(15.52, 29, 31.78) =$$

$$\begin{bmatrix} \cos(76.30) & -\sin(76.30) & 0 & 0.20\cos(76.30) + 0.30 \cos(44.52) + 0.10 \cos(15.52) \\ \sin(76.30) & \cos(76.30) & 0 & 0.20 \sin(76.30) + 0.30 \sin(44.52) + 0.10 \sin(15.52) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0H_3(15.52, 29, 31.78) =$$

$$\begin{bmatrix} \cos(76.30) & -\sin(76.30) & 0 & 0.20\cos(76.30) + 0.30 \cos(44.52) + 0.10 \cos(15.52) \\ \sin(76.30) & \cos(76.30) & 0 & 0.20 \sin(76.30) + 0.30 \sin(44.52) + 0.10 \sin(15.52) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0H_3(15.52, 29, 31.78) = \begin{bmatrix} 0.2368 & -0.9715 & 0 & 0.3576 \\ 0.9715 & 0.2368 & 0 & 0.4314 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} {}^0R_3(\theta_1, \theta_2, \theta_3) & {}^0r_{\theta_0\theta_3} \\ 000 & 1 \end{bmatrix}$$

Al tener identificado todas las articulaciones se pudo resolver la matriz de transformación homogénea que se plantea en la cinemática directa, siendo esta expresión matemática la indicadora de la localización del sistema final (H_3) con respecto al sistema de referencia de la base fija del robot (H_0) que depende de las tres articulaciones $\theta_1, \theta_2, \theta_3$ compuesta por una matriz de cambio de orientación ${}^0R_3(\theta_1, \theta_2, \theta_3)$ que describe la orientación de la pinza (rayo tractor sónico) y la posición absoluta ${}^0r_{\theta_0\theta_3}$ del origen H_0 hasta H_3 . La tabla 3.2 muestra los resultados obtenidos del modelamiento matemático de la cinemática del robot.

Tabla 3.2. Resultados obtenidos del modelamiento cinemático del Robot

Variable	Valor
c	0.4817 m
θ_1	15.52 °
θ_2	29.00 °
θ_3	31.78 °
δ	40.63 °
λ	12.63 °
${}^0R_3(\theta_1, \theta_2, \theta_3)$	$\begin{bmatrix} 0.2368 & -0.9715 & 0 \\ 0.9715 & 0.2368 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
${}^0r_{\theta_0\theta_3}$	$\begin{bmatrix} 0.3576 \\ 0.4314 \\ \mathbf{0} \end{bmatrix}$

(Diseñado por: Autores de la Tesis)

3.1.4. Modelo dinámico

Para realizar el modelo dinámico estructural mecánico de un robot de 3 GDL se plantea diferentes métodos como solución, siendo una de ellas aplicando la segunda ley de Newton (Equilibrio de fuerzas) o su equivalente para movimientos de rotación denominada Ley de Euler. Sin embargo, como método alternativo se puede usar la formulación Lagrangiana basada en tomar en consideración la energía del sistema, la ventaja de usar este método es la facilidad de modelar un sistema complejo como es el caso de un robot. La ecuación 3.13 describe los parámetros a tomar en cuenta:

$$L = E_c - E_p \quad (3.13)$$

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (3.14)$$

Donde:

- L: Función Lagrangiana
- E_c : Energía cinética
- E_p : Energía Potencial
- q_i : Coordenadas generalizadas
- τ_i : Pares que actúan sobre la articulación

Para la presente investigación se tiene la figura 3.3 como representación de un robot de 3 GDL, con masas concentradas en su centro de gravedad, el desarrollo del modelamiento dinámico se realizará sobre ello.

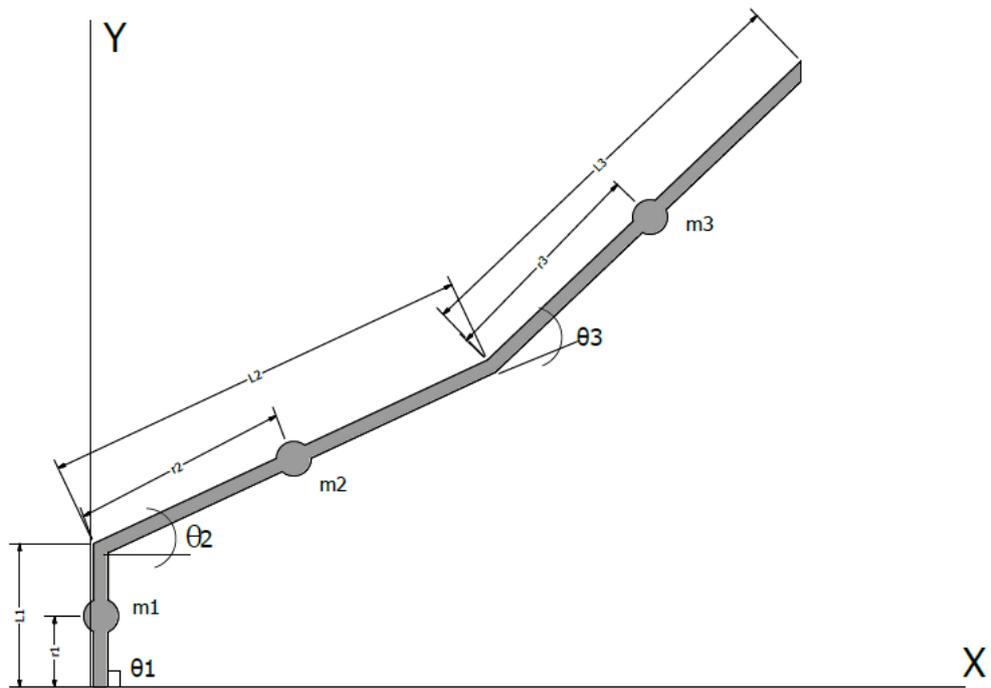


Figura 3.3. Representación gráfica en 2D del Robot manipulador (*Diseñada por: Autores de la tesis*)

Aplicando el método Lagrangiano, en primer lugar, se procede a determinar la posición (representada por x) y velocidad en cada articulación respecto a su centro de masa (representada por \dot{x}) la cual se obtiene derivando la posición respecto del ángulo para luego expresar las energías necesarias y así obtener la función Lagrangiana.

Para la masa 1 se tiene los siguientes parámetros:

$$\begin{aligned} x &= r1 \cos(\theta_1) & \dot{x} &= -r1 \sin(\theta_1) \cdot \dot{\theta}_1 \\ y &= r1 \sin(\theta_1) & \dot{y} &= r1 \cos(\theta_1) \cdot \dot{\theta}_1 \end{aligned}$$

Sabiendo que el primer ángulo θ_1 es 90° , las velocidades angulares para cada eje resultan cero debido a que existe una derivada de una constante, por lo que la expresión quedaría de la siguiente manera:

$$\begin{aligned} x &= r1 \cos(\theta_1) & \dot{x} &= 0 \\ y &= r1 \sin(\theta_1) & \dot{y} &= 0 \end{aligned}$$

Con las expresiones anteriores se representa la energía cinética y potencial de la articulación.

➤ Energía Cinética:

$$E_c = \frac{1}{2} m_1 v^2 \quad (3.15)$$

$$E_c = \frac{1}{2} m_1 (0)^2 = 0$$

➤ Energía Potencial:

$$E_p = m_1 \cdot g \cdot h \quad (3.16)$$

$$E_p = m_1 \cdot g \cdot r_1 \sin(\theta_1)$$

Se repite los mismos procedimientos para las siguientes articulaciones tomando en cuenta que la distancia es desde el punto de referencia 0 hasta la concentración de masa m_2

$$x = (L_1 + r_2) \cos(\theta_2) \quad \dot{x} = -(L_1 + r_2) \sin(\theta_2) \cdot \dot{\theta}_2$$

$$y = (L_1 + r_2) \sin(\theta_2) \quad \dot{y} = (L_1 + r_2) \cos(\theta_2) \cdot \dot{\theta}_2$$

➤ Energía Cinética:

$$E_c = \frac{1}{2} m_2 ((L_1 + r_2)^2 \sin^2(\theta_2) \cdot \dot{\theta}_2^2 + (L_1 + r_2)^2 \cos^2(\theta_2) \cdot \dot{\theta}_2^2)$$

$$E_c = \frac{1}{2} m_2 (L_1 + r_2)^2 \cdot \dot{\theta}_2^2$$

➤ Energía Potencial:

$$E_p = m_2 \cdot g \cdot (L_1 + r_2) \sin(\theta_2)$$

Para la masa 3 se obtienen las siguientes expresiones:

$$x = (L_1 + L_2 + r_3) \cos(\theta_2 + \theta_3) \quad \dot{x} = -(L_1 + L_2 + r_3) \sin(\theta_2 + \theta_3) \cdot (\dot{\theta}_2 + \dot{\theta}_3)$$

$$y = (L_1 + L_2 + r_3) \sin(\theta_2 + \theta_3) \quad \dot{y} = (L_1 + L_2 + r_3) \cos(\theta_2 + \theta_3) \cdot (\dot{\theta}_2 + \dot{\theta}_3)$$

➤ Energía Cinética:

$$E_c = \frac{1}{2} m_3 ((L_1 + L_2 + r_3)^2 \sin^2(\theta_2 + \theta_3) \cdot (\dot{\theta}_2 + \dot{\theta}_3)^2 + (L_1 + L_2 + r_3)^2 \cos^2(\theta_2 + \theta_3) \cdot (\dot{\theta}_2 + \dot{\theta}_3)^2)$$

$$E_c = \frac{1}{2} m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot (\dot{\theta}_2 + \dot{\theta}_3)^2$$

➤ Energía Potencial:

$$E_p = m_3 g \cdot (L_1 + L_2 + r_3) \sin(\theta_2 + \theta_3)$$

Por lo tanto, aplicando la fórmula de LaGrange se procede a realizar la diferencia entre la sumatoria de las energías cinéticas con la sumatoria de las energías potenciales gravitatorias.

$$L = 0 + \frac{1}{2} m_2 (L_1 + r_2)^2 \cdot \dot{\theta}_2^2 + \frac{1}{2} m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot (\dot{\theta}_2 + \dot{\theta}_3)^2 - m_1 \cdot g \cdot r_1 \sin(\theta_1) \\ - m_2 \cdot g \cdot (L_1 + r_2) \sin(\theta_2) - m_3 g \cdot (L_1 + L_2 + r_3) \sin(\theta_2 + \theta_3)$$

Luego de tener la expresión de LaGrange, se procede a realizar las derivadas parciales respecto a sus ángulos representados por q_i y \dot{q}_i .

$$\frac{\partial L}{\partial q_1} = -m_1 \cdot g \cdot r_1 \cos(\theta_1) \cdot \dot{\theta}_1$$

$$\frac{\partial L}{\partial \dot{q}_1} = 0$$

$$\frac{\partial L}{\partial q_2} = -m_2 \cdot g \cdot (L_1 + r_2) \cos(\theta_2) \cdot \dot{\theta}_2 - m_3 g (L_1 + L_2 + r_3) \cos(\theta_2 + \theta_3) \cdot \dot{\theta}_2$$

$$\frac{\partial L}{\partial \dot{q}_2} = m_2 \cdot (L_1 + r_2)^2 \cdot \dot{\theta}_2 + m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot \dot{\theta}_2$$

$$\frac{\partial L}{\partial q_3} = -m_3 g \cdot (L_1 + L_2 + r_3) \cos(\theta_2 + \theta_3) \cdot \dot{\theta}_2$$

$$\frac{\partial L}{\partial \dot{q}_3} = m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot \dot{\theta}_3$$

Para determinar el torque en cada articulación se necesita aplicar la ecuación de par expresado en la ecuación 3.14.

➤ Torque para la primera articulación:

$$\tau_1 = \frac{d}{dt}(0) - (-m_1 \cdot g \cdot r_1 \cos(\theta_1) \cdot \dot{\theta}_1)$$

$$\tau_1 = m_1 \cdot g \cdot r_1 \cos(\theta_1) \cdot \dot{\theta}_1$$

➤ Torque para la segunda articulación:

$$\tau_2 = \frac{d}{dt}(m_2 \cdot (L_1 + r_2)^2 \cdot \dot{\theta}_2 + m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot \dot{\theta}_2) - (-m_2 \cdot g \cdot (L_1 + r_2) \cdot \cos(\theta_2) \cdot \dot{\theta}_2 - \\ m_3 g (L_1 + L_2 + r_3) \cos(\theta_2 + \theta_3) \cdot \dot{\theta}_2)$$

$$\tau_2 = m_2 \cdot (L_1 + r_2)^2 \cdot \ddot{\theta}_2 + m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot \ddot{\theta}_2 + m_2 \cdot g \cdot (L_1 + r_2) \cdot \cos(\theta_2) \cdot \theta_2 + m_3 g (L_1 + L_2 + r_3) \cdot \cos(\theta_2 + \theta_3) \cdot \theta_2$$

➤ Torque para la tercera articulación:

$$\tau_3 = \frac{d}{dt} (m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot \dot{\theta}_3) - (-m_3 g \cdot (L_1 + L_2 + r_3) \cos(\theta_2 + \theta_3) \cdot \dot{\theta}_2)$$

$$\tau_3 = m_3 \cdot (L_1 + L_2 + r_3)^2 \cdot \ddot{\theta}_3 + m_3 g \cdot (L_1 + L_2 + r_3) \cos(\theta_2 + \theta_3) \cdot \dot{\theta}_2$$

Con esta última parte se ha culminado el modelo matemático del torque en cada articulación, es decir, resolviendo cada ecuación asignando valores específicos como el valor de sus masas, longitudes de la articulación, velocidades y aceleraciones angulares se puede determinar el par que se ejerce sobre la articulación y gracias a ello elegir un motor que esté a la altura de los valores obtenidos.

Para realizar la simulación del movimiento y como resultado obtener los diferentes valores de pares en un determinado tiempo es necesario satisfacer los requerimientos para el algoritmo computacional de Newton-Euler, por lo que en primer lugar se procede a realizar las matrices de rotación de la articulación respecto a su anterior.

$${}^0 R_1 = \begin{bmatrix} C1 & -S1 & 0 \\ S1 & C1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^1 R_2 = \begin{bmatrix} C2 & -S2 & 0 \\ S2 & C2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^2 R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Es necesario formar las matrices de posición de la articulación analizada respecto al punto de referencia anterior. Para el caso propio debido a estar trabajando en un plano 2D el eje z se completa con cero:

$${}^0 P_1 = \begin{bmatrix} L_1 c_1 \\ L_2 s_1 \\ 0 \end{bmatrix}$$

$${}^1 P_2 = \begin{bmatrix} L_1 c_2 \\ L_2 s_2 \\ 0 \end{bmatrix}$$

$${}^2 P_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Así mismo, se procede a formar las dos matrices que corresponden a los centros de masa de cada sistema, como se aprecia en la figura 3.1, los centros de masa están en la sección $\frac{L}{2}$.

$${}^0 P_o g_1 = \begin{bmatrix} -\frac{1}{2}L1 \\ 0 \\ 0 \end{bmatrix}$$

$${}^0 P_o g_2 = \begin{bmatrix} -\frac{1}{2}L2 \\ 0 \\ 0 \end{bmatrix}$$

Las matrices de los centros de inercia de cada articulación se representan por la siguiente matriz

$${}^0 I_c g_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{12} m_1 L_1^2 & 0 \\ 0 & 0 & \frac{1}{12} m_1 L_1^2 \end{bmatrix}$$

$${}^0 I_c g_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{12} m_2 L_2^2 & 0 \\ 0 & 0 & \frac{1}{12} m_2 L_2^2 \end{bmatrix}$$

Dentro de los valores iniciales se consideran cero a excepción de la matriz de la aceleración de la gravedad que solo tendrá magnitud en el eje Y.

$${}^0 w_o = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^0 v_o = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^0 \alpha_o = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^0 a_o = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix}$$

A continuación, se procede a añadir el siguiente código desarrollado en Matlab 2015 mostrado en la figura 3.4 para determinar el torque de cada articulación, de manera gráfica. Gracias a la función creada denominada *rne ()* se puede calcular de manera inmediata la dinámica inversa del robot, es decir, colocando los parámetros presentados como matrices en los puntos anteriores se logra obtener la gráfica de Torque vs tiempo.

```
t = [0:.056:2]; % crea un vector de tiempo
[q,qd,qdd] = jtraj(qz, qr, t); % calcula la trayectoria en coord. de las art.
tau = rne(robot, q, qd, qdd); % calcula la dinámica inversa
%Ahora pueden graficarse las torcas como función del tiempo
figure(2)
plot(t, tau(:,1),'b'), grid on
title('Torque N°1 vs Tiempo')
xlabel('Tiempo (s)');
ylabel('Torque de Articulación 1(Nm)');

figure(3)
plot(t, tau(:,2),'r'), grid on
title('Torque N°2 vs Tiempo')
xlabel('Tiempo (s)');
ylabel('Torque de Articulación 2(Nm)');

figure(4)
plot(t, tau(:,3),'g'), grid on
title('Torque N°3 vs Tiempo')
xlabel('Tiempo (s)');
ylabel('Torque de Articulación 3(Nm)');
```

Figura 3.4. Código para determinar los pares para cada articulación

Como resultado del modelo computacional se obtiene la figura 3.5 la cual representa el torque necesario para realizar un determinado movimiento establecido en el programa.

