



**UNIVERSIDAD DEL AZUAY**  
**FACULTAD DE CIENCIA Y TECNOLOGÍA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA**

**“Desarrollo e implementación de un sistema electromecánico para el control de la posición y la trayectoria de una esfera, realimentado por visión artificial.”**

**Trabajo de graduación previo a la obtención del título de:**

**INGENIERO ELECTRÓNICO**

**Autor:**

**FELIPE AGUSTÍN GARCÍA GÓMEZ**

**ZHIAKOFF EMMANUEL VÁZQUEZ TRONCOSO**

**Director:**

**ANDRÉS PATRICIO CABRERA FLOR**

**CUENCA, ECUADOR**

**2023**

# DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA ELECTROMECAÁNICO PARA EL CONTROL DE LA POSICIÓN Y LA TRAYECTORIA DE UNA ESFERA, REALIMENTADO POR VISIÓN ARTIFICIAL

En este trabajo de titulación se realizó un sistema electromecánico con dos grados de libertad (sistema bola y mesa). Dicho sistema está constituido por una placa metálica cuadrada que, mediante la acción de los servomotores, controla la posición y trayectoria de una esfera que se desplaza sobre dicha placa, manteniéndola estable ante perturbaciones. Una cámara de video con visión artificial se utilizó como retroalimentación del sistema. Se aplicó un controlador PID para posicionar a la esfera en puntos específicos dentro de la plataforma, que varía sus ángulos de inclinación realizando movimientos precisos. A través de una aplicación se puede visualizar en tiempo real el funcionamiento del sistema y su comportamiento ante ruidos y alteraciones, midiendo la respuesta del controlador.

**Palabras clave:** Control PID, Sistema bola y mesa, visión artificial.



---

Andrés Patricio Cabrera Flor  
**Director de Tesis**



---

Felipe Agustín García Gómez  
**Autor**



---

Daniel Esteban Iturralde Piedra  
**Director de Escuela**



---

Zhiakoff Emmanuel Vázquez Troncoso  
**Autor**

# DEVELOPMENT AND IMPLEMENTATION OF AN ELECTROMECHANICAL SYSTEM TO CONTROL THE POSITION AND PATH OF A SPHERE, FED BACK BY ARTIFICIAL VISION

In this work an electromechanical system with two degrees of freedom (ball and plate system) was made. This system is made up of a square metal plate that, through the action of servomotors, controls the position and trajectory of a sphere that moves on the plate, keeping it stable in the event of disturbances. A video camera with artificial vision was used as system feedback. A PID controller was applied to position the sphere at specific points within the platform, which varies its inclination angles making precise movements. Through a computer application, the operation of the system and its behavior under the influence of noise and disturbances can be viewed in real time, measuring the response of the controller.

**Keywords:** PID control, ball and table system, artificial vision.



---

Andrés Patricio Cabrera Flor

**Thesis Director**



---

Daniel Esteban Iturralde Piedra

**School Coordinator**

**Translated by:**

A handwritten signature in blue ink, appearing to read "Felipe Agustín García Gómez".

---

Felipe Agustín García Gómez

**Author**

A handwritten signature in blue ink, appearing to read "Zhiakoff Emmanuel Vázquez Troncoso".

---

Zhiakoff Emmanuel Vázquez Troncoso

**Author**



# Desarrollo e implementación de un sistema electromecánico para el control de la posición y la trayectoria de una esfera, realimentado por visión artificial.

Felipe Agustín García Gómez.  
*Ingeniería Electrónica.*  
*Universidad del Azuay.*  
Cuenca, Ecuador.  
tin13@es.uazuay.edu.ec

Zhiakoff Emmanuel Vázquez Troncoso.  
*Ingeniería Electrónica.*  
*Universidad del Azuay.*  
Cuenca, Ecuador.  
zhiakoff@es.uazuay.edu.ec

**Resumen**— En este trabajo de titulación se realizó un sistema electromecánico con dos grados de libertad (sistema bola y mesa). Dicho sistema está constituido por una placa metálica cuadrada que, mediante la acción de los servomotores, controla la posición y trayectoria de una esfera que se desplaza sobre dicha placa, manteniéndola estable ante perturbaciones. Una cámara de video con visión artificial se utilizó como retroalimentación del sistema. Se aplicó un controlador PID para posicionar a la esfera en puntos específicos dentro de la plataforma, que varía sus ángulos de inclinación realizando movimientos precisos. A través de una aplicación se puede visualizar en tiempo real el funcionamiento del sistema y su comportamiento ante ruidos y alteraciones, midiendo la respuesta del controlador.

**Palabras Clave**—PID, Control PID, Sistema bola y mesa, visión artificial.

## I. INTRODUCCIÓN.

En el entorno en el que vivimos hay muchos factores naturales que se hallan en un estado de desequilibrio. En [1] se dice que, para mejorar la comprensión del estudio de estos fenómenos, se debe fundamentar una interpretación de los sistemas no lineales al relacionarlas con ecuaciones que rigen el sistema.

El sistema ball and plate busca ayudar a los estudiantes de ingeniería a la interpretación de los sistemas de control. Se define a un sistema ball and plate, como un sistema electromecánico con instrumentación electrónica y una etapa de control, siendo un sistema que se deriva de otro llamado ball on beam, cuyo propósito es balancear una esfera sobre una viga [2].

La diferencia entre estos dos sistemas son los grados de libertad presentes en cada uno: el principal objetivo en el sistema ball and plate es estabilizar una esfera en un punto o hacerla seguir una trayectoria predeterminada dentro de un plato (la mesa) [3].

El método de control utilizado en el desarrollo de este proyecto es el control PID (control proporcional, integral y derivativo), que es una herramienta de lazo cerrado, cuyo propósito es obtener un estado de salida anhelado, siendo la más utilizada en las industrias por sus características simples

y efectivas, ofreciendo una solución eficiente a los problemas de control en el mundo por medio de sus distintos enfoques, permitiendo reaccionar ante situaciones imprevistas [4].

Acción proporcional (P): Conformada por la multiplicación de la señal de error y la constante proporcional KP, siendo en esencia un amplificador con ganancia ajustable [5].

Acción derivativa (D): Esta acción es la respuesta al cambio de error, permitiendo una repercusión rápida, utilizándose cuando el sistema requiere estabilizarse ante una oscilación excesiva [5].

Acción integral (I): La principal función de esta acción es garantizar que la salida del sistema corresponda al punto de referencia deseado, realizando acciones correctoras siempre que presente un error entre la comparación del punto de referencia y la variable a controlar [5].

El sistema se guía por la siguiente ecuación [6]:

$$c_{(T)} = K_p * e_{(t)} + K_i * \int e_{(t)} * dt + K_d * \frac{de_{(t)}}{dx} \quad (1)$$

El proyecto está desarrollado en Python, al ser un lenguaje versátil y de fácil uso que, por medio de un intérprete, permite soportar diferentes módulos y paquetes, además de ofrecer una variedad de bibliotecas, presentando una gran ventaja al momento de procesar imágenes sobre otros lenguajes de programación como MATLAB, poseyendo un mejor tiempo de respuesta [7].

Como sensor del sistema se utiliza una cámara web, que mediante el uso de visión artificial permite examinar e interpretar las imágenes, extrayendo atributos específicos y características de la imagen procesada, con el único objetivo de generar información de utilidad, permitiendo al sistema tomar decisiones veloces basada en lo que observa [8].

Para modificar la posición de la esfera y realizar acciones correctivas, el sistema cuenta con la implementación de un microcontrolador, encargado de transmitir la información a los actuadores, teniendo una comunicación maestro-esclavo con el computador. [9]

En la Fig. 1 se puede comprender de una mejor manera la arquitectura del sistema.

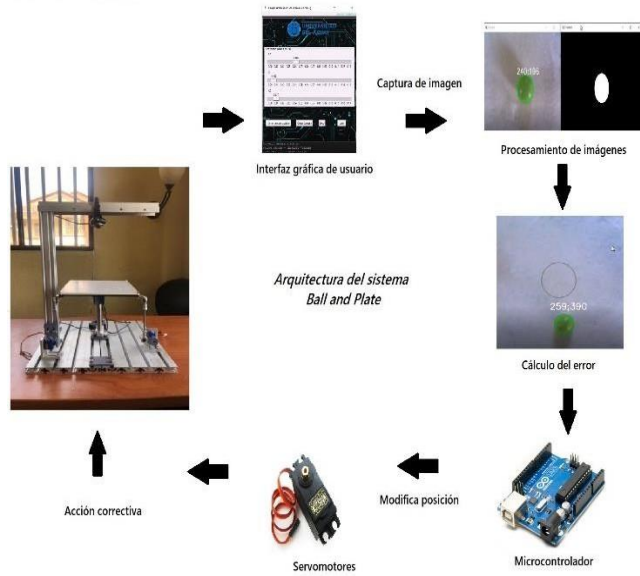


Fig. 1. Arquitectura del sistema Ball and Plate.

