

# CONTROL DE BRAZO ROBÓTICO DE TRES GRADOS DE LIBERTAD CON DE PROCESAMIENTO DIGITAL DE IMÁGENES PARA SEGUIMIENTO DE UN BRAZO HUMANO

*THREE DEGREE OF FREEDOM ROBOTIC ARM CONTROL WITH DIGITAL IMAGE PROCESSING FOR FOLLOW-UP A HUMAN ARM*

**Juan Sifuentes Mijares**

Tecnológico Nacional de México / Instituto Tecnológico de la Laguna, México  
*jsifuentesm@correo.itlalaguna.edu.mx*

**Luis Jesús Flores Fraire**

Tecnológico Nacional de México / Instituto Tecnológico de la Laguna, México  
*dbitp-metal@hotmail.com*

**Rosy Edith Castañeda De La Torre**

Tecnológico Nacional de México / Instituto Tecnológico de la Laguna, México  
*rosyedith@live.com.mx*

**Alejandro Romero Juárez**

Tecnológico Nacional de México / Instituto Tecnológico de la Laguna, México  
*peto.08090@gmail.com*

**Recepción:** 21/octubre/2019

**Aceptación:** 23/noviembre/2019

## Resumen

En este documento, se describe el desarrollo de un robot planar de tres grados de libertad, para el seguimiento de un brazo real humano mediante Procesamiento Digital de Imágenes PDI, usando una cámara. Mediante un algoritmo que resuelve el modelo cinemático inverso del robot y usando la imagen capturada, se obtienen los ángulos existentes; entre los centros de tres aéreas del brazo humano. Los ángulos son escalados y enviados a una placa arduino. Un control, tipo proporcional, utiliza los errores entre los ángulos del brazo y el robot para mover tres servomotores; los cuales están unidos a los eslabones del robot. Este sistema se realizó con el fin de tener un robot imitador de bajo costo con fines educativos y de experimentación. Como resultado, se obtuvo un brazo robótico que hace seguimiento o imita con sus tres eslabones a dedos, palma y antebrazo del usuario.

**Palabras Claves:** Brazo robótico, control, PDI.

## **Abstract**

*In this document is described the development of a project that consists of a planar robot of three degrees of freedom for the follow-up of a real human arm by means of Digital Image Processing DIP using a camera. Using an algorithm that solves the inverse kinematic model of the robot and using the captured image, the angles between the geometric centers of the human arm are obtained. These angles are scaled and sent to an Arduino board. A type of proportional control uses the angle errors between the arm and the robot to move three servo motors which are attached to the links of the robot. This system was made in order to have a low-cost mimic robot for educational and experimentation purposes. A robotic arm that follow-up or mimics with its three links to the user's fingers, palm and forearm was obtained as a result.*

**Keywords:** Control, robotic arm, DIP.

## **1. Introducción**

Un brazo robótico o robot manipulador; consiste en una estructura cuyos eslabones se asemejan a un brazo humano, debido a su acomodo. Dicho sistema cumple con diversas funciones, ya sea para la manipulación de objetos mediante su órgano terminal, o en la modificación de objetos mediante la ayuda de este. En el presente documento; se plantea el caso donde el brazo robótico intenta imitar las posiciones que toman 3 secciones de un brazo humano. Las aéreas del brazo humano que se imitan son: el antebrazo, la sección palmar y la sección dactilar. Estas aéreas son representadas posicionalmente por cada uno por de los 3 eslabones que conforman el robot manipulador.

Un trabajo similar, al que se presenta en este documento, es el trabajo titulado "An experimental testbed for visual servo control of robotic manipulators" presentado en el 2003, en EFTA IEEE Conference on Emerging Technologies and Factory Automation por P.J. Sequeira G., J.R.C. Pinto. En ese trabajo se presentó un banco de pruebas experimentales, para el servo-control visual de manipuladores robóticos;

en este la parte de realimentación visual era de 50hz y se podían probar diversos algoritmos clásicos de control [Sequeira, 2003]. Otro trabajo interesante, es el titulado “A vision-based user interface of a mobile robotic arm for people with severe disabilities” presentado por Gao F., Higa H., H., Soken T. en el International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS) en el 2015. En ese trabajo se presentó un sistema colaborativo humano – robot, basado en el seguimiento ocular, para una persona con discapacidad, usando una cámara, la cual mediante el movimiento de sus ojos podía acceder a diferentes comandos del robot [Gao, 2015]. En [Cao, 1998] presentan un entorno multirobot interactivo con un sistema de visión un mundo integrado real y simulado llamado simulación parcial. Y diversos trabajos de visión, sobre todo en robótica móvil, han sido presentados recientemente en diferentes espacios científicos [Jiang, 2018],[Huiyao, 2012], estos trabajos justifican el esfuerzo realizado, siguiendo esas líneas de investigación.

