

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ
INSTITUTO DE INGENIERÍA Y TECNOLOGÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL Y MANUFACTURA,
"Desarrollo de un algoritmo de control para un robot de 2 grados de libertad con una cámara y un láser"

Visión

Francesco García Luna

Alumnos

Martín Iván Martínez Vela-180000

Maximiliano García Carreón-180026

Edgar Irán García Ojeda-180032

Fernando Durán Romero-180079

Allan Daniel González Antonio-180083

Ciudad Juárez, Chihuahua.

4 Mayo 2022

Índice general

1. Introducción.	1
1.1. Antecedentes.	1
1.2. Planteamiento del problema.	1
1.3. Objetivo.	2
1.3.1. Objetivos específicos.	2
2. Marco Teórico.	3
2.1. Marcador ArUco	3
2.2. Controlador PID	3
2.3. Cámara	3
2.4. Servomotor	3
2.5. Sistema de control en lazo cerrado	5
2.6. OpenCV.	5
2.7. Matriz de rotación.	6
3. Metodología.	7
3.1. Diagrama de flujo.	8
3.2. Modelo 3D	10
3.3. Circuito Electrico	11
4. Resultados	12
5. Conclusion	14
6. Referencias.	15

Índice de figuras

2.1. Estructura de un servomotor MG995.	4
2.2. Estructura de un servomotor MG995.	4
2.3. Representación del algoritmo PID.	5
3.1. Diagrama de flujo empleado en Python.	8
3.2. Diagrama de flujo empleado en Arduino.	9
3.3. Modelo 3D.	10
3.4. Circuito electrico.	11
4.1. Resultados obtenidos y corrección de error.	12
4.2. Gráfica que representa la corrección del error.	13

Capítulo 1

Introducción.

En este presente documento se mostrarán los resultados del desarrollo de un algoritmo para apuntar al centro de un ArUco mediante un láser, así como la metodología empleada. El documento se encuentra seccionado por antecedentes, planteamiento del problema, objetivos, marco teórico, metodología, diagrama de flujo, resultados y conclusión.

1.1. Antecedentes.

La visión por computadora es aquel conjunto de herramientas y métodos que permiten obtener, procesar y analizar imágenes del mundo real con la finalidad de que puedan ser tratadas por un ordenador.

Una de las primeras aplicaciones de las imágenes digitales apareció en la industria de la prensa. Dichas imágenes eran enviadas a través de un cable submarino entre Londres y Nueva York. De ahí los equipos se encargaban de codificar las imágenes para transmitirla y luego se reconstruían al otro extremo. En un inicio los trabajos para mejorar la calidad visual de las imágenes se concentraban principalmente en la selección de los métodos de impresión y la distribución de los niveles de intensidad.

Para el procesamiento de imágenes se ha desarrollado bibliotecas referentes a la visión artificial, la más utilizada y popular hasta la fecha es la biblioteca abierta conocida como OpenCV que significa open computer vision (visión artificial abierta). Es utilizada en la detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, estas son algunas aplicaciones. Si bien OpenCV está desarrollado en C++ también está disponible en el lenguaje de programación de Python.

1.2. Planteamiento del problema.

La problemática de este proyecto se enfoca en diseñar un sistema capaz de apuntar a un ArUco y disparar al centro del mismo de manera automática.

1.3. Objetivo.

Desarrollar un algoritmo de control para un robot de dos grados de libertad que tendrá como tarea apuntar al centro de un ArUco por medio de un láser.

1.3.1. Objetivos específicos.

- Analizar algoritmos de sistema de visión por computadora.
- Analizar algoritmos para controlar servomotores.
- Desarrollar algoritmo de control.
- Obtener marco de referencia del láser.
- Obtener posición del ArUco con respecto al láser.
- Validar algoritmo.

Capítulo 2

Marco Teórico.

2.1. Marcador ArUco

Un marcador ArUco es un identificador único compuesto por un borde y una matriz binaria que se utiliza como un sistema de referencia, el cual permite conocer la posición y orientación utilizando la librería de OpenCV. La codificación binaria permite su identificación y la aplicación de técnicas de detección de errores.