## II. METODOLOGÍA.

### A. Modelo Matemático.

En el presente apartado se desarrolló el modelo matemático del sistema electromecánico Ball and Plate.

Para poder describir las ecuaciones físicas del movimiento de la esfera y la rotación de la plataforma en sus ejes “X” y “Y”, se hace uso de las ecuaciones de Lagrange [7], pudiendo así obtener las ecuaciones diferenciales y su representación en el espacio de estados, realizando de esta manera una aproximación lineal del modelo, a fin de poder encontrar la función de transferencia de la planta.

A fin de modelar el sistema, se hicieron algunos supuestos y condiciones con la finalidad de simplificar el análisis.

- La esfera es totalmente homogénea y simétrica.
- La fricción y el rozamiento existentes entre la esfera y la plataforma serán despreciados.
- La esfera permanece en todo momento en contacto con la plataforma.

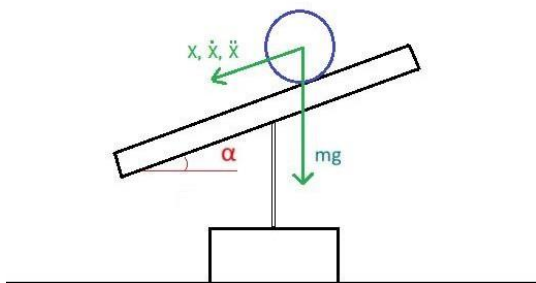


Fig. 2. Diagrama de cuerpo libre en el eje X del sistema Ball and Plate

Para un mejor entendimiento de la demostración matemática es importante identificar los parámetros y variables del sistema Tabla 1.

TABLA I. PARÁMETROS Y VARIABLES DEL SISTEMA

Simbología	Descripción	Unidad
x	Desplazamiento de la esfera en el eje “X”.	m
$\dot{x}$	Velocidad de la esfera en el eje “X”.	m/seg
$\ddot{x}$	Aceleración de la esfera en el eje “X”.	m/seg <sup>2</sup>
y	Desplazamiento de la esfera en el eje “Y”.	m
$\dot{y}$	Velocidad de la esfera en el eje “Y”.	m/seg
$\ddot{y}$	Aceleración de la esfera en el eje “Y”.	m/seg <sup>2</sup>
$\Theta_x$	Ángulo entre el eje “X” del plano con la horizontal.	radianes
$\Theta_y$	Ángulo entre el eje “Y” del plano con la horizontal.	radianes
m	Masa de la esfera.	kg
h	Altura en la que se encuentra la esfera.	m
g	Constante gravitacional.	m/seg <sup>2</sup>
R	Radio de la esfera.	m
J	Momento de inercia.	Kg m <sup>2</sup>
$\dot{s}$	Velocidad lineal por rotación.	m/seg
$\dot{r}$	Velocidad lineal de la esfera.	m/seg
Wb	Velocidad angular de la esfera.	radianes/seg
$\theta$	Velocidad angular del plano.	radianes/seg

Según la Fig.2 a fin de obtener la trayectoria de la esfera, se aplica Lagrangiano en el tiempo, tomando en cuenta que la energía cinética del sistema  $E_{c(t)}$  es la suma de la energía cinética de rotación y la energía cinética de traslación [10].

$$L(t) = E_{c(t)} - E_{p(t)} \quad (2)$$

El sistema, al estar en un plano bidimensional, posee dos tipos de cada una de las energías descritas anteriormente:

$$E_{c(t)} = E_{rot\ 1(t)} + E_{rot\ 2(t)} + E_{tras\ 1(t)} + E_{tras\ 2(t)} \quad (3)$$

Donde:

$$E_{rot\ 1(t)} = \frac{1}{2} J * Wb_{(t)}^2 = \frac{1}{2} J * \left[ \frac{\dot{r}_{(t)}}{R} \right]^2$$

$$E_{rot\ 2(t)} = \frac{1}{2} J * \theta_{(t)}^2$$

$$E_{tras\ 1(t)} = \frac{1}{2} m * \dot{r}_{(t)}^2$$

$$E_{tras\ 2(t)} = \frac{1}{2} m * \dot{s}_{(t)}^2$$

Se reemplaza en la ecuación (3), y mediante agrupación y simplificación de términos se puede expresar dicha ecuación de la siguiente manera:

$$E_{c(t)} = \frac{1}{2} J * \left[ \theta_{(t)}^2 + \left( \frac{\dot{r}_{(t)}}{R} \right)^2 \right] + \frac{1}{2} m * \left[ \dot{r}_{(t)}^2 + \dot{s}_{(t)}^2 \right] \quad (4)$$

La energía potencial de la esfera en la plataforma es:

$$E_{p(t)} = -m * g * h \quad (5)$$

Sustituyendo las ecuaciones (4) y (5) en (2) se tiene:

$$L_{(t)} = \frac{1}{2} J * [\dot{\theta}^2 + (\frac{\dot{r}(t)}{R})^2] + \frac{1}{2} m * [\dot{r}^2 + \dot{s}^2] + m * g * h \quad (6)$$

Al descomponer la velocidad angular ( $\theta$ ), la velocidad lineal ( $\dot{r}$ ), la velocidad lineal de rotación ( $\dot{s}^2$ ) y la altura ( $h$ ) en sus componentes, la ecuación (6) se puede expresar como:

$$L_{(t)} = \frac{1}{2} J * [(\dot{\theta}_x)^2 + (\dot{\theta}_y)^2 + \frac{\dot{x}^2}{R^2} + \frac{\dot{y}^2}{R^2}] + \frac{1}{2} m * [x^2 + y^2 + [X(\dot{\theta}_x) + Y(\dot{\theta}_y)]^2] + m * g * X \sin(\theta_x) + Y \sin(\theta_y) \quad (7)$$

Para poder realizar el análisis en el eje "X" y "Y" se aplica la ecuación Lagrange-Euler [11].

$$\frac{d}{dt} \left( \frac{\delta L}{\delta \dot{x}} \right) - \left( \frac{\delta L}{\delta x} \right) = Q \quad (8)$$

La ecuación (7) posee 4 variables ( $\dot{x}$ ,  $\dot{y}$ ,  $\dot{\theta}_x$ ,  $\dot{\theta}_y$ ), y se debe aplicar la ecuación (8) para cada una de las variables.

Para  $\dot{x}$ :

$$\frac{d}{dt} \left( \frac{\delta L}{\delta \dot{x}} \right) - \left( \frac{\delta L}{\delta x} \right) = 0 \quad (9)$$

Sustituyendo y despejando  $\ddot{x}$  se tiene:

$$\ddot{x} = \frac{1}{\left( \frac{1}{R^2} + m \right)} * (m * x(\dot{\theta}_x)^2 + m * y(\dot{\theta}_y))(\dot{\theta}_x) + m * g * \sin(\theta_x) \quad (10)$$

Para  $\dot{y}$ :

$$\frac{d}{dt} \left( \frac{\delta L}{\delta \dot{y}} \right) - \left( \frac{\delta L}{\delta y} \right) = 0 \quad (11)$$

Sustituyendo y despejando  $\ddot{y}$  se tiene:

$$\ddot{y} = \frac{1}{\left( \frac{1}{R^2} + m \right)} (m * y(\dot{\theta}_y)^2 + m * x(\dot{\theta}_x))(\dot{\theta}_y) + m * g * \sin(\theta_y) \quad (12)$$

Para  $\theta_x$ :

$$\frac{d}{dt} \left( \frac{\delta L}{\delta \dot{\theta}_x} \right) - \left( \frac{\delta L}{\delta \theta_x} \right) = TX \quad (13)$$

Sustituyendo y despejando  $\ddot{\theta}_x$  se tiene:

$$\ddot{\theta}_x = \frac{1}{m * x^2 + J} [TX + (m * g * X \cos(\theta_x)) - (\dot{x} * y * \dot{\theta}_x) - (\dot{y} * x * \dot{\theta}_x) - (x * y * \ddot{\theta}_x) - (2 * x * \dot{x} * \dot{\theta}_x)] \quad (14)$$

Para  $\theta_y$ :

$$\frac{d}{dt} \left( \frac{\delta L}{\delta \dot{\theta}_y} \right) - \left( \frac{\delta L}{\delta \theta_y} \right) = TY \quad (15)$$

Sustituyendo y despejando  $\ddot{\theta}_y$  se tiene:

$$\ddot{\theta}_y = \frac{1}{m * y^2 + J} [TY + (m * g * Y \cos(\theta_y)) - (\dot{x} * y * \dot{\theta}_x) - (\dot{y} * x * \dot{\theta}_x) - (x * y * \ddot{\theta}_x) - (2 * y * \dot{y} * \dot{\theta}_y)] \quad (16)$$

Las ecuaciones encontradas anteriormente describen las aceleraciones lineales de la esfera y las aceleraciones angular de la plataforma.