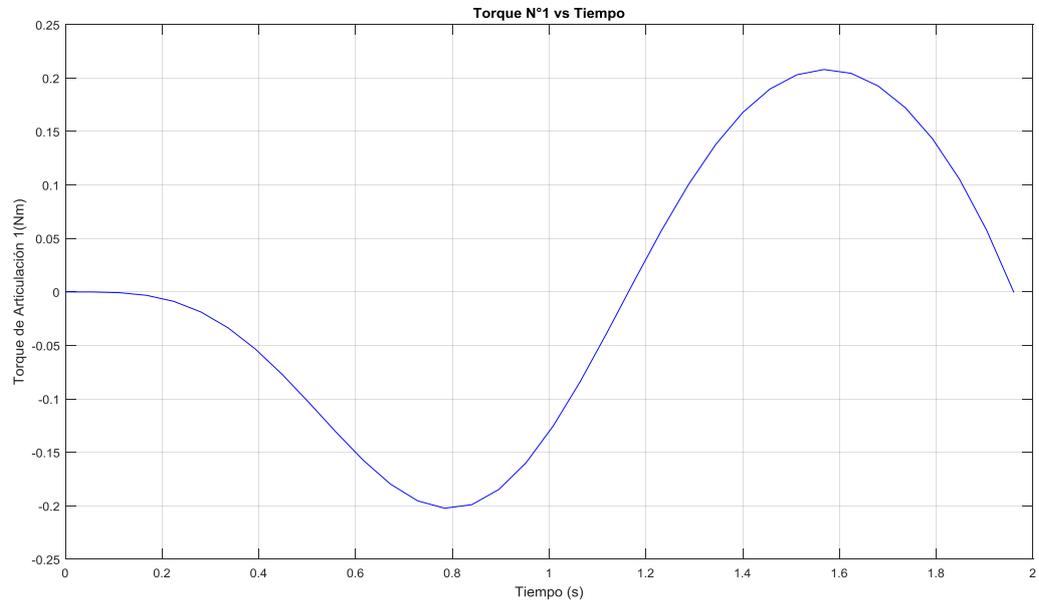


Figura 3.5. Torque gráfico de la primera articulación.

Recordemos que el tiempo está en unidades de segundos y el valor del torque en N.m (Newton metro). Al analizar la gráfica de la primera articulación, se ubica el punto con el valor máximo en valor absoluto, para que la articulación realice el movimiento es necesario generar un torque de 0.22 N.m, el torque en función del tiempo de la articulación 2 se muestra en la gráfica 3.6.

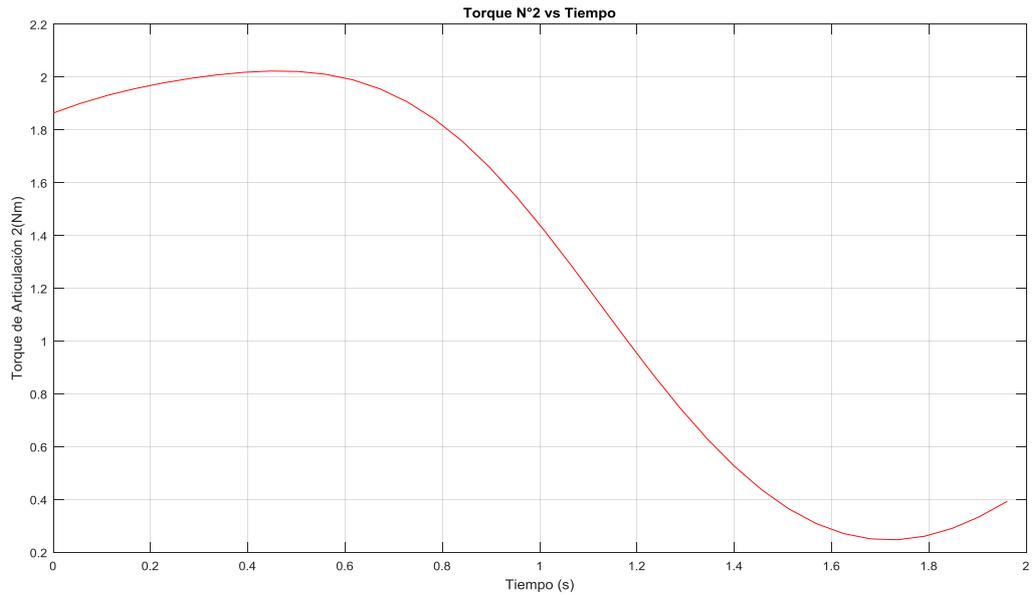


Figura 3.6. Torque gráfico de la segunda articulación

Gracias a la gráfica anterior se puede determinar que el torque necesario para mover la articulación dos es de aproximadamente 2 N.m.

Finalmente, para la articulación 3 se procede a graficar el último valor de la matriz tau, formando así la figura 3.7.

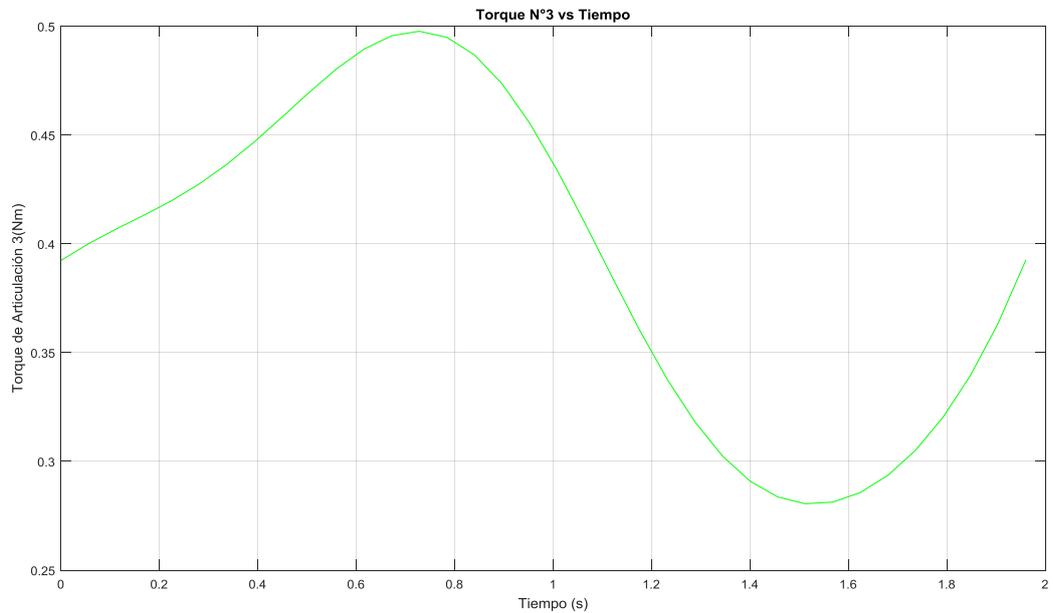


Figura 3.7. Torque gráfico de la segunda articulación

El valor del torque para la articulación 3 asciende a los 0.5 N.m aproximadamente, dicho valor no es muy elevado ya que los parámetros como la masa y la longitud del eslabón no son a gran escala.

En modo resumen se presenta la tabla 3.3 donde se aprecia el torque necesario para cada articulación obtenida del programa desarrollado, además del servomotor estándar en el mercado que se aproxima al valor deseado.

Tabla 3.3. Tabla de Torques para cada articulación

Articulación	Torque (N.m)	Servomotor Comercial (Kg.cm)
1	0.22	2.5
2	2	20
3	0.5	6

(Elaborado por: Autores de la tesis)

Debido a que en el mercado no se obtiene dichos parámetros, es necesario aproximarlos a valores comerciales y accesibles para realizar en el futuro el desarrollo del prototipo. A continuación, se muestran algunos de los servomotores con los que se puede contar para cada articulación. Como, por ejemplo, la figura 3.8 muestra un servomotor de la marca Tower Pro de serie MG90S con una capacidad de 2.5 Kg trabajando a un voltaje de 6VDC.



Figura 3.8. Servomotor de 2.5Kg.cm (*Servo data base*)

El servomotor destinado para la articulación dos está dado por la marca Dsservo Ds3218 mostrada en la figura 3.9 con una capacidad de hasta 21.5 Kg.cm trabajando con un voltaje de 6.8VDC.



Figura 3.9. Servomotor de 21.5 Kg.cm
(*Servo data base*)

En la figura 3.10 se tiene un servomotor Tower Pro SG-5010 con una capacidad entre el rango de 3. 5kg.cm para 4.8VDC hasta un torque de 6. 5Kg.cm trabajando con un voltaje máximo de 6VDC, el cual cumple con los requerimientos para la articulación 3.



Figura 3.10. Servomotor de 6.5Kg.cm (*Servo data Base*)

Gracias a la solución de la cinemática directa e inversa, es posible realizar la simulación del movimiento del robot empleando el algoritmo de Peter Corke tal como se muestra en la figura 3.11, dentro del algoritmo se ingresa los parámetros D-H, las masas correspondientes a cada eslabón, los vectores del centro de gravedad, es decir, los parámetros calculados en los subcapítulos

anteriores. El programa desarrollado se podrá visualizar por completo en los anexos de la tesis.

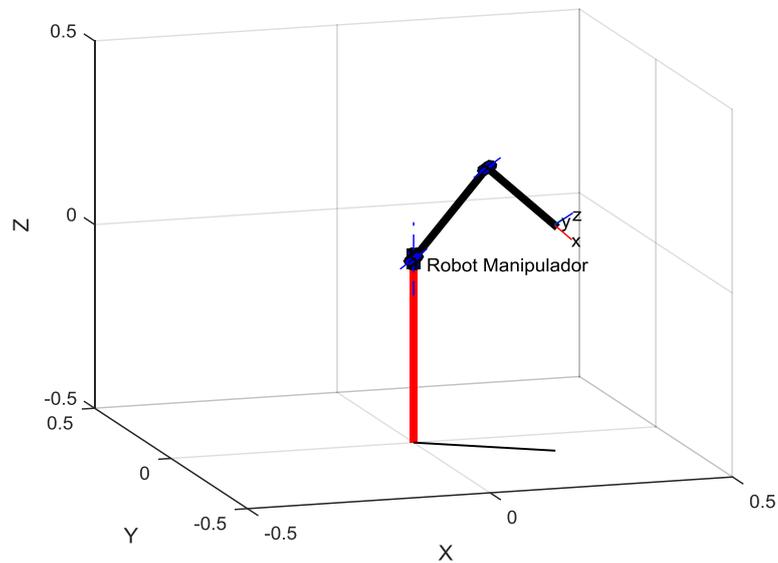


Figura 3.11. Simulación del robot manipulador en Matlab.

Por otro lado, gracias a los valores que se han ingresado para el desarrollo del modelo computacional, es posible generar un bosquejo del prototipo del robot manipulador de 3 GDL, la figura 3.12 muestra en la parte del extremo del robot el rayo tractor sónico sujeto al tercer eslabón.



Figura 3.12. Diseño Preliminar del Prototipo (*Diseñado por: Autores de la tesis*)

3.2. Diseño del algoritmo para la identificación de colores-RNA

Para realizar el diseño del algoritmo de reconocimiento de colores es necesario seguir un protocolo de desarrollo con la finalidad de organizar correctamente el trabajo realizado, el protocolo consta de las siguientes etapas de desarrollo

- Etapa de adquisición de datos.
- Etapa de extracción de características.
- Etapa de entrenamiento.
- Etapa de pruebas y resultados.

A continuación, en los siguientes subcapítulos se detallan cada una de las etapas mencionadas, dentro de ellas sus respectivos protocolos, algoritmos y diagramas de flujo para poder cumplir con cada etapa de desarrollo.

3.2.1. Etapa de adquisición de datos

El presente subcapítulo abarca el desarrollo del algoritmo para el reconocimiento de colores, para ello, es necesario poder tener una base de datos para luego realizar el entrenamiento supervisado en base a sus características y a las salidas deseadas. En la figura 3.13 se aprecia las 4 clases de colores entre ellas, rojo, verde, azul y negro; y por cada clase se tiene 5 muestras. El material de las esferas de colores es poliestireno, material que se usará para realizar las pruebas de levitación con el rayo tractor sónico.

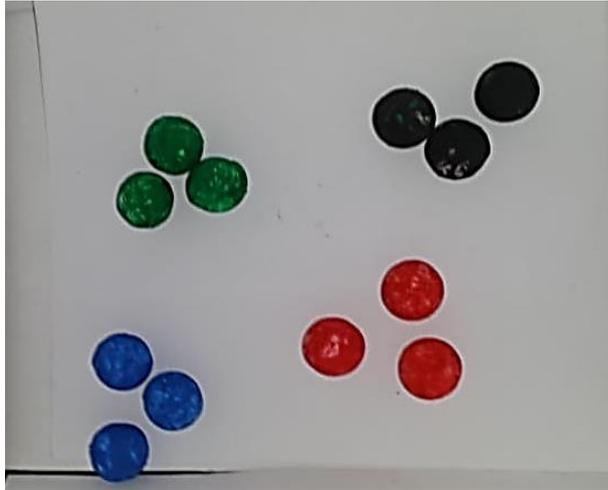


Figura 3.13. Elementos de muestras.

Para realizar las capturas de cada una de las imágenes es necesario construir un prototipo para la toma de datos, la figura 3.14 muestra un pequeño estudio de fotografía a escala de interior color blanco, iluminación led, una cámara VGA de resolución máxima de 480x360 y un mecanismo de apertura y cierre de la caja, todo ello con el fin de uniformizar la captura de imágenes y para que en el proceso de entrenamiento las clases iguales no varíen por parámetros de iluminación, enfoque, resolución etc. Se ha de tener en cuenta que, para temas de implementación, las condiciones del entorno de trabajo deberán de ser iguales a las muestras capturadas para el entrenamiento.



Figura 3.14. Estudio de toma de datos

Con todo el sistema de hardware listo para realizar las capturas, se procede a iniciar el programa desarrollado en Matlab, se realizarán 5 capturas por cada bolita de color, cada foto deberá de variar el perfil que se muestra de la esfera, en total se obtendrá 60 muestras para realizar el entrenamiento.

En la figura 3.15. se muestra el diagrama de flujo descrito para realizar las capturas constantes y almacenarlas con un nombre correlativo, para el presente caso se usará *color_n* donde *n* corresponde a la numeración de la fotografía capturada.

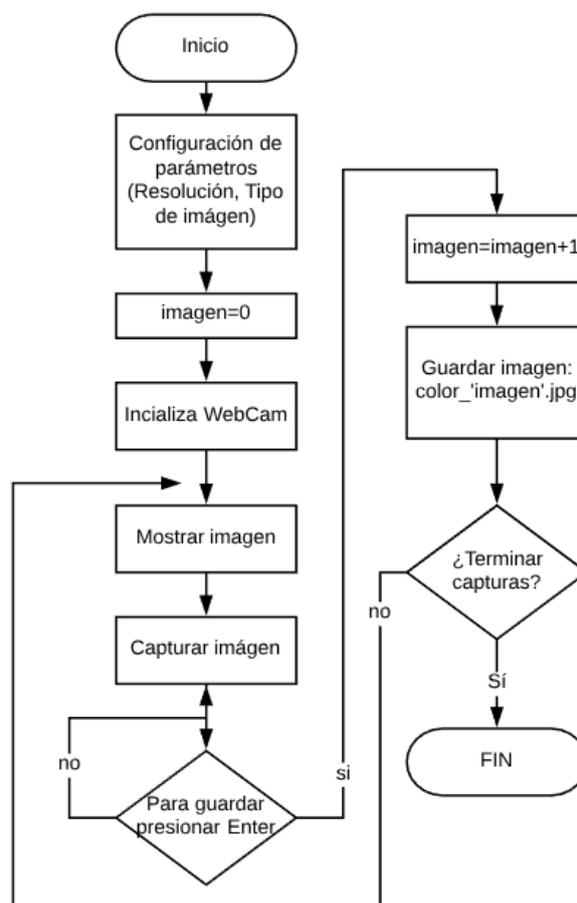


Figura 3.15. Diagrama de flujo para recolección de datos

En base al algoritmo de la figura 3.15 se procede a realizar el código de programación en Matlab 2015, recordar que la WebCam solo será reconocida en el programa si previamente se han instalado los controladores

correspondientes, estos se pueden descargar dentro de la opción *Get hardware support packages*.

Dentro del código desarrollado mostrada en la figura 3.16 se tiene la variable *x* donde se carga la entrada de imagen de la WebCam, se realiza una vista en tiempo real con la instrucción *preview(x)* y se solicita una imagen en colores RGB, se captura la imagen y se muestra en una nueva ventana de figura, dentro del bucle *While* se tiene las variables para guardar el archivo de imagen con el nombre *color_'n'.jpg* , cada vez que se presiona la tecla "Enter" la variable *n* aumenta en una unidad y la nueva imagen se guarda con el nombre seguido de la variable aumentada, para salir del bucle se presiona *q*, esto indicará el cese del programa de adquisición. Cabe recordar que solo se debe de presionar la tecla Enter cuando la nueva muestra se encuentre dentro del estudio de adquisición de datos lista para su captura.

```

%% Entorno para realizar las capturas de las muestras de datos
clear all, close all, clc;

x=videoinput('winvideo',1,'YUY2_640x480');
preview(x)
set(x,'ReturnedColorSpace','rgb');
Imagencel=getsnapshot(x);
figure(1)
captura=image(Imagencel);
done=false;
n=1;

while ~done
    a=input('Para capturar(Press Enter): ');
    if ~strcmp(a,'l')
        Imagencel=getsnapshot(x);
        set(captura,'CData',Imagencel);
        drawnow;
        filename= sprintf('color_%d.jpg',n);
        imwrite(Imagencel,filename,'png');
        n=n+1;
    end
    if strcmp(a,'q')
        done=true;
        return
    end
end
end

```

Figura 3.16. Código de programación en Matlab para las capturas de muestras.

