

## **Prototipo de robot manipulador para manejo de residuos peligrosos biológicos e infecciosos mediante detección de color**

Antonio-Alexander Benítez-Arévalo, Jesús García García,  
Guillermo-Rey Peñaloza-Mendoza

Instituto Tecnológico Superior de Pátzcuaro,  
División de Ingeniería Biomédica,  
México

guillermoreypm@hotmail.com

**Resumen.** El presente trabajo aborda la construcción y prueba de un prototipo de robot manipulador de 3 grados de libertad (GDL) para el manejo de Residuos Peligrosos Biológico e Infecciosos (RPBI), asistido por medio de visión artificial para reconocer objetos por color, y así poder clasificarlos evitando de esta manera que el personal médico y de enfermería estén en contacto con estos agentes, reduciendo los accidentes provocados por errores humanos en su manejo. El manejo del robot se realiza mediante comandos de voz empleando el lenguaje de programación Python y enlazándolo con la plataforma Arduino donde se lleva a cabo el control de las articulaciones del robot. El reconocimiento de objetos se realiza empleando la librería llamada OpenCV, de Python, con una escala de colores HSV para la calibración en la detección de objetos. Los colores que se utilizan para la clasificación son rojo, amarillo y verde, esto en base a la clasificación de los RPBI según la Norma Oficial Mexicana NOM-087-ECOL-SSA1-2002. Para el reconocimiento de voz se emplea la librería `speech_recognition` de Python que permite utilizar los servicios de Google. Los movimientos del prototipo se basan en el reconocimiento de la posición de los RPBI y la posición HOME del robot, mediante cinemática inversa se obtiene el ángulo de referencia para los servomotores, los cuales comienzan a girar y de esta manera los eslabones del robot van acomodándose hasta llegar al objeto. Como efector final, el robot cuenta con un electroimán que toma el RPBI y permite su transporte.

**Palabras clave:** Robot manipulador, RPBI, visión artificial.

### **Manipulator Robot Prototype for Biological and Infectious Hazardous Waste Management by Color Detection**

**Abstract.** The present work deals with the construction and testing of a manipulator robot prototype with 3 degrees of freedom (DOF) for the handling of Biological and Infectious Hazardous Waste (RPBI), assisted by means of

artificial vision to recognize objects by color, and thus be able to classify them, thus preventing medical and nursing staff from being in contact with these agents, reducing accidents caused by human error in handling them. The robot is handled by voice commands using the Python programming language and linking it to the Arduino platform where the control of the robot's joints is carried out. Object recognition is performed using the Python OpenCV library with an HSV color scale for object detection calibration. The colors used for the classification are red, yellow and green, based on the classification of the RPBI according to the Official Mexican Standard NOM-087-ECOL-SSA1-2002. For speech recognition, the Python speech\_recognition library is used, which allows the use of Google services. The movements of the prototype are based on the recognition of the position of the RPBI and the HOME position of the robot, through inverse kinematics, the reference angle for the servomotors is obtained, which begin to rotate and in this way the links of the robot are accommodated until you reach the object. As an end effector, the robot has an electromagnet that picks up the RPBI and allows it to be transported.

**Keywords:** Robot manipulator, RPBI, artificial vision.

## 1. Introducción

Un robot manipulador puede ser definido como mecanismo, que suele consistir en una serie de segmentos, articulados o deslizándose entre sí, con la finalidad de agarrar y mover objetos, controlado remotamente por una computadora o por un humano [1]. A lo largo de la historia, se ha buscado realizar tareas de manera automática, por lo que, el desarrollo de nuevas tecnologías ha evolucionado hasta el punto en el que hoy en día existen muchos tipos de robots que hacen diferentes tareas en beneficio de la humanidad.

En el ámbito de la salud se cuenta con un área destinada a la manipulación y control de los agentes residuos peligrosos biológico e infecciosos (RPBI), esta actividad se realiza a nivel hospitalario de forma manual. Debido a sus características infecto-contagiosas, estos residuos representan un riesgo para la salud tanto de las personas como del medio ambiente, su manejo y control adecuado representa uno de los temas con mayor controversia dentro del sector salud [2].

En la actualidad, debido a la contingencia ocasionada por el COVID-19, los desechos RPBI incrementaron exponencialmente, según datos del Instituto Nacional de Ecología y Cambio Climático se estimaba que hasta julio de 2020 habría un incremento del 3.3 al 16.5% en la cantidad de agentes RPBI por la contingencia [3]. Estos datos muestran un mayor riesgo para los trabajadores de la salud, por lo tanto, el usar la tecnología para disminuir el contacto directo resulta una gran ayuda.