El tamaño del marcador determina el tamaño de la matriz interna. Los distintos tipos de arucos contienen matrices de 4x4, 5x5, 6x6, 7x7 y funcionan con un diccionario que son un conjunto de marcadores para una aplicación específica para cada uno de los valores de las matrices.

2.2. Controlador PID

Los controladores automáticos comparan el valor real de la salida de una planta con la entrada de referencia, es decir el valor deseado, determinan el error o desviación y producen una señal de control que reduzca el error de control a cero o a un valor pequeño. una de las representaciones para el controlador PID en funcion del tiempo esta dada por la ecuación:

$$u(t) = Kp \left(e(t) + \frac{1}{T_1} \int_t^0 e(\tau) d\tau + Td \frac{de(t)}{dt} \right) \quad (2.1)$$

2.3. Cámara

Es un dispositivo encargado principalmente de tomar la imagen de seguimiento, Esta diseñada para realizar el seguimiento de color. Dicha cámara se puede usar para la localización de líneas de referencia, sigue líneas, o el seguimiento de una luz en movimiento.

2.4. Servomotor

Es un actuador rotativo o motor eléctrico, se define básicamente por una regulabilidad extrema y sus correspondientes y sus correspondientes posibilidades de control. Por lo tanto,

permite el control preciso de posición de ángulo, aceleración y velocidad. La estructura básica del servomotor incluye un sensor de posición del rotor.

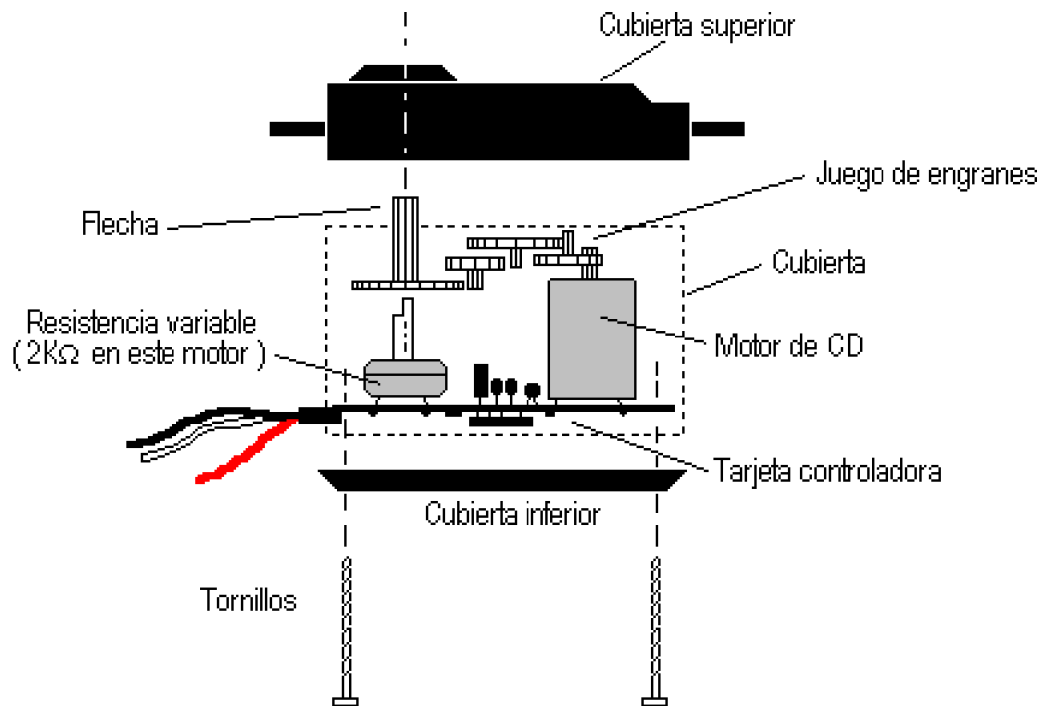


Figura 2.1: Estructura de un servomotor MG995.

