

# Robot Seguidor de Color Utilizando Matlab y el Microcontrolador PIC18F2550

Color Tracking Robot Using Matlab and PIC18F2550 Microcontroller

Richard Junior Manuel Godínez Tello<sup>1</sup>, Pablo César Córdova Morales<sup>2</sup>, Flavio Carrillo Gómero<sup>3</sup>

Facultad de Ingeniería Electrónica, Universidad Nacional Mayor de San Marcos, Lima, Perú

**Resumen**— El presente artículo muestra el desarrollo y experimento del uso de técnicas de tratamiento de imágenes aplicado a un robot para el seguimiento de un objeto mediante la identificación del color, utilizando el microcontrolador PIC18F2550 y una webcam. El modelo de robot desarrollado es capaz de identificar los colores primarios de cada pixel, captados a través de una cámara web. El primer paso en la elaboración de este proyecto fue la elección del microcontrolador PIC18F2550 debido a su versatilidad de comunicación directa vía USB con el computador. La parte central del proyecto se encuentra en el código desarrollado en Matlab, que unido con el programa del microcontrolador derivan en el accionar de los motores DC. Una vez detectados los pixeles del color preestablecidos, el robot podrá girar a la derecha o izquierda así como avanzar, retroceder o detenerse, dependiendo del movimiento y ubicación espacial del objeto en estudio y del área captada de la imagen.

**Abstract**— This article shows the development and experiment of using image processing techniques applied to robot tracking an object by color identifying, using PIC18F2550 microcontroller and webcam. The robot model developed is able to identify the primary colors of each pixel, captured through a webcam. The first step in the development of this project was the choice of the PIC18F2550 microcontroller due to its versatility of direct communication with the computer via USB. The central part of the project is developed in Matlab code which together with the microcontroller program, resulting to drive on DC motors. Once detected the preset color pixels, the robot can turn left or right and forward, rewind or stop, depending on the movement and spatial location of the object under study and the area captured the image.

**Palabras clave**— CMOS, frames, Matlab, Microcontrolador, PIC18F2550, pixel, RGB, USB.

**Key words**— CMOS, frames, Matlab, Microcontrolador, PIC18F2550, pixel, RGB, USB.

## I. INTRODUCCIÓN

La visión artificial ha cobrado gran interés por parte de los investigadores en el desarrollo de robots con el fin de dotarlos de autonomía y funciones inteligentes para la realización de tareas específicas.

El presente trabajo pretende aportar a las tecnologías de visión artificial empleando herramientas modernas y el estándar de comunicación vía USB, utilizándose para ello una webcam de tecnología CMOS (*Complementary Metal Oxide Semiconductor*) de bajo costo.

El procesamiento de imágenes en colores aborda el tema de los componentes RGB, siendo esta una combinación de 3 matrices de  $M \times N$  elementos, donde cada color de un pixel es una triple correspondencia de las componentes rojo, verde y azul de una imagen RGB en una representación espacial tal como se muestra en la Figura 1 [1].

Una imagen RGB puede ser vista como una "pila", de tres escalas, cuyo rango de valores en en cada arreglo R, G o B es de  $[0, 255]$ .

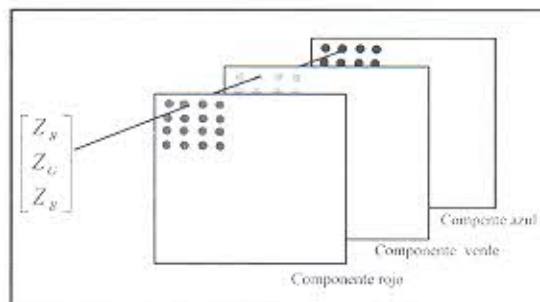


Fig. 1. Esquema de la formación de una imagen a color, mostrando los tres componentes de la imagen RGB [1].