La aplicación de la robótica para la clasificación de estos agentes RPBI es parte fundamental en su adecuado manejo y para reducir peligros por exposición. Tomando los depósitos de los agentes RPBI se puede realizar la detección de estos por medio de visión artificial, discriminando los objetos por el color del envase contenedor en un espacio de almacenaje, en base a esto, determinar el material que se contiene y por medio de un robot manipulador tomarlos del depósito primario o temporal y llevarlos al depósito principal de cada uno de los tipos de agente RPBI. Para esto se diseña y

construye un robot manipulador, el cual cuenta con un algoritmo de control, que recibe la información de la posición relativa del depósito primario y el tipo de agente RPBI detectado para determinar los movimientos necesarios para el posicionamiento del robot, desde su posición inicial hasta el agente RPBI y posteriormente transportarlo al contenedor final respectivo.

### **1.1. Clasificación de residuos peligrosos biológico-infecciosos**

Según la Norma Oficial Mexicana NOM-087-ECOL-SSA1-2002, Protección ambiental-salud ambiental-residuos peligrosos biológico infecciosos-clasificación y especificación de manejo, se deben clasificar estos agentes por su origen y su estado físico, indicando su color y tipo de envase [4].

## **2. Estado del arte**

Robot angular, antropomórfico o también conocido como robot manipulador tiene sus tres principales articulaciones de tipo rotacional, con lo cual emplea las coordenadas angulares para determinar las posiciones de su elemento terminal. Se llama antropomórfico por que simula los movimientos de un brazo humano [1]. Este robot posee gran accesibilidad y maniobrabilidad, es rápido y ocupa poco espacio en relación al campo de trabajo que abarca. Debido a sus características, lo hacen un modelo versátil con múltiples aplicaciones.

En el mercado existen diversos robots manipuladores que se usan de manera industrial, por ejemplo, el Robot paletizador de botellas RP40 de la marca Tecno el cual es flexible, fiable y sencillo [5], el Robot KUKA KR 4 AGILUS es un robot compacto ampliamente utilizado en células de automatización [6] y el Robot ABB IRB 1100 cuyo diseño es compacto y ligero [7].

Se consideran los ejemplos para visualizar el alto desempeño que brinda un robot manipulador; además se usan como base en la construcción del manipulador de estudio, reduciendo costos con el uso de software libre de alto desempeño, implementación de algoritmos inteligentes y hardware de bajo costo.

Así mismo, hoy en día, la robótica es una de las áreas más exploradas, debido a que nos permite automatizar procesos que nos brinda la posibilidad de reducir peligros por exposición, realizar tareas repetitivas, entre otras actividades. Por lo cual, existen trabajos similares a sobre el uso de robots manipuladores aplicados a la detección de color o al manejo de RPBI, ejemplos de estos desarrollos se enlistan a continuación.

- Jordán Gamito en su trabajo de fin de grado [8], desarrolla un sistema que, mediante un brazo robótico es capaz de clasificar objetos según el color de estos. Empleando una cinta transportadora prototipo, pasa objetos de distinto color, un detector de presencia detiene el avance de la banda transportadora para que el color sea determinado a través de un sensor de color que le indicará al robot hacia donde moverlo,
- En [9] se presenta un sistema teleoperado para la manipulación de residuos de riesgo biológico, este proyecto se basa en la tecnología del control teleoperado y se apoya en un brazo robótico para el contacto directo, que posee sensores

para la recolección, en este el operador desempeña su labor de manera segura y eficaz,

- El desarrollo de un robot manipulador de 4 grados de libertad con el objetivo de clasificar objetos de diferentes colores utilizando técnicas de filtrado RGB se puede revisar en el trabajo de Rosas Arias et al, [10]. En este se muestra la construcción de un brazo robótico y la programación de una cámara para detectar objetos de colores y poder realizar su separación a posiciones fijas.

### **3. Desarrollo propuesto**

Actualmente, la pandemia por el COVID-19 provoca la generación de una gran cantidad de residuos peligrosos biológico e infecciosos en la atención hospitalaria, por lo que es indispensable tomar medidas sanitarias que garanticen el correcto manejo y eliminación de los residuos para mantener mínimos riesgos de contagio para los profesionales de la salud y la población en general por estar expuesta a los RPBI que pudieran tirarse de forma clandestina, y así evitar mayor número de contagios accidentales.

#### **3.1. Planteamiento del problema**

A lo largo de aproximadamente dos años la humanidad se ha visto afectada por la pandemia generada por el COVID-19, este virus ha cobrado la vida de millones de personas, y siendo la primera línea de defensa, los que están en mayor riesgo, son los médicos y enfermeras.

Estos atienden a los pacientes que lo necesitan, por lo que, tienen contacto directo con residuos peligrosos biológico e infecciosos que se van generando en las salas de consulta y en las mismas áreas de COVID que se han implementado en los complejos hospitalarios.