El servomotor cuenta con tres terminales de conexión:

- Vcc: el cual corresponde al cable de alimentación.
- Gnd: Corresponde al cable de tierra.
- Señal: corresponde al cable de entrada de señal (PWM)

PWM=Orange (⏏)
 Vcc = Red (+)
 Ground=Brown (−)

Figura 2.2: Estructura de un servomotor MG995.

La tensión de alimentación de los servomotores suele ser de 4 a 8 voltios, en nuestro caso, los servos modelo MG995 utilizan una tensión de 4.8 a 7.2, el control del servomotor consiste básicamente en indicarle la posición en la que se debe situar.

2.5. Sistema de control en lazo cerrado

Donde $u(t)$ es la variable de control y e es el error de control. El controlador PID es la suma de tres terminos donde tenemos que el termino (P) que es proporcional al error, el termino (I) que es proporcional a la integral del error y el termino (D) que es proporcional a la derivada del error.

Los parámetros del controlador son ganancia proporcional K_p , tiempo integral representado por T_i y tiempo derivado T_d .

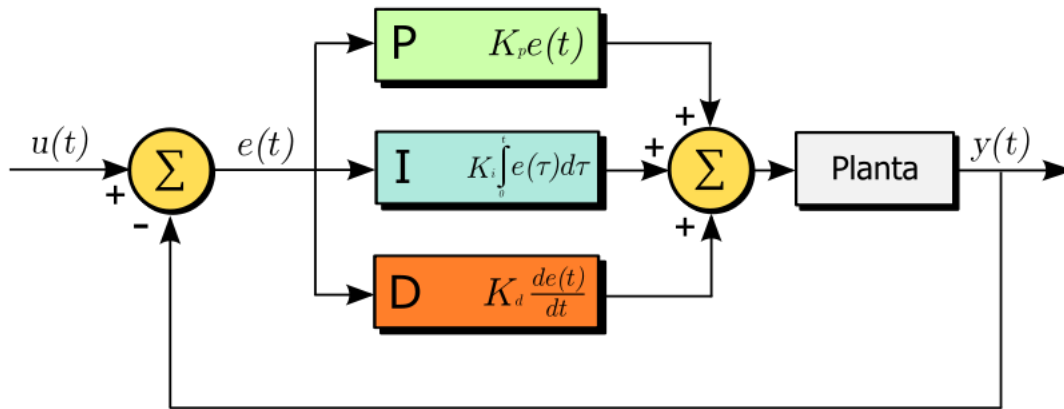


Figura 2.3: Representación del algoritmo PID.

2.6. OpenCV.

Biblioteca de código abierto para aplicaciones que utilicen visión artificial, es utilizada en varias aplicaciones como control de procesos, sistemas de seguridad con detección de movimiento, reconocimiento de objetos, robótica avanzada, etc.

Esta librería cuenta con una buena eficiencia computacional y un fuerte enfoque en aplicaciones de tiempo real. Opencv contiene también varias funciones utilizadas en distintas áreas de la visión por computadora.

Algunos ejemplos de sus aplicaciones son:

- Identificación de objetos
- Reconocimiento de rostros
- Interfaces de usuario
- Representación 3D.

2.7. Matriz de rotación.

Matriz que representa la rotación en grados del plano en sentido antihorario. La orientación de un punto puede expresarse con los tres ángulos de Euler que corresponden a la rotación respecto a los ejes x, y, z.

La rotación en cada eje puede expresarse como una matriz para cada ángulo, como se muestra a continuación:

$$Rz(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$Ry(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$Rx(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Capítulo 3

Metodología.

Para desarrollo de este proyecto tuvieron que realizar los siguientes pasos.

Primero se tuvo que calibrar la camara que se iba a utilizar, posteriormente se monto la camara en el robot de 2 grados de libertad y tambien se le implemento un laser. Una vez hecho esto, utilizamos un arduino como driver para controlar los movimientos del robot y el estado del laser.

Establecemos una comunicacion serial entre Python y el Arduino para poder enviar la señal de control que se calcula en Python y que el Arduino la reciba para así poder mover los servos y laser. Para calcular la señal de control usamos un control PI (Proporcional e Integral), obteniendo el error en relacion de la distancia que se encuentre el centro del Aruco al centro del frame que captura la camara.