<sup>1</sup> Richard J. M. Godínez Tello. E-mail: rgodinez85@gmail.com

<sup>2</sup> Pablo César Córdova Morales. E-mail: pccm\_77@hotmail.com

<sup>3</sup> Flavio Nireo Carrillo Gómero. E-mail: fcarrillog@unmsm.edu.pe

En la Figura 2 se aprecia el cubo de colores RGB que muestra los colores primarios y secundarios en los vértices. Los puntos a lo largo de la diagonal principal por ejemplo tienen valores en la escala de grises iniciándose en el negro y finalizando en el blanco (1,1,1).

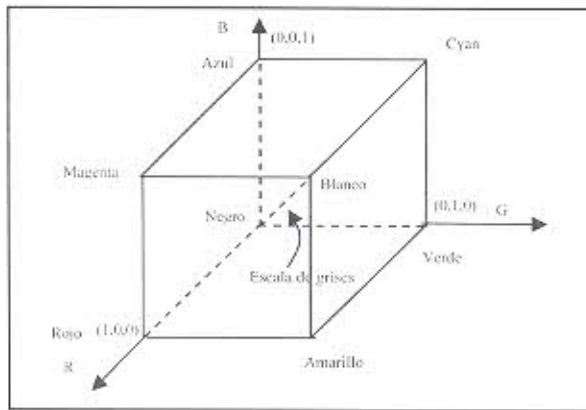


Fig. 2. Esquema del cubo de colores RGB [1].

Dado que el presente trabajo la aplicación es en tiempo real, las imágenes son *frames* captadas a través de la webcam, y utilizando Matlab se analiza la información de cada pixel, obteniendo los componentes RGB con el comando *imtool*. La Fig. 3 muestra una imagen de prueba analizada en Matlab en la que se aprecia los valores de R, G y B de cada pixel en forma cuantificada.

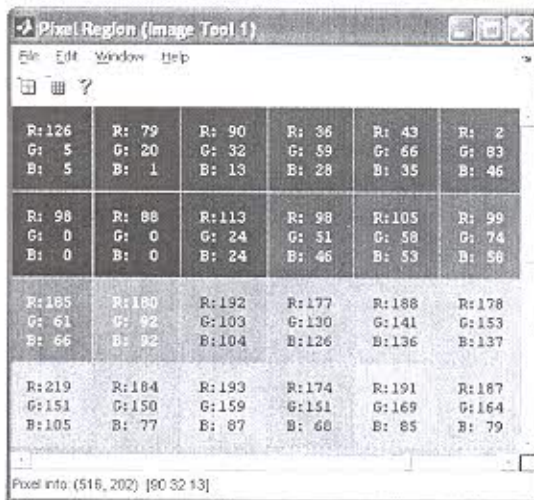


Fig. 3. Valores R, G y B de cada pixel obtenida mediante el comando *imtool*.

### II. COMUNICACIÓN PC – PIC VÍA USB

En este trabajo para las acciones de control sobre la dinámica de movimiento del robot se utilizó el microcontrolador PIC18F2550 debido a que dispone de comunicación USB, y entre sus características principales están la de tener memoria tipo FLASH,

memoria de programa de 32K, velocidad máxima de CPU de 48MHz tal que para las pruebas de laboratorio utilizamos esta velocidad con un oscilador a cristal [2].

La programación del microcontrolador se realizó en lenguaje C, compilado con el CCS Compiler basado en el protocolo USB\_ENABLE\_BULK, para el cual fue necesario el uso de los drivers de comunicación PIC-PC proporcionados por el fabricante, Microchip [3], [4].

En el proceso de operación, el computador reconocerá automáticamente el firmware programado efectuándose la comunicación entre ambos dispositivos, siendo el microcontrolador un nexo entre la cámara web y el entorno Matlab. El microcontrolador no realizará ninguna tarea hasta que no reciba datos de la PC acerca de la detección del color, tamaño y posición del objeto establecido a fin de procesar y producir las órdenes para el desplazamiento del robot.

