



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

TEMA: DESARROLLO DE ESTRATEGIAS DE CONTROL EN BASE A  
LÓGICA DIFUSA Y REDES NEURONALES EN UN ROBOT PARALELO  
TIPO STEWART PARA APLICACIÓN DE BOLA EN PLATAFORMA A  
TRAVÉS DE VISIÓN ARTIFICIAL.

AUTOR: VELA TORRES, SANTIAGO ANDRÉS

DIRECTOR: ING. IBARRA JÁCOME, OSWALDO ALEXANDER MGs.

SANGOLQUÍ  
2019



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES


CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “DESARROLLO DE ESTRATEGIAS DE CONTROL EN BASE A LÓGICA DIFUSA Y REDES NEURONALES EN UN ROBOT PARALELO TIPO STEWART PARA APLICACIÓN DE BOLA EN PLATAFORMA A TRAVÉS DE VISIÓN ARTIFICIAL.” fue realizado por el señor *Vela Torres, Santiago Andrés*, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 24 de enero del 2019

Firma:



Ing. Ibarra Jácome, Oswaldo Alexander MGs.

CC: 1719535427



## DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

### CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

#### **AUTORÍA DE RESPONSABILIDAD**

Yo, *Vela Torres, Santiago Andrés* declaro que el contenido, ideas y criterios del trabajo de titulación: DESARROLLO DE ESTRATEGIAS DE CONTROL EN BASE A LÓGICA DIFUSA Y REDES NEURONALES EN UN ROBOT PARALELO TIPO STEWART PARA APLICACIÓN DE BOLA EN PLATAFORMA A TRAVÉS DE VISIÓN ARTIFICIAL es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 24 de enero del 2019

Firma:

---

Vela Torres, Santiago Andrés

CC: 1719461848



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL

**AUTORIZACIÓN**

*Yo, Vela Torres, Santiago Andrés, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: DESARROLLO DE ESTRATEGIAS DE CONTROL EN BASE A LÓGICA DIFUSA Y REDES NEURONALES EN UN ROBOT PARALELO TIPO STEWART PARA APLICACIÓN DE BOLA EN PLATAFORMA A TRAVÉS DE VISIÓN ARTIFICIAL en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.*

Sangolquí, 24 de enero del 2019

Firma:

Vela Torres, Santiago Andrés

CC: 1719461848

# Dedicatoria

A Dios en primer lugar, por haberme dado la sabiduría para elegir esta carrera, a mí mismo, luego de tanto esfuerzo y sacrificio, a mis padres por su apoyo siempre, ¡este logro es para ustedes! Hermanas y demás familiares, novia, amigos y profesores.

*Santiago*

# Agradecimientos

En primer lugar, quiero agradecer a mi tutor el Ing. Alexander Ibarra, ya que sin él este proyecto no se habría llevado a cabo, y no solo por esto si no por su docencia, sus conocimientos impartidos dentro y fuera del aula de clase, ¡gracias! Gracias por ser de esos profesores que disfruta de su trabajo, que se esfuerza por explicar de mejor manera lo que tiene que explicar.

Agradezco a mis padres por ser ese sustento no solo económico si no, de valores, conocimiento, sabiduría, paciencia y sobre todo de amor.

A mis hermanas Criss, Mishel y demás familiares, por darme la mano en cada tropiezo que he tenido, ustedes siempre han sabido apoyarme y eso lo agradezco.

A mi novia Gissel por estar siempre allí conmigo y para mí, porque el mundo es más fácil cuando se va de a dos.

A mi gran amigo Diego por enseñarme lo que es ser amigo de verdad, y estar siempre pendiente en cada momento de esta etapa de mi vida.

A mis amigos de universidad y demás buenos profesores, por cada momento compartido en este largo caminar, dentro y fuera de la universidad.

A mis jefes Ale, Pablo y Santi, por esa paciencia con mi tiempo, por ese apoyo en mis proyectos y por ese conocimiento impartido por todo este tiempo.

A todos ustedes, muchas gracias.

## ÍNDICE DE CONTENIDO

ÍNDICE DE CONTENIDO .....	vi
ÍNDICE DE TABLAS .....	ix
ÍNDICE DE FIGURAS .....	x
RESUMEN .....	xiv
ABSTRACT .....	xv
CAPÍTULO I.....	1
GENERALIDADES.....	1
1.1 Antecedentes.....	1
1.2 Justificación e Importancia.....	3
1.3 Definición del Proyecto. ....	6
1.4 Alcance. ....	7
1.5 Objetivos.....	8
1.5.1 Objetivo General. ....	8
1.5.2 Objetivos Específicos.....	8
CAPÍTULO II .....	10
FUNDAMENTACIÓN TEÓRICA.....	10
2.1 Robots Paralelos. ....	11
2.1.1 Plataforma Stewart .....	12
2.1.2 Aplicaciones .....	13
2.2 Identificación de sistemas.....	14
2.2.1 Toma de datos .....	14
2.2.2 Elección del modelo .....	15
2.2.3 Elección de la estructura.....	16
2.2.4 Elección de algoritmo que resuelva la estructura .....	17
2.3 Control Inteligente.....	18

2.3.1 Control difuso.....	19
2.3.2 Redes neuronales.....	26
2.3.3 Control por modelo de referencia neuronal.....	32
2.4 Procesamiento Digital de Imagen.....	33
2.4.1 Digitalizado de imagen.....	33
2.4.2 Procesamiento de imagen.....	34
2.4.3 Obtención de Resultados.....	36
2.5 Estado del arte de aplicaciones “bola en plataforma”. .....	39
CAPÍTULO III .....	42
DESARROLLO DE ESTRATEGIAS DE CONTROL.....	42
3.1 Descripción de la Planta. ....	42
3.1.1 Especificaciones técnicas .....	43
3.1.2 Exactitud de la planta .....	45
3.1.3 Modificaciones a la estructura de la planta .....	45
3.1.4 Cinemática Inversa .....	48
3.2 Estrategia de detección de objeto por medio de visión artificial. ....	53
3.2.1 Filtro Ruido .....	54
3.2.2 Conversión a HSV.....	54
3.2.3 Filtro Rango de Colores .....	55
3.2.4 Conversión a Escala de Grises .....	58
3.2.5 Detección Contorno.....	59
3.2.6 Obtención Posición de la Bola .....	59
3.2.7 Bola inexistente .....	60
3.2.8 Filtro de Kalman.....	61
3.2.9 Detección Láser Rojo como Punto de referencia .....	61
3.3 Control basado en lógica difusa.....	63



3.3.1 Entradas y Salidas .....	63
3.3.2 Conjuntos difusos y funciones de pertenencia .....	64
3.3.3 Reglas difusas.....	69
3.4 Control basado en redes neuronales. ....	69
3.4.1 Obtención de datos .....	69
3.4.2 Modelo dinámico lineal de auto regresión con variables exógenas ARX.....	71
3.4.3 Modelo dinámico no lineal de auto regresión con variables exógenas NARX.....	72
3.4.4 Estrategia de control por Modelo de Referencia .....	75
CAPÍTULO IV .....	78
PRUEBAS Y ANÁLISIS DE RESULTADOS .....	78
4.1 Exactitud de la planta .....	78
4.2 Algoritmo de Visión .....	79
4.3 Controlador Difuso .....	81
4.3.1 Control difuso con punto de referencia fijo en el centro .....	84
4.3.2 Control difuso con punto de referencia variable por laser .....	90
4.4 Controlador Neuronal .....	94
4.4.1 Obtención de datos .....	94
4.4.2 Identificación de la planta .....	96
4.4.2.1 Modelo ARX .....	96
4.4.2.2 Modelo NARX .....	97
4.4.3 Controlador neuronal por modelo de referencia.....	99
CAPÍTULO V .....	103
CONCLUSIONES Y RECOMENDACIONES .....	103
5.1 Conclusiones.....	103
5.2 Recomendaciones .....	104
Bibliografía.....	106

## ÍNDICE DE TABLAS

<b>Tabla 1</b>	Ventajas y desventajas robots paralelos y seriales .....	12
<b>Tabla 2</b>	Estructura modelos lineales.....	16
<b>Tabla 3</b>	Estructura modelos no lineales.....	17
<b>Tabla 4</b>	Función lógica AND .....	30
<b>Tabla 5</b>	Especificaciones técnicas .....	43
<b>Tabla 6</b>	Conjunto Difuso Distancia.....	65
<b>Tabla 7</b>	Conjunto Difuso Velocidad.....	66
<b>Tabla 8</b>	Conjunto Difuso Ángulos.....	67
<b>Tabla 9</b>	Variación Conjunto Difuso Ángulos.....	68
<b>Tabla 10</b>	Reglas difusas.....	69
<b>Tabla 11</b>	Parámetros para experimento de toma de datos .....	71
<b>Tabla 12</b>	Pruebas Exactitud ISO9283 .....	78
<b>Tabla 13</b>	Resultados pruebas exactitud .....	78
<b>Tabla 14</b>	Datos obtenidos modelo planta .....	96
<b>Tabla 15</b>	Identificación ARX resultados .....	96
<b>Tabla 16</b>	Identificación NARX resultados .....	97
<b>Tabla 17</b>	Entrenamiento controlador.....	99

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Cinemática de un robot manipulador a) serial b) paralela .....	11
<b>Figura 2</b> Plataforma Stewart 6 GDL .....	12
<b>Figura 3</b> Robot paralelo en línea de producción de alimentos .....	14
<b>Figura 4</b> Lazo de control difuso .....	20
<b>Figura 5</b> Funciones de activación a) triangular b) trapezoidal .....	22
<b>Figura 6</b> Funciones de pertenencia – entrada .....	22
<b>Figura 7</b> Funciones de pertenencia - salida .....	23
<b>Figura 8</b> Entrada 10 pixeles.....	23
<b>Figura 9</b> Entrada 70 pixeles.....	23
<b>Figura 10</b> centroide $z=0.75$ .....	25
<b>Figura 11</b> centroide $GS= 0.33$ y $GF=0.25$ .....	25
<b>Figura 12</b> Partes de neurona a) humana b) artificial .....	27
<b>Figura 13</b> Estructura perceptrón.....	29
<b>Figura 14</b> Perceptrón para función AND .....	30
<b>Figura 15</b> AND caso 1.....	31
<b>Figura 16</b> AND caso 2.....	32
<b>Figura 17</b> Control por modelo de referencia .....	32
<b>Figura 18</b> Espacios de color de izquierda a Derecha RGB, HSV, HSI.....	36
<b>Figura 19</b> Plataforma con a) visión b) plataforma resistiva .....	39
<b>Figura 20</b> Simulador de Vuelo FUM 2400Kg.....	40
<b>Figura 21</b> Diseño digital Robot Stewart.....	42
<b>Figura 22</b> Plataforma anterior - actual.....	46
<b>Figura 23</b> Balanza digital, eslabones actuales y antiguos .....	47

<b>Figura 24</b> Brazos servomotores actuales-anteriores.....	47
<b>Figura 25</b> Sistema de Referencia.....	48
<b>Figura 26</b> Sistema de Referencia Servomotores .....	50
<b>Figura 27</b> Algoritmo cinemática .....	52
<b>Figura 28</b> Algoritmo visión artificial .....	53
<b>Figura 29</b> Filtro Gaussiano a) sin filtro b) con filtro.....	54
<b>Figura 30</b> Transformación al espacio HSV .....	54
<b>Figura 31</b> Algoritmo rango de colores .....	55
<b>Figura 32</b> Rango de colores para Verde y Azul .....	56
<b>Figura 33</b> Rango de Colores para Tomate.....	57
<b>Figura 34</b> Función In Range de OpenCV.....	57
<b>Figura 35</b> Función Bitwise_and perteneciente a OpenCV .....	58
<b>Figura 36</b> Conversión a Escala de Grises.....	58
<b>Figura 37</b> Detección de contorno de la bola.....	59
<b>Figura 38</b> Obtención posición de la bola.....	60
<b>Figura 39</b> Bola inexistente en plataforma .....	60
<b>Figura 40</b> Filtro de Kalman .....	61
<b>Figura 41</b> Algoritmo detección láser.....	62
<b>Figura 42</b> Detección láser rojo .....	63
<b>Figura 43</b> Funciones de pertenencia Distancia.....	65
<b>Figura 44</b> Funciones de pertenencia Velocidad .....	66
<b>Figura 45</b> Funciones de pertenencia Ángulos .....	67
<b>Figura 46</b> Variación funciones de pertenencia Ángulos .....	68
<b>Figura 47</b> Algoritmo toma de datos .....	70