Para obtener el error, debemos de calcular el error proporcional y el error integral. Para obtener el error proporcional, obtenemos la diferencia de pixeles entre el centro de la imagen y el centro del ArUco, con la siguiente ecuacion (3.1):

$$E_p = E * k_p \quad (3.1)$$

Donde E es el error.

Una vez obtenido esto, calculamos el error integral, el cual es el area bajo la curva multiplicada por una ganancia integral. Para conseguir el area bajo la curva, lo hacemos con esta ecuacion (3.2):

$$Area_{BC} = Area_{BC} + E \quad (3.2)$$

Una vez obtenemos el error integral, lo multiplicamos por una ganancia integral (3.3).

$$E_i = Area_{BC} * k_i \quad (3.3)$$

Despues de calcular el error proporcional y el error integral, ya sacar la señal de control con la ecuacion (3.4).

$$Señal = Ep + Ei \quad (3.4)$$

3.1. Diagrama de flujo.

En la figura 3.1 se muestra el diagrama de flujo que se utilizó para desarrollar nuestro algoritmo en Python:

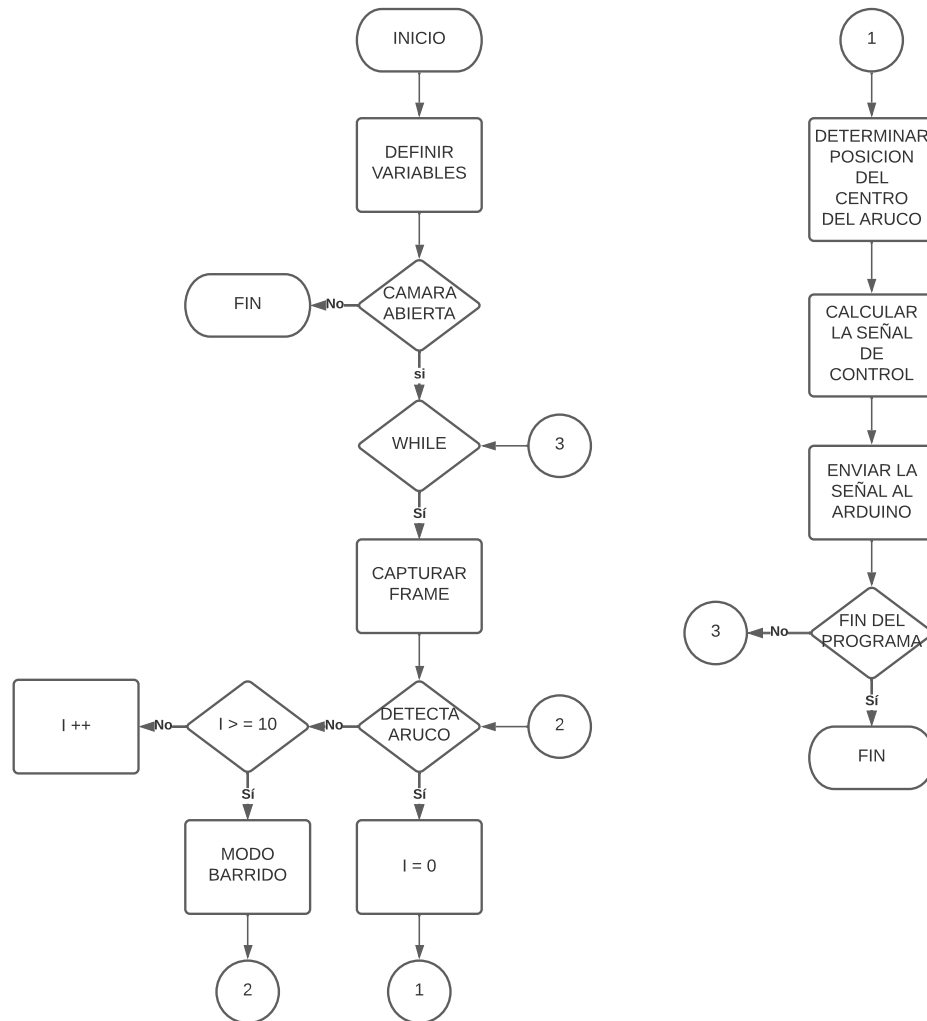


Figura 3.1: Diagrama de flujo empleado en Python.