### III. FUNCIONAMIENTO DEL SISTEMA

Teniendo en cuenta la Figura 4, donde se muestra esquemáticamente el proceso de operación del robot, en donde los *frames* captados por la cámara web, son transmitidos al computador via USB, luego cada *frame* es procesado y filtrado por el sistema construido sobre Matlab. Si el pixel de interés se encuentra en una de las tres franjas establecidas (Rojo, Verde o azul), la acción de la aplicación realizada en Matlab procesará y proporcionará dicha información al microcontrolador, el cual a su vez dispone de las directivas de movimiento para cada tipo de franja. En el sistema de filtro diseñado se han establecido como pixeles de interés para la identificación, a tres colores primarios: Rojo, Verde y Azul. Cada color tiene un rango de valores máximos y mínimos respecto a las componentes RGB, las mismas que son etiquetadas para su llamado como función mediante GUIDE de Matlab.

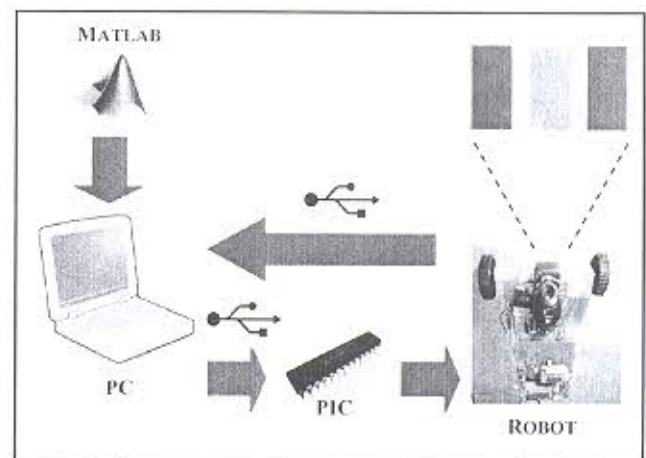


Fig. 4. Esquema de funcionamiento del sistema.



El microcontrolador recibe los datos resultado del procesado de los *frames*, que luego de tratarlo deriva en el accionamiento de los dos motores DC del robot a través del puerto B del microcontrolador, donde las funciones que realizará es la de producir instrucciones de movimiento de adelante, atrás, derecha, izquierda y pare.

A continuación se muestra el algoritmo de tratamiento de los *frames* y las reglas de movimiento para el robot:

1.<sup>er</sup> Caso:

El objeto de área  $A$  de la Figura 5(a) se encuentra frente al robot a una distancia  $d_i$ . En la imagen está registrada como área  $A'$  en píxeles.

Luego, de la Figura 5(b) el sistema evalúa:

Si  $A' > A_{ref} \rightarrow d_i$  aumenta a un valor  $d_f$ . El robot se aleja.

Si  $A' < A_{ref} \rightarrow d_i$  se reduce a un valor  $d_f$ . El robot se acerca.

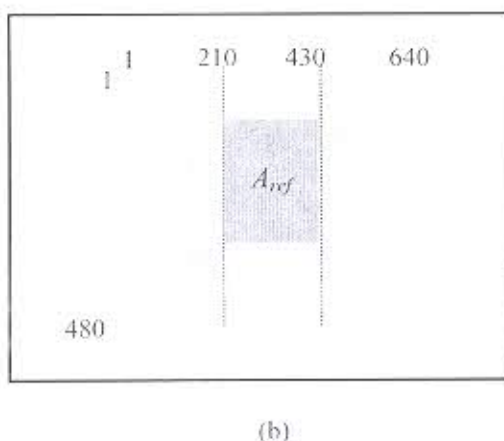
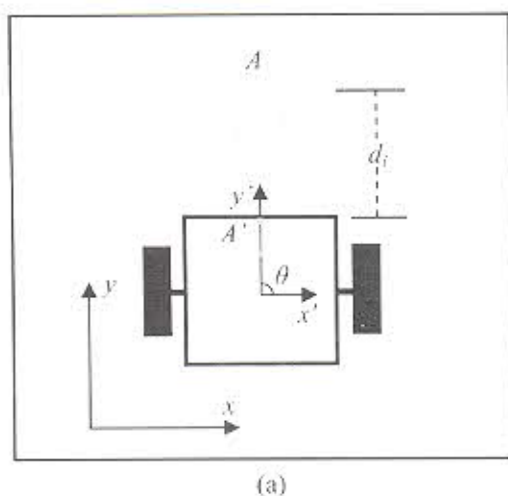