Sin embargo, el principal problema radica en la gran cantidad de residuos que se están generando y manejando dentro de las instalaciones hospitalarias, esto provoca que en ocasiones el personal no sea suficiente, además trabajando bajo presión y la sobrecarga de trabajo a la que se ven expuestos, incrementa la probabilidad de cometer errores de manejo o separación y verse infectados por los residuos biológicos e infecciosos, así como contaminar el entorno en áreas no COVID.

Actualmente, a nivel hospitalario, no se cuenta con tecnología para el manejo de los agentes RPBI dentro de la zona de almacenaje, todo el proceso, desde la generación hasta el transporte externo, se realiza por personal de salud de apoyo.

Este proceso hasta su extracción del hospital para su manejo integral dura días (varía de acuerdo al tamaño del hospital de 7 hasta 30 días), durante los cuales los agentes RPBI siguen multiplicándose, lo que genera un riesgo mayor.

#### **3.2. Solución propuesta**

Como propuesta de solución al manejo de agentes RPBI, se propone la construcción de un robot manipulador automatizado, que sirva como herramienta para la

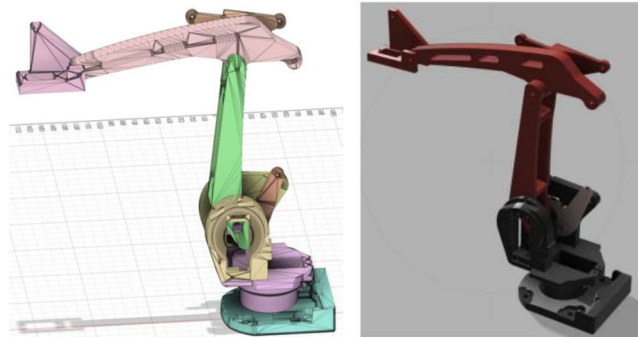


Fig. 1. Renderizado del ensamble del robot manipulador.

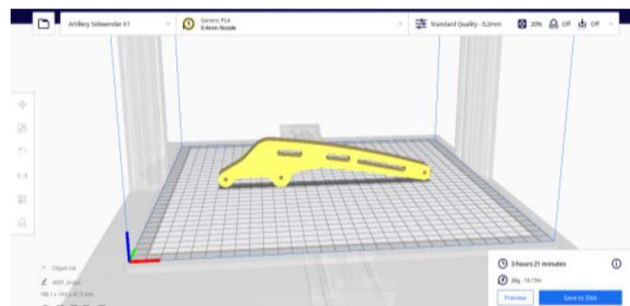


Fig. 2. Configuración de parámetros para las piezas mediante Cura Ultimaker.

clasificación y manejo de los agentes RPBI por medio de la detección del color de las envolturas de los agentes RPBI. Como metas principales se tienen:

- Construir un dispositivo robótico poliactuado, empleando tecnología 3D, con el cual se manipulen los agentes RPBI y se reduzca el riesgo de contagio en el personal de apoyo que maneja el almacenaje final de los agentes RPBI,
- Realizar la programación del robot que permita usar visión artificial para la detección y clasificación de los agentes RPBI,
- Control individual de los servomecanismos del robot para el posicionamiento en una localización de referencia (HOME) y en cada punto donde existan los agentes RPBI.

### 3.3. Diseño y construcción del robot manipulador

La implementación del prototipo se realizó mediante el diseño 3D del robot manipulador, primeramente, se realizó una investigación documental acerca de los diferentes tipos de robots manipuladores empleados en el manejo de objetos [1, 2, 6, 7, 9]. Con estos trabajos de referencia, se optó por emplear un robot manipulador básico de 3 grados de libertad, pero cuyos elementos de accionamiento no estuvieran montados sobre las articulaciones, esto para disminuir el peso del robot y permitir una mayor capacidad de carga en el efector final.



**Fig. 3.** Impresión de piezas mediante la impresora Artillery SideWinder X1.

Es importante mencionar que cada articulación está diseñada para ser controlada de manera independiente por un servomotor, el cual está conectado a una transmisión tipo biela – manivela para efectuar el movimiento a distancia de la articulación.

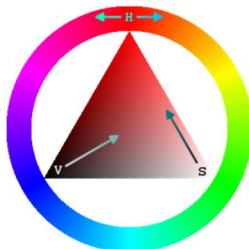
Como efector final se adapta un electroimán, el cual es un componente clave para el correcto funcionamiento del robot, ya que, se supone un elemento metálico en cada contenedor para su transporte, evitando así emplear una pinza que puede rasgar o romper los contenedores de plástico en el transporte, además un electroimán permite la limpieza de forma más sencilla que otros elementos porosos.