En la figura 3.2 se muestra el diagrama de flujo que se utilizó para desarrollar nuestro algoritmo en Arduino:

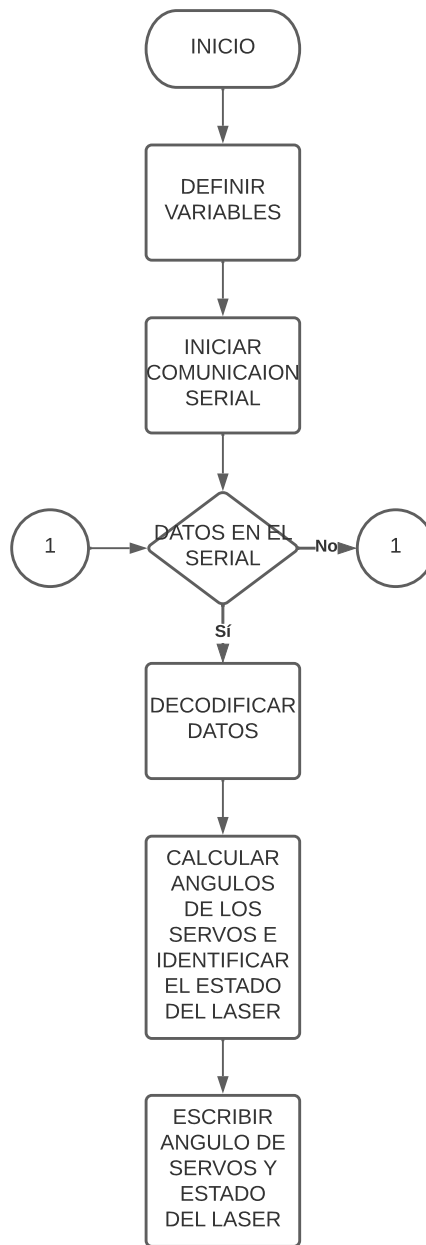


Figura 3.2: Diagrama de flujo empleado en Arduino.