Fig. 5. Posición del robot relativo al objeto. (a) Frente al objeto a una distancia  $d_i$ . (b) Imagen resultado inicial a comparar.

2.<sup>do</sup> Caso:

El objeto de área  $A$  de la Figura 6(a) se encuentra a una distancia  $d_i$  y ángulo  $\alpha$  respecto del robot. En la imagen está registrada como área  $A'$  en píxeles.

Seguidamente, de acuerdo a la imagen resultado, mostrada en la figura 6(b), el sistema evalúa:

Si  $A' > A_{ref} \rightarrow d_i$  aumenta a un valor  $d_f$ . El robot se aleja del objeto.

Si  $A' < A_{ref} \rightarrow d_i$  se reduce a un valor  $d_f$ . El robot se acerca al objeto.

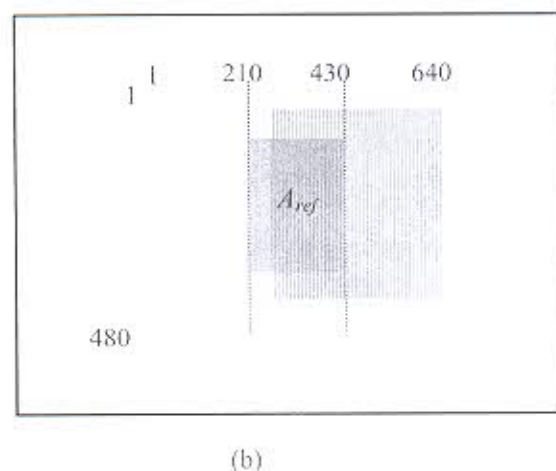
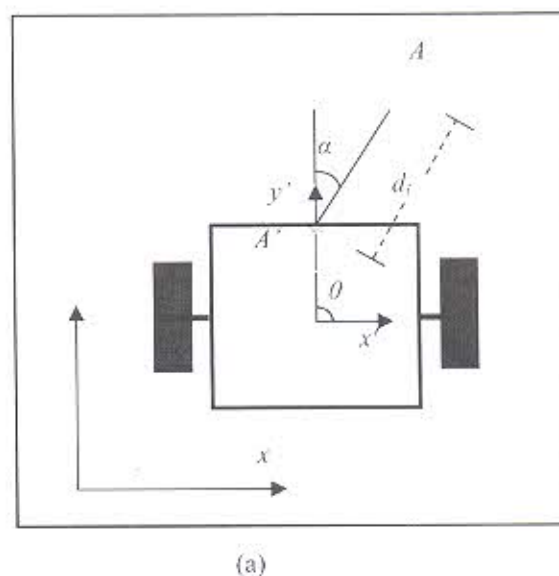
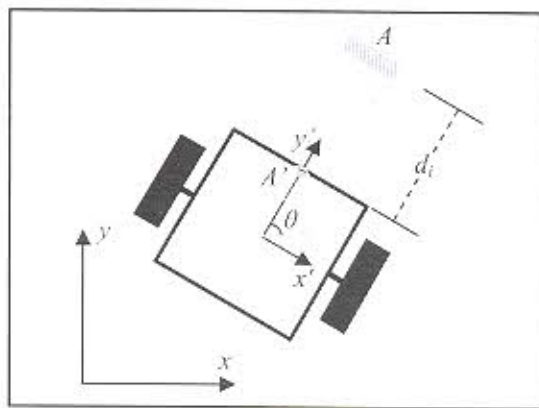