En el actual trabajo presentado, se recurrió al uso de sensores de imagen que se encargan de detectar la posición del objeto de interés y así lograr determinar la posición actual del brazo. En la parte del control del sistema, para nuestro caso en particular, se dejó fuera cualquier sensor del tipo electro-mecánico para realimentación de posición del brazo humano, debido a que se buscó hacer uso únicamente de una cámara web, como realimentación para obtener un error de posición. La cámara, mediante la toma de de fotos en serie, se utiliza para detectar la posición actual del brazo del usuario. Para la obtención de los puntos a posicionar, para el brazo robótico, se utiliza el procesamiento digital en la imagen para obtener los centroides de las áreas de interés en cada imagen. Después de la obtención de los puntos coordenados de los centroides, se recurre al uso del modelo cinemático inverso, para que, mediante las posiciones de dichos puntos, poder obtener así los ángulos con los cuales se debe posicionar a cada eslabón para imitar el brazo del usuario [Craig, 2006], [Anibal, 2001]. Los ángulos obtenidos son enviados a un microcontrolador, el cual se encarga de leerlos para así mandar como salida, una señal del tipo PWM que se encargará de posicionar a los servomotores en el ángulo adecuado, como se muestra en el diagrama de la figura 1.

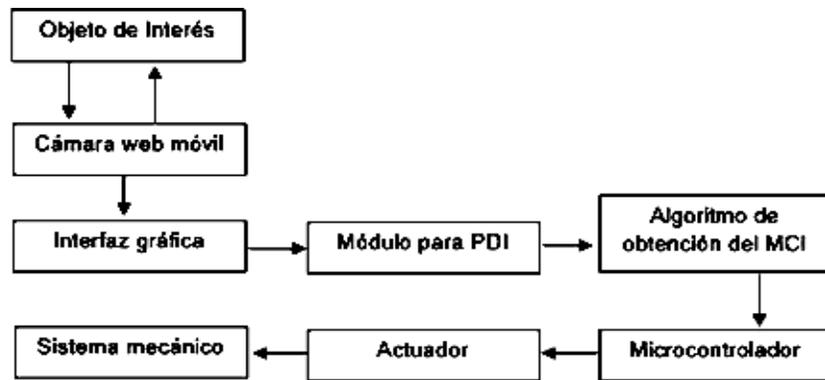


Figura 1 Diagrama del sistema.

El problema principal a resolver es: el control mediante el uso de una cámara empleando el PDI, y que en cada imagen sea posible obtener la información necesaria y suficiente del objeto de interés, sin ser necesario el uso de otro tipo de sensores que permitan suponer o determinar la posición del o los objetos en cuestión. Se espera más adelante, a futuro, poder probar otros métodos de control, tales como: el uso de técnicas de control avanzado, las cuales se ha demostrado que pueden llegar a sustituir a los controladores clásicos en distintas tareas, con resultados más que aceptables, como lo es el caso del control mediante lógica difusa, control híbrido y otros [Sifuentes, 2014], [Ogata, 2001], [Barrientos, 2007].

## 2. Métodos

A continuación, se realiza una descripción del método utilizado para la obtención de resultados, así como los materiales empleados para llevar a cabo el proyecto. También se realiza una descripción de los medios que se usaron en este trabajo.

### Conexión de los Elementos

El sistema se puede dividir en 3 bloques, los cuales son: la cámara, la computadora y la placa de desarrollo, la cual está conectada con los servomotores del brazo imitador. Dichas conexiones se detallan en la figura 2. La comunicación entre dichos elementos se realiza vía el puerto USB - serie. Para esto, es necesario configurar la comunicación serial en cada uno de los elementos con las mismas características y establecer los enlaces. La computadora se encargará de enviar las

señales de control, tanto para la cámara, como para el Arduino y la única información que recibe es la de la imagen captada.

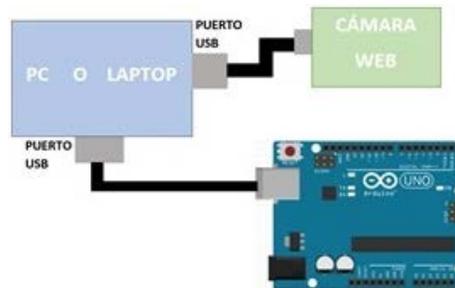


Figura 2 Sistema de conexión entre la placa, la cámara y la computadora.

La computadora se encargará de alimentar a la placa de desarrollo, junto con la cámara. Esta envía la información necesaria para el funcionamiento de todos los elementos. Para el enlace tanto de la cámara como de la placa con la computadora, es necesario contar con un software de programación que cuente con el adecuado soporte para cada uno de dichos elementos. Ya sea mediante el uso de diversos plug-in o extensiones. En este caso se decidió usar el software MATLAB debido a su facilidad de uso, además de contar con soporte para la cámara utilizada y para la placa Arduino, facilitando así la comunicación. La conexión entre la placa utilizada y los servomotores se detalla en la figura 3, y como se puede observar, la placa Arduino solo enviará las señales para los pulsos modulados en ancho de control de los servomotores, y los servomotores son alimentados por una fuente independiente externa para ayudar con la corriente, la cual se encuentra cerca de 1 ampere, demandada por los elementos.