El diseño 3D de los elementos del robot manipulador se realizaron en el Software Fusion 360 de la compañía Autodesk, el cual es uno de los softwares más empleados para el diseño y modelado 3D. Fusion 360 es un software CAD, CAM y de circuitos impresos de modelado 3D basado en la nube para el diseño y la manufactura de productos. Cada eslabón del robot manipulador fue diseñado para que en extensión máxima alcanzara 27 cm (valor tomado con una escala 1:3 del robot ABB IRB 1100). En la Fig. 1, se presenta el renderizado del modelo del brazo robótico ensamblado completamente después del diseño de cada elemento, en este ensamble se verificó la interacción de cada elemento al hacer un movimiento, para evitar daños estructurales al momento de su implementación.

Para la implementación física, se debe configurar el tipo de material a emplear, el factor de relleno de la pieza, entre otros parámetros, esto para cada uno de los elementos a imprimir. Para la configuración de la impresión 3D se utilizó el software Ultimaker CURA, en la Fig. 2 se puede observar una pieza dentro del software de configuración para la impresión.

Una vez que las piezas fueron verificadas y configuradas para la impresión, se implementó una impresora Artillery SideWinder X1 para su fabricación, su tiempo de impresión total fue aproximadamente de 34 horas, gastando un aproximado de 150 gramos de material, en este caso al ser el primer prototipo se imprimió en PLA con un relleno del 20%. Parte de este proceso de fabricación se puede observar en la Fig. 3.

Una vez terminado el proceso de impresión, se optó por darle a las piezas un tratamiento para que su aspecto fuera mejorado, esto último debido a que el acabado final que entrega la impresora es rústico al usar un relleno del 20%. Este tratamiento fue lijar las piezas para remover las imperfecciones que quedaron en el proceso de impresión, posteriormente cada pieza fue sometida a un calor constante de 60 °C y a un baño de solvente, para crear un templado superficial que permita reformar la superficie y dejar un acabado pulido, por último, se agregan diversas capas de pintura vinílica para que resaltara su aspecto.



**Fig. 4.** Distribución del espacio de color HSV.

Al ensamblar físicamente el robot, el diseño prevé la colocación de los servomotores en la base del mismo, por lo tanto, es importante mencionar que, debido al diseño, los dos eslabones de extensión no pueden rotar de manera libre en todo el rango de movimiento que ofrecen los servomotores ( $0 - 180^\circ$ ), por lo que, el rango de movimientos viene dado de la siguiente manera:

- Primer GDL (Base):  $0 - 180^\circ$ ,
- Segundo GDL (Brazo):  $100 - 0^\circ$ ,
- Tercer GDL (Antebrazo):  $0 - 40^\circ$ .

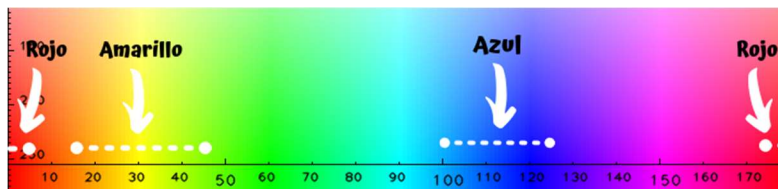
### 3.4. Programación para la detección por color de los agentes RPBI

Antes de iniciar el proceso de la detección del color de los objetos, es necesario recibir la orden del agente RPBI que se desea clasificar, esto puede ser por medio de botones, instrucciones o voz, en este caso, para disminuir el contacto físico de los usuarios, se realiza la orden por medio de voz usando los colores como referencias de las acciones a realizar.

El reconocimiento de voz, se realiza mediante Python utilizando la librería `speech_recognition`, la cual nos permite utilizar el micrófono integrado a una cámara web para adquirir la señal generada por las palabras, para posteriormente utilizar los servicios de google para el reconocimiento de voz mediante su diccionario, poder convertir el sonido a texto y este texto guardarlo como una variable que se utiliza como condicional en sentencias simples de casos `if` y `else if` donde mandamos llamar a las funciones que harán el reconocimiento por color.

Para la implementación de la detección de color se utiliza una cámara web de la marca Hisemy modelo ML2-MX-0095 que cuenta con las siguientes características: sensor de imagen de alta definición de 5 millones de píxeles, enfoque automático, calidad de cámara web 2K HD, capacidades de disparo de alta velocidad, grabación HD de  $2560 \times 1440$  a 30 fotogramas por segundo.

El código de programación se realiza mediante el lenguaje Python, en el cual se emplea el espacio de color HSV (Hue, Saturation, Value), como se ve en la Fig. 4, que significa matiz, saturación y brillo, de la librería de visión artificial de OpenCV, la diferencia con el espacio RGB, radica en usar un solo elemento, el matiz, para determinar el color a analizar en lugar de tres elementos. Los valores del HSV al igual que en el RGB, cuentan con rangos de trabajo, en este caso los canales de HSV en OpenCV son: H de 0 a 179, S de 0 a 255 y V de 0 a 255.



**Fig. 5.** Valores de matices (H) de componentes HSV. Imagen tomada del Blog Omes en el enlace <https://omes-va.com/deteccion-de-colores2/>.