Como resultado ejecutar el código se obtendrán 60 imágenes correspondientes a 4 clases de colores, para poder entrenar la red se necesita la extracción de las características de las imágenes adquiridas, este procedimiento se realiza con el diagrama de flujo de extracción de datos mostrada en la figura 3.17 de la siguiente etapa de desarrollo.

3.2.2. Etapa de extracción de características

Las características que se deben de extraer de las imágenes es el color de los píxeles, para ello, el primer paso es la descomposición de estos en sus colores bases, rojo, verde y azul para luego realizar el histograma de cada uno de ellos. Los valores de los colores oscilan desde 0 hasta el valor 255, la matriz de los histogramas será de orden 256x256, sin embargo, los datos importantes se

encuentran en las filas, por ello debemos de obtener una matriz de orden 256×1 , se necesita realizar la transpuesta de esta matriz para obtener una matriz de 1×256 , los pasos se deberán de repetir para cada histograma. Al finalizar tendremos tres matrices de 1×256 , se debe unificar las matrices para formar una sola y esta representará al mapa de colores de la imagen en sus tres tonos, la matriz característica de una imagen deberá de ser de 1×768 , el número 1 representa la muestra y los 768 valores corresponden a las características de la imagen.

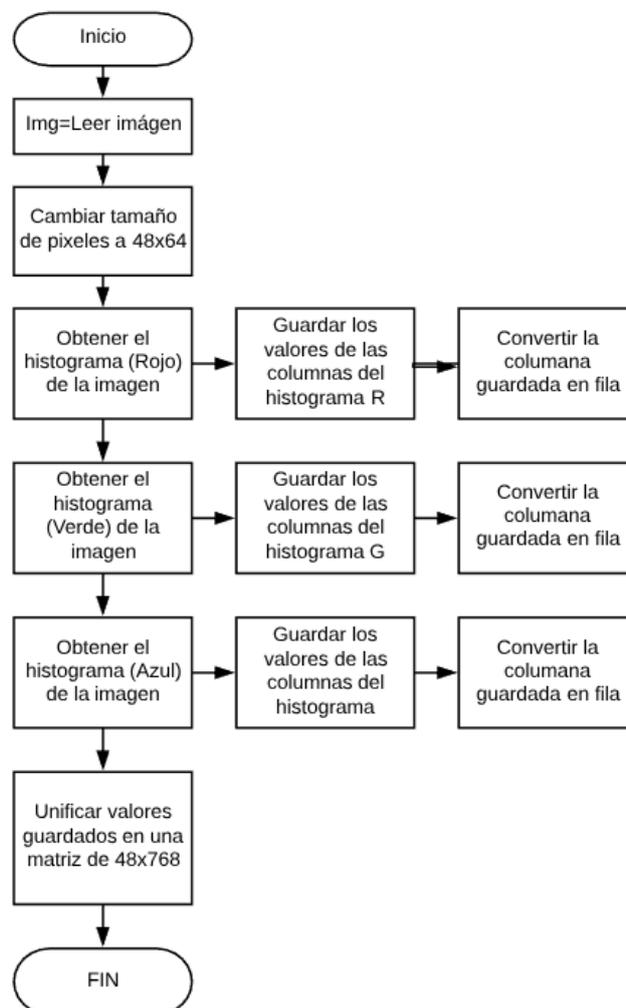


Figura 3.17. Diagrama de Flujo para obtención de características.

El diagrama de flujo mostrado, corresponde a la obtención de características para una sola imagen, sin embargo, en la figura 3.18 se muestra el código

completo para la obtención de características de todas las imágenes y como resultado final, la obtención de una sola matriz característica de 60x768 (muestras x características).

```

test=[]; % Vector vacio para agregar los datos a la matriz.

for numero=1:48;
    ObtencionImg=strcat('color_',num2str(numero),'.jpg');
    Img=imread(ObtencionImg);
    ImgResize=imresize(Img,[48,64]);

    HistogramaRojo= ImgResize(:,:,1); %Obtencion del Histograma rojo de la imagen en el bucle.
    [filas,x] = imhist(HistogramaRojo);%Retorno del vector del histograma en escala de grises.
    RedColumna=filas(1:256); %Obtenmcion del vector caracteristico rojo en columna.
    RedFila=RedColumna';%obtencion del vector caracteristico rojo en fila.
    %figure(1), imhist(HistogramaRojo);

    HistogramaVerde= ImgResize(:,:,2);%Obtencion del Histograma verde de la imagen en el bucle.
    [filas,x] = imhist(HistogramaVerde);%Retorno del vector del histograma en escala de grises.
    GreenColumna=filas(1:256);%Obtencion del vector caracteristico verde en columna.
    GreenFila=GreenColumna';%obtencion del vector caracteristico verde en fila.

    HistogramaBlue= ImgResize(:,:,3);%Obtencion del Histograma azul de la imagen en el bucle.
    [filas,x] = imhist(HistogramaBlue);%Retorno del vector del histograma en escala de grises.
    BlueColumna=filas(1:256);%Obtencion del vector caracteristico azul en columna.
    BlueFila=BlueColumna';%obtencion del vector caracteristico azul en fila.

    ConcaterRGB=[RedFila GreenFila BlueFila];%Concatenacion de los vectores de los histigramas rgb.
    test=[test;ConcateRGB]; %Concatenacion de los vectores caracteristicos de las imagenes en una matriz o-1.
end
test=test%/4991 %Muestra la matriz característica de las imagenes.

```

Figura 3.18. Código para la obtención de características

La matriz de características unificados se denomina *test*, al tener la matriz característica es necesario definir nuestras salidas deseadas *yd*, tendremos como salida de la red a 4 neuronas, una por cada clase de muestra. La tabla 3.4 muestra los valores que se deben de ingresar al algoritmo de entrenamiento de la red neuronal.

Tabla 3.4. Datos de salida deseada de la red neuronal

Color	Salida
Verde	0 0 0 1
Azul	0 0 1 0
Rojo	0 1 0 0
Negro	1 0 0 0

(Elaborada por: Autores de la tesis)

Para el presente desarrollo, todas las filas correspondientes al color verde se define la salida de la red como [0 0 0 1], todas las muestras azules con la matriz [0 0 1 0], lo mismo para los colores rojos y negros, al realizar la matriz de salida deseada se obtiene una matriz de dimensiones 60x4 siendo el número 60 a las muestras obtenidas y 4 a los valores de salidas de la red neuronal. De igual manera se procede a guardar la matriz de las salidas deseadas para realizar el entrenamiento.

3.2.3. Etapa de entrenamiento

El entrenamiento se ejecuta dentro de la ventana de comandos de Matlab escribiendo el nombre del Toolkit *nprtool*, dentro de ella, en la ventana de *Selección de datos* se cargarán las dos matrices, la matriz de características en *Inputs*, mientras que en *outputs* se deberá de cargar la matriz de las salidas deseadas. Estando en la ventana de *Validación y pruebas de datos* se eligen el porcentaje de las muestras para el entrenamiento, validación y pruebas, para el algoritmo presente, se dejarán los porcentajes por default. La siguiente ventana corresponde a la arquitectura de la red neuronal, donde las neuronas de la capa de entradas ya están definidas por las características de las imágenes, mientras que las neuronas de la capa de salida son 4, definidos por las salidas deseadas, lo que se debe de especificar son el número de neuronas en la capa oculta, si bien es cierto, no hay una forma matemática de determinar el número de capas ocultas y de neuronas ya que esto dependerá del problema a solucionar, se debe de realizar una serie de pruebas de entrenamientos. En la figura 3.19 se aprecia la configuración y arquitectura de la red neuronal para el reconocimiento de colores, con una capa oculta con 100 neuronas

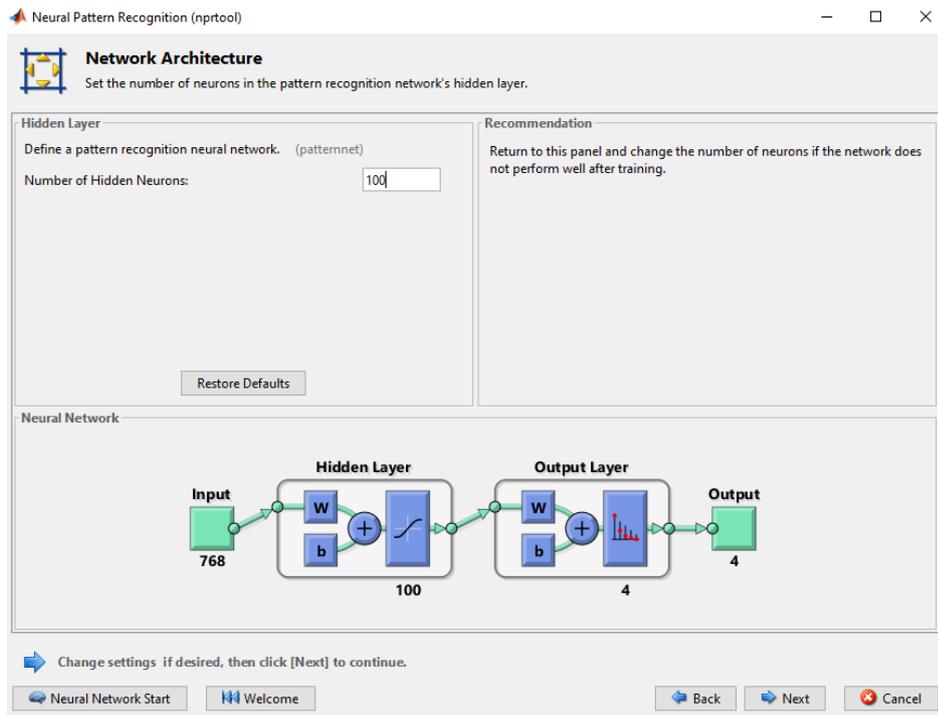


Figura 3.19. Arquitectura de la red neuronal

Finalmente, en la siguiente ventana se procede a realizar el entrenamiento presionando la opción *Train*, para corroborar el entrenamiento eficiente se procede a analizar la imagen 3.20 el cual corresponde a la opción de *plots Performance*.

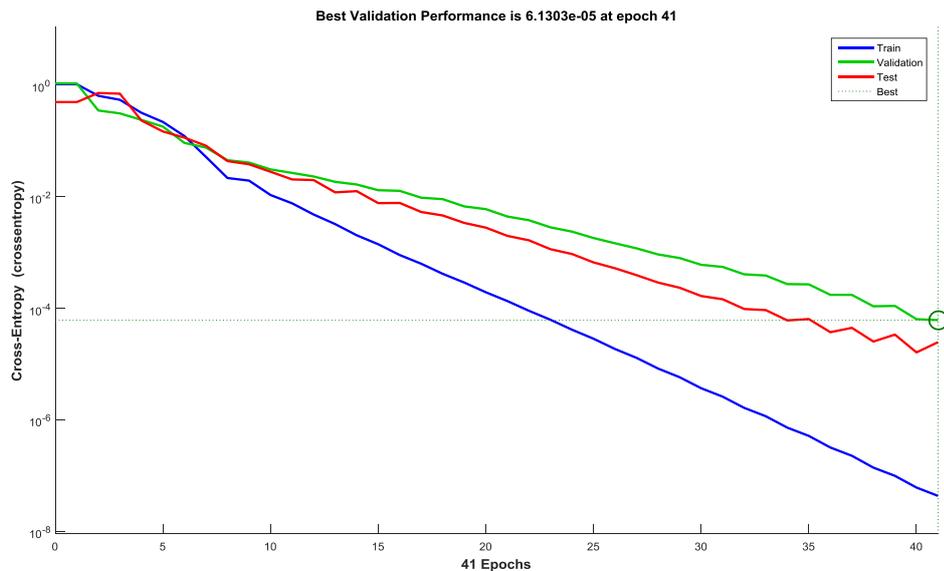


Figura 3.20. Validación del rendimiento de la Red entrenada.

La gráfica 3.19 se puede observar el desempeño de la red neuronal, se han realizado 41 iteraciones donde cada vez se han ido ajustando los pesos sinápticos para poder llegar a entrenar la red con las salidas deseadas, el punto marcado con el círculo verde indica que es la máxima eficiencia que se puede llegar con las neuronas configuradas, a partir de ese punto existiría un sobreentrenamiento el cual limitaría el desempeño de la red.

Por otro lado, la imagen 3.21 muestra la matriz de confusión de la red neuronal entrenada, esta matriz indica el porcentaje de precisión para cada salida, una precisión ideal correspondería a 25% para cada salida, sin embargo, se puede observar que para la salida 1 se obtuvo un 20.6 de precisión, para la salida 2 un 23.5%, mientras que para la salida 3 se obtuvo un 29.4% de precisión y finalmente un 26.5% de precisión para la salida 4.

Training Confusion Matrix

Output Class	1	7 20.6%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	8 23.5%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	10 29.4%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	9 26.5%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
		1	2	3	4	
		Target Class				

Figura 3.21. Matriz de confusión de la red entrenada.

Gracias a la herramienta de Matlab, se genera el código de la red neuronal en un nuevo Scrib, el cual debemos de ejecutar para poder validar la información. La figura 3.22 muestra el código del programa, donde la variable x contiene la matriz de datos de las características y la variable t contiene la matriz de las

salidas deseadas, sin embargo, para el proceso de pruebas solo se usa las líneas de *Test the Network*, donde la variable x corresponde a una matriz de 1x768 que indica que se carga una muestra para que sea reconocida por la red neuronal. El proceso de pruebas se realizará en el subcapítulo siguiente.

```
%Red Neuronal Colores
x = MatrizdeDatos';
t = Yd';
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.
% Create a Pattern Recognition Network
hiddenLayerSize = 100;
net = patternnet(hiddenLayerSize);
% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
% Train the Network
[net,tr] = train(net,x,t);
% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
% View the Network
view(net)
```

Figura 3.22. Código de la Red Neuronal

3.2.4. Etapas de pruebas y resultados

Es necesario realizar una serie de pruebas con muestras totalmente diferentes a las 48 imágenes que han servido antes para el entrenamiento de la red neuronal, esto con el fin de verificar que la red cumpla con la identificación de los colores y que exista ningún tipo de inconveniente al variar el entorno de las capturas de imágenes.

Para ello, se procede a capturar 12 imágenes en total, 3 para cada esfera de color, esto con la ayuda del código presentado con anterioridad en la figura 3.16 correspondiente a las capturas de las imágenes, teniendo la nueva base de datos se procede a realizar la extracción de características con el código de

la imagen 3.23 realizando un cambio de valores en la línea de la instrucción *For*, el número de muestras pasa de ser 60 a 12, se ha realizado también un cambio de variables para que los nuevos valores no se actualicen en los anteriores.

```

MatrizdeDatos=[]; % Vector vacio para agregar los datos a la matriz.
for numero=1:12;
    ObtencionImagen=strcat('color_',num2str(numero),'.jpg');
    Img=imread(ObtencionImagen);
    LecturaImagen=imresize(Img,[48,64]);

    HistogramaRojo= LecturaImagen(:,:,1); %Obtencion del Histograma rojo de la imagen en el bucle.
    [counts,x] = imhist(HistogramaRojo);%Retorno del vector del histograma en escala de grises.
    VecCarRedColumna=counts(1:256); %obtencion del vector caracteristico rojo en columna.
    VecCarRedFila=VecCarRedColumna';%obtencion del vector caracteristico rojo en fila.
    %figure(1), imhist(HistogramaRojo);

    HistogramaVerde= LecturaImagen(:,:,2);%Obtencion del Histograma verde de la imagen en el bucle.
    [counts,x] = imhist(HistogramaVerde);%Retorno del vector del histograma en escala de grises.
    VecCarGreenColumna=counts(1:256);%obtencion del vector caracteristico verde en columna.
    VecCarGreenFila=VecCarGreenColumna';%obtencion del vector caracteristico verde en fila.

    HistogramaBlue= LecturaImagen(:,:,3);%Obtencion del Histograma azul de la imagen en el bucle.
    [counts,x] = imhist(HistogramaBlue);%Retorno del vector del histograma en escala de grises.
    VecCarBlueColumna=counts(1:256);%obtencion del vector caracteristico azul en columna.
    VecCarBlueFila=VecCarBlueColumna';%obtencion del vector caracteristico azul en fila.

    ConcatenarRGB=[VecCarRedFila VecCarGreenFila VecCarBlueFila];%Concatenacion de los vectores de los hi
    MatrizdeDatos=[MatrizdeDatos;ConcatenarRGB]; %Concatenacion de los vectores caracteristicos de las im
end
MatrizdeDatos=MatrizdeDatos%/4991 %Muestra la matriz caracteristica de las imagenes.