### **Cámara Web y Adquisición de Fotografías**

Para facilitar la labor de binarización de la imagen tomada por la cámara, se recurre al uso de un fondo de color negro. Así como una serie de cintillas negras de velcro, que se posicionarán como se muestra en la figura 4. Esto hará que la cámara perciba mejor los contrastes de color, entre el fondo y el brazo del usuario, permitiendo así realizar una binarización de la imagen tomada, de una manera más eficiente.

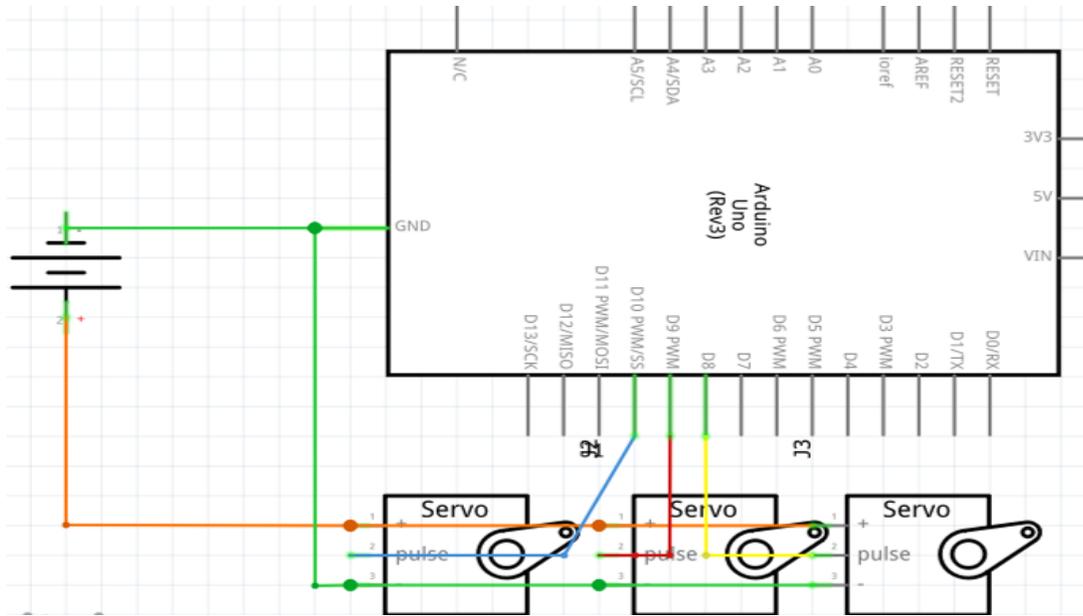


Figura 3 Sistema de conexión entre la placa Arduino UNO y los servomotores.



Figura 4 Brazo con las cintillas colocadas frente al fondo negro.

La cámara utilizada, en la captura de imágenes, es una "Microsoft lifecam vx 800". Esta cámara cuenta con una resolución máxima de 640x480 píxeles, siendo utilizada en este caso la resolución señalada a una velocidad aproximada de 24 fotogramas por segundo. Esto debido a que para la realización del proyecto no es un objetivo el uso de una alta resolución, así como una alta velocidad, ya que en caso contrario se complicaría la labor de procesamiento de la imagen tomada [Saha, 2010], [Mario, 2007].

## Interfaz Gráfica de Usuario

Para la realización del módulo para el PDI, fue necesario primero hacer la configuración para su uso en el software que se desea utilizar. En este caso se empleó el software Matlab para dicho fin; debido a su facilidad para conectarlo vía puerto serie con Arduino, y así también, como la facilidad para implementar sistemas de control rápidamente. Además, dicho software también permitió generar las aplicaciones e interfaces gráficas para control de distintos procesos, mediante la programación de eventos, en los objetos de la interfaz gráfica.

Después de establecer las comunicaciones correctamente entre la cámara y la placa Arduino con el software Matlab, se da inicio a la secuencia de programa para la creación de una interfaz gráfica de usuario, o GUI por sus siglas en inglés.

La GUI creada para este proyecto es mostrada en la figura 5, en esta se pueden observar del lado derecho 3 botones rotativos. Estos botones rotativos permiten el control individual de las articulaciones del robot, cuando se encuentra en modo manual; fijando el ángulo para referencia de cualquiera de los 3 servomotores. Del lado izquierdo se tienen otros 3 botones para las siguientes acciones:

- Botón modo: este permite seleccionar entre el modo de captura de fotografías de manera automática y el movimiento automático de los servomotores; y el modo de captura manual, mediante el cual se permite mover los servomotores con ayuda de los 3 botones rotativos.
- Botón Arduino: Enlaza imagen y software con el microcontrolador.
- Botón de Salir: cierra la interfaz.