<b>Figura 48</b> Estructura ARX de Matlab .....	71
<b>Figura 49</b> Ingreso datos entrada salida para identificación .....	72
<b>Figura 50</b> Identificación ARX.....	72
<b>Figura 51</b> Estructura NARX.....	73
<b>Figura 52</b> Neural Net Time Series Toolbox de Matlab.....	74
<b>Figura 53</b> Número de neuronas y regresores.....	74
<b>Figura 54</b> Entrenamiento de red neuronal .....	75
<b>Figura 55</b> Estructura controlador neuronal .....	76
<b>Figura 56</b> Entrenamiento de controlador + planta.....	77
<b>Figura 57</b> Posición bola en x.....	80
<b>Figura 58</b> Posición bola en y.....	80
<b>Figura 59</b> Máximo error punto de referencia en centro .....	82
<b>Figura 60</b> Máximo error punto de referencia variable .....	83
<b>Figura 61</b> Error en estado estable.....	84
<b>Figura 62</b> Prueba 1 esquina 1 .....	84
<b>Figura 63</b> Prueba 1 esquina 2.....	85
<b>Figura 64</b> Prueba 1 esquina 3 .....	85
<b>Figura 65</b> Prueba 2 punto 1 .....	86
<b>Figura 66</b> Prueba 2 punto 2 .....	87
<b>Figura 67</b> Prueba 3 punto 1 .....	88
<b>Figura 68</b> Prueba 3 punto 2 .....	88
<b>Figura 69</b> Prueba 3 punto 3 .....	89
<b>Figura 70</b> Prueba 4 .....	90
<b>Figura 71</b> Prueba Punto de referencia variable .....	91

<b>Figura 72</b> Prueba Punto de referencia variable Error.....	92
<b>Figura 73</b> Ingreso coordenadas .....	93
<b>Figura 74</b> Resultado ingreso coordenadas.....	93
<b>Figura 75</b> Datos roll .....	94
<b>Figura 76</b> Datos X.....	94
<b>Figura 77</b> Datos Pitch.....	95
<b>Figura 78</b> Datos de Y .....	95
<b>Figura 79</b> Desempeño red neuronal planta.....	98
<b>Figura 80</b> Respuesta de red neuronal en tiempo .....	98
<b>Figura 81</b> Desempeño controlador .....	99
<b>Figura 82</b> Respuesta cambios de punto de referencia planta controlada.....	100
<b>Figura 83</b> Sección 1 prueba control RNA .....	101
<b>Figura 84</b> Sección 2 prueba control RNA .....	101
<b>Figura 85</b> Sección 3 prueba control RNA .....	102

## **RESUMEN**

Los sistemas de equilibrio han sido siempre motivo de estudio en la teoría de control. En el presente trabajo se han desarrollado estrategias de control inteligentes para controlar la posición de una bola sobre una plataforma en un robot paralelo tipo Stewart de 6 grados de libertad por medio de visión artificial. Partiendo de modificaciones realizadas a la estructura de la planta para mejorar su desempeño y de mejoras realizadas al método de procesamiento de imagen, se implementó una estrategia de control en base a lógica difusa, la cual permite equilibrar la bola no solo en el centro si no en cualquier punto de la plataforma, logrando un tiempo de equilibrio de la bola de 6s como el más bajo registrado. Se diseñó además una estrategia de control neuronal en base a un controlador por modelo de referencia, mediante la adecuada identificación de la planta por medio de una red neuronal NARX.

### **PALABRAS CLAVE:**

- **ROBOT PARALELO**
- **CONTROL INTELIGENTE**
- **VISIÓN ARTIFICIAL**
- **CONTROL DIFUSO**
- **REDES NEURONALES**

## **ABSTRACT**

Balance systems have always been important in control theory. In the present work, intelligent control strategies have been developed to control the position of a ball on a platform in a Stewart-type robot with 6 degrees of freedom by artificial vision. First, some structure modifications of the plant were implemented for improving the plant response and an algorithm to upgrade the image processing capacity. Then, a control strategy was implemented based on fuzzy logic, which permits to balance the ball not only in the center and in any point of the platform, achieving a balance time of 6s ball as the lowest recorded. A neural control strategy was also designed based on a controller by reference model, through the appropriate identification of the plant by means of a NARX neural network.

### **KEYWORDS:**

- PARALLEL ROBOT**
- INTELLIGENT CONTROL**
- ARTIFICIAL VISION**
- FUZZY CONTROL**
- NEURAL NETWORKS**



# CAPÍTULO I

## GENERALIDADES

### 1.1 Antecedentes.

En la Universidad de las Fuerzas Armadas – ESPE, en el año 2017 fue presentado el trabajo de titulación, previo a la obtención del título de Ingeniero en Mecatrónica del Departamento de Ciencias de la Energía y Mecánica con el Tema: “Diseño, construcción e implementación de una plataforma tipo Stewart para control de una aplicación bola en plataforma a través de visión artificial” por parte de los alumnos: Bueno Sánchez Cristian Alejandro y Maldonado Soria Alejandro Emmanuel, bajo la dirección del Ing. Loza Matovelle David César MSc.

En este trabajo se desarrolló el diseño y la construcción de la plataforma Stewart, mediante el uso de software CAD/CAE, procesos de manufactura, corte láser e impresión 3D. Además, se dio solución a la cinemática del robot paralelo de 6 grados de libertad (Alejandro & Emmanuel, 2017). Se utilizó un control retroalimentado por medio de visión artificial, el cual permite saber la posición de la bola mediante el procesamiento digital de la misma.

Parte de las conclusiones de este trabajo de titulación fueron:

Al lograr el control de la plataforma y la aplicación de dos técnicas de control se corrobora el buen funcionamiento del sistema. Además, se demuestra que puede ser controlado mediante diversas técnicas ya que existe evidencia de esto en investigaciones anteriores. Aquello da apertura a la investigación en el campo de control mediante la manipulación del sistema (Alejandro & Emmanuel, 2017).

Mientras que una de las recomendaciones en el trabajo fueron:

Existen ciertos aspectos en los cuales se pueden utilizar modificaciones para optimizar su funcionamiento. Por ejemplo: implementar otra forma de detectar la posición de la bola podría ser muy importante. Se sugiere probar otras técnicas de control, capaces de aprovechar y optimizar las capacidades y prestaciones de la plataforma (Alejandro & Emmanuel, 2017).

Son varias las investigaciones realizadas a lo largo de los años desde que la plataforma Stewart fue inventada, conforme ha ido avanzando la tecnología en sus diferentes aplicaciones, la mayoría de ellas orientadas a estudiar técnicas de control clásicas, modernas o inteligentes.

En Julio del 2017, el artículo “Control de equilibrio con realimentación visual de un manipulador robótico para un sistema de bola en plataforma” fue presentado. En este artículo, se describe un sistema de control en equilibrio de bola en plataforma robotizada guiada por visión, que consiste en un manipulador de robot (actuador), un sistema de bola en plataforma (planta) y un sistema de visión artificial (retroalimentación). El sistema de visión artificial retroalimenta los datos de posición de la bola a una velocidad de 50 cuadros por segundo. En base a los datos de retroalimentación, el efector final de un manipulador robótico manipula para controlar la posición de la pelota al maniobrar el ángulo de inclinación del sistema de motores y engranes. El sistema de control general se implementa con dos chips FPGA, uno para el procesamiento de la visión artificial, y uno para los controladores PID de los motores (Shih, Hsu, & Chang, 2017).

Sin embargo, la visión artificial no es la única estructura que presenta estos robots/plataforma Stewart. Varios diseños han sido llevados a cabo mediante pantallas resistivas o *touch*, en las que se obtiene la posición y velocidad de la bola directamente de la

plataforma, y con ello se ejecutan las acciones de control para posicionar la bola en el punto deseado.

En el 2015, el Instituto Politécnico Rensselaer, dentro del Departamento de Ingeniería Mecánica, Ingeniería Aeronáutica y Mecánica de Estados Unidos, publicó el artículo “*Diseño mecatrónico de una bola en plataforma*”. En este artículo se discute el desarrollo de un sistema bola en plataforma para un robot Stewart basado en principios de diseño mecatrónico. Se presentó un sistema dinámico completo que incluye diseño de hardware, selección de sensor y actuador, modelado de sistema, parámetro de identificación, diseño del controlador y prueba experimental. Se utilizó un sensor *touch pad*, para la obtención de la posición de la bola en plataforma. Una vez construido y probado se concluyó que puede ser utilizado como un excelente banco de pruebas para verificar diversas teorías de control. Aunque los controladores basados en el modelo lineal funciona extremadamente bien, será interesante aplicar los principios de no linealidad al control y busca cualquier otra mejora en el rendimiento de sistema (Craig, 2015).

## **1.2 Justificación e Importancia.**

En noviembre del 2017 El Gobierno Ecuatoriano estableció el Plan Nacional de Desarrollo 2017-2021, donde se pretende garantizar el acceso progresivo de las personas a sus derechos durante todo el ciclo de vida, a través de políticas públicas y programas para el desarrollo social y humano. Este documento, además, orienta las acciones del Gobierno actual para los próximos cuatro años.

El Plan Nacional de Desarrollo 2017-2021 se organiza en 3 Ejes y 9 Objetivos. El objetivo 5 trata sobre impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria, y en este contexto se menciona:

Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades (SENPLADES, 2017).

El presente proyecto, se justifica con el objetivo 5 del Plan de Desarrollo 2017-2021, al ser un proyecto de investigación y desarrollo, de estrategias de control aplicadas a robótica; tema que pretende promover la transferencia tecnológica dentro de la universidad.

La plataforma/robot Stewart de 6 grados de libertad ofrece un diseño de prueba utilizado para investigación y educación, en los campos de ingeniería de modelado matemático, robótica, control de variables múltiples, control no lineal, clásico y moderno (S. Christensen, 2014). La plataforma no está destinada a una tarea específica, sus aplicaciones son totalmente diferentes unas de otras.

Una de las posibles aplicaciones, son los sistemas de equilibrio. Estos son uno de los problemas más desafiantes en el campo de control. El reto de balancear ha sido motivo de constante investigación y desarrollo para aplicaciones de robótica, transporte, simulación, etc (Awtar et al., 2002). Es por esto que estos sistemas han brindado la oportunidad para desarrollar nuevas técnicas y algoritmos de control.

El sistema de equilibrio bola en plataforma, basado en el robot Stewart, es un sistema que puede usarse para verificar la eficiencia de muchos sistemas de control y lograr el sistema de estabilización. También puede ser usado al reemplazar la bola por otro objeto cualquiera en aplicaciones robóticas mecatrónicas o industriales (Bdoor, Ismail, Roman, & Hendawi, 2016). La importancia de este robot para la universidad a nivel de pregrado, está en la relación que

tiene principalmente con las materias de Sistemas de Control, Control Digital, Control Inteligente y Robótica entre otras, en donde se estudian diferentes métodos de control que pueden ser probados, evaluados y comparados en la misma planta.

El control ha desempeñado un papel vital en el avance de la ingeniería y la ciencia. Este se ha convertido en una parte importante e integral en los sistemas de vehículos espaciales, en los sistemas robóticos, en los procesos modernos de fabricación y en cualquier operación industrial que requiera el control de temperatura, presión, humedad, flujo, nivel, etc. Es deseable que la mayoría de los ingenieros y científicos estén familiarizados con la teoría y la práctica del control (Ogata, 2013).