3.2. Modelo 3D

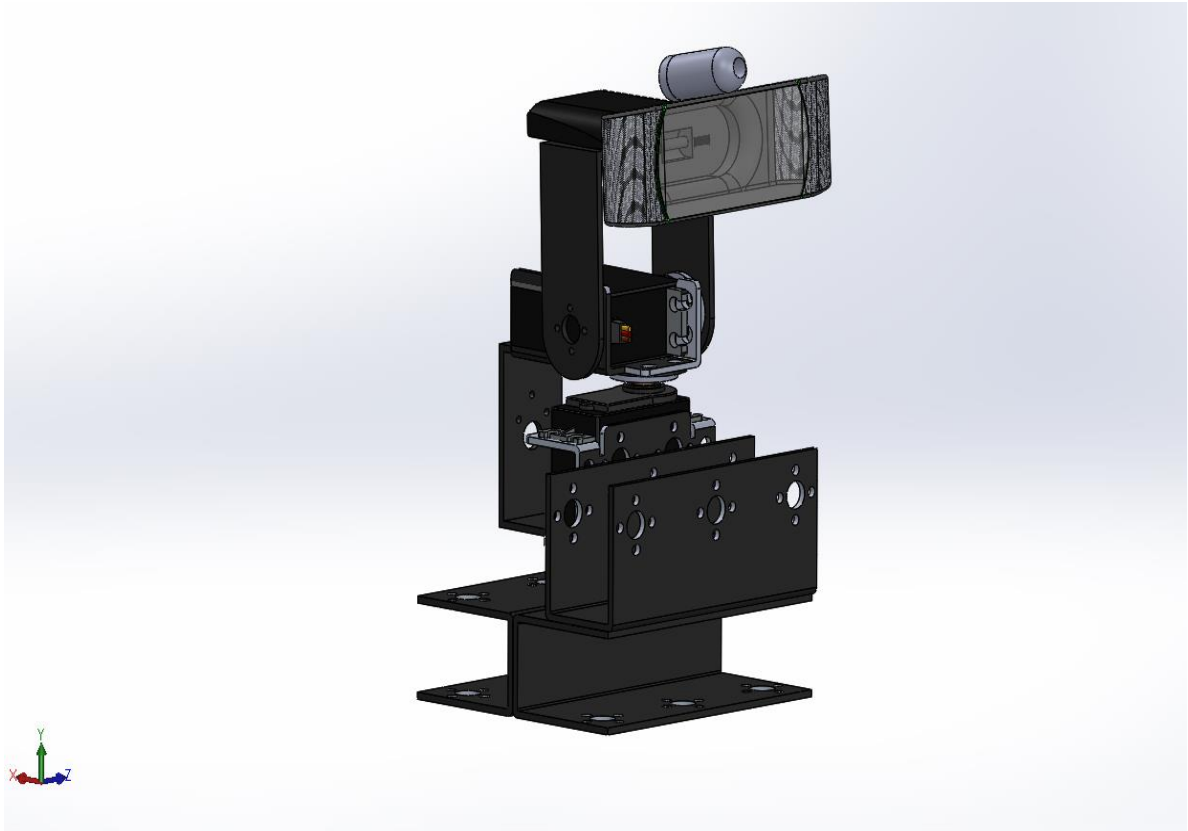


Figura 3.3: Modelo 3D.