### B. Modelamiento del sistema en variables de estado

A partir de este punto se realiza el modelamiento del sistema, con el objetivo de poder prescindir del número de entradas y salidas. [12]

Las entradas de control están definidas como:

$$U = [U_A \ U_B]^T$$

Que corresponden a  $\theta_x$  y  $\theta_y$  respectivamente. Se debe realizar un cambio de variable a variables de estado donde:

$$\begin{aligned} x_1 &= x & x_4 &= \theta_x & x_7 &= \theta_y \\ x_2 &= \dot{x} & x_5 &= y & x_8 &= \dot{\theta}_y \\ x_3 &= \theta_x & x_6 &= \dot{y} & & \end{aligned}$$

Sustituyendo estas variables en las ecuaciones (10), (12), (14) y (16), se obtiene las ecuaciones que explican la dinámica del sistema en el espacio de estados, además de deducir que el sistema no es lineal, representando de una forma matricial se tiene:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} F_{x1} & F_{x2} & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5(x_1 * (x_4)^2 + g * \sin(x_3)) & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5(x_5 * (x_8)^2 + g * \sin(x_7)) & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_A \\ U_B \end{bmatrix} \quad (17)$$

Se aplica de esta manera para poder eliminar el acoplamiento existente entre las velocidades angulares, haciendo que la aceleración angular sea aproximadamente igual a cero [13].

### C. Linealización del sistema.

Al ser un sistema no lineal lo que se debe realizar es una linealización respecto al punto de equilibrio. Para poder linealizar hace uso del método de Lyapunov: Sea  $x = 0$  un punto de equilibrio de  $\dot{x} = f(t, x)$  en tiempo igual a cero, donde  $x$  es el vector de estado, y sea  $f$  una función vectorial no lineal continua y diferenciable, para esta condición existe una solución única [14].

Entonces:

- $\dot{x}_1 = x_2 = 0$
- $x_2 = \frac{5}{7}(x_1 * (x_4)^2 + x_5 * x_4 * x_8 + g * \sin(x_3)) = 0$
- $\dot{x}_3 = x_4 = 0$
- $x_4 = 0$
- $x_5 = x_6 = 0$

- $x_6 = \frac{5}{7}(x_5 * (x_8)^2 + x_1 * x_4 * x_8 + g * \sin(x_7)) = 0$
- $x_7 = x_8$
- $x_8 = 0$

Para poder realizar la aproximación lineal a la función en un punto, se utiliza la matriz jacobiana formada por las derivadas parciales.

$$Jf = \begin{pmatrix} \frac{dF_1}{dx_1} & \dots & \frac{dF_1}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{dF_n}{dx_1} & \dots & \frac{dF_n}{dx_n} \end{pmatrix} \quad (18)$$

Remplazando los valores obtenidos en la matriz jacobiana y subdividiéndola para cada eje se puede representar de la siguiente manera:

Aproximación lineal para el eje "X":

$$\begin{bmatrix} \dot{x}_1 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 5/7(g) & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [U_A] \quad (19)$$

Aproximación lineal para el eje "Y":

$$\begin{bmatrix} \dot{x}_5 \\ \dot{\theta} \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 5/7(g) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [U_B] \quad (20)$$

#### D. Función de transferencia.

En el sistema electromecánico los actuadores son servomotores donde su salida será un ángulo de rotación, entonces se debe plantear los ángulos del movimiento de la plataforma como una nueva variable de control.

Si:  $x_3 = \theta_x$  y  $x_7 = \theta_y$

El sistema dinámico puede ser representado en un espacio de estados, especificado a través de un conjunto de entradas, salidas y variables de estado.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

Si se realiza el cambio de variable donde  $x_3 = U_x$  y  $x_7 = U_y$  de esta manera el nuevo espacio de estados será:

Para el eje "X":

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 5/7g \end{bmatrix} U_x \quad (21)$$

Para el eje "Y":

$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 5/7g \end{bmatrix} U_y \quad (22)$$

Si se remplaza la gravedad:

$$\frac{5}{7}g \rightarrow \frac{5}{7} * 9.8 = 7$$

El sistema presenta un doble integrador definido en espacio de estados como:

$$[x(t)] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} U_x \quad (23)$$

Así las matrices de estado A, B, C y D son:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = [1 \quad 0] \quad D = [0] \quad (24)$$

La función de transferencia puede obtenerse mediante el uso de:

$$G(s) = C(sI - A)^{-1} * B + D \quad (25)$$

Remplazando, resolviendo cada término se tiene:

$$G(s) = C(sI - A)^{-1} = [1 \quad 0] \frac{1}{(s^2)} \begin{bmatrix} s & 1 \\ 0 & s \end{bmatrix}$$

$$G(s) = C(sI - A)^{-1} = \frac{1}{(s^2)} [s \quad 1] * \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$G(s) = C(sI - A)^{-1} = \frac{7}{(s^2)} \quad (26)$$

Siendo la ecuación 26 la función de transferencia del sistema ball and plate.

#### E. Sistema de control.

Una vez obtenido el modelo matemático del sistema, se procede al diseño del controlador PID. En este apartado se demuestra los diferentes pasos para obtener el diseño de un controlador idóneo para la planta. La programación se realizó en el ambiente de MATLAB, al ser un entorno que permite simular el sistema, además de contar con funciones que facilitan el proceso de cálculo mediante el uso de sus herramientas.

El controlador PID debe promover una acción de control para los requerimientos del proceso. Existen muchas maneras de sincronizar sus parámetros, pero en el presente trabajo se utilizó el método Ziegler-Nichols, al ser un método confiable y de los más usados.

Para hacer uso de este método se debe seleccionar los valores para las constantes KP, Ti, Td con el fin de obtener un comportamiento de respuesta anhelado ante un error.

TABLA II. SINCRONIZACIÓN DEL PID

Controlador	KP	Ti	Td
<b>P</b>	0.5 * Kcr	$\infty$	0
<b>PI</b>	0.45 * Kcr	$\frac{Pcr}{1.2}$	0
<b>PID</b>	0.6 * Kcr	0.5 * Pcr	0.125 * Pcr

Para poder determinar el valor de Pcr y Kcr se utiliza la Fig. 3, obtenida en Simulink a partir de un sistema realimentado, introduciendo la ecuación de la función de transferencia según la tabla 2.

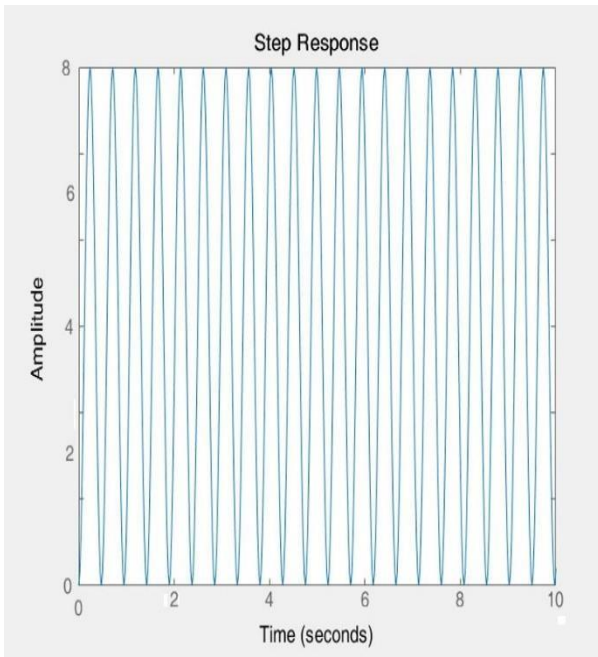


Fig. 3. Respuesta del sistema en Simulink.

Como se ve en la Fig. 3, el valor de Pcr corresponde a 1.190 segundos y para que la respuesta al impulso oscile de manera periódica el valor Kcr es 4.

Haciendo uso de un controlador PID en paralelo, por su fácil implementación y su amplio uso en las industrias se tiene:

$$G_{(s)} = K_p + \frac{K_i}{s} + K_d * s \quad (27)$$

Según la Table 2, se pueden expresar a las siguientes ecuaciones:

$$K_i = \frac{K_p}{T_i} \quad (28)$$

$$K_d = K_p * T_d \quad (29)$$

Desarrollando (28):

$$T_i = 0.5 * P_{cr} = 0.595$$

$$K_i = \frac{2.4}{0.595} = 4.033$$

Desarrollando (29)

$$T_d = 0.125 * P_{cr} = 0.148$$

$$K_d = 2.4 * 0.148 = 0.355$$

A partir de los valores obtenidos en las ecuaciones (28) y (29) se poseen los valores para la ecuación (27):

$$G_{(s)} = 2.4 + \frac{4.033}{s} + 0.355 * s \quad (30)$$

Con la ecuación (3.4), se soluciona la Tabla 2, obteniendo un controlador mediante el método de Ziegler-Nichols.

TABLA III. SINCRONIZACIÓN DE LOS VALORES REALES DEL PID

Controlador	KP	Ti	Td
<b>P</b>	2	$\infty$	0
<b>PI</b>	1.8	0.991	0
<b>PID</b>	2.4	0.595	0.148

<sup>b</sup>. Sincronización de los valores reales del PID.