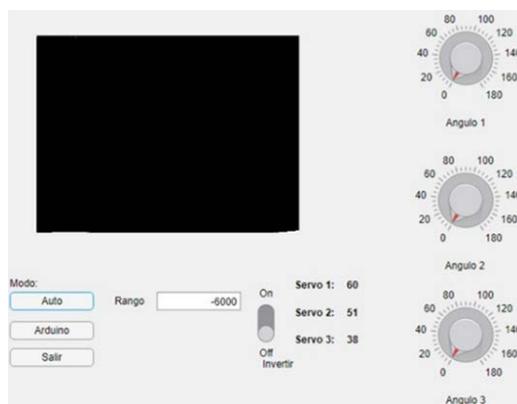


Figura 5 Interfaz gráfica de usuario generada.

Al lado derecho a estos 3 botones, se encuentra una casilla con el señalamiento: “Rango”. Dicha casilla permite introducir un valor, con el cual se puede delimitar el tamaño de los objetos, que puede o no detectar el software. Y así permitir al usuario indicar las dimensiones mínimas, para que un objeto capturado en la imagen, sea considerado como tal y no como un defecto en la captura.

Al lado de la casilla mencionada anteriormente, se encuentra un interruptor de 2 posiciones con la señalización “Invertir”. Este interruptor se encarga de controlar la orientación del robot, permitiendo al usuario cambiar los ángulos con los cuales se mueve el robot, como si fuera una relación tipo espejo. Este botón se utiliza con el fin de evitar la necesidad de girar el manipulador en caso de que se encuentre orientado del lado incorrecto a la cara frontal, ya que en tal caso se observaría al brazo realizando movimientos con ángulos iguales al complemento del ángulo real. Aun lado del interruptor selector, se observan 3 etiquetas con las palabras “servo 1:”, “servo 2:” y “servo 3:”. Estas etiquetas muestran al usuario los ángulos calculados por el algoritmo del modelo cinemático inverso para cada uno de los servomotores. Esto con el fin de mostrar el correcto funcionamiento de cada uno de los servomotores.

Por último, se observa un cuadro color negro en la esquina superior izquierda de la pantalla, este cuadro muestra las imágenes captadas por el software después de pasar por el módulo para el PDI.

Para mostrar al espectador y al usuario de la interfaz diversos datos tales como:

- Las imágenes tomadas por la cámara ya binarizadas.
- Los objetos detectados por el PDI con sus centroides marcados.

## **EI PDI**

EL procesamiento digital de imágenes “PDI”, en este caso consta de diversas etapas u operaciones, necesarias para lograr el correcto funcionamiento, ver el diagrama mostrado anteriormente en la figura 1.

Primero se obtiene la imagen tomada por la cámara. Después se guarda en una matriz con las dimensiones configuradas en la cámara, en este caso es una matriz de 640 por 480 con tres capas, haciéndola de 3 dimensiones.

Las 3 dimensiones, son debidas a que la cámara utiliza una dimensión para el color rojo, otra para el verde y otra para el azul. La matriz de 3 dimensiones es llamada matriz "RGB". En estas matrices, se almacenan los valores para cada uno de los 3 colores de cada pixel de la imagen. Si se combinan los valores de estas matrices con sus respectivos colores, se obtendrá el tono de color real captado por cada pixel. Por motivos de optimización del programa, solo se utiliza una dimensión de dichas matrices.

De los valores de la matriz, podemos decir, que un valor de 0 corresponde a la completa ausencia de ese color, para un valor más grande en la matriz corresponde a una tonalidad más grande de dicho color. Para fines prácticos, se realiza una binarización de la imagen, para manejar así solo 2 posibles valores en la matriz, los cuales serán uno y cero. De estos valores, el uno será el color blanco mientras que el cero será el color negro. Este proceso facilita en gran medida la metodología a realizarse para los puntos siguientes.

Tras realizar, de manera efectiva, la binarización, se procede a realizar la detección de las figuras en la imagen. Esto se logra mediante la generación de barridos a través de la imagen, los cuales detectan los cambios de color en la imagen. Estos barridos ayudan en 2 objetivos claves para el programa:

- El primero de estos objetivos; es la obtención de los centros de masas de las figuras en blanco. Los centros de masas de las figuras: son los puntos en los cuales puede suponerse que son las posiciones equivalentes a la masa total concentrada para cada objeto. Estos permiten al usuario obtener de manera precisa, las distancias promedio entre cada una de las figuras, mediante una simple operación aritmética entre dichos puntos.
- El segundo de estos objetivos; es obtener el posicionamiento relativo entre los centros de masas, conforme son encontrados por el programa. El programa realiza un barrido de abajo hacia arriba de la matriz, con el fin de obtener un marco o punto de referencia. Se toma como punto de referencia el primer centro de masa capturado. Y después de este se distribuyen los centros de masas, de donde serán tomadas sus posiciones y ángulos, para representar a cada uno de los eslabones del robot.