```

Figura 3.23. Código para la extracción de características de imágenes de pruebas

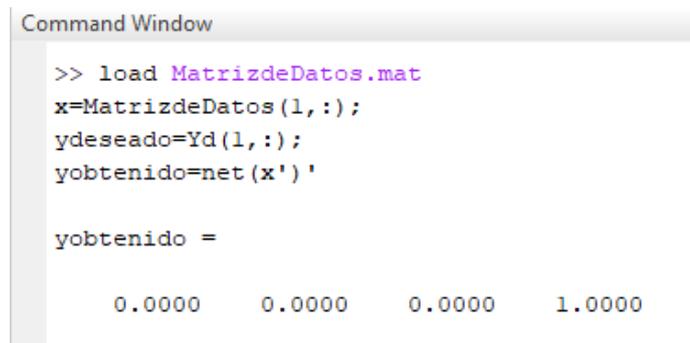
Ejecutando el código se obtiene una nueva matriz de 12x768 denominada *MatrizdeDatos*, se procede a guardar para poder realizar las pruebas correspondientes, los primeros 3 corresponden al color verde, los siguientes 3 a muestras de color rojo, siguen las azules y finalmente las últimas 3 filas corresponden a las características de 3 muestras del color azul.

Para poder ejecutar las pruebas es necesario realizar el código de la figura 3.24 donde se carga (load) el nombre de la nueva matriz de características (*MatrizdeDatos.mat*), en la variable *x* se guarda la primera fila y todas las columnas de *MatrizdeDatos*, esto para evaluar la primera muestra y finalmente, estas características se evalúan con el parámetro *net*.

```
load MatrizdeDatos.mat
x=MatrizdeDatos(1,:);
ydeseado=Yd(1,:);
yobtenido=net(x)'
```

Figura 3.24. Código para testear la red neuronal.

Se espera obtener como salida un valor de [0 0 0 1] correspondiente al valor de la salida deseada para el color verde. Al ejecutar el código anterior se obtiene los parámetros de la figura 3.25.



```
Command Window
>> load MatrizdeDatos.mat
x=MatrizdeDatos(1,:);
ydeseado=Yd(1,:);
yobtenido=net(x)'

yobtenido =

    0.0000    0.0000    0.0000    1.0000
```

Figura 3.25. Datos obtenidos de la prueba para la primera muestra

El resultado indica que la red ha reconocido la imagen como un color verde, esto indica que la red neuronal cumple con el propósito de reconocer los colores entrenados, para cerciorarse, se completa las pruebas para las 12 imágenes, la tabla 3.5 muestra un resumen de las pruebas realizadas con las nuevas muestras, donde el número de muestras corresponde a la fila de la muestra a evaluar, se tiene en la segunda las salidas deseadas para realizar una comparativa con los valores que se registran luego de ejecutar la red para el reconocimiento de las nuevas muestras.

Tabla 3.5. Salidas obtenidas de las 12 muestras de pruebas

Número de Muestras	Salidas deseadas	Salidas Obtenidas
x=MatrizdeDatos(1,:)	[0 0 0 1]	[0.0000 0.0000 0.0000 1.0000]
x=MatrizdeDatos(2,:)	[0 0 0 1]	[0.0385 0.0260 0.0009 0.9346]
x=MatrizdeDatos(3,:)	[0 0 0 1]	[0.0000 0.0000 0.0000 1.0000]
x=MatrizdeDatos(4,:)	[0 0 1 0]	[0.0001 0.0000 0.9999 0.0000]
x=MatrizdeDatos(5,:)	[0 0 1 0]	[0.0000 0.0340 0.9659 0.0001]
x=MatrizdeDatos(6,:)	[0 0 1 0]	[0.0000 0.0001 0.9999 0.0000]
x=MatrizdeDatos(7,:)	[0 1 0 0]	[0.0000 1.0000 0.0000 0.0000]
x=MatrizdeDatos(8,:)	[0 1 0 0]	[0.0000 1.0000 0.0000 0.0000]
x=MatrizdeDatos(9,:)	[0 1 0 0]	[0.0000 1.0000 0.0000 0.0000]
x=MatrizdeDatos(10,:)	[1 0 0 0]	[0.8358 0.0076 0.0169 0.1397]
x=MatrizdeDatos(11,:)	[1 0 0 0]	[0.5021 0.1579 0.3293 0.0107]
x=MatrizdeDatos(12,:)	[1 0 0 0]	[0.5495 0.0351 0.3913 0.0241]

(Elaborada por: Autores de la tesis)

Como muestra la tabla 3.5 Se han obtenidos resultados favorables en cuanto a las pruebas del reconocimiento de colores, dentro de las nuevas muestras de pruebas se ha variado la iluminación, posición y grado de color del objeto. Finalmente, el algoritmo desarrollado para el reconocimiento de colores se puede ampliar para la identificación de más clases, teniendo en cuenta que el número de muestras deberá de incrementarse para que la red pueda tener una mayor base de datos para realizar el entrenamiento.

3.3. Diseño del algoritmo para el control servo visual

A continuación, se procederá a realizar el diseño del algoritmo mediante procesamiento de imágenes para determinar las posiciones en 2d del objeto dentro del área de trabajo, para posteriormente, realizar el escalado de píxeles a metros y así poder cargar los valores al algoritmo desarrollado, dentro de este subcapítulo también se determinan los valores que los ángulos deben de tener para llegar a alcanzar el objeto sobre la mesa.

El presente subcapítulo será dividido en varias etapas para poder seguir un orden correlativo que conlleve al desarrollo correcto del algoritmo computacional, en las siguientes líneas se mencionan dichas etapas.

- Etapa de adquisición y procesamientos de datos.
- Etapa de extracción de características y entorno de trabajo.
- Etapa de simulación y resultados.

3.3.1. Etapa de Adquisición y procesamiento de datos

En primer lugar, para realizar la etapa de adquisición se debe de contar las dimensiones del área de trabajo definidos, por lo que, en el desarrollo de la cinemática se determinó que el máximo alcance del robot en centímetros es de 0.48, en base a esto, el área de trabajo queda definido por las dimensiones de 0.51x0.38, se está considerando una tolerancia debido a que no se requiere que el robot llegue a su máximo desplazamiento en cuanto al eje y.

Posteriormente, la cámara se debe de ubicar perpendicularmente al área de trabajo, teniendo en consideración que, la cámara deberá de captar todo el entorno establecido, la base deberá de ser de color uniforme blanco para evitar el ruido innecesario al momento de realizar el procesamiento de las imágenes, se realiza las capturas correspondientes y se almacenan dentro de una carpeta única para para desarrollar todo el algoritmo en Matlab.

Para realizar el procesamiento de la imagen es necesario ejecutar el código desarrollado presentado en la figura 3.26, el cual ejecuta el comando de lectura de la imagen para guardarla en una variable *Img*, la imagen es convertida a escalas de grises y binarizada a un umbral menor que 85, se filtran los píxeles sueltos que son el ruido de la imagen con el comando *bwareaopen* () donde sus parámetros son, la imagen binarizada y el número de conjunto de píxeles que se quiere filtrar. La dilatación de la imagen es un proceso que se utiliza para poder rellenar aquellas partes huecas de la geometría analizada, por ser el objeto de estudio es una esfera, se utiliza el comando '*Disk*' y se indica el radio en píxeles que se desea rellenar con el valor de 1.

```
Img=imread('color_1.jpg');
Igris=rgb2gray(Img);
Ibin=Igris<85;
Ibin2=bwareaopen(Ibin,5);
se = strel('disk',10);

%Proceso para dilatación de imagen
BW2 = imdilate(Ibin2,se);
figure(1)
imshow(BW2), title('Imagen Binarizada');
```

Figura 3.26. Código para el procesamiento de imagen.

Finalmente, se aplica la dilatación con los parámetros descritos con anterioridad para obtener la figura 3.27. Para pruebas del algoritmo se ha tomado 6 muestras de las capturas para la etapa de entrenamiento de la red neuronal para el reconocimiento de colores.

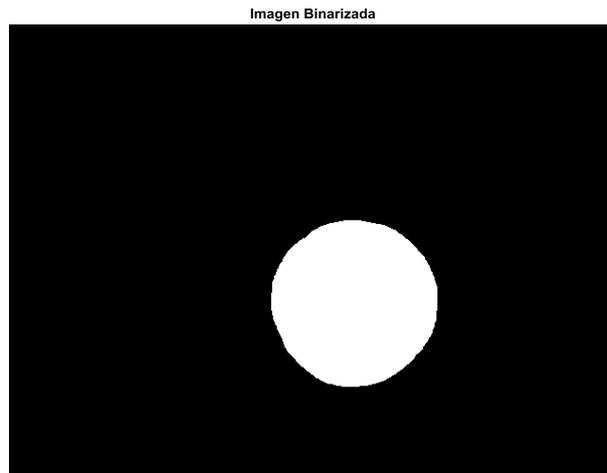


Figura 3.27. Imagen proceda de la muestra

Teniendo la imagen totalmente binarizada y filtrada, se puede pasar a la siguiente etapa de desarrollo.

3.3.2. Etapa de extracción de las características y entorno de trabajo

Para determinar las coordenadas del objeto de la figura 3.27 es necesario simplificar toda la geometría a un solo punto, para ello, se identifica el centroide la figura y se obtiene las coordenadas en ese punto, la imagen 3.28 muestra el código para identificar el centroide de la figura.

```
%proceso para determinar las propiedades de la figura
stats = regionprops('table',BW2,'Centroid',...
    'MajorAxisLength','MinorAxisLength')
centers = stats.Centroid;
diameters = mean([stats.MajorAxisLength stats.MinorAxisLength],2);
radii = diameters/2;
hold on
viscircles(centers,radii);
hold off
```

Figura 3.28. Código para determinar las propiedades de la figura

Con el código mostrado se obtienen los parámetros de, una matriz de 1x2 donde se almacena el centroide de la figura, además realiza una comparación del diámetro de acuerdo a los valores de los ejes x, y tomados desde el centroide de la figura hasta el contorno exterior, se restringe los valores y finalmente el diámetro corresponde al número máximo de ambos valores.

Como resultado se obtiene que el centroide de la imagen se ubica en las coordenadas (365.04, 297.28) por ser una imagen, las unidades están en pixeles, lo que corresponde a realizar es almacenar estos valores en dos variables, una para cada eje y posteriormente realizar el gráfico de estas coordenadas para determinar la posición exacta.

Sin embargo, debemos de escalar la resolución de la cámara de 640x480 a las dimensiones del área de trabajo de 51x38cm, en la gráfica 3.29. se muestra el espacio de trabajo y el punto x sobre la imagen indica el objeto posicionado sobre el plano.

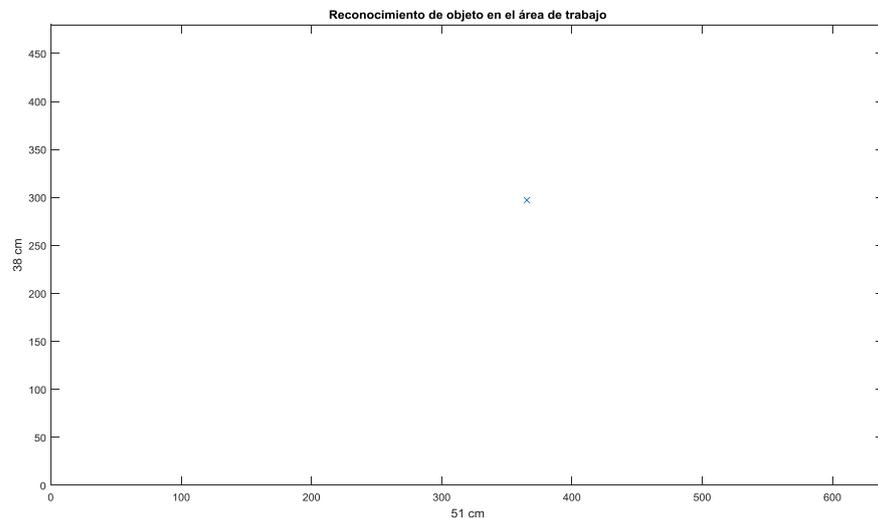


Figura 3.29. Posición del objeto en el plano 2d.

Ya posicionado el objeto dentro del área, se necesita conocer sus coordenadas en el sistema métrico para poder llevarlo a la realidad, para ello se tiene la figura 3.30 donde se aprecia las distancias de cada coordenada, como también la posición del robot frente al área de trabajo.

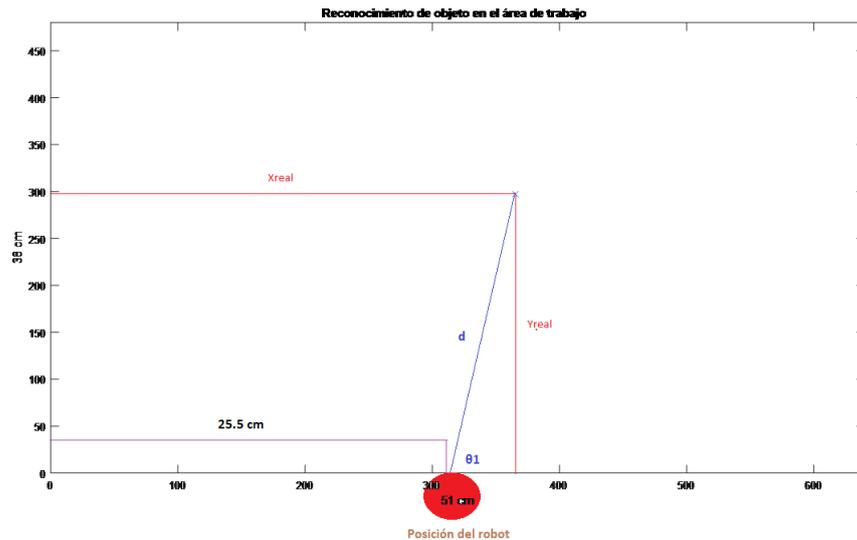


Figura 3.30. Posición del robot respecto al objeto sobre el plano

Las variables X_{real} y Y_{real} corresponden a las coordenadas en píxeles del objeto, como se mencionó previamente, se debe de tener en cuenta que:

$$480 \text{ píxeles} = 38 \text{ cm}$$

$$640 \text{ píxeles} = 51 \text{ cm}$$

Del enunciado anterior se determina mediante una regla de 3 las coordenadas en centímetros para cada eje, como ejemplo se toma el valor del eje x de la imagen binarizada.

$$640 \text{ píxeles} = 51 \text{ cm}$$

$$365.04 \text{ píxeles} = x \text{ cm}$$

Como resultado se obtiene son 29 cm, lo mismo se realiza para el eje y

$$480 \text{ píxeles} = 38 \text{ cm}$$

$$297.08 \text{ píxeles} = y \text{ cm}$$

EL resultado obtenido es de 23.5 cm, teniendo las coordenadas en el sistema métrico, se puede determinar la variable d , correspondiente a la distancia que existe entre el robot hasta la posición del objeto, debido a que la geometría es

un triángulo rectángulo, se puede determinar dicha distancia por medio de la ecuación de Pitágoras.

$$d = \sqrt{(29 - 22.5)^2 + 23.5^2}$$

$$d = 24.38 \text{ cm}$$

La distancia d corresponde a la longitud total que se debe de tener con ambos eslabones, además, de la gráfica se puede determinar el ángulo de rotación realizando:

$$\theta_1 = \arctg\left(\frac{23.5}{(29 - 25.5)}\right)$$

$$\theta_1 = 81.52^\circ$$

Este valor obtenido, es la rotación en grados del robot para poder posicionarse y llegar al objeto en el plano 2D.

Realizando los cálculos por medio de la ley de cosenos, se puede determinar los ángulos que deben de adoptar θ_2 y θ_3 para alcanzar la distancia d , para ello, en la gráfica 3.31 se aprecia la geometría que debe de tomar los eslabones del robot para alcanzar al objeto.

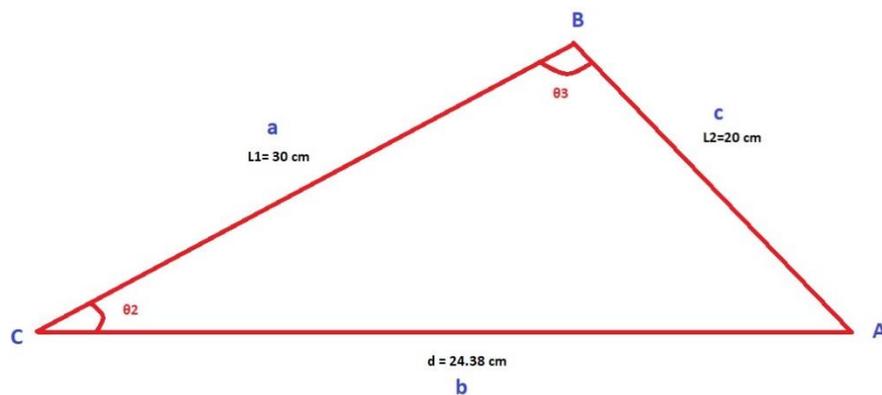


Figura 3.31. Geometría de la posición de los eslabones

Aplicando la ley de cosenos para determinar los valores de los ángulos, para el ángulo θ_2 se tiene:

$$\theta_2 = \arccos\left(\frac{0.30^2 + 0.2438^2 - 0.20^2}{2 * 0.30 * 0.2438}\right)$$

$$\theta_2 = 41.56^\circ$$

Finalmente se aplica la misma solución para determinar el ángulo θ_3

$$\theta_3 = \arccos\left(\frac{0.30^2 + 0.20^2 - 0.2438^2}{2 * 0.30 * 0.20}\right)$$

$$\theta_3 = 54.07^\circ$$

El ángulo θ_1 está condicionado de acuerdo a la posición del objeto dentro del plano 2D, en la figura 3.32 se observa los dos posibles casos de posicionamiento del robot, de 0 hasta 90° y de 91 hasta 180°.