Se procede a graficar los resultados en la tabla 3, para verificar si los resultados obtenidos son ideales o en su defecto se debe modificar los valores para garantizar un óptimo rendimiento.

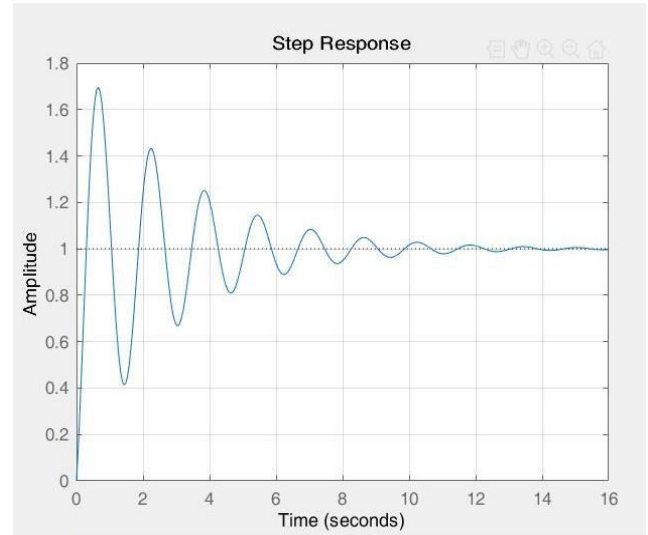


Fig. 4. Respuesta del controlador PID a los valores de la tabla 3.

Como se aprecia en la Fig. 4, los valores obtenidos por el método Ziegler-Nichols cumplen con el objetivo de estabilizar el sistema, pero el tiempo en el que lo hacen es demasiado alto (16 segundos). Además, el porcentaje de sobre pico es muy pronunciado y son muchos los picos existentes dentro de la gráfica, por lo tanto, el resultado no es idóneo.

Con ayuda de MATLAB, se hace uso de la herramienta AUTOTUNING, que permite una sincronización de los parámetros de manera automática, a partir de la función de transferencia, tratando de mejorar así el tiempo de estabilidad del sistema y reducir el porcentaje de sobre pico.



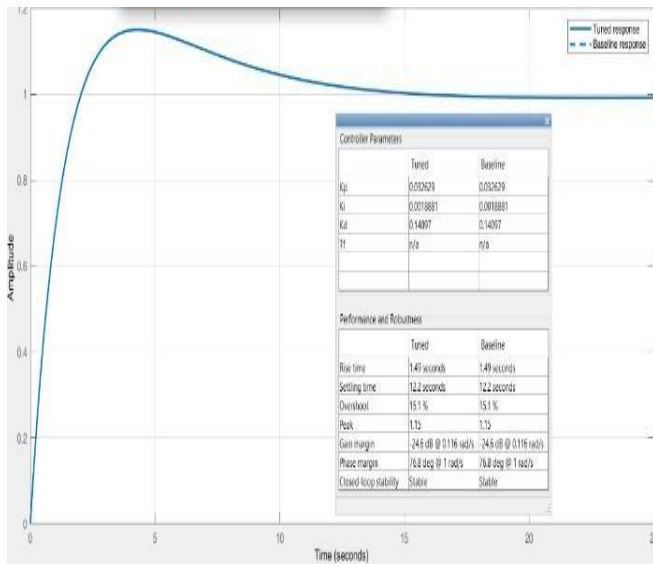


Fig. 5. Respuesta del controlador PID obtenida a partir del AUTOTUNING.

Como se aprecia en la Fig. 5, el porcentaje de sobre pico disminuyó considerablemente, aunque el tiempo de estabilidad aún es demasiado alto. Una ventaja de esta herramienta es que permite probar con diferentes valores y así obtener una respuesta óptima.

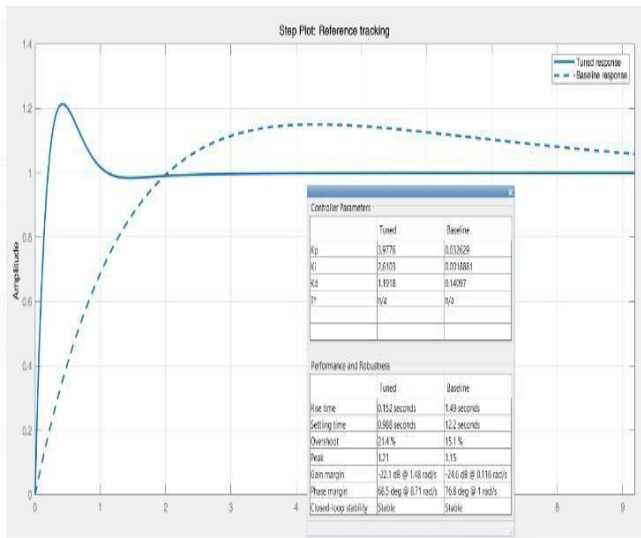


Fig. 6. Respuesta del controlador PID obtenida a partir del AUTOTUNING mejorado.

Como se aprecia en la Fig. 6, aunque el porcentaje de sobre pico aumentó comparado con el de la anterior figura, el tiempo de estabilidad se redujo, pasando de 15 segundos a 2 segundos, obteniendo una estabilidad veloz ante un impulso, siendo valores idóneos para el controlador PID, tabla 4.

TABLA IV. COMPARACIÓN DE LOS MÉTODOS UTILIZADOS

Controlador PID	AUTOTUNING	AUTOTUNING Mejorado	Ziegler-Nichols
$K_p$	0.032	1.279	2.4
$K_i$	0.018	0.130	4.033
$K_d$	0.140	0.881	0.355

<sup>c</sup> Comparación de los métodos utilizados

Mediante el uso de la herramienta Simulink se procede a realizar la simulación del método que entregó el mejor resultado como se aprecia en la Fig.7 y Fig.8.

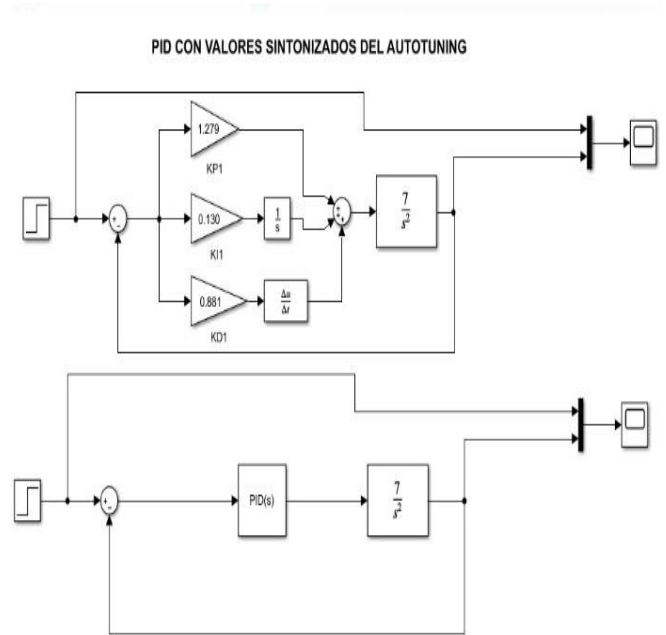


Fig. 7. Controlador PID método AUTOTUNING mejorado.

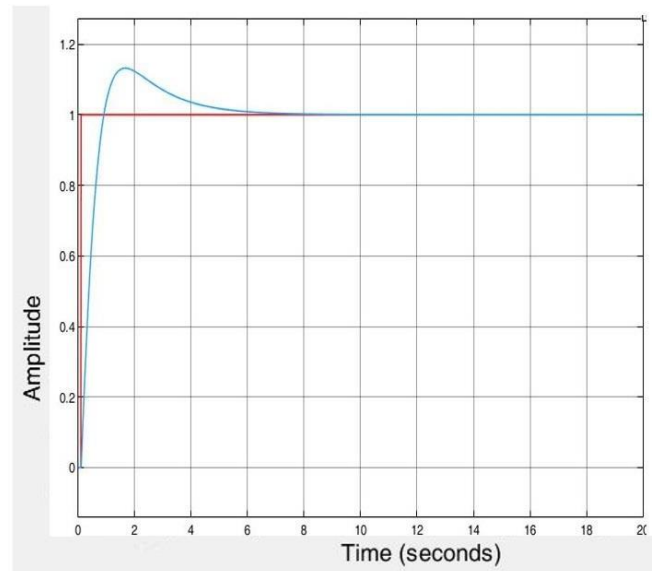


Fig. 8. Respuesta del controlador PID mediante el método AUTOTUNING mejorado.

#### F. Simulación de un modelo tridimensional.

Mediante el uso de la herramienta vrbuid2 (VIRTUAL REALITY BUILD), se crea un entorno de trabajo virtual con el objetivo de analizar el comportamiento del controlador PID ya diseñado, facilitando la planificación y creación de un modelo tridimensional virtual del sistema a ball and plate.