3.3. Circuito Electrico

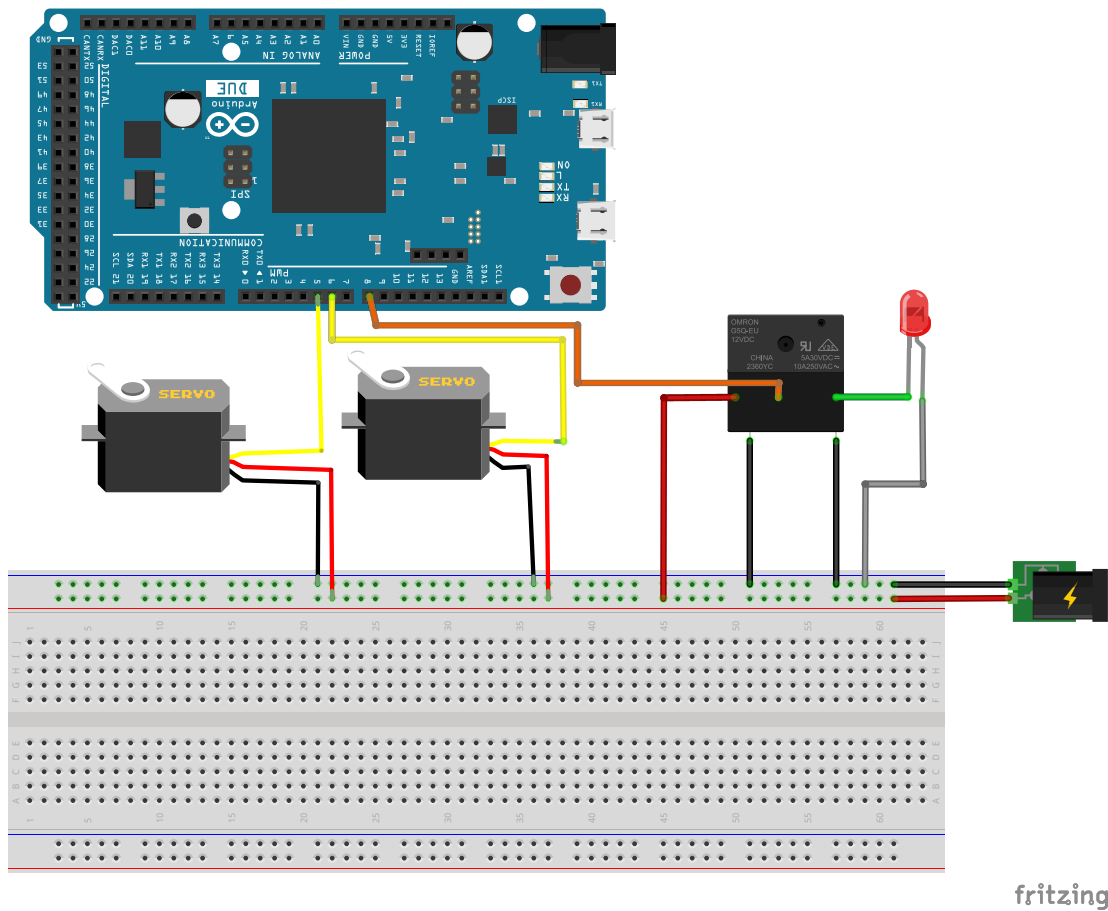


Figura 3.4: Circuito electrico.

Capítulo 4

Resultados

Como resultado obtuvimos un sistema que logra apuntar hacia un ArUco de manera satisfactoria, corrigiendo su posición cada vez que el error sale de la tolerancia.

Aquí, podemos ver cómo funciona en la figura 4.1.

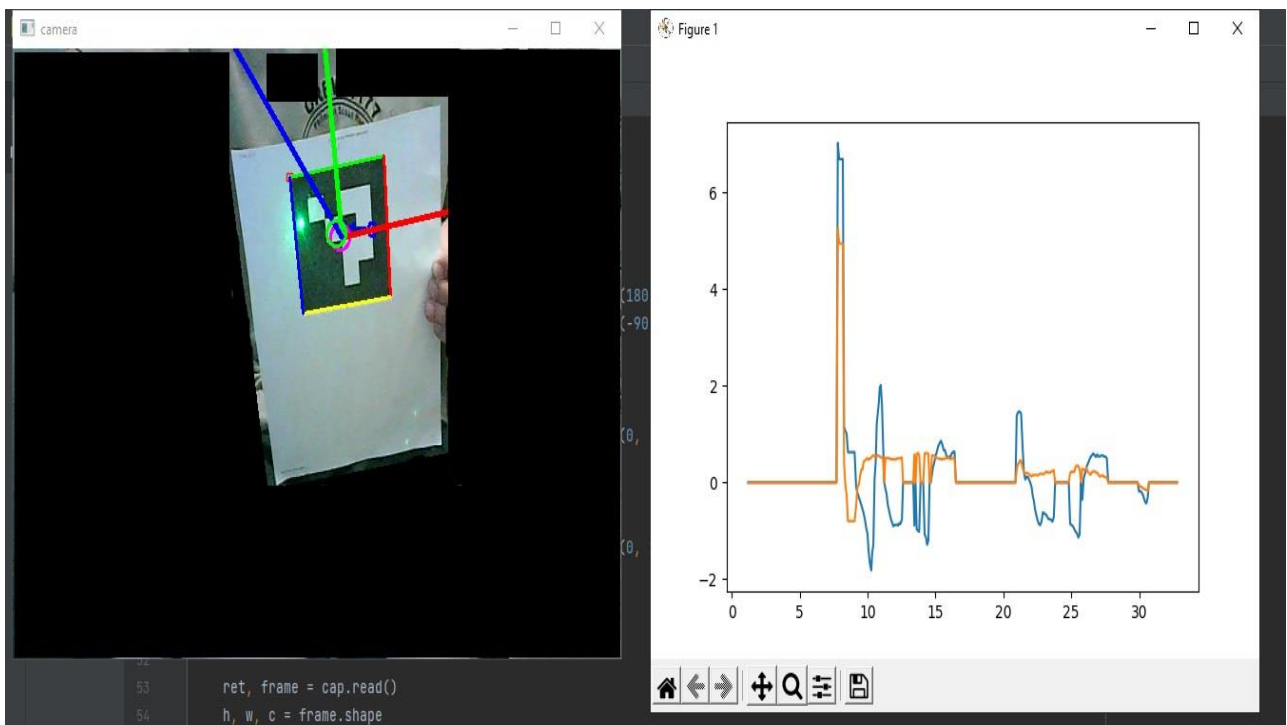


Figura 4.1: Resultados obtenidos y corrección de error.