Para determinar un color se toma el componente H que corresponde al matiz, que en este caso se puede ver en la Fig. 5, que este componente va cambiando de rojo, amarillo, verde, violeta hasta llegar nuevamente a rojo, siendo los matices que nos interesan el verde, rojo y amarillo. Bajo este esquema, se realizan 4 pasos generales para la detección del color, descritos a continuación.

Paso 1.- Tomar la imagen a procesar: Primeramente se debe leer la imagen que se desea procesar para detectar los colores, en este caso particular, se toma la imagen desde el video adquirido por la cámara web, esto se realiza con las siguientes instrucciones en Python. Para capturar el video se emplea la instrucción `cap = cv2.VideoCapture(0)`, posteriormente se debe capturar un fotograma para analizar el contenido, esto se hace con la instrucción `ret,frame=cap.read()`.

Paso 2.- Transformar de BGR a HSV: Al leer los fotogramas del video, por defecto se toma en valores BGR, por lo tanto, para hacer un manejo más simple de los colores, se procede a transformar el espacio a valores HSV, donde se discriminará por matices. Esta conversión se realiza con la instrucción `cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)`, como primer argumento se tiene la imagen adquirida en el primer paso llamada "frame", esta corresponde a la que se desea convertir a valores HSV, el segundo argumento "cv2.COLOR\_BGR2HSV" realiza la transformación de valores RGB a HSV.

Paso 3.- Establecer rango de valores de los matices para los colores deseados: Usando como referencia la Fig. 5, el rango de valores del matiz (H) va de 0 a 179, el rango de saturación (S) va de 0 a 255 y el brillo (V) se mantiene constante en 255, sin embargo, este último parámetro también puede variarse de 0 a 255, considerando que los valores cercanos a 0 oscurecen el matiz cercano a negro. Para implementar la selección de los valores HSV en Python se debe crear un array con numpy, donde se especifiquen como argumentos separados los valores H, S y V, seguido de `np. uint8`, por lo que la instrucción base para la detección de cada color es `ColorBajo=np.array([Hcb, Scb, Vcb], np. uint8)` y `ColorAlto=np.array([Hca, Sca, Vca], np. uint8)`, donde una instrucción limita el valor mínimo del color y la otra el valor máximo.

Particularizando a este caso, los rangos de valores H para cada uno de los colores empleados en el etiquetado de agentes RPBI son: Rojo de 0 a 10 para el primer rango mostrado en la Fig. 6 y de 170 a 179 para el segundo rango de valores. Amarillo de 25 a 35 en el rango principal. Verde, este es el color de etiquetado neutro, se trabaja en valores de 50 a 70. Así mismo, los parámetros de saturación y brillo se ajustan para las condiciones de iluminación existentes en el área de trabajo, quedando estos en los límites, S de 100 a 255 y V de 20 a 255.



**Tabla 1.** Parámetros DH del robot manipulador diseñado.

Articulación	$\theta$	$d$	$a$	$\alpha$
1	$\theta_1$	$L_1 = 3cm$	0	$90^\circ$
2	$\theta_2$	0	$L_2 = 12 cm$	0
3	$\theta_3$	0	$L_3 = 12 cm$	0

Paso 4.- Visualización: La visualización de la imagen puede ser presentada de diversas maneras, esto debido a que la selección por color emplea una binarización de la imagen por matiz, por lo tanto, puede ser visualizada como imagen blanco y negro o en términos del color de selección. En este caso particular, se prefiere visualizar el color real de selección por lo que se emplea la instrucción `maskColorvis = cv2.bitwise_and (frame, frame, mask= maskColor)`, donde a la imagen se le aplica la máscara de color seleccionada. Una vez que se tiene la detección del color de los agentes RPBI, se requiere conocer la posición del color, es decir, del agente RPBI, para que el robot manipulador viaje a esa posición para recogerlo.

Esta actividad se realiza por medio de geometría, se identifican cuatro puntos en la imagen binarizada por color que se obtiene del proceso anterior, estos cuatro puntos corresponden a los píxeles con los valores mínimos y máximos de coordenadas (x,y) de la zona detectada con el color deseado. Utilizando estos cuatro puntos se encuentra el centro del objeto,  $(X_r, Y_r)$ , por medio de las expresiones de la ecuación 1. Esta información es referenciada al plano de la imagen, por lo que solamente generará el movimiento horizontal y vertical del robot, no el de profundidad o extensión del mismo:

$$\begin{aligned} X_r &= \frac{X_{m\acute{a}x} + X_{m\grave{i}n}}{2}, \\ Y_r &= \frac{Y_{m\acute{a}x} + Y_{m\grave{i}n}}{2}. \end{aligned} \tag{1}$$

Esta posición toma como referencia u origen la esquina superior izquierda de la imagen, lo cual es necesario referenciar posteriormente a la posición del robot manipulador.