Al obtener datos mediante el procesamiento de imágenes capturadas, se procede a la sustitución de variables para la realización del modelo cinemático Inverso.

### Algoritmo para el Modelo Cinemático Inverso

Tras la obtención y acomodo de los centros de masas de las figuras encontradas, se procede a hacer uso del modelo cinemático inverso para la obtención de los ángulos. Con estos ángulos se moverán cada uno de los servomotores, con el fin de realizar el seguimiento o imitación del brazo humano.

En nuestro caso, en la aplicación del modelo cinemático inverso, se hace uso del algebra para encontrar una relación de distancia entre 2 puntos o más, y así poder obtener el ángulo o ángulos entre dichos puntos. En este modelo solo se resuelve el caso entre 2 puntos, con el fin de evadir ciertos grados de incertidumbre que pueden llegar a presentarse en el uso de dicho modelo para más de 2 figuras en movimiento. La metodología usada para esto es la siguiente:

El software selecciona, de entre el vector con los centroides detectados, el que encontró en la primera posición de abajo hacia arriba, y toma el centroide que le sigue en segunda posición de abajo hacia arriba. Después, se realiza una resta aritmética vectorial de las posiciones, en donde se encontraron dichos puntos, logrando así, obtener la distancia entre dichos puntos. Con estos datos y el teorema de Pitágoras, el cual nos permite utilizar los valores encontrados para formar un triángulo como el que se muestra en la figura 6.

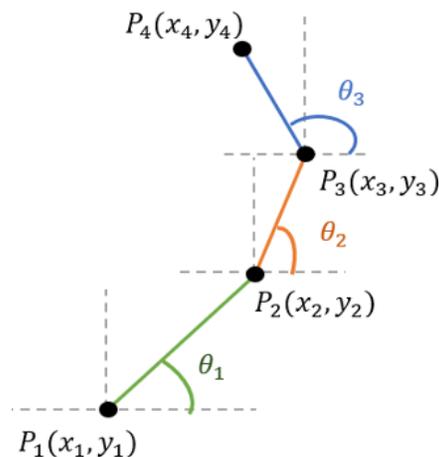


Figura 6 Modelo cinemático inverso.

Se obtiene la distancia en x, que está dada por ecuación 1.

$$x = x_2 - x_1 \quad (1)$$

Así también, la distancia en y se encuentra dada por ecuación 2.

$$y = y_2 - y_1 \quad (2)$$

En cambio, la hipotenusa del triángulo formado, o la distancia entre los 2 centros de masas se encuentra dada por ecuación 3.

$$h = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

Es posible entonces despejar la función de ángulo, mediante las distintas funciones trigonométricas dadas por el teorema de Pitágoras, para triángulos rectángulos, ecuaciones 4, 5 y 6.

$$\text{seno}(\theta) = \frac{\text{cateto opuesto al ángulo}}{\text{hipotenusa}} \quad (4)$$

$$\text{coseno}(\theta) = \frac{\text{cateto adyacente al ángulo}}{\text{hipotenusa}} \quad (5)$$

$$\text{tangente}(\theta) = \frac{\text{cateto opuesto al ángulo}}{\text{cateto adyacente al ángulo}} \quad (6)$$

Se elige usar la función coseno (ecuación 5), para el desarrollo del algoritmo. Teniendo en cuenta que el cateto adyacente al ángulo será la distancia en x de los puntos, entonces se tiene al despejar de la función coseno (ecuación 5) mostrada anteriormente, que el primer ángulo estará dado por la ecuación 7. Siendo la ecuación 8, la que regirá el comportamiento angular de los eslabones, conforme se mueva el brazo del usuario.

$$\theta = \text{coseno}^{-1}\left(\frac{\text{cateto adyacente al ángulo}}{\text{hipotenusa}}\right) \quad (7)$$

$$\theta = \text{coseno}^{-1}\left(\frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}\right) \quad (8)$$

La operación se repite para cada uno de los centros de masa encontrados, usando los centroides actual y siguiente, que encontrados de abajo hacia arriba. Permitiendo así encontrar los ángulos de manera más eficiente.

Al tener dichos ángulos, es posible ahora realizar el posicionamiento de los eslabones conforme a los ángulos. Estos ángulos son enviados por el software de manera automática, vía puerto serie, para ser capturados por la placa Arduino al leer dicho puerto.