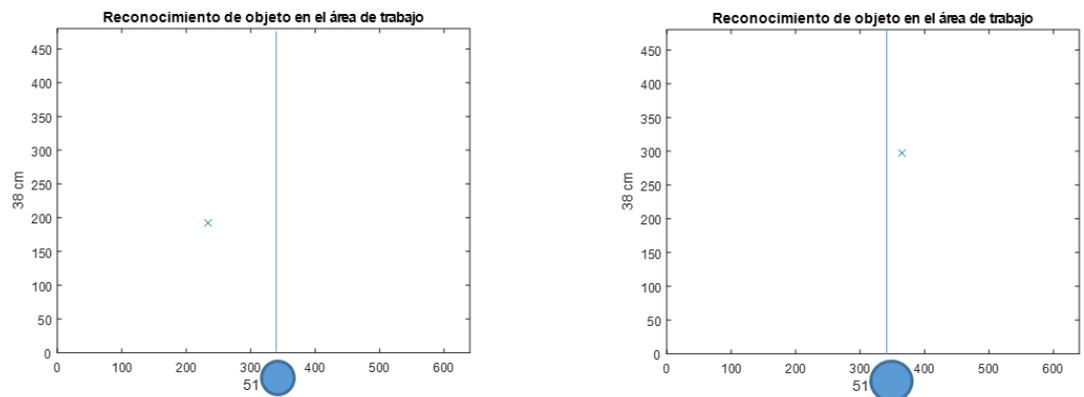


Figura 3.32. Posición del objeto en diferente cuadrantes.

De estar el objeto posicionado en el segundo cuadrante, es decir, cuando el ángulo θ_1 sea mayor que 90 grados se debe de realizar una condición para el algoritmo, el cual consiste en restar el ángulo llano de $180 - \theta_3$ para obtener el ángulo correcto de desplazamiento.

3.3.3. Etapa de simulación

Gracias a los parámetros de la etapa anterior, se puede desarrollar el algoritmo para la obtención de los ángulos de cada articulación. Para ello, el código de la figura 3.33 resume los cálculos de la etapa anterior para la obtención de los parámetros necesarios.

```
%Conversión de unidades de pixeles a cm
%area de trabajo 51*38 cm
xmetros=round((xpos*51)/640,1);
ymetros=round((ypos*38)/480,1);

%distancia del robot al objeto
distobjeto=round(sqrt(xmetros^2+ymetros^2),2)

%Determinando el ángulo de giro para theta 1
if xmetros<25.5
    thetal=atan(ymetros/xmetros);
    thetal=pi()-thetal
%    thetal=thetal*(180/pi());
%    tehtal=round(180-thetal)

    else
        thetal=atan(ymetros/xmetros);
%        thetal=thetal*(180/pi());
%        thetal=round(thetal)
end

theta2=acos((30^2+distobjeto^2-20^2)/(2*30*distobjeto));
% theta2=theta2*(180/pi());
% theta2=round(theta2,2)
theta3=acos((30^2+20^2-distobjeto^2)/(2*30*20));
%theta3=theta3*(180/pi());
%theta3=round(theta3,2)
```

Figura 3.33. Algoritmo para la conversión y obtención de ángulos.

Finalmente, para realizar la simulación se hace uso del Toolkit de robótica proporcionado por Peter Korque denominado *Startup_rvc*, la figura 3.34 muestra el código desarrollado para hacer uso de la herramienta, donde los valores de los ángulos de θ_1 , θ_2 y θ_3 obtenidos por medio del procesamiento de imágenes, se llevan al entorno de simulación, dentro de la matriz Link se procede a modificar los grados de libertad para la simulación y estos grados son definidos por la tabla de parámetros de Denavit-Hartenberg, como paso siguiente, se unifica todos los eslabones con el parámetro SerialLink, se define el nombre del robot y se ingresa la posición inicial que debe adoptar el robot,

es decir, se ingresa las grados correspondientes a la zona segura. Dentro de cada for se genera la animación del movimiento para cada articulación, la variable th1, th2 y th3 corresponden a los ángulos que se irán incrementando con un paso de 0.1 hasta llegar los ángulos generados por el procesamiento de imágenes los cuales son theta1, theta2 y theta3. Finalmente se realiza una pause después de cada ciclo de for para poder observar la animación del movimiento.

```

%% Algoritmo para la animación del movimiento del robot manipulador
L1=5; L2=31; L3=20;
%L=Link([th d a alph])
L(1)=Link([0 L1 0 pi/2]);
L(2)=Link([0 0 L2 0]);
L(3)=Link([0 0 L3 0]);
rob=SerialLink(L)
rob.name='Manipulador';
th1=pi()/2; th2=pi/2; th3=45;
%rob.plot([pi/2, th2, th3])
%rob.fkine([th1,th2, th3])  matriz de homogeneidad
%syms th1 th2 th3
rob.fkine([0 0 0])
th1=pi/2; th2=0; th3=0
%prueba del primer ángulo
for th1=0:0.1:thetal
    figure(3)
    rob.plot([th1,0,0]);
    pause(0.005);
end
for th2=0:0.1:thetal
    figure(3)
    rob.plot([th1,th2,0]);
    pause(0.005);
end
for th3=0:-0.1:-theta3
    figure(3)
    rob.plot([th1,th2,th3]);
    pause(0.005);
end

```

Figura 3.34. Código para la simulación del movimiento

Como resultado de la animación se obtiene que el movimiento descrito por el robot llega al punto en donde se encuentra el objeto identificado, la figura 3.35 muestra al robot cuando ha llegado a su posición final.

Reconocimiento de objeto en el área de trabajo

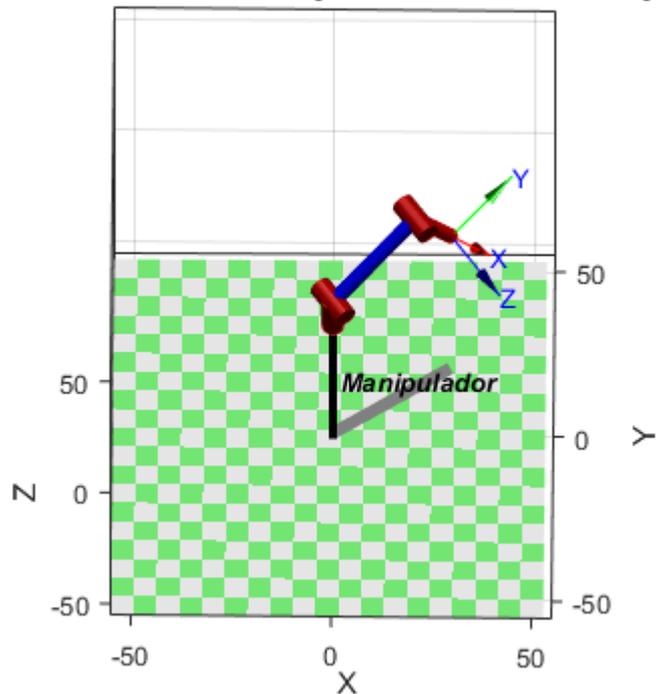


Figura 3.35. Posición final del robot manipulador

Para una mejor visualización se tiene la figura 3.36 la cual es la nueva muestra para la obtención de las coordenadas y ángulos respectivos, por los que en el espacio 2D se posiciona tal como lo muestra.

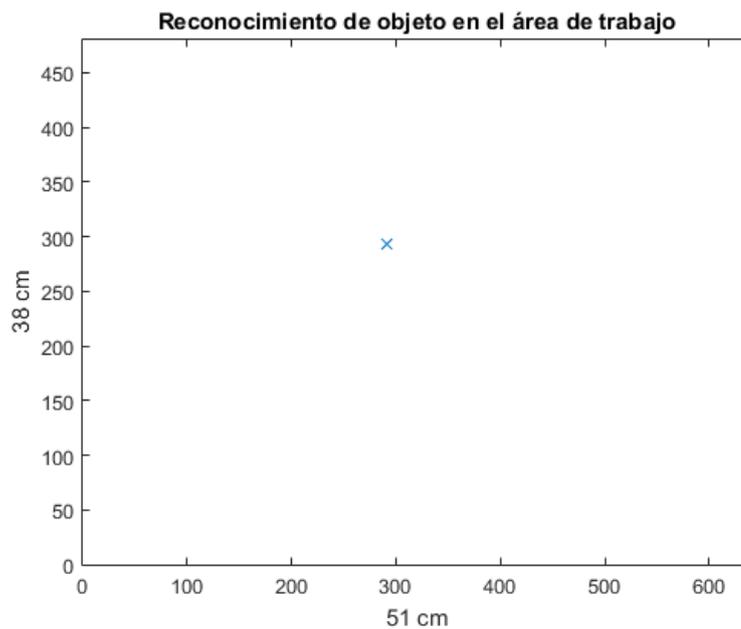


Figura 3.36. Nueva muestra en coordenadas 2D.

Y como resultado de ejecutar el código para la simulación se obtiene que, el robot llega a la posición del objeto, tal lo muestra la figura 3.37.

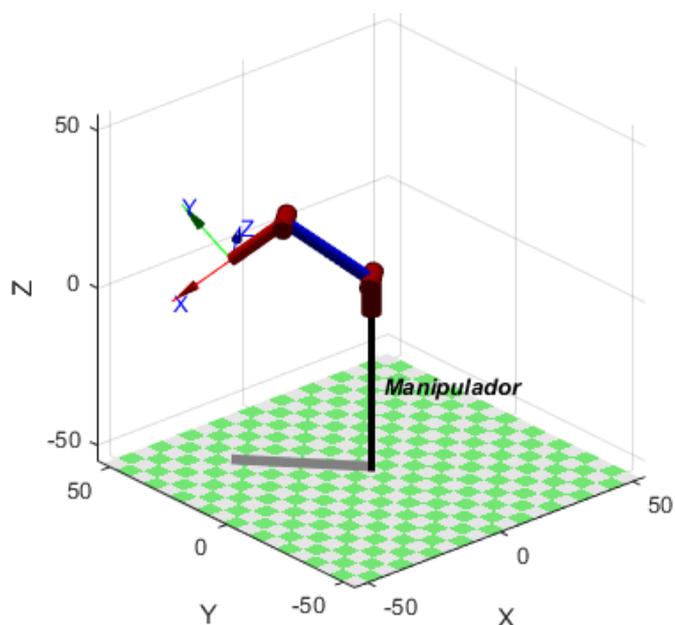


Figura 3.37. Posición del robot manipulador en coordenadas de la nueva muestra.

Gracias al algoritmo desarrollado se puede determinar las coordenadas del objeto dentro del área de enfoque de la cámara, el sistema cumple con el propósito de poder realizar un servo control visual para un robot de 3 GDL.

3.4. Diseño del Rayo tractor sónico

En primer lugar, una oscilación que se propaga en un medio se denomina onda. En este caso, la relación que existe entre el sentido de la oscilación y el de la propagación es de ondas longitudinales, esto quiere decir que el sentido de la oscilación coincide con el de la propagación de la onda.

Las ondas sonoras no se propagan en el vacío, necesitan de un medio, el aire. Este medio posee distintas características de importancia para que las ondas de sonido se puedan dispersar, las cuales son:

- Propagación lineal: las ondas de sonido propagarse por el mismo espacio al mismo tiempo sin afectarse mutuamente.
- Medio no dispersivo: las ondas de sonido se propagan a la misma velocidad independientemente de su frecuencia o amplitud.
- Medio homogéneo: el sonido se propaga esféricamente, es decir, en todas las direcciones, generando lo que se denomina un campo sonoro.

Teniendo en cuenta estos principales conocimientos se procede a plantear el diseño del rayo tractor sónico, empezando por la delimitación del objeto a manipular por medio de hologramas acústicos.

3.4.1. Delimitación de objeto a manipular:

Si se desea con certeza obtener resultados óptimos en el proceso de levitación se deben considerar los factores del objeto (representación de los elementos químicos) a levitar como la masa, tamaño y densidad factores importantes para el diseño del equipo, existiendo por otro lado características del objeto que son irrelevantes, debido a que no afectan a este proceso experimental de levitación. Por otro lado, es de importancia la forma que va a poseer este elemento, debido a que el holograma acústico emitido debe propagarse de manera uniforme en todo el contorno del cuerpo para que el rayo tractor sónico pueda mantenerlo suspendido en el aire. Este contorno debe de ser homogéneo, por lo tanto, se escogerá una esfera, debido a que su superficie es tal como se requiere, permitiendo una distribución de la energía más estable a comparación de otras formas.

El material de las esferas a analizar es de poliestireno expandido, el cual presenta las siguientes características:

Tabla 3.6. Elemento a manipular

Material	Forma	Diámetro (m)	Densidad (kg/m ³)
Poliestireno expandido	Esférica	0,003	19.00

(Elaborado por: Autores de la tesis)

3.4.1.1. Volumen del elemento a manipular

Tendiendo los datos principales del cuerpo que se someterá a la energía del rayo tractor sónico, a partir de los valores del diámetro y la densidad de la esfera, se obtendrá el volumen de esta.

$$V_{volumen} = \frac{4}{3}\pi r^3 (m^3) \quad (3.17)$$

$$V_{volumen} = \frac{4\pi}{3} \left(\frac{0.003}{2}\right)^3 (m^3)$$

$$V_{volumen} = 1.4137 * 10^{-8} (m^3)$$

3.4.1.2. Masa del elemento a manipular

$$m = \rho * V_{volumen} (Kg) \quad (3.18)$$

$$m = 19 * 1.4137 * 10^{-8} (Kg)$$

$$m = 2.68 * 10^{-7} (Kg)$$

Datos obtenidos:

Tabla 3.7. Tabla de resultados del elemento

Material	Forma	Volumen (Kg/m ³)	Masa (Kg)
Poliestireno expandido	Esférica	1.4137*10-8	2.68*10-7

(Elaborado por: Autores de la tesis)

Por lo tanto, la muestra para el diseño del equipo levitador por hologramas acústicos es una esfera de poliestireno expandido.

3.4.2. Datos para al diseño del Rayo Tractor Sónico

3.4.2.1. Frecuencia de Resonancia:

Al obtener ya las características necesarias del objeto a manipular es posible determinar la frecuencia que se debe generar por el sistema para la levitación.

Sin embargo, se debe considerar la siguiente restricción:

$$\emptyset_{\text{diametro del objeto}} \leq \frac{1}{2} \lambda \quad (3.19)$$

Dónde:

λ : Longitud de onda

La relación entre el diámetro del objeto y la longitud de onda, es que esta última debe ser mucho mayor que la primera característica con el fin de que la presencia de la partícula no afecte al campo acústico a generar.

Teniendo en cuenta que el diámetro de la esfera es de 0.003 m, se halla la longitud de onda:

$$0.003 \leq \frac{1}{2} \lambda$$

$$\lambda \geq 0.006 \text{ m}$$

A esta longitud de onda se le aumentará una tolerancia de 40%, lo que corresponde a un margen aproximado de 0.0025 m, esta tolerancia se justifica debido a que no se desea que las perturbaciones externas no interfieran con el campo acústico.

De acuerdo a la restricción previamente planteada se halla la frecuencia:

$$f = \frac{v}{\lambda} \text{ (Hz)} \quad (3.20)$$

Dónde:

v: Velocidad del sonido

λ : Longitud de onda

$$f = \frac{340}{0.0085} \text{ (Hz)}$$

$$f = 40 \text{ K (Hz)}$$

3.4.2.2. Potencia Acústica

Con las últimas características definidas, es de importancia la obtención de la energía acústica necesaria para contrarrestar la gravedad de dicho material a manipular.

Para obtener la energía necesaria, se debe hallar el número de ondas:

$$k = \frac{2\pi}{\lambda} \quad (3.21)$$

$$k = \frac{2\pi}{0.0085}$$

$$k = \frac{2\pi}{0.0085}$$

$$k = 739,20$$

Ecuación para la energía:

$$E_{min} = \frac{4\rho g}{5k} (J)$$

Dónde:

$$E_{min} = 4 * \frac{19 * 9.806}{5 * 739.20}$$

$$E_{min} = 0.2015 \text{ J}$$

Al convertirlo en Watts*s:

$$E_{min} = 0.2015 \text{ Ws}$$

Potencia acústica:

$$L_w = 10 * \log\left(\frac{W_1}{W_0}\right)$$

Dónde:

W_1 : Potencia hallada en Watts

W_2 : Potencia umbral de audición

$$L_w = 10 * \log\left(\frac{0.2015}{10^{-12}}\right)$$

$$L_w = 10 * \log\left(\frac{0.2015}{10^{-12}}\right)$$

$$L_w = 10 * \log\left(\frac{0.2015}{10^{-12}}\right)$$

$$L_w = 113.04 \text{ dB}$$

3.4.2.3. Presión Acústica

Sabiendo lo siguiente:

$$L_p = L_w = 20 * \log\left(\frac{P}{P_0}\right) = 10 * \log\left(\frac{W}{W_0}\right) \quad (3.22)$$

Dónde:

P_0 : Presión audible = 20u Pa

Teniendo el resultado anterior:

$$L_w = 113.04 \text{ dB}$$

$$113.04 = 20 * \log\left(\frac{P}{20 * 10^{-6}}\right)$$

$$P = 20 * 10^{-6} * 10^{113.04/20}$$

$$P = 8,975 \text{ Pa}$$

3.4.2.4. Fuerza del Rayo Tractor Sónico

$$P = \frac{F}{A} \text{ (Pa)} \quad (3.23)$$

Para el área de la esfera donde se aplica la fuerza a buscar:

$$A = 4\pi r^2 \text{ (m}^2\text{)}$$

$$A = 4\pi(0.0015)^2 \text{ (m}^2\text{)}$$

$$A = 2,83 * 10^{-5} \text{ m}^2$$

Volviendo a la fuerza del transductor:

$$F = P * A (N)$$

$$F = 8,975 * 2,83 * 10^{-5} (N)$$

$$F = 2,538 * 10^{-4} (N)$$

Valor del peso de la esfera:

$$w = m * g (N)$$

$$w = 2,68 * 10^{-7} * 9.806 (N)$$

$$w = 2,638 * 10^{-6} (N)$$

Se determina lo siguiente: la fuerza hallada que posee el rayo tractor sónico es sumamente mayor al peso del objeto a manipular, por lo que se debe contrarrestar con otras fuerzas emitidas por otros altavoces para así suspender el objeto en el nodo de intersección formado por todas las fuentes de sonido que emiten una frecuencia en fase de 40KHz.