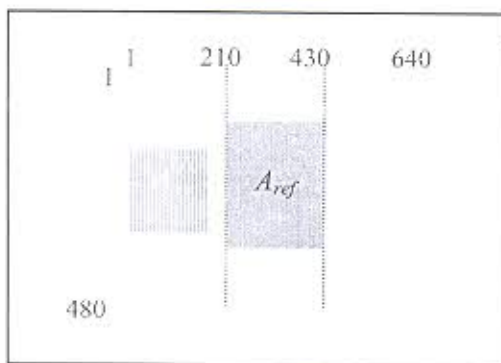
Fig. 6. Posición del robot relativo al objeto. (a) A una distancia  $d_i$  y ángulo  $\alpha$ . (b) Imagen resultado a comparar.

Seguidamente, de acuerdo a la Figura 7(a) en caso que:

Si  $A'$  está a la izquierda de  $A_{ref} \rightarrow$  El robot gira a la derecha un ángulo, tal que  $\theta = \alpha$ .



(a)



(b)

Fig. 7. Giro de un ángulo  $\alpha$  del robot. (a) Nueva posición final frente al objeto a una distancia  $d_i$ . (b) Imagen resultado inicial antes del giro.

Si  $A'$  está a la derecha de  $A_{ref} \rightarrow$  El robot gira a la izquierda un ángulo tal que  $\theta = \alpha$ .

Para el procesamiento de los *frames*, se estableció una matriz estándar de 640x480 píxeles, la misma que fue dividida en 3 franjas, tal como se muestra en la Figura 5(b).

Como consecuencia del algoritmo propuesto, se desarrolló el diagrama de flujo del proceso, mostrado en el diagrama de la Figura 8, y luego el código de las funciones del robot, utilizando Matlab:

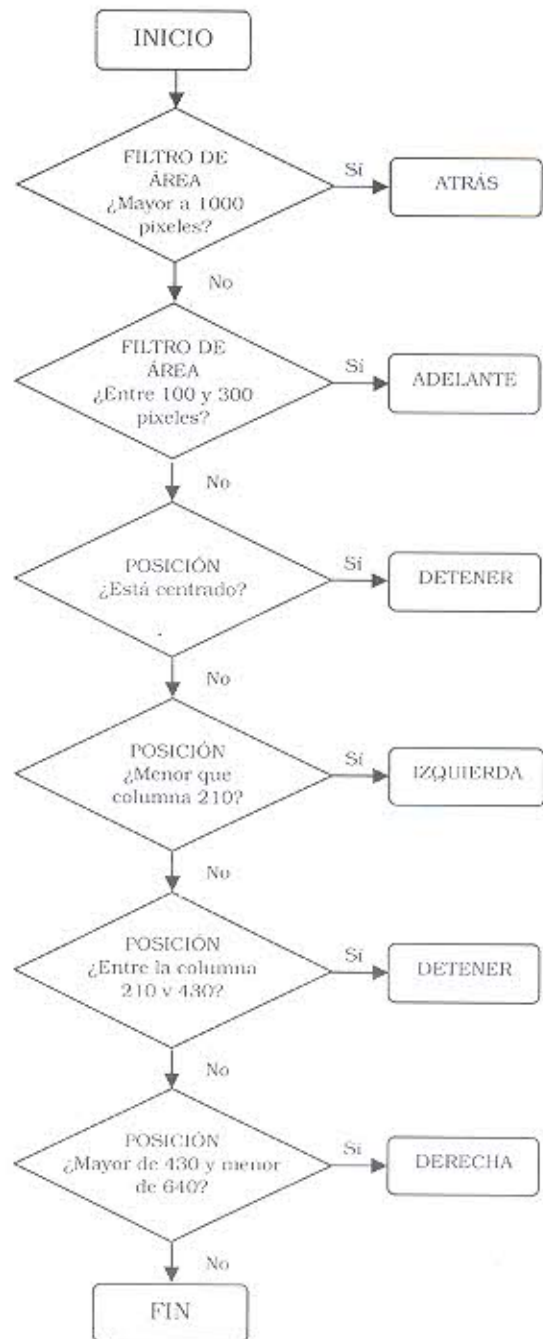


Fig 8. Diagrama de flujo del modelo del sistema.

```

if area_max>10000
    botar=6;
    set(handles.text5,'String','Atras')
else
    if area_max>100 && area_max<300
        botar=9;
        set(handles.text5,'String','Adelante')
    else
        if Centro==0
            botar=0;
            set(handles.text5,'String','Detenerse')
        else
            if Centro<210
                botar=5;
                set(handles.text5,'String','Izquierda')
            end
        end
    end
end

```

```

end
  if Centro >= 210 && Centro < 430
    botar = 0;
    set(handles.text5, 'String', 'Detenerse');
  end
  if Centro >= 430 && Centro <= 640
    botar = 10;
    set(handles.text5, 'String', 'Derecha');
  end
end
end
end
end

```

Una vez identificado el objeto de acuerdo al color, para el procesado de posición del objeto y el robot, la imagen es convertida a escala de grises, utilizando el comando *rgb2gray* y la función *regionprops* para la determinación del área de la imagen objeto.

Por conveniencia, adoptamos que el envío de las señales de la Pc al PIC sea de forma intermitente para ejercer un mejor control sobre la velocidad de los dos motores DC.

IV. DIAGRAMAS ESQUEMÁTICOS DESARROLLADOS

En la Figura 9 se presenta la imagen del robot prototipo implementado, donde se muestra una placa conteniendo el PIC18F2550, el CI L293b (driver puente H), una webcam, dos motores DC y finalmente una batería de 9 voltios

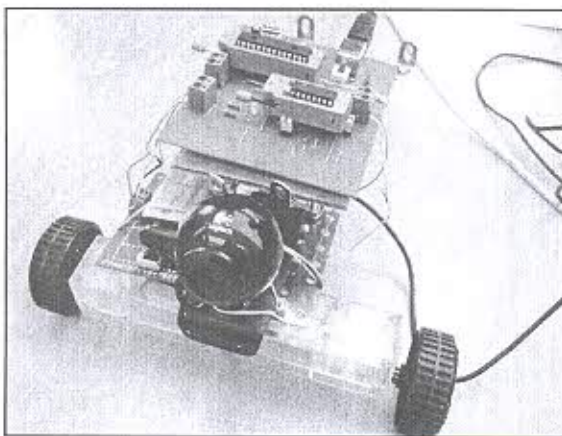
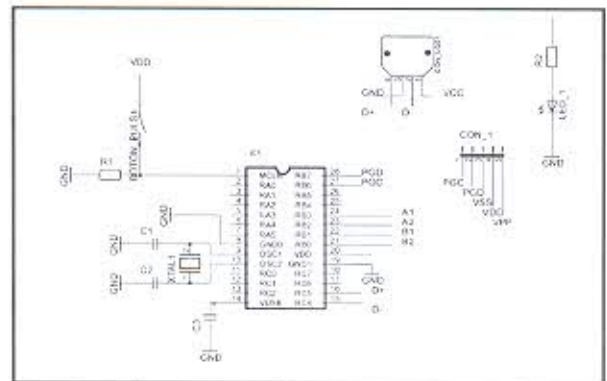


Fig. 9. Hardware del robot prototipo desarrollado.