### **Microcontrolador**

Para el microcontrolador se eligió usar una placa Arduino modelo UNO, debido a su facilidad de manejo, así como la posibilidad de conectar con distintos paquetes de software utilizados en la realización del proyecto. Tras capturar los ángulos, la placa se encarga de leer los datos e identificarlos en el orden correcto, para después encargarse de enviar una señal modulada en ancho de pulso, que es usada para el control de posición de los servomotores.

### **Diagramas de Flujo**

En esta sección se explica la parte de la programación y su funcionamiento dentro del sistema completo, mediante 2 diagramas de flujo.

En uno se explica el funcionamiento del programa que está corriendo en una computadora personal; la cual funciona como la interface entre el usuario, el arduino y la cámara. Además, también se encarga de procesar parte de la información de control y enviarla a la placa de Arduino. En el primer diagrama de flujo, que se muestra en la figura 7, se detalla la programación realizada en la computadora con Matlab. Mientras que en el otro diagrama se muestran los pasos en la programación de la placa Arduino, con el fin de controlar a los servomotores. Para la programación de la placa Arduino los pasos requeridos son menos. El programa creado tiene la función de recibir los datos de ángulo, por parte de la computadora, una vez iniciada la comunicación con la misma. Una vez recibido al menos un dato de ángulo, el programa se encargará de transmitir la señal de control al servomotor correspondiente, Si se recibe un solo un dato, solo se podrá controlar el eslabón inferior, debido al orden jerárquico establecido y seleccionado al realizar la programación. El orden es importante, ya que este indica que eslabón se mueve con cada uno de los datos, siendo el primer dato el que mueve el eslabón inferior,

el segundo mueve el eslabón intermedio y el tercero mueve el eslabón superior. La estructura de funcionamiento del es mostrada en figura 8.