Este gran avance en teorías de control ha permitido desarrollar aplicaciones en distintas áreas de la robótica, entre ellas:

- Manufactura: tareas de precisión de posicionamiento y ensamblaje de elementos con alta rapidez y eficiencia.
- Medicina: sistemas adaptados a la cirugía oftalmológica, laparoscópica, que permiten a los cirujanos realizar tareas de precisión y rapidez con escaso error.
- Laboratorios: aplicados a tratamientos de pruebas químicas y muestras de patógenos.
- CNC: sistemas continúan en desarrollo, los resultados reflejan acabados de excelente calidad.
- Militares: simuladores de vehículos tripulados, jets de combate o tanques de asalto, entrenamiento de pilotos.
- Espacial: exploración espacial y simulación de vuelo.
- Educación: estudio de comportamiento de mecanismos complejos usados en aviones, motores, maquinaria y también en la industria cinematográfica (Nikolas, 2016).

La Universidad de las Fuerzas Armadas – ESPE, ofrece la maestría en Ciencias Militares con mención en Estrategia Aeroespacial. Las aplicaciones de la plataforma Stewart en el campo de simuladores de vuelo y militares están en continuo desarrollo, y se puede decir que es el campo donde más se la ha utilizado desde su invención. Es por ello que este proyecto beneficia también a posgrado, al ofrecer una planta con las prestaciones adecuadas para probar y desarrollar técnicas de control avanzadas y complejas en los campos mencionados.

### **1.3 Definición del Proyecto.**

En 1965, Stewart propuso el uso de una estructura paralela de seis grados de libertad para generar movimiento en el espacio, como la solución más práctica en la construcción de simuladores de vuelo. La plataforma Stewart es un diseño de control de posición y movimiento. A partir de este diseño, se han realizado aplicaciones en varias ramas de la industria, tales como la aeronáutica, automotriz, marina y de maquinaria industrial (González Álvarez & Reinoso Mendoza, 2011).

En el presente proyecto se pretende desarrollar estrategias de control en base a lógica difusa y redes neuronales, para la plataforma Stewart de 6 grados de libertad en una aplicación llamada “bola en plataforma” o más conocida en inglés como “*ball on plate*”. Este sistema de equilibrio no lineal, debe posicionar la bola en un punto definido o punto de referencia, en el menor tiempo posible.

Mediante el procesamiento de imágenes, que posibilita la cámara ubicada en la parte superior de la plataforma como se observa en la figura 1, se obtiene la posición actual de la bola, señal que retroalimenta al sistema para que este tome las acciones necesarias para llevar la bola al punto definido.

Para esto, técnicas de control clásicas se han desarrollado, mediante PI o PID. Sin embargo, es objeto de estudio para personas que desean desarrollar nuevas técnicas de control moderno o inteligentes, debido a la no linealidad del sistema, y las prestaciones que este ofrece por su estructura y aplicaciones en la vida real.

#### **1.4 Alcance.**

En el presente trabajo, se estudiaron estrategias de control en base a lógica difusa y redes neuronales, las cuales ofrecen una posible solución al problema bola en plataforma.

La lógica difusa, consiste en la toma de decisiones asociadas a conceptos que pueden tomar cualquier valor de veracidad dentro de un margen establecido (Vidal, 2014). Difuso es sinónimo de borroso, es por ello que permite tratar información imprecisa y controlar procesos en los que no se conoce claramente el modelo. Una de las mayores complicaciones del robot paralelo Stewart es la obtención del modelo dinámico, por ello esta estrategia brinda una solución al control de la misma.

Por otro lado, las redes neuronales pretenden imitar el comportamiento del cerebro humano, requiriendo una etapa de entrenamiento para su posterior implementación en el sistema (Vidal, 2012). Esta estrategia de control es ideal para modelos que no cambian, y si bien el modelo de la plataforma Stewart es difícil de obtener, este no cambia y por ello las redes neuronales ofrecen una solución.

Posterior a ello, se realizaron simulaciones para probar la respuesta de cada algoritmo en la plataforma Stewart. Se analizará el desempeño de cada una de ellas para ser implementadas en el robot.

Cabe recalcar, que el algoritmo para obtener la posición de la bola mediante visión artificial está resuelto, sin embargo, se investigaron nuevos métodos de procesamiento de imágenes, los cuales brinden mejores resultados en la obtención de la posición de la bola.

En el artículo “Sistema de control bola en plataforma basada en lógica difusa en tiempo real mediante procesamiento de imágenes”, se obtuvieron 20 pixeles en error de estado estacionario y un tiempo de establecimiento de 1.9 segundos. Este artículo sirve como punto de referencia para diseñar las estrategias de control y comparar los resultados obtenidos en la plataforma.

## **1.5 Objetivos.**

### **1.5.1 Objetivo General.**

Desarrollar estrategias de control en base a lógica difusa y redes neuronales en un robot paralelo tipo Stewart para aplicación de bola en plataforma a través de visión artificial.

### **1.5.2 Objetivos Específicos.**

- Consultar sobre estrategias de control en base a lógica difusa y redes neuronales, mediante artículos científicos, libros, para el desarrollo de las mismas aplicadas a la plataforma Stewart.
- Optimizar el algoritmo de la cámara, mediante técnicas de procesamiento digital de imágenes, para la detección de la posición de la bola en plataforma.
- Controlar la velocidad y ángulo de los servomotores con exactitud, mediante drivers de alto rendimiento, para tener una adecuada respuesta en el posicionamiento y orientación de la plataforma.



- Diseñar un algoritmo de control, que permita el cambio del punto de referencia en la posición de la bola en plataforma, mediante técnicas de control inteligente, para aprovechar las capacidades y prestaciones de la plataforma.

## CAPÍTULO II

### FUNDAMENTACIÓN TEÓRICA

Durante siglos, el ser humano ha buscado la manera de facilitar y optimizar el trabajo que debe realizar, tener herramientas que le permitan aprender, descubrir, innovar. En 1920 aparece la palabra robot, utilizada por primera vez en una obra de teatro llamada "Robots Universales Rossum" escrita por el dramaturgo checo Karel Capek. Robot se deriva de la palabra checa "robotnik" y significa, siervo, servidor o trabajador forzado (La Izquierda Diario).

Sin embargo, mucho antes de haber utilizado esta palabra varios mecanismos fueron diseñados con este propósito. En 1300 a. C. en Egipto se construyó la estatua de Memon, la cual emitía sonidos con la luz del sol al amanecer (Sánchez-Martín, 2007). Otros elementos matemáticos muy sofisticados fueron inventados como el ábaco o reloj de arena.

La clasificación de los robots es sumamente grande y a nivel general se pueden clasificar en tres grupos:

- Humanoides
- Móviles
- Industriales

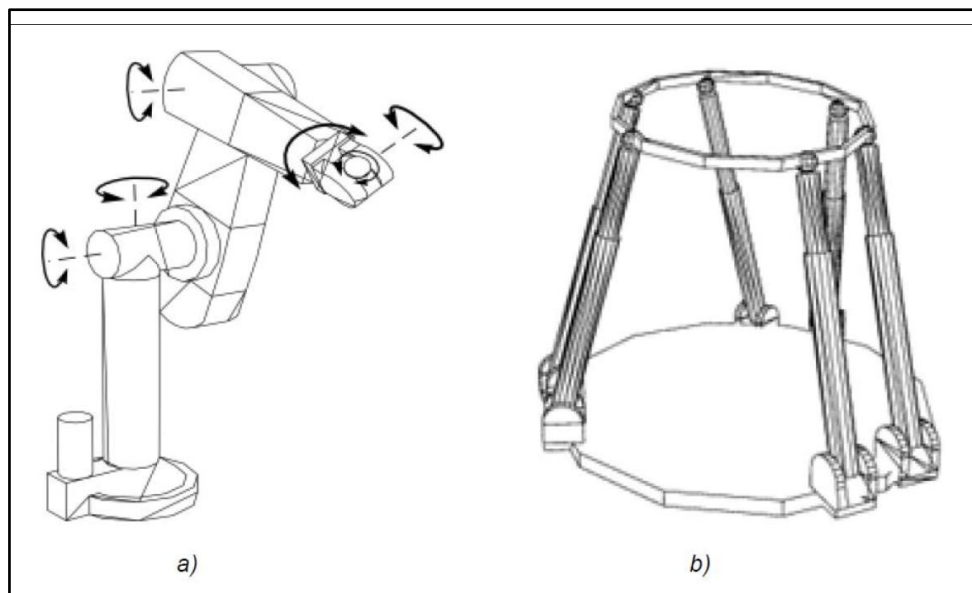
Son varias las definiciones de un robot industrial, una de ellas las propone la ISO y dice:

Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas (Víctor R. González, 2002).

Dentro de este grupo de robots industriales se encuentran los robots paralelos, los cuales son objeto de estudio de este trabajo y a continuación se explicará con más detalle.

## 2.1 Robots Paralelos.

Existen dos tipos de robots manipuladores: seriales y paralelos. La diferencia es la conexión de sus cadenas cinemáticas, los seriales se encuentran conectados en forma secuencial mientras que los paralelos como su nombre lo dice, se encuentran paralelas unas a otras. En la Figura 1 se observa un ejemplo de la cinemática de cada robot.



**Figura 1** Cinemática de un robot manipulador a) serial b) paralela

Fuente: (Sanz, 1990)

En el artículo Robots Paralelos: Máquinas con un Pasado para una Robótica del Futuro, se propone una definición para el robot paralelo:

Un robot paralelo es aquel cuya estructura mecánica está formada por un mecanismo de cadena cerrada en el que el efector final se une a la base por al menos dos cadenas cinemáticas independientes (Aracil, Saltaron, & others, 2006).

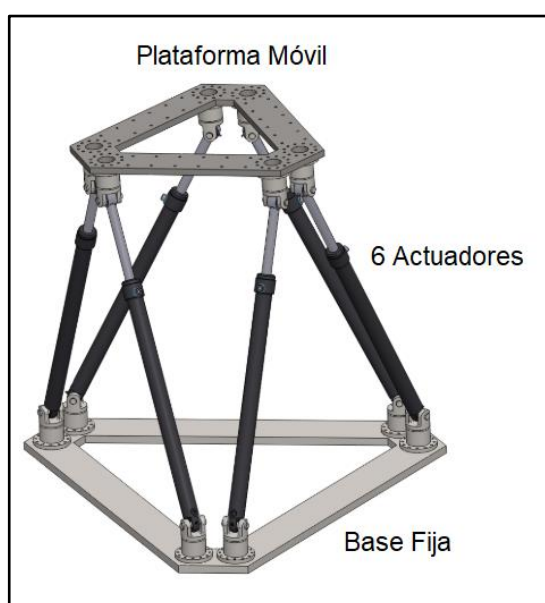
Los robots paralelos presentan ciertas ventajas frente a los seriales, como se observa a continuación:

**Tabla 1**  
*Ventajas y desventajas robots paralelos y seriales*

<b>Ventajas</b>	<b>Desventajas</b>
<b>Alta precisión</b>	Cinemática compleja
<b>Estabilidad</b>	Poca área de trabajo
<b>Peso ligero</b>	Configuración compleja
<b>Mayor aceleración</b>	Inexistencia de un modelo dinámico

### 2.1.1 Plataforma Stewart

En 1956 Stewart propuso una plataforma de 6 grados de libertad (Figura 2), con la que era posible simular el vuelo de un avión. Esta propuesta se basó en el modelo de Cough en 1947, el cual se trataba de un hexápodo el cual simulaba el proceso de aterrizaje de un avión (Vivas Albán, 1989).



**Figura 2** Plataforma Stewart 6 GDL

Fuente: (S. Christensen, 2014)

La plataforma Stewart tenía la capacidad para simular los complejos movimientos que un avión realiza durante todo su proceso de vuelo. Su configuración paralela brinda una gran precisión y sensibilidad, por lo que sus aplicaciones luego fueron desarrolladas en otros campos.

El peso o carga que puede soportar es otra ventaja que presenta esta plataforma, al igual que todos los robots paralelos como ya se mencionó anteriormente. En este caso, la carga se distribuye en sus 6 actuadores de modo que cada uno de ellos soporta  $1/6$  el peso o carga; así si el robot pesa 35kg, tiene una capacidad de carga de hasta 600kg (González Álvarez & Reinoso Mendoza, 2011).