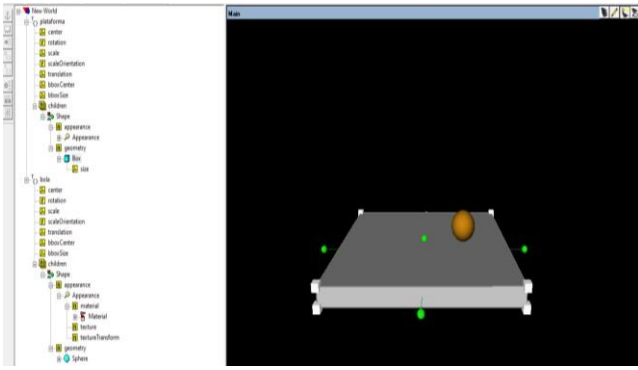


Fig. 9. Sistema ball and plate simulado en vrbuild2.

El sistema permite la selección de elementos básicos Fig.9, posibilitando modificar aspectos como: son su forma, escala, orientación, para el presente se creó una plataforma con 21 centímetros por lado, y una esfera con 4 centímetros de diámetro, se debe tratar de generar un contraste entre la plataforma y la esfera.

La ventaja de usar esta herramienta es la compatibilidad con otros instrumentos dentro del paquete de MATLAB que con solo agregar bloques extras al controlador PID diseñado en Simulink Fig. 7, facilita la creación de un potente y complejo modelo virtual accesible, permitiendo la simulación de una amplia variedad de situaciones y escenarios.

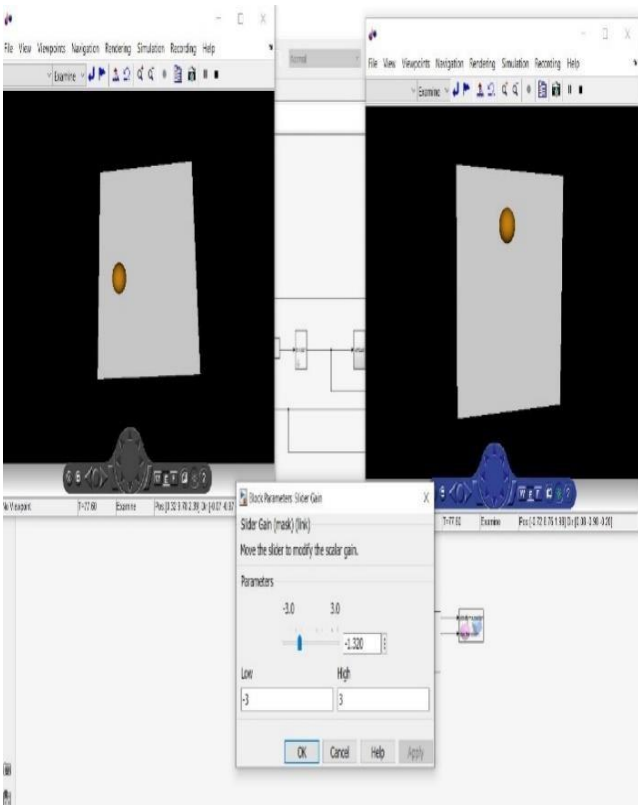


Fig. 10. Entorno virtual del sistema ball and plate

Como se aprecia en la Fig. 10, la esfera se mueve a través de varios puntos de equilibrio manteniéndose estable a lo largo del tiempo, al funcionar ambos ejes de manera conjunta y con iguales características presentan la misma respuesta de salida ante un impulso Fig.11.

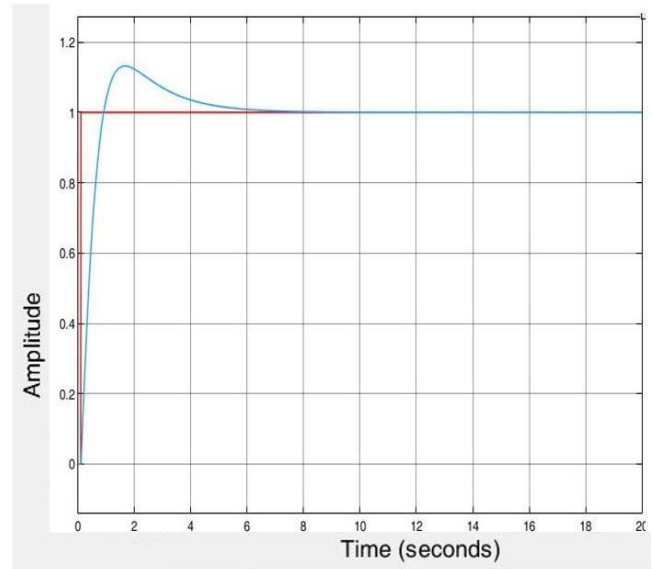


Fig. 11. Respuesta del controlador PID del sistema 3D.

### G. Arquitectura del Diseño.

En el presente apartado se describe el hardware del sistema detallando los componentes que lleva integrados, para garantizar así la funcionalidad del sistema electromecánico.

Para el controlador del sistema se utiliza un Arduino uno, al ser una placa de circuito impreso de un fácil uso, y permitir una programación sencilla por medio del IDE Arduino [15], este componente funciona en modo esclavo del software de Python, para el control de los servomotores.

El sistema electromecánico al poseer 2 grados de libertad requiere de actuadores para el movimiento de la plataforma en los ejes “X” y “Y”, para ello se hace uso de 2 servomotores uno para cada eje, colocados de manera perpendicular el uno respecto al otro, que sean capaces de soportar el peso de la plataforma sin perder el control del sistema.

Los elementos, pieza y ensamblajes para el diseño y construcción del sistema ball and plate, están presentes en la siguiente Fig. 12.

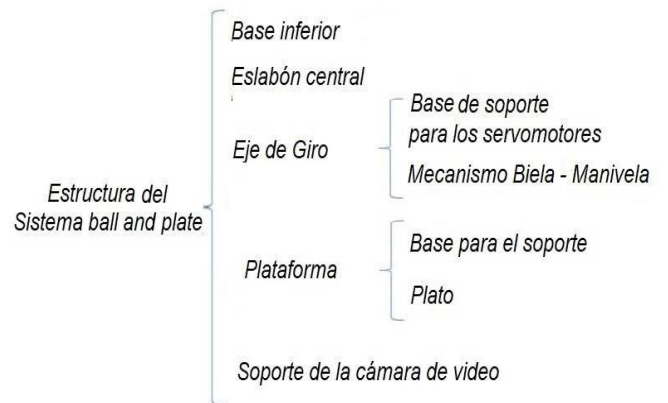


Fig. 12. Estructura del sistema ball and plate.

La base inferior, es una superficie de aluminio adecuada para la distribución de todos los elementos, se necesita que la base será sólida para así evitar ruidos como son vibraciones por el efecto del movimiento de los servomotores.

Eslabón central, es un elemento de hierro que cumple con el objetivo de soportar la mayor cantidad del peso de la plataforma y el de permitir girar su rotación.

Mecanismo biela-manivela, es un componente mecánico cuyo objetivo es transmitir el movimiento de rotación de los servomotores a los ejes de la plataforma, con el menor esfuerzo, con este mecanismo se pretende conseguir la transmisión de un ángulo muy grande en un ángulo pequeño [16], ya que la plataforma debe estar limitado por un ángulo máximo y uno mínimo para cada eje Fig.13.

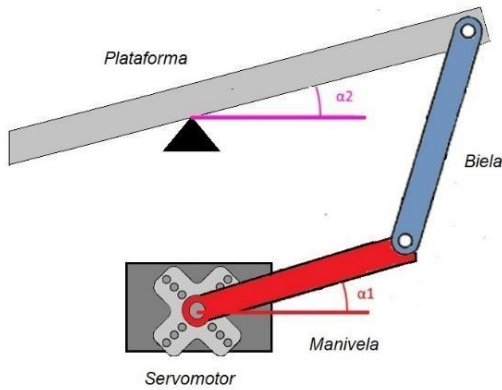


Fig. 13. Mecanismo biela-manivela.

Plataforma, es la superficie por donde la esfera se desplazará para ser controlada por la acción de los servomotores, siendo una superficie cuadrada lisa de un material de acrílico con 3 milímetros de espesor y 21 cm por lado, la plataforma no debe ser exageradamente grande ni pesada, puesto que los actuadores deberán ser capaces de moverla con facilidad.

Soporte de cámara de video, conformada por dos tubos de aluminio unidos en forma de “L”, con una altura de 50 centímetros, posibilita a la cámara web captar la superficie de la plataforma en su totalidad.