### 3.5. Control del accionamiento de las articulaciones del robot manipulador

Para el funcionamiento del brazo robótico, se emplea el uso de su cinemática, la cual determina el movimiento del robot a partir de su geometría, este tema es ampliamente abordado en literatura de robótica [1, 8, 11, 12], por lo que, solamente se bosqueja la solución implementada en particular para este robot.

Empleando el algoritmo Denavit-Hartenberg (DH) se obtiene la cinemática directa del robot esto se reduce a encontrar una matriz homogénea de transformación T que relacione la posición y orientación del extremo del robot respecto del sistema de referencia fijo situado en la base del mismo, la ecuación 2 representa la matriz T de transformación, la cual puede ser descrita como el producto de matrices A referentes a las características físicas de cada eslabón:

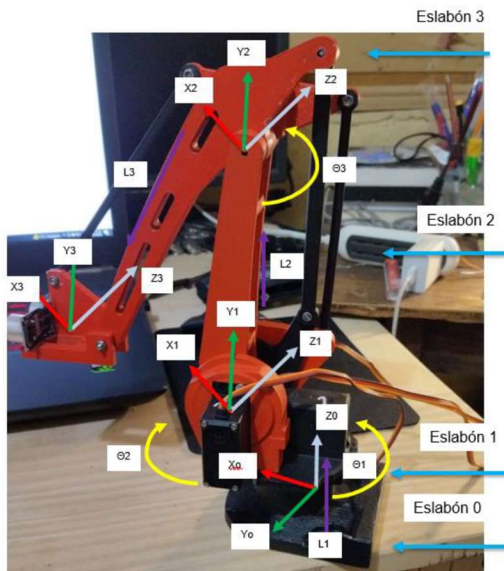


Fig. 6.- Representación de los etiquetados de Denavit-Hartenberg.

$$T = \begin{bmatrix} Rotación & Traslación \\ Perspectiva & Escala \end{bmatrix} = \begin{bmatrix} R_{3x3} & P_{3x1} \\ f_{1x3} & w_{1x1} \end{bmatrix} = \begin{bmatrix} n_x & o_x & q_x & p_x \\ n_y & o_y & q_y & p_y \\ n_z & o_z & q_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

Con base en el algoritmo DH, se determinan las matrices A con los parámetros DH  $a_i, \alpha_i, d_i$  y  $\theta_i$  inherentes al robot manipulador propuesto. La ecuación 3 muestra la definición de las matrices A en términos de los parámetros DH, mientras que la ecuación 4, muestra la relación entre el producto de las matrices A y la matriz de transformación general. La Tabla 1 muestra los parámetros DH para cada eje de referencia en nuestro robot como se encuentran etiquetados en la Fig. 6:

$$A_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

$$T = A_1^0 A_2^1 A_3^2. \quad (4)$$

Definiendo todas las matrices A (ecuación 3) en términos de los parámetros DH, y conociendo la localización destino del extremo final del robot manipulador, que será la posición otorgada por el algoritmo de la detección del color, en términos de n, o, q y p de la matriz T (ecuación 2), se pueden manipular las 12 ecuaciones resultantes de la igualación de las ecuaciones 1 y 2, llamada cinemática inversa, para determinar el valor de rotación de cada articulación para lograr el posicionamiento deseado del robot, particularmente se utilizan las ecuaciones igualadas a cero o a la coordenada final deseada representada por  $(p_x, p_y, p_z)$ . Este proceso se realiza numéricamente mediante la ayuda del software MATLAB basándose en lo implementado en [1, 11, 12].

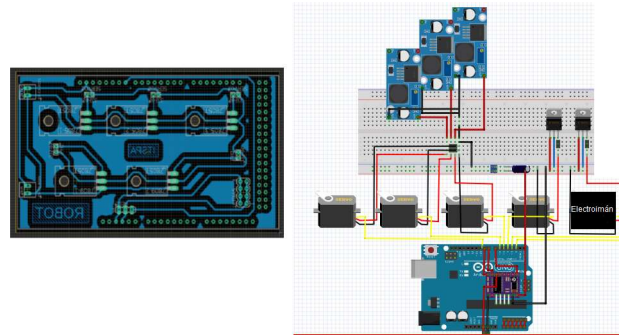


Fig. 7.- Circuito y diagrama electrónico implementado.



Fig. 8. Prototipo final.