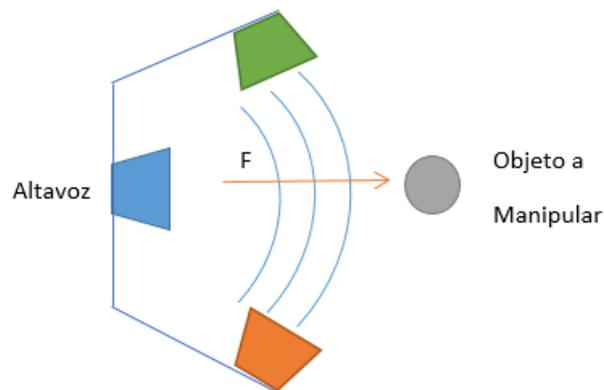


Figura 3.38. Funcionamiento del rayo tractor sónico

Selección de dispositivo:

En el resumen de la tabla 3.8 de parámetros obtenidos, se decide bajo los requisitos planteados, por escoger el siguiente altavoz:

Tabla 3.8. Parámetros para la elección de los dispositivos

Item	Transmisor	Receptor
Modelo	MSO-PT1040H07R	
Frecuencia central	40.0±1.0KHz	
Presión sonora	≥110dB	-----
Sensibilidad	-----	≥-67dB
Directividad	60°±15°	
Capacitancia	2300±30%pF	
Material de la carcasa	Plástico	
Distancia entre puntos	0.2~18m	
Voltaje de conducción	≤40Vrms	
Temperatura de funcionamiento	-20~+70°C	
Temperatura de almacenamiento	-40~+85°C	

(Elaborado por: Autores de la tesis)

3.4.3. Plano de Altavoz Ultrasónico

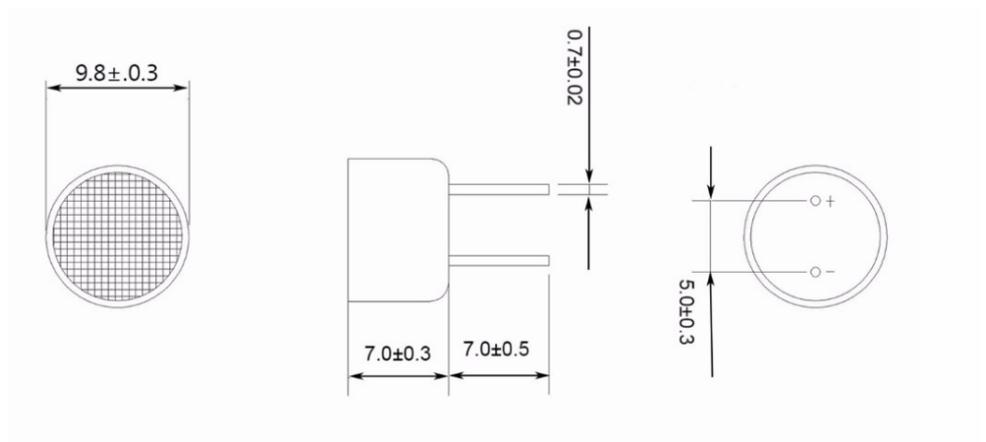


Figura 3.39. Diseñado por el fabricante

CAPÍTULO IV

ANÁLISIS Y RESULTADOS

4.1. Simulaciones y Validaciones

Las simulaciones computacionales del sistema de posicionamiento del robot manipulador fueron concluyentes y certeros, el algoritmo desarrollado lleva al tercer eslabón sobre el objeto a manipular. Sin embargo, es necesario realizar la implementación de un prototipo a escala para corroborar y validar los datos numéricos obtenidos en las simulaciones. En la figura 4.1 muestra el prototipo implementado, el cual consta de una cámara con resolución VGA en la parte final del tercer eslabón, el objetivo de la cámara es capturar el elemento y posteriormente identificarlo, además, se tiene diodos leds al lado de la cámara para mejorar el entorno de la captura de imágenes, los servomotores empleados son los más comerciales en los centros de electrónica y robótica, de costo accesible.

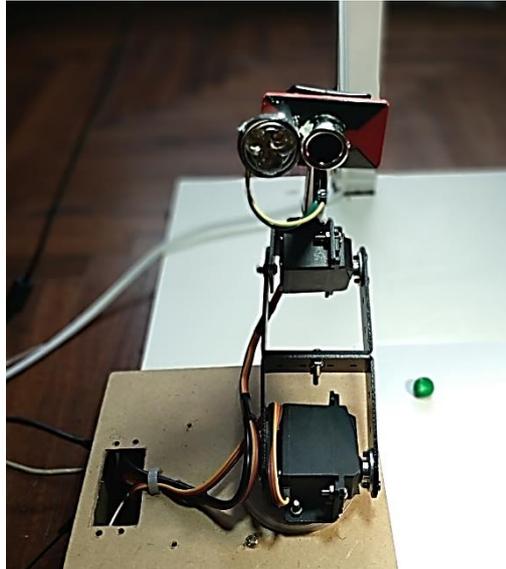


Figura 4.1. Prototipo del robot manipulador de 3 GDL

Así mismo, para la captura de la imagen del área de trabajo es necesario que una segunda cámara se posicione en la parte superior a una altura aproximada de 60 cm, para ello, se tiene un pedestal que sujeta la cámara con resolución VGA con la finalidad de hacer posible una captura uniforme y sin perturbaciones, así mismo, al lado se tiene un bulbo led de 9W a 220 VAC de color blanco, todos estos elementos se visualizan en la figura 4.2.



Figura 4.2. Vista general del prototipo para las pruebas.

A continuación, se procede a realizar las simulaciones y validaciones para dos aspectos desarrollados en esta investigación, Posicionamiento del robot manipulador por medio del control servo visual y el reconocimiento de los elementos sobre el área de trabajo para su posterior clasificación.

4.1.1. Simulación y Validación de la adquisición de imagen para el posicionamiento del robot:

El algoritmo desarrollado en el programa Matlab determinar en primer lugar, la distancia que existe desde la base del robot hasta el centroide del elemento a manipular, posteriormente, se calcula los ángulos para el posicionamiento en base a la longitud de sus eslabones. Para este caso, el prototipo cuenta con una altura de la base de 7 cm, el primer eslabón tiene una longitud de 11 cm y el segundo de 14 cm, el espacio de trabajo del manipulador es de 25x18 cm. Se han realizado 65 simulaciones para el posicionamiento del robot, por ejemplo, como resultado de una de ellas se tienen la figura 4.3 el cual muestra la posición representado por un punto en el plano, la recta indica la posición que debe de portar el robot para alcanzar al objeto.

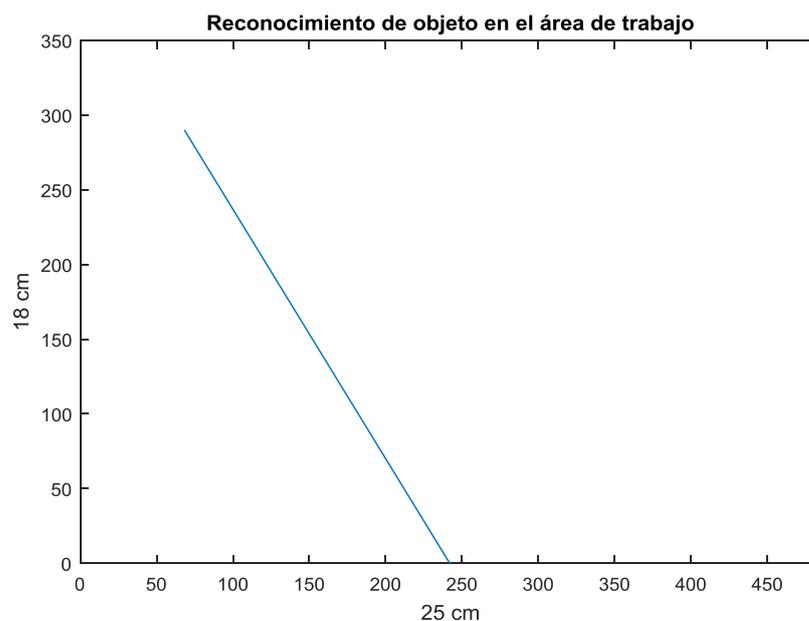


Figura 4.3. Reconocimiento del objeto en el mapa, prueba 1.

Como resultado del procesamiento del algoritmo, en pantalla nos indica que el primer ángulo denominado *theta1* debe realizar un giro de 103.14° para posicionarse tal como está representado en la figura 4.3, sin embargo, lo que ocurre en el movimiento es un cierto desfase por cada posición reconocida, para este caso, la figura 4.4 muestra el objeto remarcado por una circunferencia roja, ya que la cubierta de la cámara tapa el elemento.

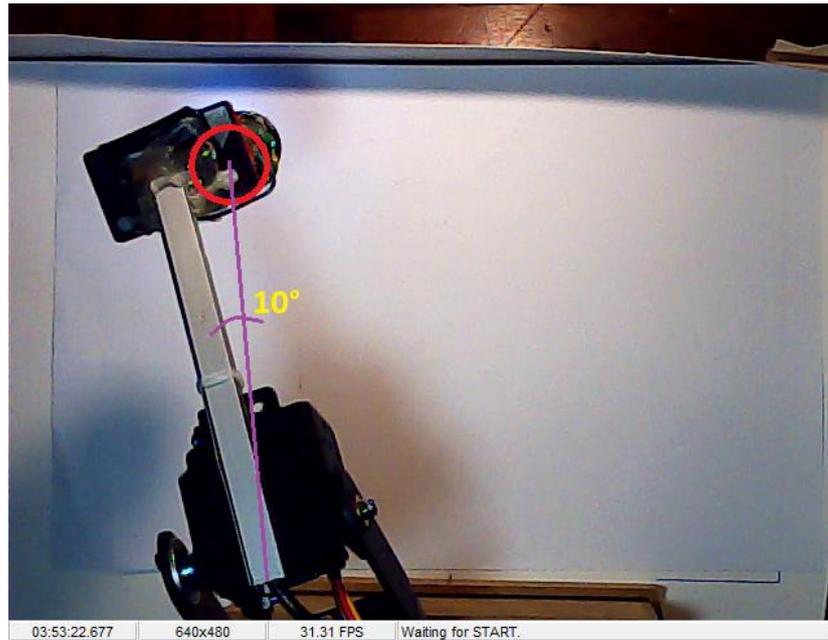


Figura 4.4. Captura del desfase de la posición, prueba 1.

Se tiene 10 grados como desfase al querer posicionarse en el centroide de la figura, para cada simulación, los ángulos de desfase son diferentes, presentando una segunda muestra de la simulación, se obtiene una siguiente posición, la figura 4.5 muestra la nueva posición en el plano de un elemento, como se visualiza está dentro del primer cuadrante de movimiento.

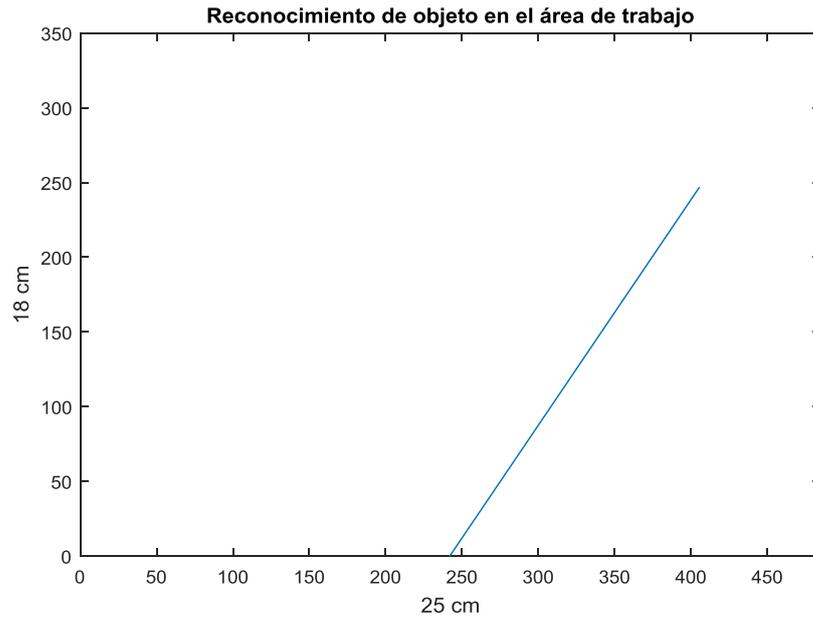


Figura 4.5. Reconocimiento del objeto en el mapa, prueba 2.

Una vez más, gracias a las condiciones del algoritmo es posible visualizar los valores de los ángulos, para esta posición, θ_1 deberá de girar 58.32° . Sin embargo, al realizar la captura con la misma cámara ya posicionada para la recolección de muestras, se puede observar un desfase, la figura 4.6 muestra la posición a la que ha llegado el robot.

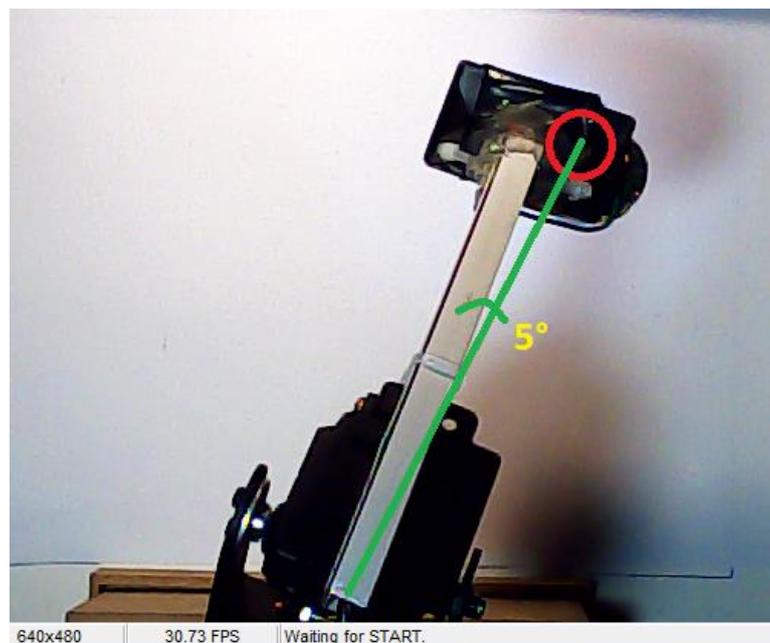


Figura 4.6. Captura del desfase de la posición, prueba 2.

El ángulo que se ha determinado como desfase corresponde a 5° , como ya se mencionó, el total de las simulaciones en tiempo real fueron de 65, en cada uno de ellas se han tomado los respectivos datos para formar la tabla 4.1 en donde se muestra los valores del primer ángulo θ_1 y el desfase que presenta cada prueba.

Tabla 4.1. Simulaciones y desfases de ángulos

prueba	theta 1	desfase
1	103	10
2	58.32	5
3	120	11
4	53	12
5	60	9
6	113	10
7	14	7
8	107	9
9	93	7
10	39	5
11	170	5
12	137	9
13	92	11
14	78	5
15	91	14
16	37	14
17	89	8
18	57	7
19	96	13
20	32	9
21	140	11
22	51	5
23	57	9
24	117	8
25	48	6
26	100	8
27	121	12
28	51	13
29	21	5
30	77	11

31	45	6
32	49	12
33	109	9
34	119	13
35	71	12
36	86	11
37	120	6
38	100	13
39	86	4
40	80	8
41	146	5
42	137	12
43	106	13
44	89	8
45	72	10
46	80	9
47	114	12
48	120	10
49	72	6
50	117	5
51	76	10
52	98	5
53	85	10
54	78	7
55	156	8
56	111	9
57	108	8
58	125	10
59	127	7
60	145	7
61	85	5
62	94	13
63	143	9
64	132	12
65	64	7

(Elaborado por: Autores de la tesis)

Se realizará un promedio de los ángulos de θ_1 y del desfase, esto con la finalidad de obtener un grado de validación para el método planteado para el posicionamiento del robot

$$\text{Prom Theta 1} = 91.8^\circ$$

$$\text{PromDesfase} = 8.9^\circ$$

$$\%Validación = 100\% - \frac{8.9 * 100}{91.8} \times 100$$

$$\%Validación = 90.2\%$$

Finalmente, se tiene un porcentaje de validación de un 90.2% en cuanto al posicionamiento del robot gracias al sistema servo visual desarrollado.