### **2.1.2 Aplicaciones**

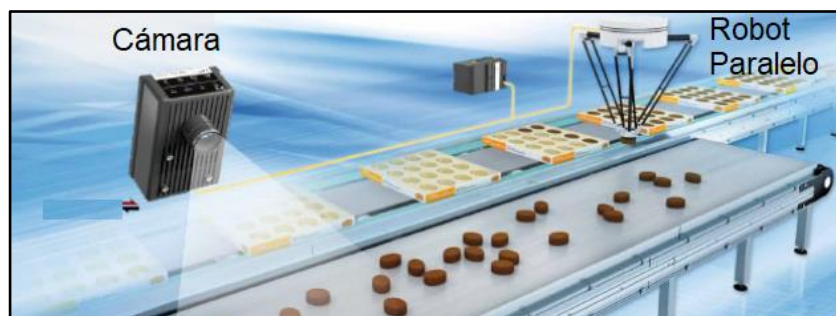
Las aplicaciones de los robots paralelos han ido desarrollándose con el pasar de los años. Entre las más conocidas están:

- Simuladores de vuelo
- Simuladores de manejo (coches)
- Posicionamiento de telescopios y antenas
- Aplicaciones médicas: cirugías y rehabilitación ortopédica
- Prueba de nuevas de técnicas de control

Dentro de la industria, son varias las funciones que estos robots realizan:

- Tareas de precisión de posicionamiento y ensamblaje.
- Empaquetado.
- Seleccionador y apilado de material (pick and place).

La mayoría de ellas están integradas a un sistema de visión el cual sirve como sensor para realizar las funciones designadas dentro de una línea de producción, como se observa en la Figura 3.



**Figura 3** Robot paralelo en línea de producción de alimentos

Fuente: (Cazalilla, 2017)

## 2.2 Identificación de sistemas

La identificación es la técnica de construir un modelo matemático de sistemas dinámicos a partir de variables medidas del mismo (Garrido, 1999). El modelo de un sistema no es de interés para el campo industrial solamente, la mejora continua e innovación de procesos de todo tipo hacen que el hombre entienda mejor los sistemas, logrando así controlarlos, regularlos y hasta predecirlos (Sedano & Ramón, 2005). Así, por ejemplo, día a día estudios del cambio climático son realizados, convirtiendo así un sistema desconocido en un sistema identificado, mediante el cual se obtienen datos actuales, pasados e incluso futuros (predicciones de clima).

El proceso de identificación se desarrolla en varias etapas:

### 2.2.1 Toma de datos

Un punto muy importante en la identificación de sistemas es la toma de datos. Éstos deben ser lo mayor significativos respecto al sistema a identificar, significa que las señales de

entrada deben tener los suficientes componentes para excitar todos los modos dinámicos posibles de la planta.

Señales de entrada más comunes:

- Ruido blanco gaussiano
- Señales aleatorias binarias
- Señales pseudo aleatorias binarias
- Señales aleatorias
- Escalón

### **2.2.2 Elección del modelo**

Se puede identificar 3 tipos de plantas:

1. Caja negra: se desconoce el modelo o en su efecto, es complicado resolverlo por lo que se opta por el método de identificación.
2. Caja ploma: algunas partes del sistema son modelables y otras no.
3. Caja blanca: se conoce el modelo matemático y se lo quiere comprobar con métodos de identificación.

Los tipos de modelos son:

1. Paramétricos: describen las relaciones entrada-salida mediante ecuaciones matemáticas.
2. No Paramétricos: describen las relaciones entrada-salida mediante gráficos, diagramas o representaciones.

De igual manera se debe identificar el tipo de sistema y para ello existen:

1. Dinámicos: la salida depende de la entrada en ese instante de tiempo y de estados anteriores en el tiempo.

2. Estáticos: la salida depende únicamente de la entrada en ese instante de tiempo.
3. Determinista: existe una ecuación exacta que representa la relación entrada-salida
4. Estocástico: sinónimo de aleatorio, el sistema se puede representar por medios probabilísticos o estadísticos.

### 2.2.3 Elección de la estructura

Para modelos lineales entre las más comunes son:

**Tabla 2**  
*Estructura modelos lineales*

Nombre	Ecuación	Estructura
<b>Error de salida</b> <b>OE</b>	$y(t) = \frac{B(q^{-1})}{F(q^{-1})} * u + e(t)$	
<b>Autoregresivos</b> <b>con variables</b> <b>exógenas ARX</b>	$A(q^{-1})y(t) = B(q^{-1}) * u(t) + e(t)$	
<b>Autoregresivos</b> <b>de media móvil</b> <b>y variables</b> <b>exógenas</b> <b>ARMAX</b>	$A(q^{-1}) * y(t) = B(q^{-1}) * u(t) + C(q^{-1}) * e(t)$	

Mientras que para modelos no lineales se presentan similares a los lineales con el complemento de la no linealidad, como se muestra a continuación:



**Tabla***Estructura modelos no lineales***3**

Nombre	Ecuación	Estructura
<b>NARX</b>	$y(t) = f(y(t-1), \dots, y(t-n), u(t-d), \dots, u(t-d-m))$	
<b>NARMAX</b>	$y(t) = f(y(t-1), \dots, y(t-n), u(t-d), \dots, u(t-d-m), e(t-1), \dots, e(t-k)) + e(t)$	

**2.2.4 Elección de algoritmo que resuelva la estructura**

Son varios los métodos numéricos que ofrecen solución a diferentes estructuras de sistemas, tanto para lineales como para no lineales:

- Matrices Polinomiales
- Mínimos cuadrados
- Método de la variable instrumento
- Máxima verosimilitud
- Redes neuronales, entre otros

Parte del objetivo de este trabajo es aplicar redes neuronales, así que este será el método que resuelva la estructura del modelo elegido.

### **2.3 Control Inteligente.**

El control ha representado un papel fundamental en el desarrollo de la ciencia y tecnología. Esta teoría de control se encuentra desarrollada y dividida básicamente en dos partes: clásica y moderna. Por años fue el control clásico que cubrió la mayoría de necesidades de control, debido a la simplicidad de los sistemas: una entrada y una salida (Zacapoaxtla, 2013).

Los sistemas son cada vez más complejos, múltiples entradas y salidas son parte de sus características y es aquí donde el control moderno, inteligente, artificial desempeña un papel fundamental.

En la tesis doctoral sobre Arquitectura de Control Inteligente (Sanz, 1990), se define al control inteligente como:

El control inteligente comprende una serie de técnicas —tomadas fundamentalmente de la inteligencia artificial— con las que se pretenden resolver problemas de control inabordables por los métodos clásicos.

Una característica de la plataforma Stewart es la complejidad de obtener su modelo cinemático, el cual es base para diseñar controladores clásicos. Por ello el control inteligente es una solución apropiada para esta problemática. Cabe recalcar que existen proyectos de control de plataformas Stewart resueltos con métodos de control clásicos, algunos de ellos serán citados en el apartado 2.4. Sin embargo, ese no es el objeto de estudio de este trabajo.

Algunos problemas que el control inteligente puede solucionar son:

- Procesos fuertemente no-lineales.

- Procesos con incertidumbre en su comportamiento dinámico o con derivas de los parámetros de proceso.
- Procesos con variables ocultas, esto es variables no medibles.
- Procesos con sensibilidad elevada.
- Procesos con perturbaciones aleatorias intensas.
- Procesos con retardos elevados

### **2.3.1 Control difuso**

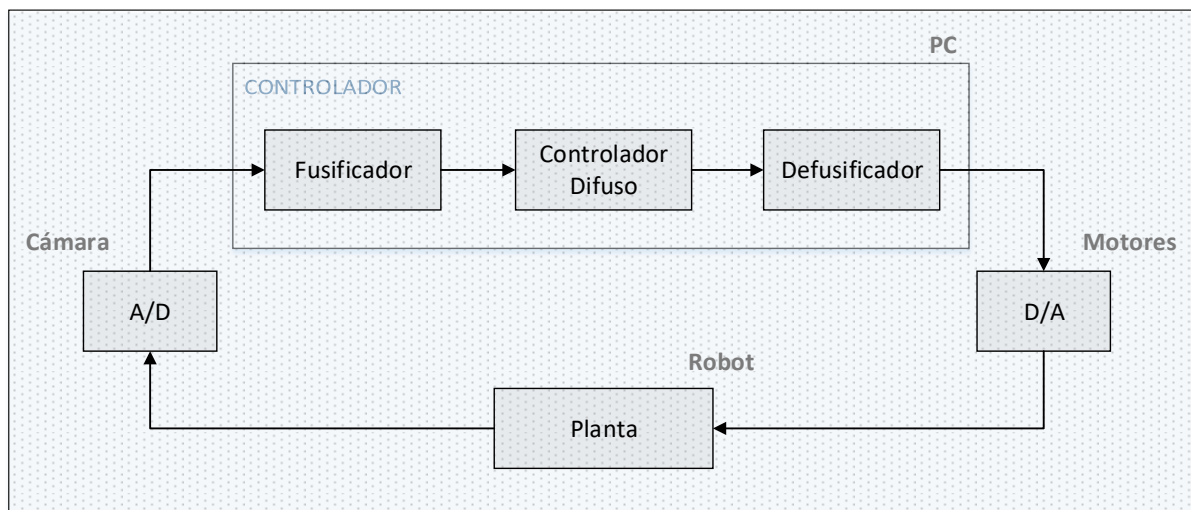
La Lógica difusa o también conocida como borrosa, puede ser definida como un sistema que no tiene valores, rangos o límites matemáticamente establecidos. Así por ejemplo si se habla de la distancia de una bola a un punto fijo, controlado por el ángulo de la base, no se hace referencia al valor en centímetros o pixeles de la bola al punto, si no a términos como: muy lejos, cerca, muy cerca, etc.

El control difuso aparece por primera vez en 1965, en la Universidad de California por el profesor Lofti A. Zadeh, en el que se menciona la primera teoría sobre conjuntos borrosos. Con el tiempo esta teoría fue aceptada e implementada en distintas industrias a nivel mundial (Eduardo Aznar Colino, 2007).

Samir Kouro R. y Rodrigo Musalem M. en su artículo Control mediante lógica difusa definen:

La lógica difusa puede ser descrita como un sistema interpretativo, en el cual los objetos o elementos son relacionados con conjuntos de fronteras no nítidamente definidas otorgándoles un grado de pertenencia relativa o graduada y no estricta como es de costumbre en la lógica tradicional (Kouro & Musalem, 2002).

Debido a que un controlador difuso no utiliza valores matemáticos, si no palabras que definan al estado del sistema y la retroalimentación (sensor) que recibe es de manera matemática, en este caso la distancia de la bola al centro o punto de referencia de la plataforma en pixeles, es necesario realizar una transformación y a eso se lo conoce como fusificación. En la Figura 4 se observa el esquema de un lazo de control difuso.



**Figura 4** Lazo de control difuso

Fuente: (Kouro & Musalem, 2002)

Para el caso de la plataforma Stewart, el A/D o sensor será una cámara, el dato entregado será distancia medida en pixeles, mientras que el D/A o actuador serán los 6 servomotores que se encarguen de brindar movimiento a la plataforma.

A continuación, se describen las principales partes de un controlador difuso con su respectivo ejemplo:

### 1. Fusificación

En esta etapa se asignan grados de pertenencia a la variable hacia los conjuntos difusos del sistema. Para ello es necesario haber definido previamente los conjuntos difusos y las funciones de pertenencia a utilizar.