El sistema electromecánico ball and plate implementado se aprecia en la siguiente Fig.14

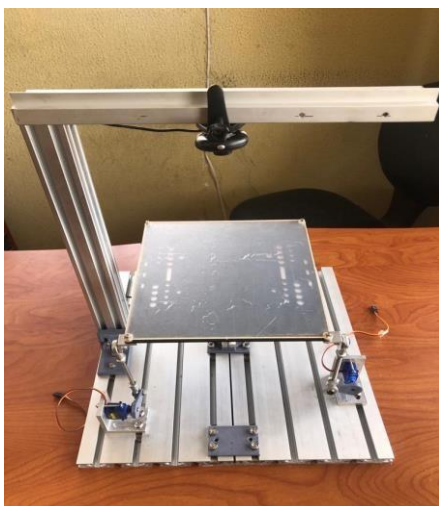


Fig. 14. Sistema ball and plate.

#### H. Procesamiento digital de imágenes.

Consiste en diferentes métodos y técnicas con el objetivo de aislar las características particulares, realizando ciertos

detalles y extrayendo atributos presentes en las imágenes, teniendo en claro las áreas de interés [17].

Dentro de este proyecto ball and plate, la cámara utilizada es capaz de registrar un alto número de fotogramas por segundo, obteniendo así una detección visual en tiempo real del movimiento de la esfera, pudiendo así atravesar las etapas del procesamiento de imágenes presentes en Fig.15.

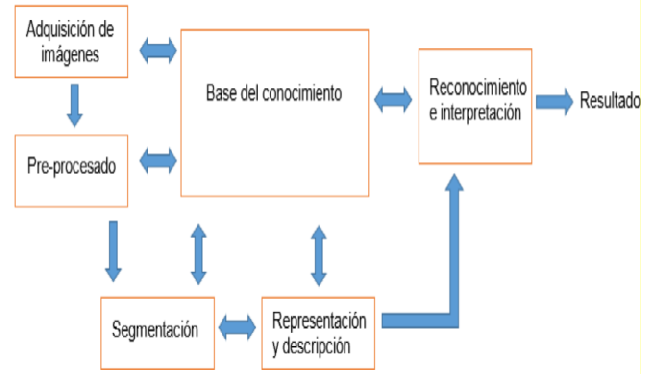


Fig. 15. Etapas fundamentales en el procesamiento de imágenes [18].

Se procede con las etapas del tratamiento de imágenes, obteniendo la visualización de la esfera. Antes de comenzar con el proceso de binarización se necesita conocer el umbral a utilizar, para esto, con la ayuda de un histograma se puede determinar el mejor valor Fig.16.

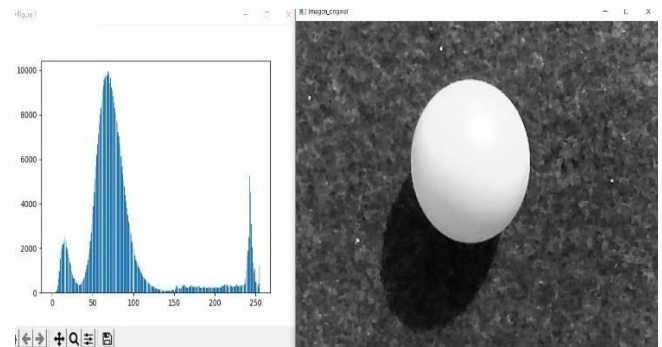


Fig. 16. Histograma de la esfera.

Obteniendo el valor de umbralización de 0.5 se procede a realizar la binarización de la imagen, además de aplicar un filtro para eliminar zonas en blancos que no corresponden a la esfera, y de rellenar los huecos existentes dentro de la misma, alcanzando una sola área que representa la esfera y obteniendo una imagen limpia. De no seguir estos pasos, al trabajar con una imagen no tratada el programa puede presentar fallas al momento de probar el sistema.

A continuación, se procede al reconocimiento de los bordes exteriores de la esfera para que, si en determinado caso se utilice otro tipo de esfera de diferente tamaño, el programa no presente errores. Además del borde exterior se marcará también el centro de la esfera con coordenadas que indique en qué posición de la plataforma se encuentra.

Debido a los reflejos que provoca la luz que incide sobre la plataforma, se producen errores, provocando que el programa no sepa cómo reaccionar ante los cambios de luz, bajando el rendimiento. Para contrarrestar estos errores se colocó un paño de color blanco sobre la plataforma, lo cual no

genera reflejos de luz, permitiendo que el sistema funcione ante cualquier condición.

En este proyecto se utiliza una esfera de color verde para que contraste con el fondo blanco de la plataforma Fig. 17.

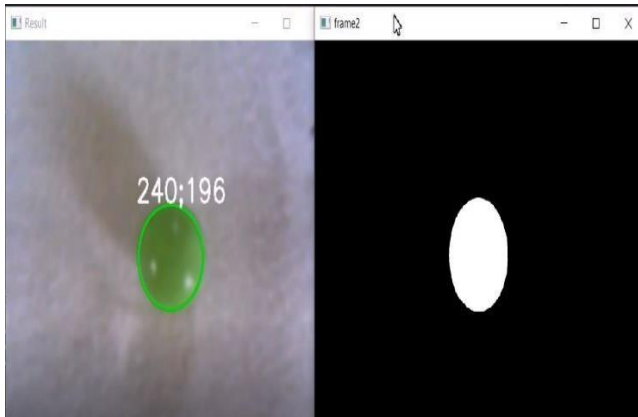


Fig. 17. Obtención del contorno y centro de la esfera.

Se realiza una pequeña modificación en el código para que aparte del color verde, y de la forma esférica, también guarde el valor del área mayor.

### I. Implementación del PID al entorno de programación

Para la implementación del PID se utiliza la ecuación 1, considerando las dos variables: la posición y velocidad de la esfera.

Para determinar la posición de la esfera se realiza una calibración de coordenadas, sabiendo que OpenCV toma las coordenadas de una manera inversa Fig.18. Además, no se puede trabajar con píxeles y se debe realizar una conversión a la unidad de longitud, con esto también se define la posición central setpoint de la plataforma [19].

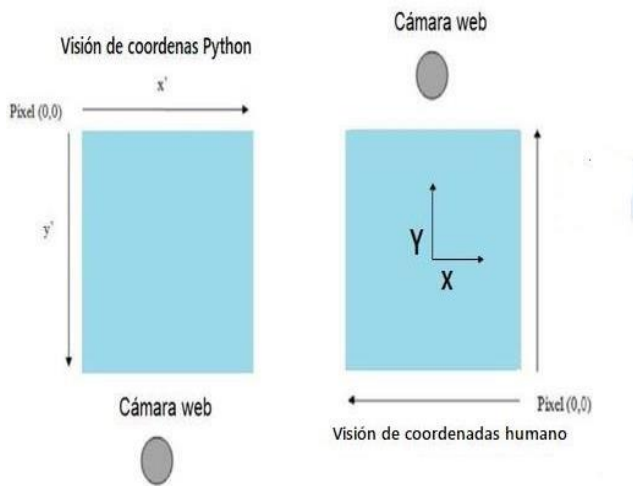


Fig. 18. Visión de coordenadas.

Para la velocidad de la esfera, OpenCV maneja una relación entre la posición de los fotogramas con el tiempo de adquisición del mismo (tiempo de servicio) para obtener la velocidad de la esfera [19].

Sacando provecho de lo anterior, se realiza el cálculo del error, considerando la posición actual de la esfera y la posición deseada. Estos valores se rempazan en la ecuación 1, y se integra el controlador PID al programa.

La comunicación con la placa Arduino se realiza por medio del puerto serial, permitiendo una transmisión y recepción de datos entre el computador y la placa, indicándole a los servomotores en qué posición deben ubicarse, a través de librerías que posee Python.

Adicional a esto se debe limitar el rango de movimiento de los servomotores, otorgando un ángulo máximo y un ángulo mínimo, para evitar daños y reajustes a la plataforma.

Para facilitar el manejo del programa se diseñó una interfaz gráfica (GUI), que permite tener un control sencillo a través del uso del ratón [20].

La GUI creada consta de tres diferentes sliders, uno para cada constante dentro del control PID. Además, contiene 4 botones cada uno con una funcionalidad en específico dentro del sistema Fig. 19.

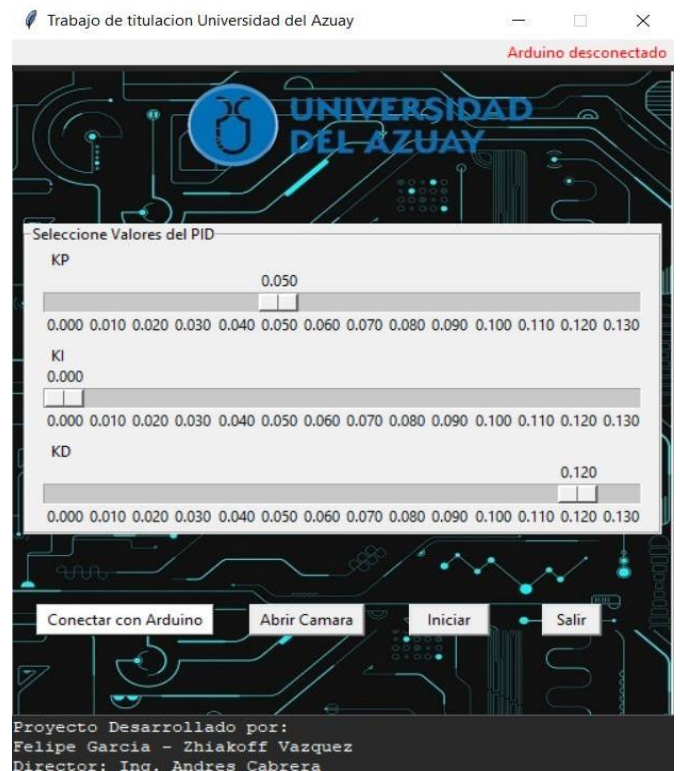


Fig. 19. Interfaz gráfica de usuario.