4.1.2. Simulación de la Red Neuronal Artificial para el reconocimiento de elementos a clasificar.

La simulación de la red neuronal artificial capacitada para la clasificación de elementos químicos, tuvo ciertos parámetros claves que cumplir con anterioridad con el fin de tener un correcto funcionamiento de acuerdo a lo entrenado. Es necesario en esta etapa considerar los aspectos que interfieren en el proceso de adquisición de muestras, si bien se sabe, se evalúa en esta investigación cuatro tipos de clases que corresponden a cuatro elementos químicos quienes son representado por cuatro colores: verde, rojo, azul y negro, con 15 muestras cada una generando 60 en total. Estas capturas deben poseer una correcta y estable iluminación ya que son datos registrados para que la red neuronal artificial se entrene bajo un criterio único y luego determine la identificación bajo lo que ha aprendido anteriormente.

Cumpliendo con los parámetros básicos de configuración previa, se procede a la captura de las 60 muestras, que, por medio de un código desarrollado en MATLAB, genera una base de datos analizada por las capas de colores que poseen las clases siendo una matriz de [60x768]. La data obtenida se empareja debidamente con las salidas esperadas o deseadas, en este caso bien sea identificado como verde, rojo, negro o azul, quienes serán parte fundamental del entrenamiento donde utilizando 150 de neuronas en la capa oculta, se da por inicio al aprendizaje de la red neuronal artificial otorgando un 70% del total

de las muestras que equivale aproximadamente a 42 muestras del total para el entrenamiento, mientras que un 15% del restante del total para la validación que son aproximadamente 9, del mismo modo se realizó para el testeo. Otorgando toda esta evaluación en matrices de confusión para poder determinar que tanto la red ha aprendido de los grados de que equivale a la cual entrega distintos datos fundamentales que permiten saber si la red se encuentra apta para cumplir su funcionamiento o no.

Unos puntos indispensables producto de las pruebas son la verificación total de este proceso, el cual está determinado por las condiciones naturales del sistema implementado en el prototipo, el número adecuado de neuronas en la capa oculta o si la red se encuentra correctamente entrenada.

Por último, el sistema tiene un grado de inteligencia capaz de identificar los elementos según el color al que ha sido asociado y clasificarlos debidamente, este grado de inteligencia se ha llevado a cabo gracias al número de neuronas en la capa oculta, el cual es de 150 neuronas, se observa en la matriz de confusión de la figura 4.3 que en el entrenamiento se ha obtenido el 100%, siendo la clase 21.4% perteneciente la más baja en el rendimiento. Sin embargo, estos valores no difieren en gran magnitud y por ello, se puede afirmar que la red es eficiente.

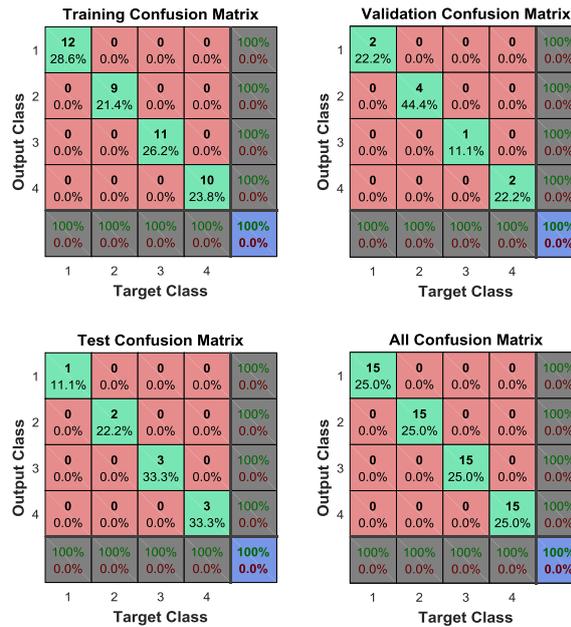


Figura 4.3. Matrices de resultados del entrenamiento de la red neuronal

Disponiendo del prototipo ya implementado se realizaron 48 pruebas de reconocimiento de colores, de las cuales 8 de ellas no acertaron en el resultado esperado, dichos valores, se visualizan en la tabla 4.2.

Tabla 4.2. Resultados de las simulaciones para el reconocimiento de elementos.

Prueba	Color de Elemento	Matriz de resultados
1	verde	[0.10 0.40 0.10 0.85]
2	verde	[0.30 0.67 0.40 0.27]
3	verde	[0.00 0.00 0.00 1.00]
4	verde	[0.60 0.28 0.00 0.94]
5	verde	[0.20 0.31 0.11 0.64]
6	verde	[0.10 0.28 0.40 0.76]
7	verde	[0.60 0.82 0.13 0.71]
8	verde	[0.12 0.25 0.32 0.45]
9	verde	[0.48 0.16 0.40 0.61]
10	verde	[0.22 0.27 0.15 0.70]
11	verde	[0.60 0.28 0.09 0.94]
12	verde	[0.60 0.28 0.16 0.94]
13	azul	[0.23 0.13 0.71 0.61]
14	azul	[0.20 0.23 0.75 0.26]
15	azul	[0.95 0.27 0.81 0.32]
16	azul	[0.17 0.32 0.59 0.22]
17	azul	[0.13 0.20 0.72 0.45]
18	azul	[0.16 0.25 0.60 0.20]
19	azul	[0.19 0.42 0.78 0.62]

20	azul	[0.24 0.44 0.85 0.23]
21	azul	[0.18 0.11 0.75 0.32]
22	azul	[0.56 0.47 0.62 0.43]
23	azul	[0.29 0.31 0.60 0.20]
24	azul	[0.10 0.12 0.36 0.11]
25	rojo	[0.12 0.14 0.31 0.78]
26	rojo	[0.15 0.86 0.44 0.23]
27	rojo	[0.45 0.95 0.29 0.47]
28	rojo	[0.22 0.71 0.57 0.13]
29	rojo	[0.18 0.73 0.45 0.19]
30	rojo	[0.54 0.49 0.60 0.45]
31	rojo	[0.32 0.89 0.62 0.36]
32	rojo	[0.33 0.86 0.19 0.30]
33	rojo	[0.37 0.80 0.27 0.16]
34	rojo	[0.45 0.96 0.36 0.12]
35	rojo	[0.38 0.71 0.82 0.18]
36	rojo	[0.16 0.85 0.46 0.51]
37	negro	[0.76 0.54 0.23 0.42]
38	negro	[0.53 0.31 0.29 0.16]
39	negro	[0.69 0.44 0.36 0.11]
40	negro	[0.81 0.27 0.38 0.09]
41	negro	[0.65 0.19 0.74 0.56]
42	negro	[0.96 0.10 0.36 0.67]
43	negro	[0.65 0.41 0.53 0.13]
44	negro	[0.73 0.62 0.24 0.59]
45	negro	[0.82 0.47 0.39 0.60]
46	negro	[0.75 0.05 0.41 0.11]
47	negro	[0.66 0.13 0.79 0.25]
48	negro	[0.92 0.18 0.26 0.71]

(Elaborado por: Autores de la tesis)

Gracias a la tabla 4.2 de resultados de simulación de RNA es posible determinar el porcentaje de validación de las pruebas en base al número de veces que la RNA ha acertado, las marcas amarillas indican los errores que ha presentado la red al momento de identificar, para 48 pruebas realizadas 8 de ellas no fueron certeras.

$$\%Validación = \frac{40 * 100}{48}$$

$$\%Validación = 83\%$$

El 17% restante corresponde al error producido por ciertos factores externos como, la calidad de la iluminación de la imagen capturada, la posición de los elementos, el ruido de la imagen, etc.

Por consiguiente, se afirma bajo el resultado de validación del 85% que el proceso elaborado para el entrenamiento, testeo y validación del aprendizaje de la RNA cumple con los requerimientos necesarios con el propósito de reconocer y clasificar cada elemento expuesto en el área de trabajo, determinando exclusivamente el color del objeto sin falla alguna al momento de analizar y procesar la información adquirida.

CONCLUSIONES

1. Se ha comprobado que el modelamiento matemático para un robot de tres grados de libertad ha llevado a realizar una correcta simulación aplicando el toolbox desarrollado por el autor Peter Corke, permitiendo visualizar de manera virtual el correcto desplazamiento del brazo manipulador en base a los parámetros previamente calculados. Asimismo, este resultado es corroborado gracias al prototipo físico, el cual cumple con las posiciones según los datos de simulación.
2. El número de neuronas en la capa oculta utilizadas para desarrollar la red neuronal artificial se ha determinado en base a pruebas en el entrenamiento, analizando resultados por cada número de neuronas entrenadas se comparó porcentajes de confusión para elegir el número correcto. Esto permite un mejor entrenamiento a la red, reduce errores y aumenta el rendimiento de aprendizaje.
3. Es posible el posicionamiento de un robot de tres grados de libertad por medio de un control servo visual, debido a que es capaz de obtener datos numéricos y plasmarlo en coordenadas de posicionamiento a base del centroide reconocido del objeto visualizado, por lo que se redirige el brazo robótico a la posición deseada, cumpliendo con todas las propiedades matemáticas que involucran al funcionamiento del robot.
4. Es posible la identificación y clasificación de los elementos a manipular, en este caso la aplicación esta direccionada a la industria química o laboratorios químicos, donde una red neuronal artificial utilice su entrenamiento para la clasificación a base de un reconocimiento de colores [r,g,b] permitiendo una automatización computacional a un proceso regularmente manual en la industrial permitiendo el aislamiento de operador ante cualquier sustancia la cual puede ser dañina para la salud.

RECOMENDACIONES

1. Para construir el prototipo utilizar servomotores de precisión y que cuenten con un sistema de engranaje de aleación de metales, debido a que los servomotores convencionales no se posicionan correctamente, además por tener engranajes de plástico, con la fricción tienen a dañarse fácilmente.
2. Para capturar la imagen del elemento es necesario contar con una correcta iluminación que reduzca al mínimo las sombras de los objetos sobre el área de trabajo.
3. Es necesario contar con una cámara de alta definición para la captura de imagen del objeto dentro del área de trabajo, ya que en el desarrollo del programa se está realizando una escala de valores de los píxeles de la imagen respecto a la dimensión del tablero. A mayor sea el número de píxeles el margen de error se reduce y así se evita el redondeo innecesario; al resolver las ecuaciones planteadas de la geometría; que genera ciertos desfases en los ángulos de posicionamiento.

BIBLIOGRAFÍA

Barrientos, A., Peñín, L., Balaguer, C., & Aracil, R. (2007). Fundamentos de Robótica (2da ed.). Madrid, España: S.A.U.

BASOGAIN, Xavier (2008). Redes Neuronales artificiales y sus aplicaciones. España: UPV-EHU.

CORKE Peter (2015) Machine vision toolbox for MATLAB (3ra edition). Australia: Springe.

CORKE Peter (2015) Robotics toolbox for MATLAB (9na edition). Australia: Springe.

ANDRADE M., BERNASSAU A. y ADAMOWSKI J. (2016) Acoustic levitation of a large solid sphere. EEUU: AIP. Disponible en: <https://aip.scitation.org/doi/10.1063/1.4959862>

BARRAZA Arturo (2007). La consulta a expertos como estrategia para la recolección de evidencias de validez basadas en el contenido. Durango, México: UPD. Disponible en: <https://dialnet.unirioja.es/descarga/articulo/2358908.pdf>

BARRIOS Guillermo (2007). Levitación en partículas en ondas de sonido usando el método de la ecuación de Boltzmann en redes. México D.F, México: UNAM. Disponible en: http://oreon.dgbiblio.unam.mx/F/1IQYR3ALSH6ALQVE9N1CTAESYJKEAJJKQD29J6PIY4G5UILYBP-44704?func=full-set-set&set_number=001974&set_entry=000003&format=999

BOULLOSA R., PÉREZ A. y DORANTES R. (2013). An Ultrasonic Levitator. México D.F., México: UNAM. Disponible en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-64232013000600007&lang=pt.

CAICEDO E. y LÓPEZ J. (2009). Una Aproximación Práctica a las Redes Neuronales Artificiales. Santiago de Cali, Colombia: PEUV. Disponible en: <http://bibliotecadigital.univalle.edu.co/handle/10893/10330>

DÍAZ José (2008). Sistematización del diseño de dispositivos de levitación superconductor por efecto Meissner. Madrid, España: UC3M. Disponible en: <https://e->

archivo.uc3m.es/handle/10016/5136

FORESTI D., BJELOBRK N., NABAVI M., y POULIKAKOS D. (2011). Investigation of a line-focused acoustic levitation for contactless transport of particles. Zúrich, Suiza: JAP. Disponible en: https://www.researchgate.net/publication/234982031_Investigation_of_a_line-focused_acoustic_levitation_for_contactless_transport_of_particles.

GARCÍA F., CUEVAS A. y NIÑO Y. (2016). Segmentation of color images by chromaticity features using self-organizing maps. Texcoco, México: UAEM. Disponible en: 10.15446/ing.investig.v36n2.55746.

GÓMEZ J., SÁNCHEZ J., OCAMPO A. y RESTREPO J. (2012). Aplicación de redes neuronales en la clasificación de arcillas. Medellín, Colombia: Revista EIA, núm. 17. Disponible en: <http://web.b.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=10&sid=c955b470-bea7-4e6c-a2f6-adafacb9f88a%40sessionmgr102>.

HERNÁNDEZ G., RÍOS L. y BUENO M. (2016). Implementación de un controlador de posición y movimiento de un robot móvil diferencial. Bogotá, Colombia: UDFJC. Disponible en: 10.14483/udistrital.jour.tecnura.2016.2.a09

JAMALI A., MAT I., MOHD P. y TOKHI M. (2017) Intelligent modeling of double link flexible robotic manipulator using artificial neural network. Skudai, Malasia: UTM. Disponible en: https://www.researchgate.net/publication/324134028_Intelligent_modeling_of_double_link_flexible_robotic_manipulator_using_artificial_neural_network

LUYO Christian (2015). Diseño e implementación de un sistema de control e interfaz para un brazo robótico de 5 GDL. Lima, Perú: PUCP. Disponible en: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6450>

MARZO A., ANN S., DRINKWATER B., RANJAN D. y LONG B. (2015) Holographic acoustic elements for manipulation of levitated objects. Pamplona, España: UNAV. Disponible en: DOI:10.1038/ncomms9661

MARZO A., BARNES A. y DRINKWATER B. (2017) TinyLev: A multi-emitter single-axis acoustic levitator. Bristol, Reino Unido: UoB. Disponible en: DOI:10.1063/1.4989995

MARZO A., CALEAP M. y DRINKWATER B. (2018) Acoustic Virtual Vortices with Tunable orbital angular momentum for trapping of mie particles. Bristol, Reino Unido: UoB. Disponible en: DOI: 10.1103/PhysRevLett.120.044301

MATEUS S., GONZÁLEZ N. Y BRANCH J. (2014). Aplicación de Redes Neuronales Artificiales en Entornos Virtuales Inteligentes. Medellín, Colombia: UNC. Disponible en: <https://scielo.conicyt.cl/pdf/infotec/v25n5/art15.pdf>

MEDRANO D. y RUIZ D. (2010) Diseño y construcción de un levitador acústico, mediante el principio de onda estacionaria, aplicado a transporte sin contacto. Bogotá, Colombia: USB. Disponible en: <http://hdl.handle.net/10819/5975>

Ortiz N. y Vinueza R. (2006). Diseño y construcción de un sistema semiautomático de limpieza por ultrasonido para muestras metalográficas y fractográficas con aplicación a un banco de pruebas y limpieza de inyectores a gasolina para el laboratorio de metalurgia de la ESPE. Sangolquí, Ecuador: ESPE. Disponible en: repositorio.espe.edu.ec/xmlui/handle/21000/824

PANDEY Shalini (2018) Concept for automated classification by means of a robotic arm. Gujarat, India: UBV. Disponible en: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7589474&filter%3DAND%28p_IS_Number%3A7724213%29%26rowsPerPage%3D100&pageNumber=1&rowsPerPage=100

RODRÍGUEZ R., SÁNCHEZ M., RAVICHAGUA M. y CASTILLO S. (2011). Control del brazo robótico de seis grados de libertad, mediante retroalimentación visual, para la cosecha del ají paprika. Lima, Perú: UNI. Disponible en: <http://cybertesis.uni.edu.pe/handle/uni/13699>

SEGURA Ángela (2003). Diseños Cuasi-Experimentales. Antioquia, Colombia: UNA. Disponible en: http://www.sld.cu/galerias/pdf/sitios/renacip/disenos_cuasiexperimentales.pdf

TRIANA D., ROA S. y FORERO C. (2015). Desarrollo y control de un brazo robótico mediante la adquisición de datos en tiempo real hacia un espacio no real. Bucaramanga, Colombia: UAB. Disponible en: https://www.researchgate.net/publication/317732439_Desarrollo_y_control_de_un_brazo_robotico_mediante_la_adquisicion_de_datos_en_tiempo_real_hacia_un_espacio_no_real

VÁSQUEZ Juan (2014). Red Neuronal Feedforward como estimador de patrones de corrientes en el interior del puerto de manzanillo sujeto a la acción de tsunamis. México DF, México: IMT. Disponible en: <https://www.imt.mx/archivos/Publicaciones/PublicacionTecnica>

ZABALA S., MARVIN J. y YASSER G. (2009). Metodología para la implementación de un sistema de reconocimiento de objetos mediante técnicas de visión artificial aplicada a manipuladores robóticos. Bucaramanga, Colombia: UDI. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=4991532>

DONG, G.; ZHU, Z. H. Predictive visual servo kinematic control for autonomous robotic capture of non-cooperative space target. Acta Astronautica, [s. l.], v. 151, p. 173–181, 2018. Disponible en: <http://search.ebscohost.com/login.aspx?direct=true&db=afh&AN=132426431&site=ehost-live>

BUITRAGO-SALAZAR, G.; RAMOS-SANDOVAL, O. Sistema de servo control visual empleando redes neuronales y filtros en el dominio de CIELAB. Ingeniería Mecánica, [s. l.], v. 18, n. 2, p. 100–108, 2015. Disponible en: <http://search.ebscohost.com/login.aspx?direct=true&db=fap&AN=108882797&site=ehost-live>. Acceso el: 10 diez. 2018.