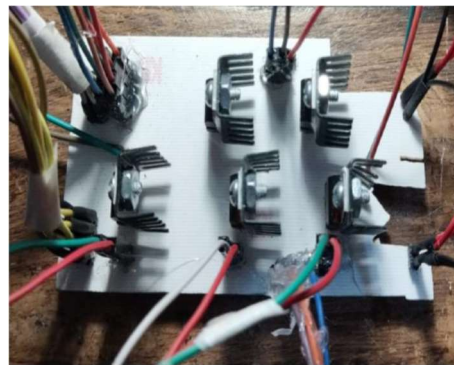
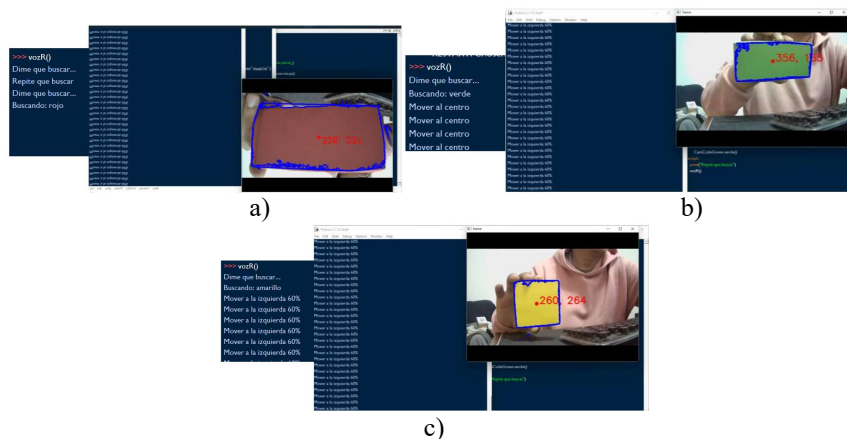


Fig. 9. Circuito electrónico.

La cinemática inversa se implementa dentro de la programación física de la plataforma Arduino, esto completa el ciclo de recolección de información, desde que la imagen es tomada y procesada en Python, el dato de posición del agente RPBI se manda por comunicación serial a Arduino implementando, por medio de las ecuaciones de cinemática inversa, el cálculo de los ángulos de rotación de cada articulación, y sea el mismo Arduino que controle los servomecanismos. La cinemática inversa permite conocer la localización cartesiana del robot y así determinar los ángulos que el robot debe girar, desde su posición actual, para que su elemento final se coloque en la coordenada deseada.

### 3.6. Implementación física de los componentes electrónicos

En este punto, se tienen todos los elementos necesarios para la detección del color, la programación para el movimiento del robot y la implementación física de los elementos del robot. Por lo tanto, solamente falta implementar los actuadores que le dan movimiento físico al robot.



**Fig. 10.** Pruebas de detección de color. a) Detección de Rojo, b) - Detección de Verde, c) Detección de Amarillo.

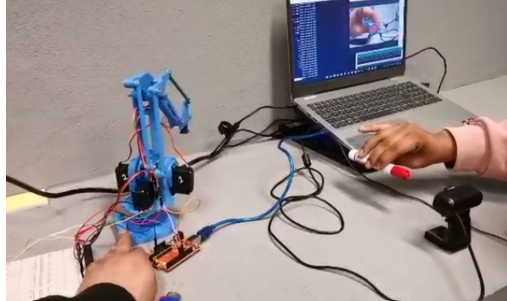
En este caso, por su costo se implementan servomotores TowerPro MG995 que son económicos, pero con alto torque de trabajo de hasta 11Kg-cm, estos para el movimiento del robot, como elemento final, se implementa un electroimán de bajo consumo de energía, todo ello controlado y programado a través de la tarjeta de desarrollo Arduino. Debido a que se trabaja con la tarjeta de desarrollo Arduino, la corriente que este otorga no es suficientes para operar todos los dispositivos electrónicos que requiere el robot. Para ello se diseñó un circuito impreso (PCB por sus siglas en inglés Printed Circuit Board), mostrado en la Fig. 7, el cual es alimentado con 9V y permite alimentar a la tarjeta Arduino, a los 4 servomotores implementados (uno para las articulaciones 1 y 3, dos para la articulación 2 del robot para otorgar mayor fuerza al extender el robot al máximo) y al electroimán implementado.

#### 4. Resultados

El diseño del robot manipulador se ensambló y se verifico antes de la impresión 3D, mediante simulación se encontró que no se presentaba ninguna interferencia en el mecanismo y que se podía mover libremente.

El resultado de la impresión mostró un prototipo viable para aplicarlo como demostración del funcionamiento de la clasificación por color. La fig. 8 muestra el prototipo final implementado. La escala que se emplea para el desarrollo del prototipo es la adecuada para simular el funcionamiento real de este, ya que trabajarlo como prototipo es más viable que hacerlo en escala real, para la validación de la estrategia. En la Fig. 9 se muestra la implementación de los elementos electrónicos necesarios para la alimentación de los servomotores empleados. Una vez finalizado el ensamblaje del robot, se procedió a programar cada uno de los motores que conforman al robot.

La programación se hizo con el software Arduino. Al momento de estar programando el robot se fueron realizando pruebas individuales para ir verificando que cada uno de los componentes del robot estuviera realizando de manera correcta la función con la que debería cumplir.