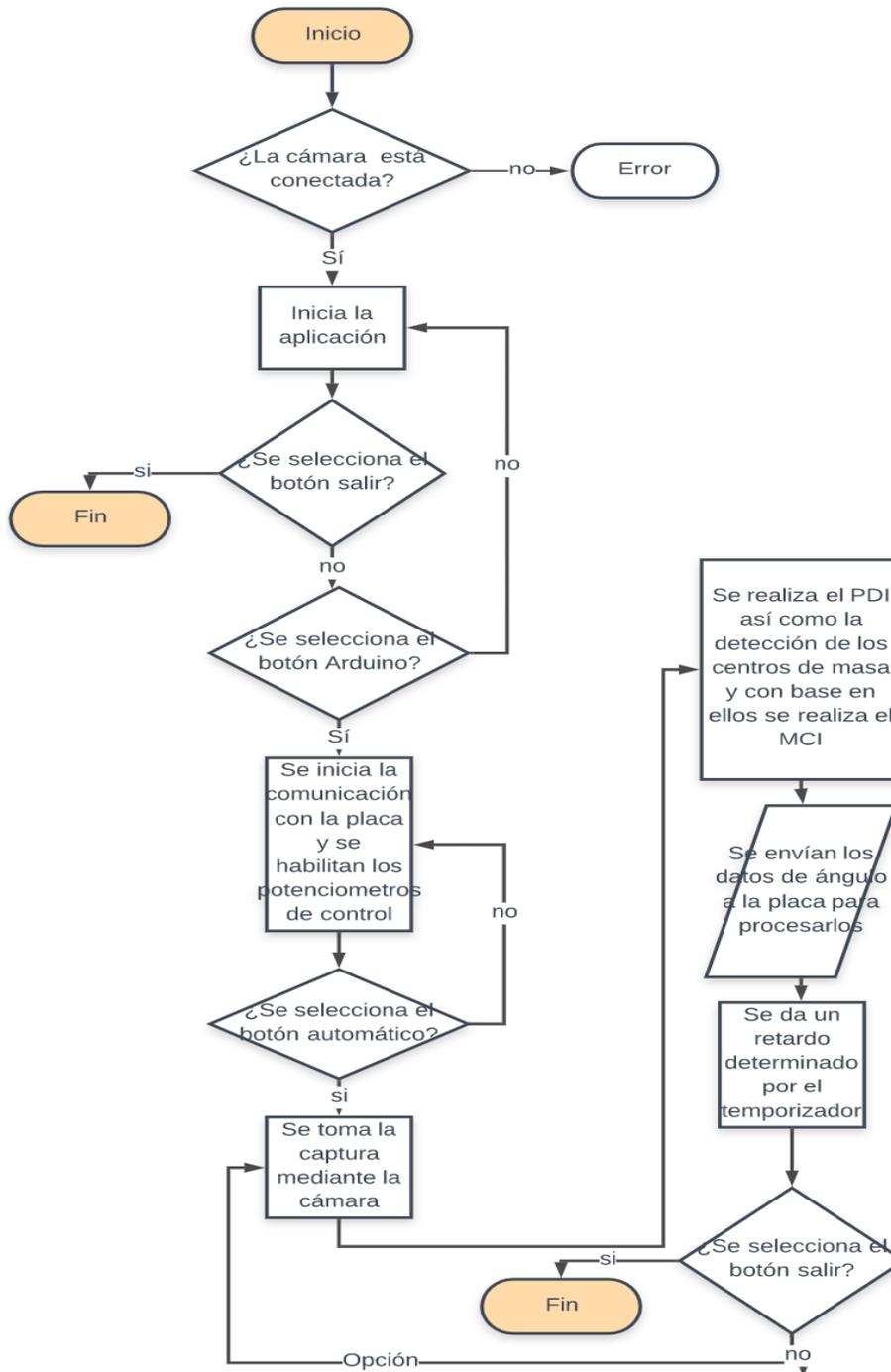


Figura 7 Diagrama de flujo del programa corriendo en la computadora

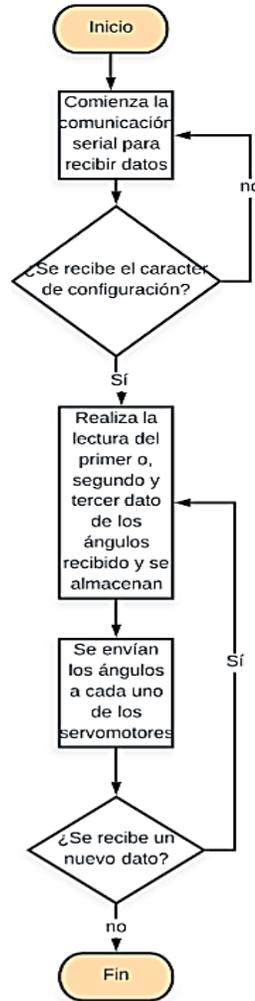


Figura 8 Diagrama de flujo del programa corriendo en la placa Arduino.

### Estructura Mecánica

El sistema mecánico, desarrollado para el manipulador, consta de 3 eslabones fabricados en metacrilato de 6mm. Se eligió el uso de dicho material debido a su facilidad de manejo, bajo peso y gran resistencia. Se cortaron con medidas adecuadas para poder montar los motores en los mismos eslabones, y tener 10 centímetros desde el centro del eje del motor al otro centro del eje del otro motor, esto en el eslabón más grande. Mientras que para los otros 2 eslabones se dejaron 5 centímetros entre los dos ejes. Haciendo así que esto fuera de alguna forma similar, a escala, de las dimensiones de las secciones del brazo que serán identificadas por la cámara. El servomotor más bajo, se encuentra unido a una base

fabricada en madera, para añadir peso y evitar así la caída del sistema mecánico. Este servo es el que sostiene la suma de todo el peso del robot.

### **Servomotores**

Para la realización del proyecto en cuestión, se eligió el uso de 3 servomotores Mg995. Los servomotores cuentan con la ventaja de tener integrado en el mismo encapsulado con un circuito de control, un encoder y un potenciómetro unido al eje del motor. Este dispositivo al ser accionado por la señal pwm y moverse, mueve el eje del potenciómetro y mediante una realimentación de posición, permite ubicar la posición actual del motor en relación a la deseada, y de esta manera llevarlo a esa posición angular. Esto elimina la necesidad de uso de un medidor externo para poder determinar la posición actual del motor, y hace más fácil el control de los motores, ya que estos solo requieren una alimentación de 5 volts de C.D. y la señal de control. El eje de cada uno de los servomotores está unido, mediante un grupo de tornillos, a cada uno de los eslabones del manipulador, permitiendo así el movimiento en conjunto para su posicionamiento.

### **3. Resultados**

Tras el diseño del brazo, se esperaba que este tuviera una movilidad eficiente, y por el bajo al peso y la potencia adecuada de los motores, no se entorpeciera en ningún momento el movimiento para alcanzar el ángulo a imitar. Se cumplió con las expectativas previstas, además como se observa en la figura 9, se logra identificar el centro geométrico de cada parte del brazo del usuario. Además, en la pantalla de la interfaz gráfica, se observa adecuadamente la imagen captada por la cámara después de ser binarizada. El manipulador debe mover sus eslabones adecuadamente para tener los ángulos correspondientes a centroides calculados. El comportamiento del brazo no es exactamente igual al que se presenta en el modelo matemático. Existen limitaciones físicas que están presentes por utiliza servomotores que solo trabajan entre 0 y 180 grados, teniéndose que aplicar consideraciones restrictivas en el control, cuando los ángulos sobrepasan el área de trabajo del brazo robótico.