En la figura 4.2, podemos observar la grafía del error que genera este sistema, podemos ver al principio como tiene un overshoot debido al gran tamaño del error, pero conforme el sistema trabaja, este mismo se va a estabilizar poco a poco.

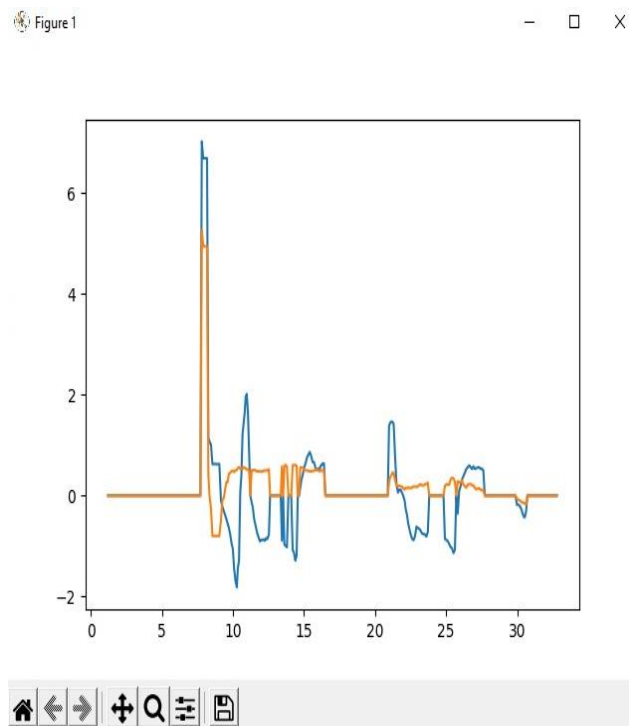


Figura 4.2: Gráfica que representa la corrección del error.

Podemos ver dos líneas independientes, cada línea representa el error de cada grado de libertad del robot.

Capítulo 5

Conclusion

El sistema cumple con la tarea que fue definida, pero la precisión con la que trabaja el sistema depende de muchos factores, como la iluminación, la calidad de la cámara, la posición en donde se encuentra el ArUco, la resolución de los servomotores, etc. Es por ello que un trabajo a futuro, se buscaría de trabajar en los factores que afectan al sistema para que ofrezca una mejor precisión y mejorar el sistema de control agregando el control derivativo.

Capítulo 6

Referencias.

- [1] About-OpenCV". OpenCV (8 Oct 2021). [online], disponible: <https://opencv.org/about/>
- [2] R. Jiménez, "Sistema de seguimiento de objetos usando OpenCv, ArUco y Filtro de Kalman extendido", tesis, Universidad de Sevilla, SE, España, 2018.
- [3] J. N. A. Rivero, «Robot Dirigido Por Voz,» Universidad de las Américas Puebla, Puebla, 2007.
- [4] E. A. Prieto y P. A. G. Quintero, «Diseño e implementación de un robot móvil autónomo con detección de color,» Universidad De San Buenaventura, Bogotá, 2006.
- [5] G. Viera-Maza, «Procesamiento de imágenes usando OpenCV aplicado en Raspberry PI para la clasificación del cacao,» Universidad De Piura, Piura, 2017.
- [6] USON, «USON,» 2010. [En línea]. Available: <http://tesis.uson.mx/digital/tesis/docs/22170/Capitulo2.pdf>. [Último acceso: 6 Abril 2022].
- [7] OpenCV, «OpenCV,» 6 Abril 2022. [En línea]. Available: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html. [Último acceso : 6Abril2022].

[8] A. E. Benavides, «Sistema de control dinámico del manipulador de un robot industrial en empresas troqueladoras de 4 grados de libertad,» Universidad Rafael Bellosó Chacín, Maracaibo, 2013.

[9] M. E. P. Chin, «Desarrollo de un sistema de visión computacional para el vuelo autónomo de un Vehículo Aéreo No Tripulado,» Universidad Autónoma de Yucatán, Mérida, 2019.