El primer botón tiene la funcionalidad de conectarse con Arduino, realizando todo el proceso del puerto serial de una manera automática, es el primer botón ya que, si no se realiza una conexión, el programa no realizará ninguna.

El segundo botón permite abrir la cámara y visualizar la esfera dentro la plataforma, incorporando todo el procesamiento de imágenes detallado en este documento.

El tercer botón permite ejecutar el proyecto, realizando el cálculo del error entre la posición de la esfera y el valor del setpoint, comunicándose con el microprocesador para que este indique la acción correctiva que deben tomar los actuadores para así modificar la posición de la esfera, esta acción se repite hasta que la esfera se encuentre dentro de un margen de error previamente programado.

Y el ultimo botón tienen la funcionalidad de cerrar la interfaz gráfica.



### III. RESULTADOS.

#### A. Pruebas de Funcionamiento.

Para poder realizar las pruebas del sistema electromecánico, antes se debe efectuar una conexión eléctrica entre los servomotores y la placa de Arduino, ya que la potencia del microcontrolador no será suficiente para que los dos actuadores trabajen de una manera correcta Fig. 20.

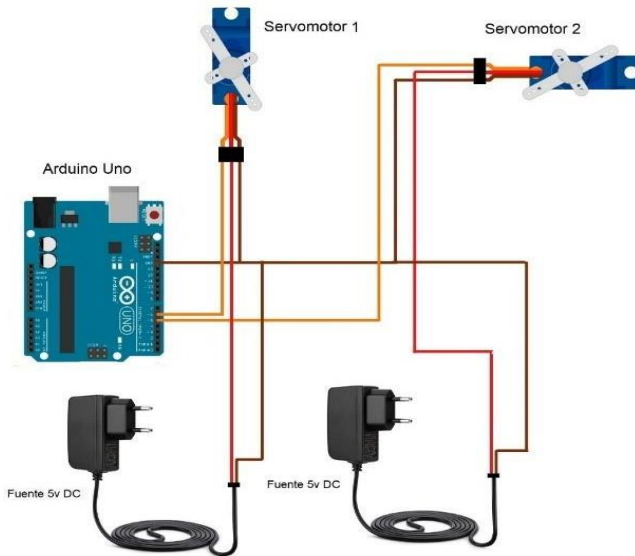


Fig. 20. Esquema de conexión.

Una vez realizados los pasos mencionados, se procedió con las pruebas de funcionamiento, donde el programa busca siempre desplazar a la esfera hacia el centro de la plataforma en el menor tiempo posible y mantenerla en ese punto a lo largo del tiempo durante la ejecución del programa, así como realizar acciones correctivas ante perturbaciones presentadas durante las pruebas.

Los resultados obtenidos del sistema electromecánico se pueden ver en Fig. 21 y Fig. 22.



Fig. 21. Sistema ball and plate en equilibrio.

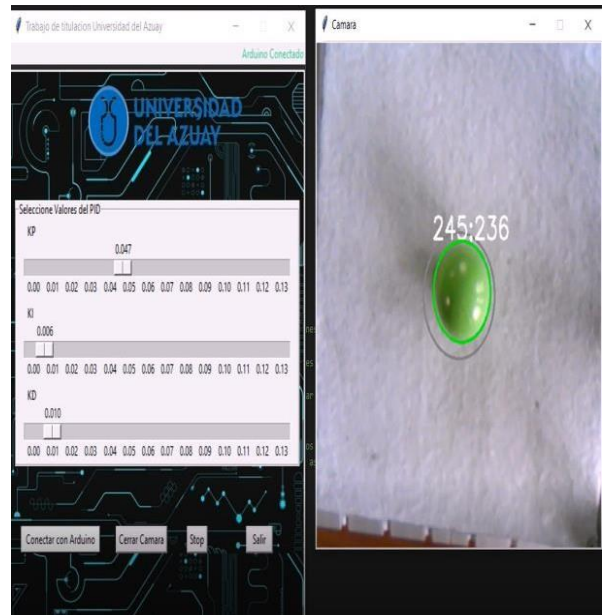


Fig. 22. Esfera ubicada en el setpoint.

Exponiendo los resultados, se puede afirmar que el sistema desarrollado es capaz de detectar una esfera y llevarla hacia el setpoint en el menor tiempo posible, incluso cuando se provoca escenarios imprevistos tales como:

- Generar una perturbación manual golpeando la esfera en una dirección aleatoria.
- Introducir la esfera en la plataforma mediante un lanzamiento con rebote.
- Introducir la esfera en la plataforma con violencia.
- Introducir en la plataforma otra esfera del mismo color, de color diferente y de diferentes tamaños.

En todos estos casos el sistema reacciona de una manera satisfactoria, reponiéndose, corrigiendo la perturbación y el error, volviendo a ubicar la esfera en el setpoint de la plataforma, manteniéndose allí a lo largo del tiempo, hasta que se cierre el programa de manera manual, siendo un sistema estable y confiable.

#### B. Problemas obtenidos durante el proceso.

Como ya se mencionó, la luz juega un papel importante en el control del sistema, afectando en forma directa el funcionamiento. Con la ayuda del paño blanco se pudo reducir los reflejos de luz sobre la plataforma, pero la luz del ambiente afecta al procesamiento de imágenes, ocasionando que se tenga que hacer un reajuste manual de la calibración de los valores del controlador PID.

Mientras se realizaba pruebas técnicas, ocurría muy seguido que los servomotores se trababan por su ubicación dentro de la base. Se buscó una mejor posición para evitar así este problema, pero los servomotores apenas podían mover la plataforma, o tener un buen tiempo de respuesta. Al final se remplazaron por servomotores con una mayor potencia.

Cuando se volvió a probar, éstos se movían de una forma errática provocando daños en la infraestructura, se dio este error ya que inicialmente no se consideró que el ángulo de la

cámara no es el ángulo en el que el servomotor debe posicionarse, para ello se creó un archivo que permite realizar una transformación del ángulo de la cámara (Alpha) en un ángulo para los servomotores (Theta).

La altura de la cámara también fue una variable a considerar ya que, si se ubica muy cerca de la plataforma, esta no capta la esfera cuando se acerca a los bordes dejándola caer, y si, por el contrario, la cámara se la ubicaba a una gran altura, por efecto de la luz, a veces generaba tonos verdes que confundían al programa. Para ello se debe buscar la altura ideal para que ninguna variable externa afecte al sistema.

Se debe tener en cuenta que la memoria del controlador Arduino Uno es limitado y se puede saturar si se envían muchos datos al mismo tiempo. Para ello se debió optimizar el programa enviando sólo la información necesaria por el puerto serial.

### C. Trabajos futuros.

Haciendo como punto de partida este proyecto y centrándose en los sistemas inestables, existen opciones de modificación al incorporar una función de auto calibración para los valores del controlador PID que se adapte a los diferentes entornos de luz presentes en el ambiente.

También se puede tratar de controlar la plataforma de manera remota utilizando una aplicación móvil.

Recordando que este trabajo está orientado en hacer más didáctico el aprendizaje de los estudiantes, sería llamativo realizar el mismo trabajo, pero con otro tipo de sensor como por ejemplo una pantalla táctil o utilizar otro tipo de control, y poder realizar una comparación entre estos, determinando cual es el mejor método.

Para finalizar, con los posibles trabajos a futuro se puede tratar de generar figuras geométricas a partir de la trayectoria de la esfera.

## IV. CONCLUSIONES.

El sistema ball and plate desarrollado posee infinitos puntos naturales de equilibrio, ya que cualquier posición de la esfera será un punto de equilibrio, siempre que la velocidad angular de la plataforma, el ángulo de giro de la misma, y la velocidad lineal de la esfera sean nulos. Si un punto cumple estas condiciones, este es un punto de estabilidad del sistema.

Es fundamental obtener el modelo matemático de la planta, ya que este puede predecir de una manera óptima el comportamiento del sistema, y en base a este se puede diseñar un controlador PID que trabaje de manera eficaz.

Los requisitos y condiciones que se debe considerar para que el sistema funcione de una manera correcta son: la iluminación, siendo constate y regulada, la ubicación de la cámara, estando en una posición que pueda captar la plataforma en su totalidad, la capacidad del microcontrolador utilizado y la potencia de los servomotores para que puedan reaccionar de una manera correcta ante las perturbaciones.