## 2. Conjuntos Difusos

A los conjuntos difusos se los define con las variables lingüísticas, que no son más que adjetivos que describen al conjunto. Estos conjuntos permiten describir el estado en el que se encuentra el sistema a controlar, como se observa en los siguientes ejemplos:

Conjunto de entrada IN:

- PL: positivo lejos
- PC: positivo cerca
- Z: cero

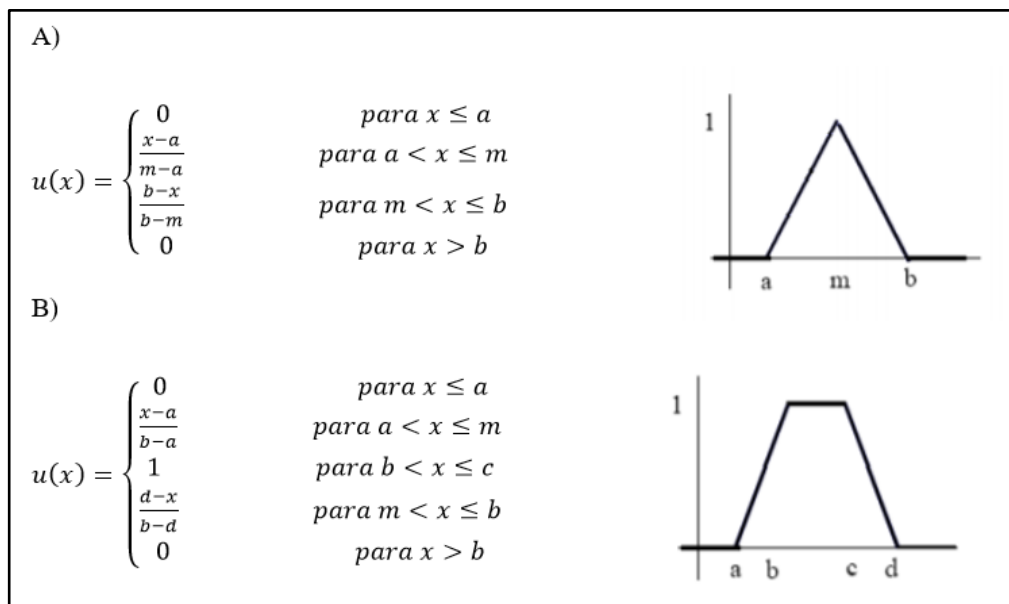
Conjunto de salida OUT:

- GF: giro fuerte
- GS: giro suave
- Z: cero

Difuso o borroso se refiere a que no es claro, es por ello que el sistema no se encuentra necesariamente en un solo estado o conjunto, si no puede pertenecer a varios. Para determinar a qué conjunto pertenece y con qué magnitud a cada uno de ellos se utilizan las funciones de pertenencia.

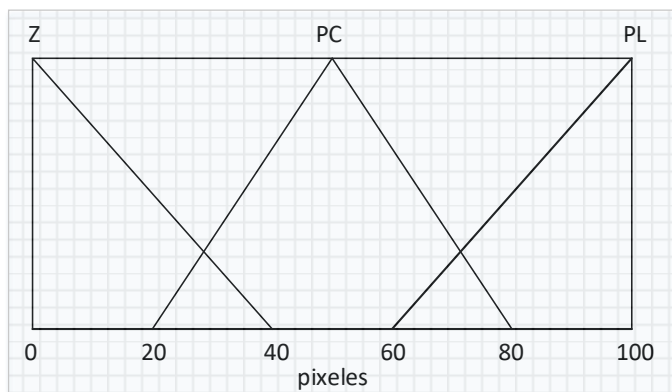
## 3. Funciones de Pertenencia

Generalmente, cualquier función puede ser utilizada para definir un conjunto difuso o variable lingüística, sin embargo, la triangular y trapezoidal son más utilizadas en lógica difusa.



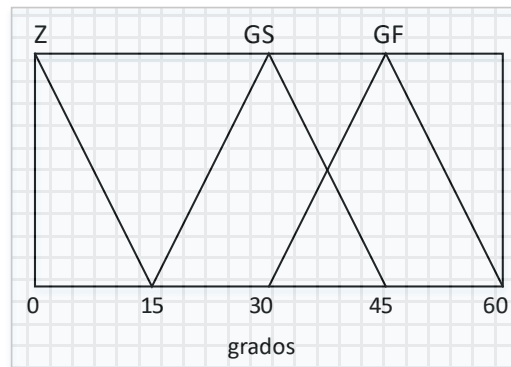
**Figura 5** Funciones de activación a) triangular b) trapezoidal

Así para el ejemplo mencionado de la distancia de una bola a un punto fijo y los conjuntos difusos definidos previamente, para un rango de entrada de 0 a 100 pixeles se definen las siguientes funciones triangulares de pertenencia:



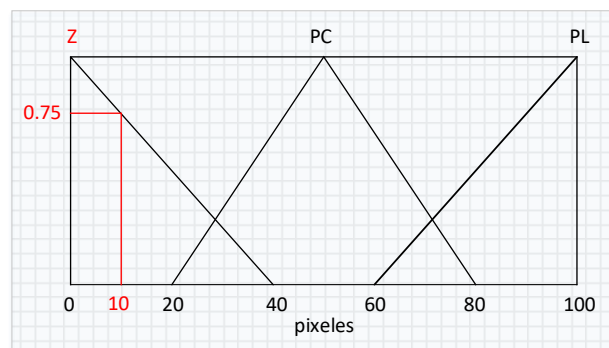
**Figura 6** Funciones de pertenencia – entrada

Para la salida definida entre 0 y 60 grados, se definen los siguientes conjuntos de pertenencia:



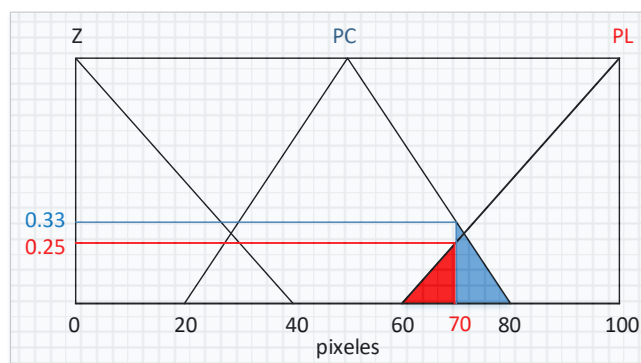
**Figura 7** Funciones de pertenencia - salida

Para un valor de entrada de 10 pixeles, la entrada fusificada tendrá un valor de “0.75Z”, como se observa en la siguiente figura:



**Figura 8** Entrada 10 pixeles

Mientras que, para un valor de entrada de 70 pixeles, la entrada fusificada tendrá un valor de “0.33PC” y “0.25PL”, como se observa en la siguiente figura:



**Figura 9** Entrada 70 pixeles

Así se muestra que una entrada, puede pertenecer a uno o más conjuntos difusos dependiendo sus límites definidos previamente y las funciones de pertenencia asignados.

#### 4. Reglas difusas

Son reglas que generalmente tienen la forma “Si X entonces Y”. Pueden ser reglas simples del tipo Si A entonces B o más complejas con operadores lógicos como AND, OR, etc.

Para el ejemplo en desarrollo, se definen tres reglas difusas:

- a) Si la bola está lejos IN: PL, entonces girar fuerte OUT: GF.
- b) Si la bola está cerca IN: PC, entonces girar suave OUT: GS.
- c) Si la bola está en cero IN: Z, entonces girar cero OUT: Z.

#### 5. Defusificación

Una vez fusificada la variable medida y aplicada las diferentes reglas de fusificación, se debe defusificar para proceder a realizar la acción de control correspondiente. Métodos matemáticos como el del centroide, promedio ponderado, membresía del medio del máximo son aplicados para obtener de las variables lingüísticas los valores numéricos que serán la entrada al actuador. El centroide es el más conocido y aplicado por distintos controladores difusos.

Para una entrada IN: 10 pixeles, su valor fusificado IN:  $0.75Z$ , cumple con las reglas en las siguientes proporciones:

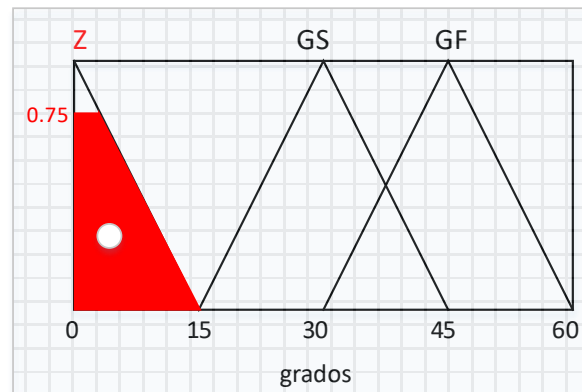
Regla 1: 0%

Regla 2: 0%

Regla 3: 75%

Como es Z en 0.75, la salida será el equivalente al centroide mostrado en la Figura 10.





**Figura 10** Centroide  $z=0.75$

Salida=3.75 grados

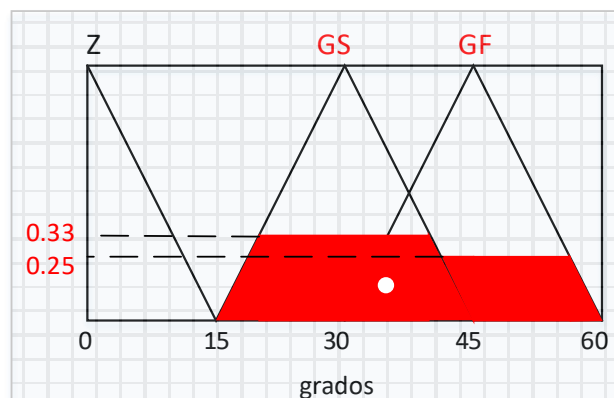
Para una entrada IN:70 pieles, su valor fusificado IN: 0.33PC y IN: 0.25PL cumple con las reglas en las siguientes proporciones:

Regla 1: 25%

Regla 2: 33%

Regla 3: 0%

En este caso la salida será  $OUT=0.33GS$  y  $OUT=0.25GF$ . Para hallar el valor defusificado se calcula el centroide del área sombreada, como se muestra en la siguiente figura:



**Figura 11** Centroide  $GS= 0.33$  y  $GF=0.25$

Salida=38.5 grados

### 2.3.2 Redes neuronales

Cuando se lee el término redes neuronales, se relaciona con el cerebro humano y el conjunto de neuronas que este posee. Pues precisamente de eso se trata, es un sistema que pretende imitar el comportamiento neuronal humano, de una manera más sencilla.

En 1936, fue Alan Turing quien empezó a analizar el cerebro humano como una especie de computadora. Posterior a ello, el neurofísico Warren McCulloch y el matemático Walter Pitts crearon una teoría sobre las neuronas, vistas como un sistema de control (Matich, 2001).

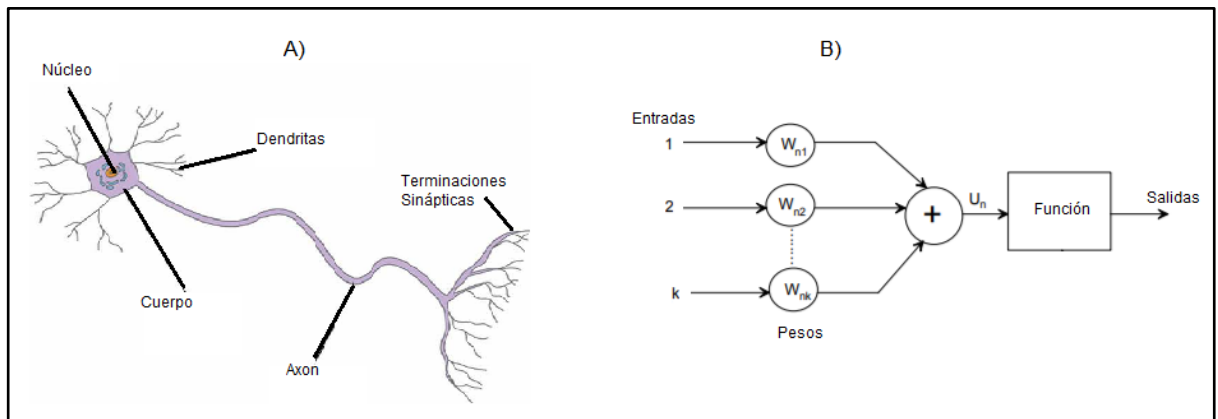
Hilera G. y Martínez H. en su artículo: Redes neuronales artificiales, fundamentos, modelos y aplicaciones definen a una red neuronal como:

Un sistema de computación que consiste en un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas (Hilera & Martínez, 2000).

Debido a su semejanza al cerebro humano, las RNA o redes neuronales artificiales presentan ciertas ventajas como sistema:

- Capacidad de entrenar y aprender.
- Buena respuesta a fallos.
- Procesamiento en tiempo real.
- Compatible con tecnología existente.