**Fig. 11.-** Pruebas de detección de color con el robot manipulador.

Con respecto a la identificación de los colores, las pruebas se realizaron con los comandos de voz, en este trabajo se emplearon solamente 5 comandos de voz, los cuales son:

- Inicio: Permite energizar todo el robot manipulador y activarlo para que realiza alguna actividad. Activa la cámara para que realice la adquisición del espacio de trabajo alrededor del robot,
- Detener: Este comando apaga el robot manipulador, dejando activo solamente los recursos mínimos para poder activarse en el momento deseado,
- Rojo, Verde y Amarillo: Son los comandos que permiten activar la detección de los colores empleados para la clasificación.

En la Fig. 10 a) se muestra el reconocimiento del comando “Rojo” y la detección del color realizada, con la cámara. En esta misma se sobrepone la coordenada central del contorno de color detectado. En las Fig. 10 b) y c), muestran lo mismo para los comandos “Verde” y “Amarillo”, respectivamente. La Fig. 11 muestra evidencia de la implementación del robot haciendo pruebas con el reconocimiento de color.

## **5. Conclusiones y trabajo a futuro**

La programación de robot manipuladores no es una tarea sencilla, ya que se deben tener en cuenta muchos aspectos tanto mecánicos, como de programación. Sin embargo, el prototipo construido se desempeña de forma correcta, realizando las actividades de detección de color, movimiento a la posición de detección, toma de objeto y movimiento a la posición final de almacenaje. La propuesta, al generar un manejo a distancia y automatizado, disminuye significativa la interacción del usuario con la manipulación de los objetos, ya que el usuario funge como observador y no como manipulador de los objetos.

El trabajo presenta un prototipo a escala, por lo que actualmente cuenta con limitaciones tecnológicas para su implementación, sin embargo, como trabajo futuro se plantea la implementación de un robot manipulador de mayor capacidad para el manejo de objetos y su clasificación.

## Referencias

1. Sandler, B. Z.: Robotics: designing the mechanisms for automated machinery. 2nd Ed. Academic Press, (1999)
2. Morelos, R., Ramírez, M., Sánchez, G., Chavarín, C., Meléndez, E.: El trabajador de la salud y el riesgo de enfermedades infecciosas adquiridas. Revista de la Facultad de Medicina de la UNAM, vol. 57, no. 4, pp. 34–42 (2014)
3. Instituto Nacional de Ecología y Cambio Climático: Panorama de la generación y manejo de residuos sólidos y médicos durante la emergencia sanitaria por COVID-19. (2020)
4. Norma Oficial Mexicana NOM-087-ECOL-SSA1-2002. Obtenida del sitio <http://www.salud.gob.mx/unidades/cdi/nom/087ecolssa.html> (2003)
5. Tecno SA de CV. Obtenido en línea del enlace <http://manipulacion-ergonomica.tecnoco.com.mx/equipo/robot-paletizador-de-botellas/>
6. Kuka Robotics México. Obtenido en línea del enlace <https://www.kuka.com/es-mx/productos-servicios/sistemas-de-robot/robot-industrial/kr-4-agilus>
7. ABB Robotics. Obtenido en línea del enlace <https://new.abb.com/products/robotics/es/robots-industriales/irb-1100>
8. Jordán Gamito, A.: Diseño y realización de un sistema automatizado de clasificación por colores basado en brazo robótico y el TMS320F28335. TFG de Ingeniería en Tecnologías Industriales, Departamento de Ingeniería Electrónica Escuela Técnica Superior de Ingeniería Universidad de Sevilla, Sevilla, (2020)
9. Astete Rupay, J., Gallo Torres, D., Torres Hinostroza, A., Peña Romero, A., Sarmiento Cornejo, M.: Sistema teleoperado aplicado a la manipulación de residuos hospitalarios con riesgo biológico. Revista Ingenium, vol. 1, no. 2 (2016) ISSN en línea 2519-1403
10. Rosas-Arias, L., Vallejo-Meraz, J., Pérez-Bailón, W., Rojas-Cid, J. D.: Robot clasificador de objetos de color utilizando técnicas de filtrado RGB. Revista de Prototipos Tecnológicos, vol.3 no.10, pp. 50–59 (2017) ISSN-2444-4995
11. HengHui, S., MinZhou, L., Wei, L., Xiang, D.: Virtual link-lengths approximating method for forward kinematics of multiple motion coupled linkage mechanism. In: International Conference on Electronic & Mechanical Engineering and Information Technology, pp. 2318–2323 (2011)
12. Hendarto, H. Munadi, A., Setiawan, J. D.: ANFIS application for calculating inverse kinematics of programmable universal machine for assembly (PUMA) robot. In: The 1<sup>st</sup> International Conference on Information Technology, Computer, and Electrical Engineering, pp. 35–40 (2014)