Los valores de las variables del controlador PID obtenidas mediante cálculos o por medio del uso de la herramienta AUTOTUNING no sirvieron en un entorno real, volviendo inestable al sistema, además de causar daños físicos a la infraestructura. Esto es porque en un ambiente real, las

condiciones no son las ideales, así como en la simulación no se considera la fricción, desgaste mecánico, y las perturbaciones del sistema.

Para solucionar este error, se tomó como base los valores obtenidos del AUTOTUNING mejorado y se los dividió para 10, obteniendo así valores de las constantes muy pequeños, pero logrando estabilizar el sistema.

Durante las pruebas siempre hay que tener en cuenta que se debe hacer una sincronización manual de las constantes del PID, ya que siempre se presenta cambios de luz en el ambiente.

Los resultados obtenidos muestran que efectivamente el sistema logra la estabilidad en un tiempo adecuado, aprobando todas las pruebas realizadas, y alcanzando resultados favorables y satisfactorios.

Las conclusiones extraídas dentro del presente trabajo pueden contribuir significativamente a la comprensión y desarrollo de proyectos similares.

## REFERENCIAS.

- [1] V. E. S. Sarabia, Desarrollo de un controlador en espacio de estados para un sistema Ball and Plate realimentado por visión artificial, Quito: Universidad Politécnica Salesiana, 2019.
- [2] G. S. X. Arroyo, Diseño, implementación y comparación de los algoritmos de control SMC, Fuzzy y FSMC, de un sistema de balance Ball & Plate (Esfera & Plato) para el control de posición y seguimiento de camino de una esfera, Quito: Escuela Politécnica Nacional, 2019.
- [3] O. Durán, I. Leal, E. Vidal, J. Salgado, R. Segur, C. Casillo y J. Oliver, «Diseño, construcción y control de una plataforma Bola-Plano.» *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, pp. 7ma edición, 88-95, 2019.
- [4] S. F. A. Sepulveda, Desarrollo de una estrategia de control PID para el sistema Bola y Viga, Ocaña Colombia: Universidad Francisco de Paula Santander Ocaña, 2021.
- [5] C. W. G. Guambo, Estudio comparativo de un control predictivo y PID aplicado a un proceso didáctico de caudal, Latacunga: Universidad Técnica de Cotopaxi, 2021.
- [6] K. Ogata, «Modern Control Engineering (5th Edition),» *Prentice Hall*, p. 905, 2009.
- [7] C. Pineda, Aprendizaje automático y profundo en python: Una mirada hacia la inteligencia artificial, Bogota: Ediciones de la U, 2021.
- [8] G. Viera Mazam, «Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao,» *Universidad de Piura*, 2017.
- [9] S. Leybold, «Ball & Plate System,» 29 Julio 2017. [En línea]. Available: <https://www.leybold-shop.com/ball-and-plate-system-33-240.html>.

- [10] S. Álvarez Serrano, «MODELADO Y DISEÑO DE UN SISTEMA BAILE & PLATE,» *Universidad Politécnica de Catalunya*, 2015.
- [11] A. Mohammadi y J. Ryu, «Neural network-based PID compensation for nonlinear systems: Ball-on-Plate example.,» *International Journal of Dynamics and Control*, pp. 178-188, 2020.
- [12] J. M. Sánchez Manzo, «Diseño y Construcción de un sistema Mecatrónico de Bola y Plato (Ball and Plate),» *Tecnológico Nacional De México, Instituto Tecnológico De Tuxtla Gutiérrez*, 2018.
- [13] E. Kofman, «Modelado y simulación de sistemas dinámicos: Métodos, algoritmos y herramientas,» *Universidad Nacional de Rosario*, 2017.
- [14] M. M. Seron, «Sistemas No Lineales,» *Universidad Nacional de Quilmes*, 2001.
- [15] Datasheet, «Arduino Uno,» *Arduino*, 2022.
- [16] Datasheet, «Micro Servo MG995,» *Micro Servo MG995*, 2022.
- [17] V. Arévalo, J. González y G. Ambrosio, «La librería de visión artificial opencv. aplicación a la docencia e investigación,» *Base Informática*, pp. 40, 61-66., 2004.
- [18] E. Guerra, «Procesamiento de imágenes y análisis espectral de Fourier para la detección de cáncer de piel,» *Universidad de Sonora*, 2017.
- [19] R. C. Gonzales y R. E. Woods, «Digital Image Processing,» *Pearson Education*, 2018.
- [20] J. Grayson, «Python and Tkinter programming.,» *Manning Publications Co. Greenwich.*, 2000.

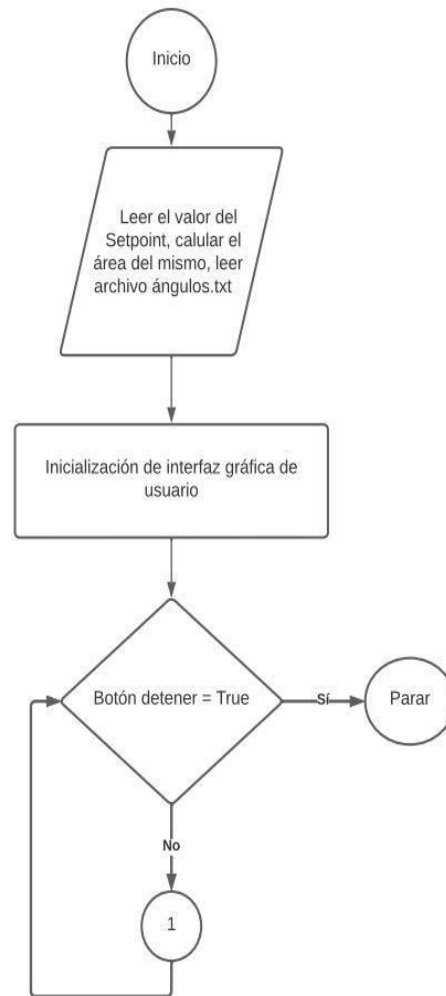


Fig. 23. Diagrama de flujo de la GUI.

A. Anexos.

Para un mejor entendimiento del desarrollo del proyecto, es preciso presentar un diagrama de flujo, Fig. 23, Fig. 24 y Fig. 25.

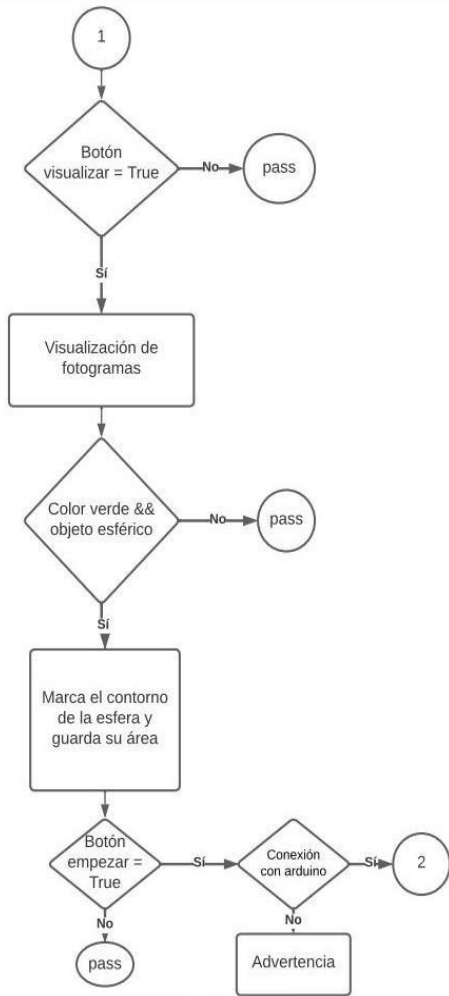


Fig. 24. Diagrama de flujo del procesamiento de imágenes.

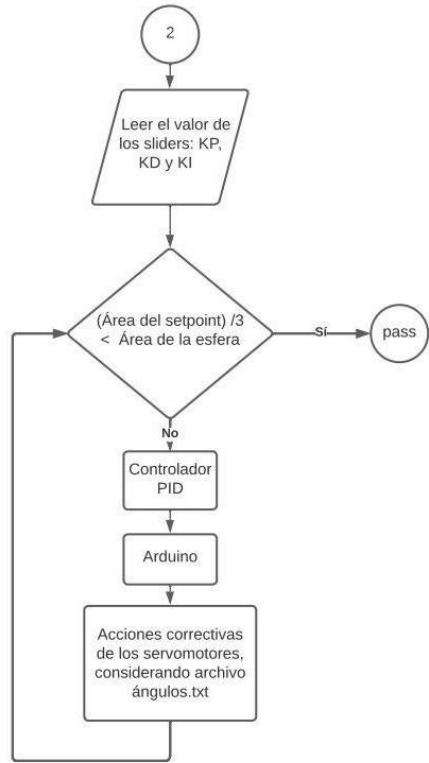


Fig. 25. Diagrama de flujo del control PID.