DONG, G.; ZHU, Z. H. Incremental visual servo control of robotic manipulator for autonomous capture of non-cooperative target. Advanced Robotics, [s. l.], v. 30, n. 22, p. 1458–1465, 2016. Disponible en: <http://search.ebscohost.com/login.aspx?direct=true&db=afh&AN=118223475&site=ehost-live>.

ANEXO 1

FICHA DEL TRABAJO DE INVESTIGACIÓN

FACULTAD: Ingeniería de Sistemas y Electrónica

CARRERA: Ingeniería Mecatrónica

1. Título del Trabajo de Investigación propuesto

Aplicación de redes neuronales a la robótica

2. Indica la o las competencias del modelo del egresado que serán desarrolladas fundamentalmente con este Trabajo de Investigación:

Desarrollo del modelo cinemático y dinámico de un robot, procesamientos digitales de imágenes, desarrollo de redes neuronales artificiales para satisfacer los problemas planteados

3. Número de alumnos a participar en este trabajo. (máximo 2)

Número de alumnos: 2

4. Indica si el trabajo tiene perspectivas de continuidad, después de obtenerse el Grado Académico d Bachiller, para seguirlo desarrollando para la titulación por la modalidad de Tesis o no.

SI

5. Enuncia 4 o 5 palabras claves que le permitan realizar la búsqueda de información para el Trabajo en Revistas Indizadas en WOS, SCOPUS, EBSCO, SciELO, etc., desde el comienzo del curso y obtener así información de otras fuentes especializadas.

Ejemplo:

Palabras Claves	SciELO	EBSCO	IEEE
Palabra 1: Redes neuronales	exitoso	exitoso	exitoso
Palabra 2: Control de Robots	exitoso	exitoso	exitoso
Palabra 3: Modelado de un robot manipulador	exitoso	exitoso	exitoso
Palabra 4: Procesamiento de señales	exitoso	exitoso	exitoso
Palabra 5: Control Servo visual	exitoso	exitoso	exitoso

6. Como futuro asesor de investigación para titulación colocar:

(Indique sus datos personales)

a. Nombre: José Ambrosio Machuca Mines

b. Código Docente: C08292

c. Correo: C08292@utp.edu.pe

d. Teléfono:

e. Otro:

7. Especifica si el Trabajo de Investigación:

(Marca con un círculo la que corresponde, puede ser más de una)

a. Contribuye a un trabajo de investigación de una Maestría o un doctorado de algún profesor de la UTP.

b. Está dirigido a resolver algún problema o necesidad propia de la organización.

c. Forma parte de un contrato de servicio a terceros.

d. Corresponde a otro tipo de necesidad o causa (explicar el detalle):

Búsqueda de soluciones para el control de un robot manipulador con fines de aplicación a la industria química o de manufactura.

8. Explica de forma clara y comprensible los objetivos o propósitos del trabajo de investigación

Para la programación de robots se emplean las coordenadas cartesianas del espacio de trabajo del mismo, realizando el controlador la tarea de convertir la información de las coordenadas en variables articulares y motoras para los movimientos del robot. Es decir, que se necesita pasar del espacio físico al de las variables articulares o motoras. Se han aplicado funciones como la cinemática inversa y la dinámica inversa para abordar la temática planteada, pero estas pueden presentar el problema de la no linealidad. Las redes neuronales resultan prometedoras en cuanto a los resultados esperados para afrontar dicha problemática, ofreciendo una estructura de control que se adapte a las variaciones.

9. Brinde una primera estructuración de las acciones específicas que debe realizar el alumno para que le permita iniciar organizadamente su trabajo.

- Investigue sobre las funciones de cinemática inversa y dinámica inversa en la robótica.
- Investigue sobre las limitaciones de las funciones de cinemática inversa y dinámica inversa en la robótica.
- Investigue sobre las aplicaciones de las redes neuronales en la robótica.

10. Incorpora todas las observaciones y recomendaciones que consideres de utilidad para el alumno y a los profesores del curso con el fin de que desarrollen con éxito todas las actividades

Identificar información específica sobre las limitaciones de las funciones de cinemática inversa y dinámica inversa en la robótica.

Identificar investigación sobre las aplicaciones de las redes neuronales en la robótica.

11. Fecha y docente que propone la tarea de investigación

Fecha de elaboración de ficha: 27/03/2018

Docente que propone la tarea de investigación: Mercedes Zambrano Onofre

12. Esta Ficha de Tarea de Investigación ha sido aprobada como Tarea de investigación para el Grado de Bachiller en esta carrera por:

(Sólo para ser llenada por la Facultad)

Nombre:

Código:

Cargo:

Fecha de aprobación de ficha (día/mes/año): _____ / _____ / _____

ANEXO 3

Programa desarrollado en Matlab 2015 para las pruebas en conjunto.

Programa desarrollado en Matlab 2015 para las pruebas en conjunto.

```
%%% BOZZETA DE LA CRUZ FERNANDO
%%% BRUNO VILLENA RONNY
%%%
close all, clear all, clc; clear port;
run('REDNEURONALCOLORES.m')
%%Config servo serial
load rect.mat
ardu=arduino('COM5','uno');
s1=servo(ardu,9)
s2=servo(ardu,10)
s3=servo(ardu,11)
writeDigitalPin(ardu,2,0);

while true
    writePosition(s1,0);
    pause(0.25);
    writePosition(s2,0.5);
    pause(0.25);
    writePosition(s3,0.5);
    pause(0.25);
    break
end
webcam=videoinput('winvideo',4,'YUY2_640x480')
preview(webcam)
pause(1)
set(webcam, 'ReturnedColorSpace', 'RGB');
[y, Fs] = audioread('CAPTURA.mpeg');
volum = y*5; % Incrementar volumen
sound(volum, Fs, 16)
a=input('Para capturar(Press Enter): ');
if ~strcmp(a,'1')
    prueba = getsnapshot(webcam);
    % [J, rect] = imcrop (prueba);
    I2 = imcrop (prueba, [90 90 485 350]);
    df=flipdim(I2,1);
    imshow(I2)
    display('capturado')
else return
end
closepreview(webcam)
th1=0; th2=0; th3=0;
% Img=imread('color_6.jpg'); %Lee imagen
Igris=rgb2gray(df); %Elige (columnas, filas, 1=red)
Ibin=Igris<85; %Crea la imagen binaria
Ibin2=bwareaopen(Ibin,45); %Filtra los pixeles fuera del
área de trabajo
se = strel('disk',8);

%Proceso para dilatación de imagen
BW2 = indilate(Ibin2,se);
figure(1)
imshow(BW2), title('Imagen Binarizada');
```

```

[y, Fs] = audioread('RECONOCIENDO POS.mpeg');
volum = y^5; % Incrementar volumen
sound(volum, Fs, 16)
pause(4)

%proceso para determinar las propiedades de la figura
stats = regionprops('table',BW2,'Centroid',...
    'MajorAxisLength','MinorAxisLength')
centers = stats.Centroid;
diameters = mean([stats.MajorAxisLength stats.MinorAxisLength],2);
radii = diameters/2;
hold on
viscircles(centers,radii);
hold off

%Guardar variables de centroide en coordenadas
xpos=centers(1,1);
ypos=centers(1,2);

%Dibujar punto de centroide con coordenadas de resolución
figure(2)
plot([xpos 242],[ypos 0])

%plot(xpos,ypos,'x');
title('Reconocimiento de objeto en el área de trabajo');
axis([0 485 0 350]);
xlabel('25 cm');
ylabel('18 cm');
pause(1);
line(3,4);

%Conversión de unidades de píxeles a cm
%area de trabajo
xmesa=25;
ymesa=18;

xpixel=485;
ypixel=350;

ymetros=round((ypos*ymesa)/ypixel,3);
xmetros=round((xpos*xmesa)/xpixel,3);

[y, Fs] = audioread('DETERMINANDO COORDENADAS.mpeg');
volum = y^5; % Incrementar volumen
sound(volum, Fs, 16)
pause(4)

%Determinando el ángulo de giro para theta 1
if xmetros>=(xmesa/2)
    xmetros=xmetros-(xmesa/2);%-pi()/6;
    thetal=round(atan(ymetros/xmetros),3);
    thetal_ardu=round((thetal*180/pi)/180,3)+0.01%-0.05;
else
    xmetros=(xmesa/2)-xmetros;
    thetal=round(pi()-atan(ymetros/xmetros),3);%-pi()/15;
    thetal_ardu=round((thetal*180/pi)/180,3)-0.1%-0.08
    %-0.1%-0.05%-0.2;

```

```

end

%distancia del robot al objeto
distobjeto=sqrt(xmetros^2+ymetros^2)+1;

%determinando los ángulos theta2 y theta3
%para ello se desarrolla geométricamente las ecuaciones que describen
%la distancia total de un triangulo oblicuo, por ley de cosenos

% Distancia de los eslabones
esla2=11;
esla3=14;

theta2=round(acos((esla2^2+distobjeto^2-
esla3^2)/(2*esla2*distobjeto)),3);
theta2_ardu=1-round((theta2*180/pi)/180,3)+0.06;

theta3=round(pi-acos((esla2^2+esla3^2-distobjeto^2)/(2*esla2*esla3)),3);
theta3_ardu=1-round((theta3*180/pi)/180,3)+0.06;

[y, Fs] = audioread('SIMULACION MOVIMIENTO.mpeg');
volum = y*5; % Incrementar volumen
sound(volum, Fs, 16)

%% Algoritmo para la animación del movimiento del robot manipulador

L1=7; L2=esla2; L3=esla3;
%L=Link([th d a alph])
L(1)=Link([0 L1 0 pi/2]);
L(2)=Link([0 0 L2 0]);
L(3)=Link([0 0 L3 0]);
rob=SerialLink(L)
rob.name='Manipulador';

th1=pi()/2; th2=pi/2; th3=45;
figure(3)
rob.plot([pi/2, th2, th3])
%rob.fkine([th1,th2, th3])  matriz de homogeneidad
%syms th1 th2 th3

rob.fkine([0 0 0])
th1=0; th2=0; th3=0;

for th1=0:0.1:theta1
    figure(3)
    rob.plot([th1,0,0])
    pause(0.005);
end

for th2=0:0.1:theta2
    figure(3)
    rob.plot([th1,th2,0]);
    pause(0.005);
end

for th3=0:-0.1:-theta3

```

```

    figure(3)
    rob.plot([th1,th2,th3]);
    pause(0.005);
end

display('fin de la simulación')

%% Envio de angulos serial

[y, Fs] = audioread('DESPEJAR LA ZONA.mpeg');
volum = y*5; % Incrementar volumen
sound(volum, Fs, 16)
pause(3);
writeDigitalPin(ardu, 2, 1);
pause(0.1);
% for angulo1=0:0.005:thetal_ardu
%     display(angulo1)
%     pause(0.05)
% end

% for angulo1=0:0.005:thetal_ardu
%     writePosition(s1, angulo1);
%     pause(0.005)
% end

while true
    writePosition(s1, thetal_ardu);
    pause(0.25);
    writePosition(s2, theta2_ardu);
    pause(0.25);
    writePosition(s3, theta3_ardu);
    pause(0.25);
    break
end
pause(5)

%% %% Entorno para realizar las capturas de las muestras de datos
webcam=videoinput('winvideo',1,'UYV2_640x480')
preview(webcam);
pause(2)
set(webcam, 'ReturnedColorSpace', 'rgb');
[y, Fs] = audioread('CAPTURAR ELEMENTO.mpeg');
volum = y*5; % Incrementar volumen
sound(volum, Fs, 16)
a=input('Para capturar(Press Enter): ');
pause(1)
if ~strcmp(a, '1')
    ImagenColor=getSnapshot(webcam);
    display('capturado')
else return
end
closepreview(webcam)
writeDigitalPin(ardu, 2, 0);

%% Bloque de procesamiento de imagenes
% Igris=rgb2gray(ImagenColor); %Elige (columnas,
filas, 1=red)
% Ibin=Igris<85; %Crea la imagen binaria

```

```

% Ibin2=bwareaopen(Ibin,45); %Filtra los píxeles fuera del
área de trabajo
% se = strel('disk',1);

%Proceso para dilatación de imagen
% BW2 = imdilate(Ibin2,se);
% figure(1)
% imshow(BW2), title('Imagen Binarizada');
%
% stats = regionprops('table',BW2,'Centroid',...
% 'MajorAxisLength','MinorAxisLength')
% centers = stats.Centroid;
% diameters = mean([stats.MinorAxisLength],2);
% radii = diameters(1)/2;
% hold on
% viscircles(centers,radii);
% hold off
%
%
% ejex=centers(1,1)
% ejey=centers(1,2)
%
% xrec=ejex-radii-20;
% yrec=ejey-radii-20;
%
% xrec2=ejex+radii;
% yrec2=ejey;
%
% I2 = imcrop(Imagencolor, [xrec yrec xrec2 yrec2]);
% figure (1)
% imshow(I2)
%
% figure (2)
% imshow(Imagencolor)

imshow(Imagencolor)
MatrizdeDatos2=[];
LecturaImagen=imresize(Imagencolor,[480,640]);
HistogramaRojo= LecturaImagen(:,:,1); %Obtencion del Histograma rojo de
la imagen en el bucle.
[counts,x] = imhist(HistogramaRojo);%Retorno del vector del histograma en
escala de grises.
VecCarRedColumna=counts(1:256); %obtencion del vector caracteristico rojo
en columna.
VecCarRedFila=VecCarRedColumna';%obtencion del vector caracteristico rojo
en fila.
%figure(1), imhist(HistogramaRojo);

HistogramaVerde= LecturaImagen(:,:,2);%Obtencion del Histograma verde de
la imagen en el bucle.
[counts,x] = imhist(HistogramaVerde);%Retorno del vector del histograma
en escala de grises.
VecCarGreenColumna=counts(1:256);%obtencion del vector caracteristico
verde en columna.
VecCarGreenFila=VecCarGreenColumna';%obtencion del vector caracteristico
verde en fila.

HistogramaBlue= LecturaImagen(:,:,3);%Obtencion del Histograma azul de la
imagen en el bucle.

```

```

[counts,x] = imhist(HistogramaBlue);%Retorno del vector del histograma en
escala de grises.
VecCarBlueColumna=counts(1:256);%obtencion del vector caracteristico azul
en columna.
VecCarBlueFila=VecCarBlueColumna';%obtencion del vector caracteristico
azul en fila.

```

```

ConcateRGB=[VecCarRedFila VecCarGreenFila VecCarBlueFila];%Concatenacion
de los vectores de los histogramas rgb.
MatrizdeDatos2=[MatrizdeDatos2;ConcateRGB]; %Concatenacion de los
vectores caracteristicos de las imagenes en una matriz 0-1.

```

```

reconocimiento=net(MatrizdeDatos2)';

```

```

close all;
negro=reconocimiento(1,1);
rojo=reconocimiento(1,2);
azul=reconocimiento(1,3);
verde=reconocimiento(1,4);
elemMax=max(reconocimiento);

while true
    if verde==elemMax
        display('COLOR: VERDE')
        [y, Fs] = audioread('VERDE.mpeg');
        volum = y*5; % Incrementar volumen
        sound(volum, Fs, 16)

        while true
            writePosition(s1,0.9);
            pause(0.25);
            writePosition(s2,0.45);
            pause(0.25);
            writePosition(s3,0.2);
            pause(0.25);
            break
        end
        break
    end
end

if rojo==elemMax
    display('COLOR: ROJO')
    [y, Fs] = audioread('ROJO.mpeg');
    volum = y*5; % Incrementar volumen
    sound(volum, Fs, 16)

    while true
        writePosition(s1,0.9);
        pause(0.25);
        writePosition(s2,0.65);
        pause(0.25);
        writePosition(s3,0.3);
        pause(0.25);
        break
    end
    break
end
end

```

```

if negro==elemMax
    display('COLOR: NEGRO')
    [y, Fs] = audioread('NEGRO.mpeg');
    volum = y*5; % Incrementar volumen
    sound(volum, Fs, 16)

    while true
        writePosition(s1,0.9);
        pause(0.25);
        writePosition(s2,0.75);
        pause(0.25);
        writePosition(s3,0.4);
        pause(0.25);
        break
    end
    break
end

if azul==elemMax
    display('COLOR: AZUL')
    [y, Fs] = audioread('AZUL.mpeg');
    volum = y*5; % Incrementar volumen
    sound(volum, Fs, 16)

    while true
        writePosition(s1,0.9);
        pause(0.25);
        writePosition(s2,0.85);
        pause(0.25);
        writePosition(s3,0.5);
        pause(0.25);
        break
    end
    break
end

end
end
pause(3);
while true
    writePosition(s2,0.5);
    pause(0.25);
    writePosition(s3,0.5);
    pause(0.25);
    writePosition(s1,0);
    pause(0.25);
    break
end
end
[y, Fs] = audioread('FIN.mpeg');
volum = y*5; % Incrementar volumen
sound(volum, Fs, 16)

msgbox('Fin del proceso','Success')

```