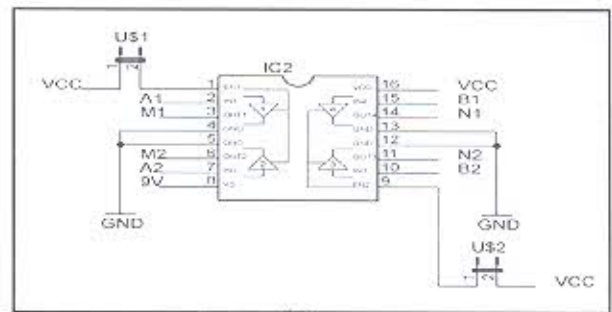
En el diseño del robot se utilizó el puerto B del microcontrolador a través del cual se realizan las conexiones hacia las entradas del driver L293b. La tensión USB de la PC alimenta únicamente al microcontrolador y al driver L293b, y la batería de 9 voltios energiza las acciones de los dos motores DC.

En la Figura 10(a), se muestra el diagrama de circuitos diseñado de la parte de control con el PIC18F2550 y las figuras 10 (b) y 10(c) muestran los diagramas de circuito de potencia administrada por el CI driver push pull de 4 canales L293b. Se eligió dicho

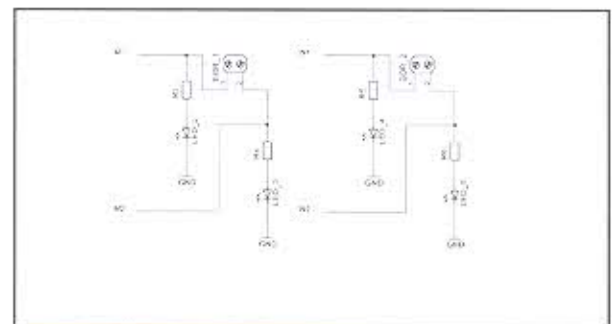
CI porque requiere una corriente máxima de 1Amp., en consideración de que hemos trabajado con dos motores pequeños DC de bajo consumo.



(a)



(b)



(c)

Fig. 10. Diagramas esquemáticos desarrollados del robot prototipo. (a) Configuración del PIC18F2550 con el puerto B como salida. (b) Configuración del driver L293b [5]. (c) Esquema de conexión de los dos motores DC con led de señalización.

V. CONCLUSIONES

El prototipo de robot desarrollado cumple con el algoritmo propuesto acerca de identificar el color del objeto preestablecido y ubicar al robot en la posición deseada, utilizando técnicas de tratamiento digital de imágenes. El factor de iluminación es una variable que se debe considerar con la finalidad de fijar con las condiciones necesarias para que el sistema defina adecuadamente el color prefijado, ya sea mejorando el



umbral del filtro o condicionando la intensidad de iluminación del medio de prueba del sistema.

Cabe señalar que este trabajo fue presentado en el Encuentro Tecnológico de Robótica PERÚ 2010, organizado por el Colegio de Ingenieros del Perú en la que mostró una interfaz desarrollada con Matlab, tal como se aprecia en la Figura 11.



Fig. 11. Interfaz visual de Matlab implementada.

Como consecuencia de este trabajo, se observó que es necesario desarrollar este sistema utilizando el

lenguaje de programación C, así como la plataforma OpenCV, que proporciona un entorno de desarrollo eficiente con programación en código C optimizado la que tendrá como consecuencia mejorar la velocidad de respuesta en tiempo real del robot.

#### REFERENCIAS

- [1] Gonzales, Rafael, Richard Woods y Steven Eddins. Digital Image Processing using Matlab. Pp. 194-241, 2003.
- [2] Datasheet de microcontrolador PIC18F2550. Microchip.  
<http://ww1.microchip.com/downloads/en/devicedoc/39632d.pdf>
- [3] Comunicación entre Matlab y PIC de Microchip usando puerto USB. Gabriel Pool. 2010  
<http://www.mathworks.com/matlabcentral/fileexchange/24417>
- [4] Microchip Technology. Tópicos de USB.  
[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1486](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1486)
- [5] Datasheet de integrado L293b. Puente H de SGS Thomson.  
[http://www.todopic.com.ar/utiles/l293b\\_driver\\_en\\_puente.pdf](http://www.todopic.com.ar/utiles/l293b_driver_en_puente.pdf)