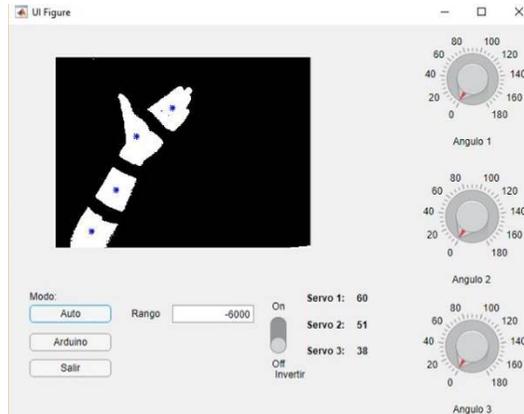


Figura 9 Interfaz gráfica del usuario, con brazo binarizado y centroides.

La figura 10 muestra el brazo robótico debidamente posicionado, y gracias a que resultó ser un robot muy ligero y con baja latencia, al correr el programa de PDI, entre cada procesamiento es posible capturar aproximadamente 24 imágenes por segundo, por lo cual se logra apreciar un movimiento suave y a la par con el del brazo fotografiado.



Figura 10 Sistema de brazo robótico imitador de 3g.d.l.

#### 4. Discusión

En primera instancia se ha creado un sistema de comunicación con el motivo de lograr un procesamiento digital de imágenes, el cual se logra al tener conectada una cámara web a la PC, donde gracias a esta se obtiene una imagen para ser binarizada. En este sistema queda claro que la cámara y la computadora deben ser mejoradas, en caso de que se necesite mayor velocidad, pero para los fines didácticos no es necesario. Otra mejora podría ser que se puede incrementar el tamaño del robot, pero dado que la única función en este momento es imitar, se

puede considerar adecuado, ya que para esto no necesita cargar ningún objeto. Hay que mencionar que el mismo sistema mecánico realizado y descrito en este documento es susceptible de ser modificado, con el fin de cambiar el campo de aplicación, ya que los brazos robóticos son altamente versátiles en muchas áreas.

## **5. Conclusiones**

Actualmente el brazo robótico cumple su tarea de imitar de una manera muy simple, por lo que la exactitud de los ángulos que se envían a los servomotores no siempre resulta exacta y precisa, esto se debió muchos otros factores que no fueron considerados, P. ej. no existe una segunda cámara que le tome fotos al brazo robótico para tener una redundancia en el error de posición. Pero de igual forma la exactitud de los ángulos enviados a los servomotores es suficientemente exacta para cumplir con el objetivo.

Un aspecto importante del prototipo es el procesamiento de la imagen, el algoritmo para obtener los ángulos utilizando el modelo cinemático inverso entre 2 puntos es la forma más ventajosa que se pudo utilizar, ya que, sin mucho procesamiento, se logra el objetivo de imitar el brazo de una persona.

Es claro que el prototipo puede mejorarse en diversos aspectos, tanto como de precisión, velocidad de respuesta, control, etcétera, pero como se encuentra actualmente permite replicar el movimiento que detecte la cámara de la mejor manera posible.

## **6. Bibliografía y Referencias**

- [1] Barrientos A., Peñín L.F., Balaguer C., Aracil R. Fundamentos de robótica. MCGRAW-HILL, España, 2007.
- [2] Cao F., Shepherd B. MIMIC: a robot planning environment integrating real and simulated worlds. Proceedings IEEE International Symposium on Intelligent Control, USA, 1989.
- [3] Chacon M. Procesamiento Digital de Imágenes, TRILLAS, México, 2007.
- [4] Gao F., Higa H., Uehara H. Soken T. A vision-based user interface of a mobile robotic arm for people with severe disabilities, International Conference on

- Intelligent Informatics and Biomedical Sciences (ICIIBMS), 172-175, Japón, 2015.
- [5] Craig J.J. Robótica. PEARSON EDUCACIÓN, México, 2006.
- [6] Jiang L., Wang W., Chen Y., Jia Y. Personalize Vision-based Human Following for Mobile Robots by Learning from Human-Driven Demonstrations. Proceedings of the 27th IEEE International Symposium on Robot and Human Interactive Communication, 726 - 731, China, 2018.
- [7] Ogata, K. Ingeniería de control moderna. PEARSON, México, 2010.
- [8] Ollero Baturone A. Robótica: Manipuladores y robots móviles, ALFAOMEGA, España, 2001.
- [9] Saha S.K. Introducción a la Robótica, Mc Graw-Hill, México, 2010.
- [10] Sequeira P.J., Caldas J.R. An experimental testbed for visual servo control of robotic manipulators. EFTA IEEE Conference on Emerging Technologies and Factory Automation Proceedings, 377-382, Portugal, 2003.
- [11] Sifuentes-Mijares J.; Santibanez V.; Meza J. L. A globally asymptotically stable nonlinear PID regulator with fuzzy self-tuned PD gains, for robot manipulators. World Automation Congress (WAC), USA, 2014.
- [12] Wu H., Cai Z., Wang Y. Vision-based auxiliary navigation method using augmented reality for unmanned aerial vehicles, IEEE 10th International Conference on Industrial Informatics, 520-525, China, 2012.