Si se realiza una comparación entre una neurona humana y una artificial, se pueden plantear las siguientes partes:



**Figura 12** Partes de neurona a) humana b) artificial

Fuente: (Hilera & Martínez, 2000)

Cada parte de la neurona realiza una función específica, las cuales se detallan a continuación:

### 1. Entradas

Son las señales eléctricas que la neurona recibe para luego procesarla.

### 2. Pesos

Aquí cada entrada es ponderada con su respectivo peso, conocido como peso sináptico. Es esta quizás, la parte más importante ya que aquí la señal es amplificada o atenuada según el peso, y es por esto que las redes se entrenan previamente antes de implementarse en el sistema, ya que los pesos deben tener los valores adecuados para obtener los resultados esperados. El entrenamiento de la red neuronal consiste en asignar el peso adecuado a cada uno de los pesos sinápticos.

### 3. Función

La función más conocida como función de activación, debe evaluar si la neurona se activa o no, al ser un sistema MISO (múltiples entradas, una salida), si la neurona es binaria su acción

solo puede ser de carácter todo o nada. Además, esta función toma en cuenta el estado anterior de activación.

Distintas funciones son utilizadas en esta parte, entre las más conocidas están:

- Escalón
- Lineal
- Sigmoide
- Gaussiana
- Tangente Hiperbólica

#### 4. Salidas

Más conocida como función de salida, la cual se encarga de cambiar el estado anterior al actual de la neurona. Igualmente se utilizan entre otras, las funciones ya mencionadas.

Hasta el momento se ha descrito la neurona, sus partes y funciones, sin embargo, una sola no podría realizar procesos de control avanzados, por tanto, las redes neuronales presentan un conjunto de neuronas interconectadas entre sí de distintas maneras, con lo que se obtiene mejores resultados a nivel de control.

#### 5. Arquitectura de una red neuronal

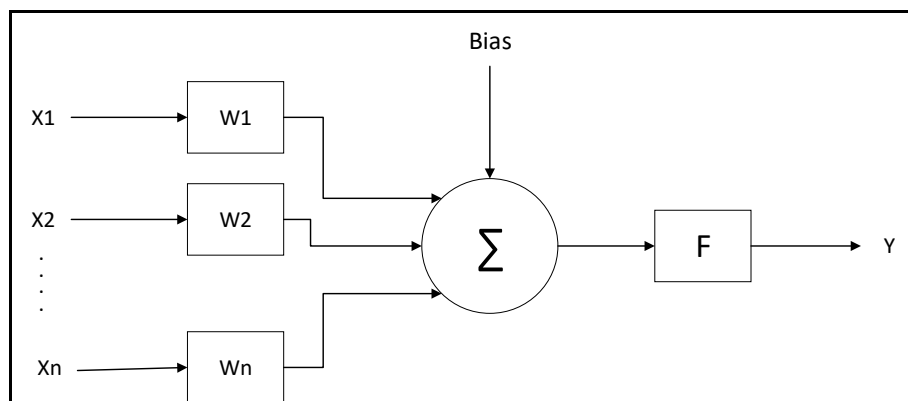
Se refiere a la manera de cómo están conectadas las neuronas entre sí dentro de la red, para ello se han definido distintas topologías de red:

- Según el número de capas: monocapas, multicapa.
- Según el tipo de conexiones: no recurrentes, recurrentes.
- Según el grado de interconexión: totalmente conectadas, parcialmente conectadas (Serrano J, Soria, & Martín, 2009).

## El Perceptrón

Fue Inventado en 1957 por Frank Rosenblatt. El perceptrón es una neurona artificial, cuyo objetivo es discriminar linealmente sus entradas (Larrañaga & Iñaki). La idea no fue bien aceptada debido a sus pocas aplicaciones, ya que se limitada a trabajar con datos en sus entradas que sean linealmente separables. Años después se creó una red de perceptrones (red neuronal) que permitía clasificar sus entradas no solo lineal, cuadrática ni exponencial, si no en cualquier grado posible. Desde entonces su estructura ha sido base para otros tipos de redes y sus aplicaciones han ido aumentando.

La estructura básica del perceptrón se muestra a continuación:



**Figura 13** Estructura perceptrón

Fuente: (Mejia, I)

Donde:

- X: entradas (de 1 a n)
- W: pesos sinápticos
- Bias: valor de corrección
- $\Sigma$ : función suma
- F: función de perceptrón
- Y: salida del perceptrón

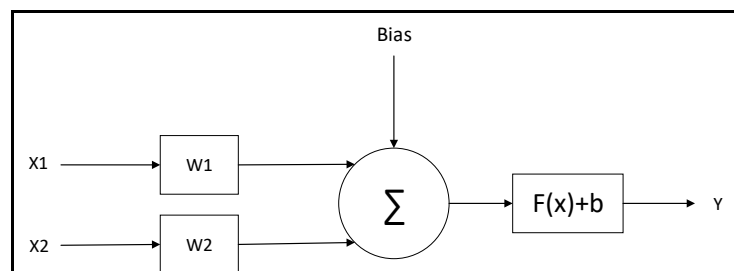
### Ejemplo de aplicación

Mediante un perceptrón con una función de activación escalón, se va a imitar la función de una compuerta lógica AND, para lo cual es necesaria la tabla de verdad de la función:

**Tabla 4**  
*Función lógica AND*

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Se conoce la estructura del perceptrón, para lo cual necesitamos dos entradas una salida y el bias.



**Figura 14** Perceptrón para función AND

Resolviendo el sistema, resulta la ecuación:

$$x1 * w1 + x2 * w2 + b = F(x) = Y \quad (2.1)$$

Reemplazando los valores de la Tabla 1 resulta el siguiente sistema de ecuaciones:

$$b = 0 \quad (2.2)$$

$$w2 + b = 0 \quad (2.3)$$

$$w_1 + b = 0 \quad (2.4)$$

$$w_1 + w_2 + b = 1 \quad (2.5)$$

Para el sistema de ecuaciones se obtiene dos soluciones, las cuales resuelven el problema de la compuerta AND.

Solución 1:

$$w_1 = 1$$

$$w_2 = 1$$

$$b = -1.1$$

Solución 2:

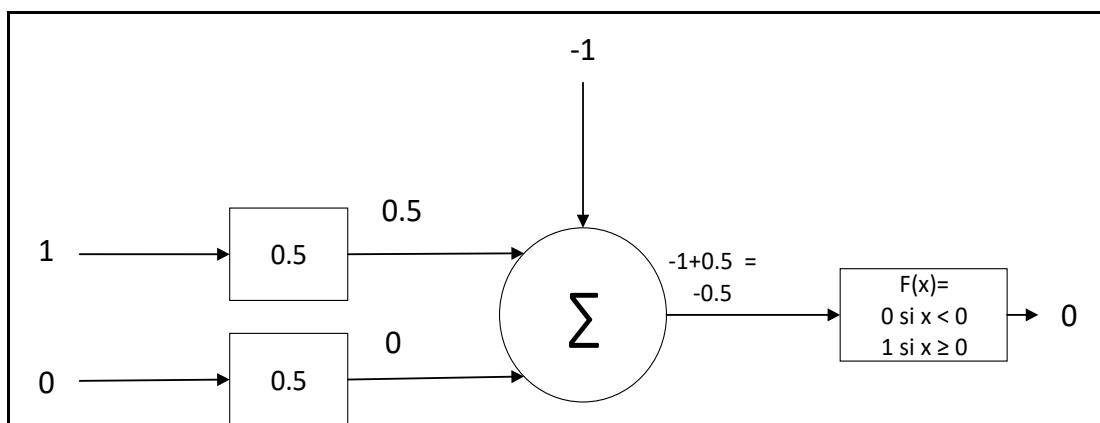
$$w_1 = 0.5$$

$$w_2 = 0.5$$

$$b = -1$$

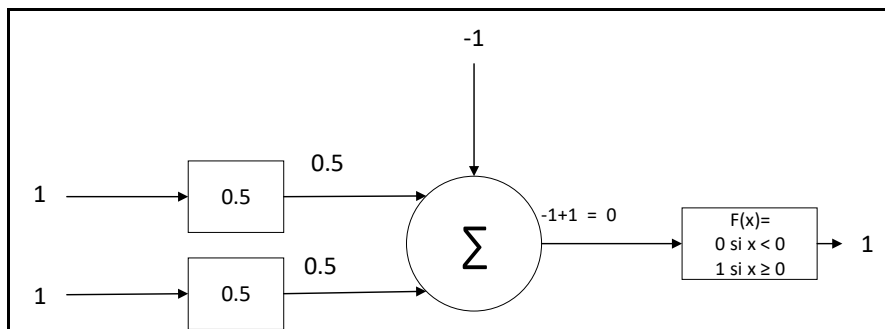
Tomando la solución 2 se obtienen los siguientes resultados

Para  $x_1=1$   $x_2=0$   $Y=0$ :



*Figura 15* AND caso 1

Para  $x_1=1$   $x_2=1$   $Y=1$ :



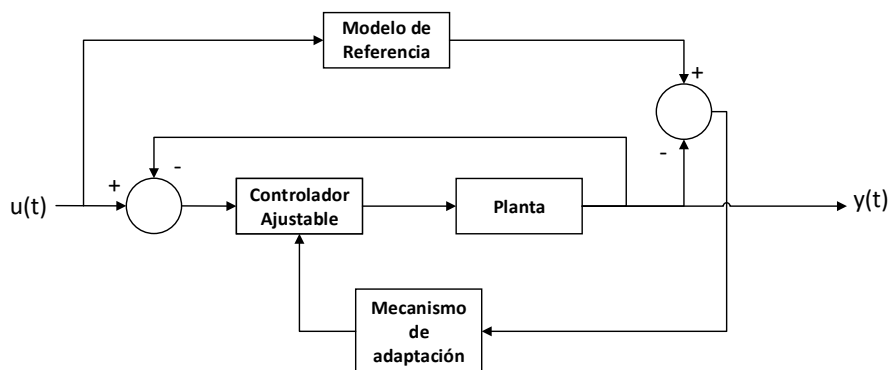
**Figura 16** AND caso 2

Como se mencionó anteriormente, si se quisiera implementar un perceptrón para representar a la función EX OR, no es posible debido a que sus salidas no son linealmente separables. Este problema se resuelve con una red de dos perceptrones.

### 2.3.3 Control por modelo de referencia neuronal

El control por modelo de referencia es una estrategia de control adaptativo. El modelo de referencia sirve como modelo de rendimiento de controlador, lo que le permite al controlador ajustar sus valores hasta obtener un rendimiento similar en la planta implementada vs el modelo de referencia (Rodríguez Rubio & López Sánchez, 1996)

La estructura de controlador se muestra a continuación:



**Figura 17** Control por modelo de referencia

Fuente: (Rodríguez Rubio & López Sánchez, 1996)



La arquitectura de este controlador utiliza dos redes neuronales, la primera, encargada de modelar la planta, y la segunda encargada de ajustar los parámetros del controlador. Por ello, la planta es en primera instancia identificada, y luego se procede a entrenar el controlador

## **2.4 Procesamiento Digital de Imagen.**

El humano es el sistema que posee el mejor procesamiento de imagen, con este es capaz de detectar, analizar, recordar y aprender. La visión artificial parte de la inteligencia artificial, la cual busca obtener la mayor cantidad de información de una imagen, a mayor velocidad.

Las principales aplicaciones de visión artificial son:

- Control de calidad
- Reconocimiento caracteres
- Control tráfico
- Robótica\*
- Biomédicas
- Militares

Un sistema de visión artificial realiza ciertos procesos para lograr su objetivo: capturar imagen en tiempo real, digitalizar, procesar, obtener resultados, parámetros, etc. (Gobierno de España & Union Europea, 2012).

### **2.4.1 Digitalizado de imagen**

Existen distintos tipos de videocámaras, las cuales están diseñadas para aplicaciones específicas. Cada una de ellas capturan imagen en tiempo real y las convierten en señales eléctricas para su posterior uso. Entre las más comunes están:

- Lineales: obtienen la imagen por medio de un mapa a través de líneas de desplazamiento longitudinal.
- Matriciales: leen los píxeles de una imagen en forma de matriz.
- Color: obtienen la imagen a color, por lo que proporciona mayor información que otras cámaras.
- Blanco y negro: obtiene la imagen en tonos blancos, negros y grises.
- Infrarrojas: genera una imagen a partir de la emisión de radiación infrarroja.
- Ópticas: se obtiene la imagen mediante la transmisión de la luz al sensor de la cámara, en cantidades controladas.

#### **2.4.2 Procesamiento de imagen**

Aquí se aplican ciertas técnicas con el objetivo de mejorar y simplificar la información requerida de la imagen. Existen un sinnúmero de filtros y transformaciones, que han sido desarrollados con distintos propósitos y para aplicaciones específicas, a continuación, se revisarán algunas de ellas:

- Suavizado

Es un filtro, utilizado generalmente para eliminar el ruido de una imagen, ya sea por un error en su lectura o por problemas de captura del dispositivo.

- Transformación a otros espacios de color

El color que obtiene una videocámara de color es en RGB (rojo, verde y azul), este es el espacio de color más conocido y utilizado. Sin embargo, resulta a veces complejo reconocer ciertos elementos de interés, además que un cuadro de imagen o *frame* tiene un peso mayor en RGB que en otros espacios de color, lo que implica una necesidad de mayor procesamiento para analizar toda la imagen.

**Escala de Grises:** A partir de una escala de claridad, será posible clasificar cualquier muestra de color de acuerdo a su parecido con uno u otro de los grises de la escala. La ecuación para transformar a escala de grises una imagen tiene en cuenta solo la componente Y del modelo YIQ, la cual debe ser aplicada a cada pixel de la misma. Se muestra en la ecuación 2.1

$$Y = R * 0.229 + G * 0.587 + B * 0.114 \quad (2.6)$$

**HSV:** este espacio parte de los valores RGB para la transformación a HSV. Implica una imagen liviana, reduciendo la cantidad de procesamiento necesaria para obtener elementos. Las ecuaciones para transformar de RGB a HSV se muestran a continuación.

$$H = \cos^{-1} \left[ \frac{\frac{1}{2} [(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right] \quad (2.7)$$

$$S = 1 - \frac{3\min(R, G, B)}{R + G + B} \quad (2.8)$$

$$V = \max(R, G, B) \quad (2.9)$$

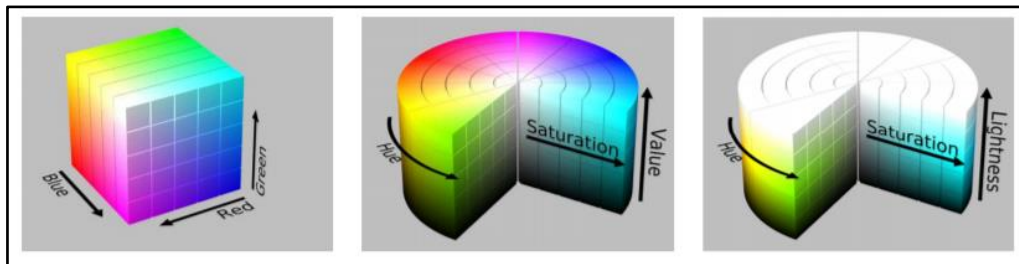
Los valores RGB se obtienen dividiendo su valor para 255.

$$\begin{aligned} R &= R/255 \\ G &= G/255 \\ B &= B/255 \end{aligned} \quad (2.10)$$

**HSI:** muy utilizado en procesamiento digital de imágenes debido a la simplicidad para mejorar la imagen o extraer algún elemento de ella (Alonso Pérez, 2009). Las ecuaciones para transformar de RGB a HSI son similares a las de HSV, variando el último parámetro *Value* por *Lightness*, se muestra a continuación.

$$I = \frac{1}{3}(R + G + B) \quad (2.11)$$

El espacio RGB es un cubo, mientras que HSV y HSI son cilíndricos, se observa gráficamente en la siguiente figura: **Figura 18**



**Figura 18** Espacios de color de izquierda a Derecha RGB, HSV, HSI

Fuente: (Alejandro & Emmanuel, 2017)

### 2.4.3 Obtención de Resultados

Existen varias opciones para hacer seguimiento a un objeto, más conocido como tracking. Los métodos revisados serán basados en el seguimiento de un color específico y en seguimiento de contornos de un objeto.

OpenCV (*Open Source Computer Vision Library*) es una herramienta con la que se pueden realizar distintas aplicaciones en visión artificial en tiempo real. Es de uso libre y se lo puede usar en distintos lenguajes de programación (OpenCV Team, 2018).

- Rango de colores

Si se desea seguir un color específico, existen funciones desarrolladas para este objetivo. Lo que hacen es comparar el color del pixel con un rango de colores previamente definidos. Así si se quiere rastrear el color verde entre un verde claro y oscuro, y el pixel analizado está dentro del rango, la función devolverá un valor de activación y si no está dentro, devolverá uno de desactivación. Finalmente se sabrá que áreas de la imagen están dentro y fuera del rango.

Cuando se trabaja en otros espacios como HSV es más fácil para este tipo de funciones transformar los rangos de color definidos al espacio utilizado y analizar dentro del espacio

transformado, la ventaja es que, por ser un espacio más simple, con menos colores, se demora menos en analizar cada pixel.

- Seguimiento de contornos

Existen varios métodos propuestos por diferentes autores, para encontrar el contorno de un objeto. Los contornos de un objeto de una imagen se pueden definir como transiciones entre dos regiones de niveles de gris o de un color significativamente distintos. Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc. (Universidad de Jaén, 2006)

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises y de color de la imagen.

En el caso de funciones bidimensionales discretas  $f(x,y)$ , la derivada es un vector que apunta en la dirección de la máxima variación de  $f(x,y)$  y cuyo módulo es proporcional a dicha variación. Este vector se denomina gradiente y se define:

$$\text{Para una imagen } z = E(x, y) \quad (2.12)$$

$$\text{El gradiente } \text{grad}(E) = \left[ \frac{\partial E}{\partial x}, \frac{\partial E}{\partial y} \right]^T \quad (2.13)$$

$$\text{Donde la orientación del gradiente } \alpha(x, y) = \tan^{-1} \left( \frac{Gy}{Gx} \right) \quad (2.14)$$

$$\text{Donde la magnitud del gradiente } \nabla f = [Gx^2 + Gy^2]^{\frac{1}{2}} \quad (2.15)$$

$$\text{Para lo que } Gx = \frac{\partial E}{\partial x}, Gy = \frac{\partial E}{\partial y} \quad (2.16), (2.17)$$

- 2.4.3.3 Seguimiento de objetos

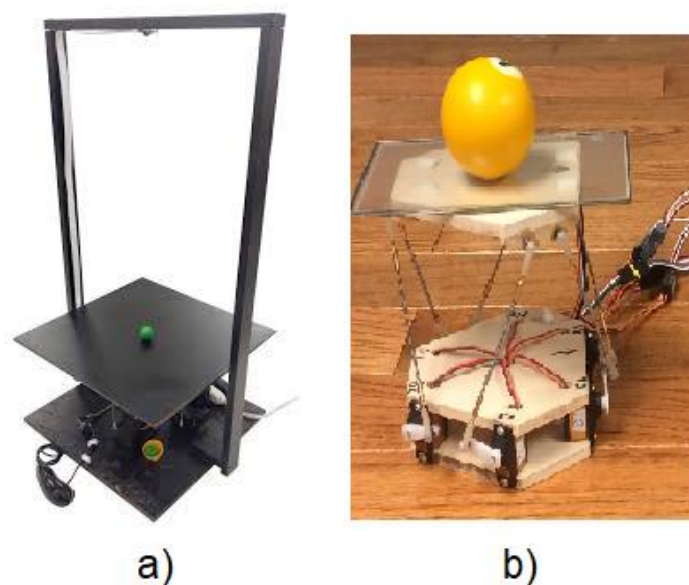
Como ya se mencionó, existen varios algoritmos que permiten realizar el tracking o seguimiento a objetos. Entre los más conocidos están:

- Flujo óptico denso: estos algoritmos ayudan a estimar el vector de movimiento de cada píxel en un cuadro de video.
- Flujo óptico disperso: estos algoritmos, como el rastreador de características de Kanade-Lucas-Tomashi (KLT), rastrean la ubicación de algunos puntos de característica en una imagen.
- Filtrado de Kalman: un algoritmo de procesamiento de señales muy popular utilizado para predecir la ubicación de un objeto en movimiento según la información de movimiento anterior.
- Meanshift y Camshift: estos son algoritmos para localizar los máximos de una función de densidad. También se utilizan para el seguimiento.
- Rastreadores de un solo objeto: en esta clase de rastreadores, el primer cuadro se marca con un rectángulo para indicar la ubicación del objeto que queremos rastrear. Luego se rastrea el objeto en cuadros posteriores utilizando el algoritmo de seguimiento. En la mayoría de las aplicaciones de la vida real, estos rastreadores se utilizan junto con un detector de objetos.
- Algoritmos de búsqueda de pistas de objetos múltiples: en los casos en que tenemos un detector de objetos rápido, tiene sentido detectar múltiples objetos en cada cuadro y luego ejecutar un algoritmo de búsqueda de pistas que identifique qué rectángulo en un cuadro corresponde a un rectángulo en el cuadro siguiente (Satya Mallick, 2017).

## 2.5 Estado del arte de aplicaciones “bola en plataforma”.

Los sistemas de balanceo han sido siempre de gran interés para el desarrollo de estrategias de control. Desde antes del año 2000 varios artículos han sido publicados sobre estrategias de control de sistemas de bola en viga o bola en plataforma (Ramón, 1999) tanto clásicas como modernas, las cuales brindan una solución al problema de equilibrio (Matsuo, Tsuruta, & Suemitsu, 1999).

La aplicación bola en plataforma al ser un sistema altamente no lineal, brinda las prestaciones para desarrollar nuevas técnicas de control (Yaovaja, 2018). Existen variaciones en cuanto a la construcción de este particular sistema: plataformas de dos (Ali & Aphiratsakun, 2015), tres (Cazalilla, 2017) o seis grados de libertad (Bang & Lee, 2018), y dentro de estas difiere la manera de la obtención de la posición de la bola en la plataforma, si es por lectura de la plataforma que puede ser resistiva, inductiva, *touch* (Kassem, Haddad, & Albitar, 2015) o mediante visión artificial (Shih et , 2017).



**Figura 19** Plataforma con a) visión b) plataforma resistiva

Los resultados obtenidos en esta aplicación basada en una plataforma Stewart con visión artificial son positivos, tanto en métodos de control clásicos: PID (Garzón Mancera & Garzón Melo, 2016) (Ali & Aphiratsakun, 2015) ; como modernos: difuso, RNA, modo deslizante, etc (Yaovaja, 2018)(Kuncan & Ertunç, 2016). Para controladores PID tiempos de estabilización de la bola en plataforma hasta 33 segundos (Ali & Aphiratsakun, 2015), mientras que en controladores difusos tiempos de 4 a 6 segundos (Yaovaja, 2018) y en controladores modo deslizante 4 segundos de estabilización (Bang & Lee, 2018).

La aplicación bola en plataforma no solo consiste en equilibrar la bola en el centro de la plataforma, existen trabajos publicados sobre cómo mover la bola en trayectorias geométricas como cuadrados, o círculos (Craig, Kevin, 2008), además de poder variar el punto de referencia y que no sea solo en el centro. Con esto se pretende desarrollar nuevos controladores, con mejores desempeños y más robustos.



**Figura 20** Simulador de Vuelo FUM 2400Kg



Las aplicaciones del robot Stewart van más allá de sistemas de equilibrio, incluso en la industria tienen varias funciones importantes dentro de líneas de producción y éstas igualmente se retroalimentan a través de visión (Vivas Albán, 1989). Sin embargo, originalmente fue creado como simulador de vuelos, debido a su gran precisión (Villacís et al., 2018). Actualmente la mayoría de simuladores de vuelos, sean comerciales, helicóptero o militares poseen la configuración Stewart o Gough-Stewart.

## CAPÍTULO III

### DESARROLLO DE ESTRATEGIAS DE CONTROL

#### 3.1 Descripción de la Planta.

El robot Stewart de 6 grados de libertad, es fabricado en base a distintos materiales: Plástico ABS, acero, acrílico, goma. En la Figura 21 se muestra el diseño digital del robot y sus partes.



*Figura 21* Diseño digital Robot Stewart



Fuente: (Alejandro & Emmanuel, 2017)

### 3.1.1 Especificaciones técnicas

En la Tabla 5 se resume el hardware y software utilizado en el robot.






**Tabla 5**

*Especificaciones técnicas*

N°	Elemento	Descripción	Fotografía
1	Cámara	Marca y modelo: Logitech C170. Conexión: puerto USB. Resolución Max: 720p/30 fps. Tipo de enfoque: foco fijo Campo visual: 58°	
2	Riel	Material: Aluminio. Longitud mínima: 64cm. Longitud máxima: 100cm.	
3	Plataforma	Material: Acrílico 6mm negro. Peso: 880g. Dimensiones: 35x35 cm.	
4	Motores	Tower Pro Servo high-torque MG996R. Peso: 55g c/u. Torque: 9.4kgf.cm a 4.8V. Voltaje: 4.8 – 7.2 V. Velocidad respuesta: 0.17s/60° a 4.8V.	
5	Pelotas	-Tipo 1: plásticas Colores: verde, azul, tomate Peso 2.7g Diámetro: 40mm -Tipo 2: metálica Colores: verde, azul, tomate Peso 35.7g y 95g Diámetro: 20 y 35mm	

Continúa

**Tabla 5**  
*Especificaciones técnicas*

N°	Elemento	Descripción	Fotografía
6	Brazos Servos	Material: Aluminio Color: rojo y celeste Rosca: m3 Distancia al centro: 20 y 24mm	
7	Eslabones	Juntas esféricas Material: Plástico y aluminio Medida: LST/2 Aft	
8	Driver Servomotores	Pololu Micro Maestro 6-Channel USB Servo Controller. Polarización: 5-16V. Control: USB serial RS232. Resolución pulso: 0.25uS. Rango de pulso: 64 – 3280uS. Canales: 6 servo motores.	
9	PC	Marca: Lenovo Ideapad 320. Procesador: AMD FX9800P 2.7GHz. RAM: 12Gb. VGA: R8 M535DX 4Gb	
10	Software utilizado	Sistema Operativo: Ubuntu 16.04 lts 64bits. Lenguaje de programación: Python 2.7 Librerías: OpenCV, Pololu maestro controller. Matlab R2017b	

### 3.1.2 Exactitud de la planta

La norma ISO9283 indica la siguiente fórmula para determinar la exactitud de la orientación, aplicada a una prueba de 30 repeticiones para 5 posiciones diferentes (International Organization for Standardization, 1998):

$$Apo = \sqrt{(\bar{\psi} - \psi_c)^2 + (\bar{\theta} - \theta_c)^2 + (\bar{\varphi} - \varphi_c)^2} \quad (3.1)$$

$$\bar{\psi} = \frac{1}{n} \sum_{j=1}^n \psi_j \quad (3.2)$$

$$\bar{\theta} = \frac{1}{n} \sum_{j=1}^n \theta_j \quad (3.3)$$

$$\bar{\varphi} = \frac{1}{n} \sum_{j=1}^n \varphi_j \quad (3.4)$$

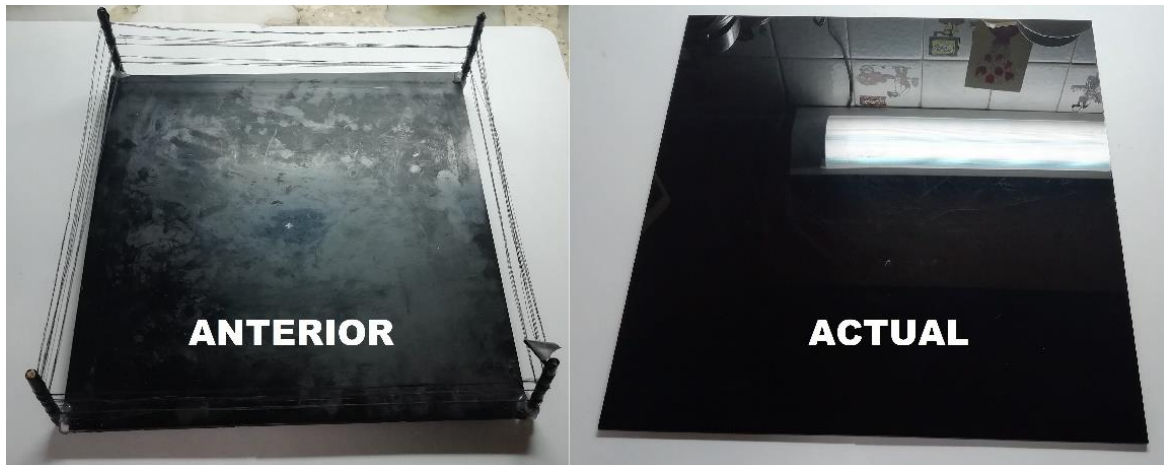
Donde:

- $Apo$  = Exactitud de la orientación al punto de prueba
- $\bar{\psi}$  = Promedio de orientaciones en Roll
- $\bar{\theta}$  = Promedio de orientaciones en Pitch
- $\bar{\varphi}$  = Promedio de orientaciones en Yaw
- $n$  = número de pruebas
- $\psi_c, \theta_c, \varphi_c$  = Coordenadas enviadas
- $\psi_j, \theta_j, \varphi_j$  = Coordenadas medidas

### 3.1.3 Modificaciones a la estructura de la planta

Se realizaron ciertas modificaciones a la planta con el fin de mejorar su desempeño, estética y estrategia de control.

1. Se reemplazó la plataforma anterior que fue impresa en 3d con plástico ABS en dimensiones 30x30cm. Actualmente la plataforma mide 35x35cm teniendo más espacio, con lo que se puede observar mejor el funcionamiento, sobre todo cuando se varía el punto de referencia. Además, la plataforma actualmente es de acrílico negro totalmente brillante, por lo que mejora el robot en apariencia y presentación.



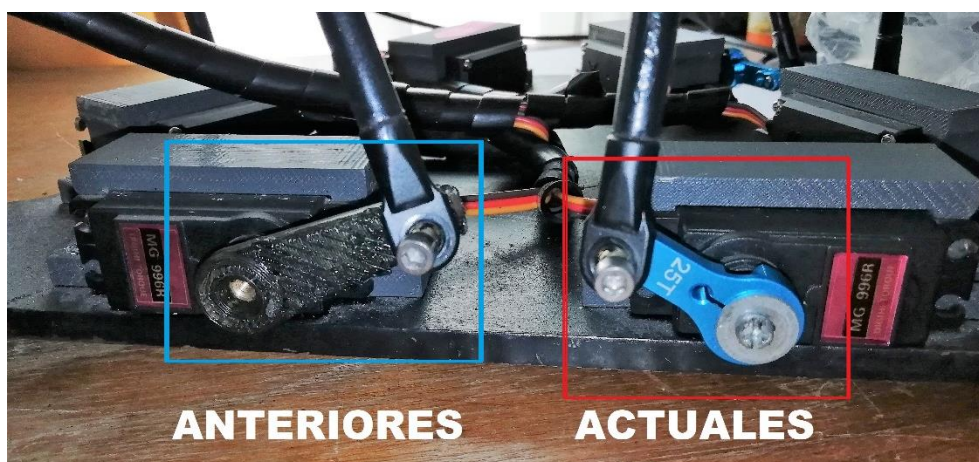
*Figura 22* Plataforma anterior - actual

2. Se reemplazó los eslabones antiguos que eran de 20cm por otros de 10cm, con la plataforma ubicada más abajo, se gana estabilidad al momento de moverla, y se reduce la fuerza que ejercen los brazos de los servos al momento de moverla a otra posición. Además, los eslabones anteriores estaban contruidos con metal, por lo que se redujo el peso considerablemente (250g) con los actuales contruidos con plástico y aluminio, lo que implica un menor esfuerzo y desgaste para los motores.



*Figura 23* Balanza digital, eslabones actuales y antiguos

3. Se reemplazó los brazos de los servos, los cuales fueron impresos en 3d con plástico ABS, por lo que no tenían un adecuado acople al servomotor (los dientes del engrane principal no se pueden imprimir en 3D, son muy pequeños). Los nuevos servos están diseñados para el servomotor utilizado, y es de aluminio lo que garantiza un perfecto acople con el servomotor y una buena resistencia.

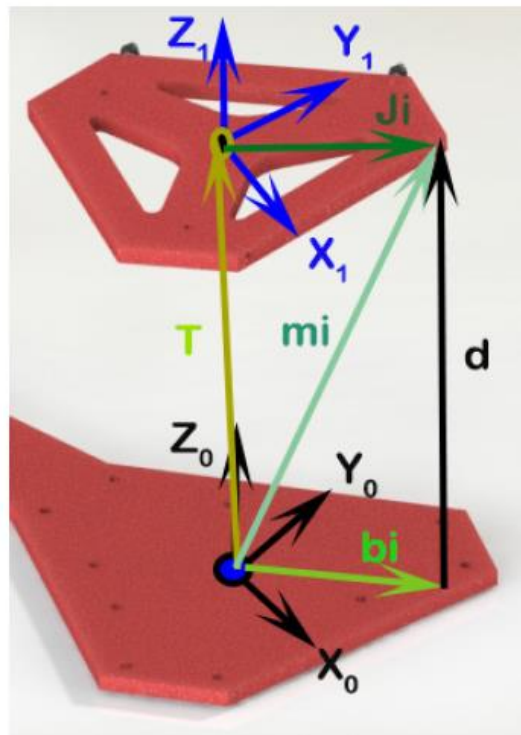


*Figura 24* Brazos servomotores actuales-antiguos

### 3.1.4 Cinemática Inversa

La Cinemática inversa permite determinar el movimiento que deben hacer los actuadores, en este caso 6 servomotores, para mover el actuador final (la plataforma) a cierta posición u orientación.

A continuación, se presenta el sistema de ecuaciones que resuelve la cinemática inversa del robot paralelo tipo Stewart de 6 grados de libertad (Alejandro & Emmanuel, 2017):



**Figura 25** Sistema de Referencia

Fuente: (Alejandro & Emmanuel, 2017)

Donde:

- $X_0, Y_0, Z_0 =$  Sistema de referencia fijo del manipulador
- $X_1, Y_1, Z_1 =$  Sistema de referencia móvil del manipulador
- $i =$  número de brazo o pierna (1,2..6)



- $T$  = vector de traslación
- $m_i$  = Vector de movimiento de la junta superior con respecto al sistema de referencia fijo.
- $d_i$  = Distancia que separa el brazo del servomotor con la junta superior
- $B_i$  = Coordenadas del brazo del servomotor con respecto al sistema de referencia fijo
- $J_i$  = Coordenadas de la junta superior con respecto al sistema de referencia móvil.

Las coordenadas del vector  $m_i$  con respecto al sistema fijo de referencia, parte de la siguiente relación:

$$\vec{T} + \vec{J}_i = m_i \quad (3.5)$$

Para hallar el valor de  $\vec{J}_i$  se procede a calcular la matriz de rotación del sistema:

$$R_t = R(z) * R(y) * R(x) \quad (3.6)$$

Por lo que la matriz resulta:

$$R_t = \begin{bmatrix} \cos(\gamma)\cos(\theta) & -\text{sen}(\gamma)\cos(\varphi) + \cos(\gamma)\text{sen}(\varphi)\text{sen}(\theta) & \text{sen}(\varphi)\text{sen}(\gamma) + \cos(\varphi)\text{sen}(\theta)\cos(\gamma) \\ \cos(\theta)\text{sen}(\gamma) & \cos(\varphi)\cos(\gamma) + \text{sen}(\theta)\text{sen}(\varphi)\text{sen}(\gamma) & \cos(\gamma)\cos(\varphi) + \text{sen}(\theta)\text{sen}(\gamma)\cos(\varphi) \\ -\text{sen}(\theta) & \text{sen}(\varphi)\cos(\theta) & \cos(\varphi)\cos(\theta) \end{bmatrix} \quad (3.7)$$

Una vez definida  $R_t$  se calcula  $\vec{J}_i$

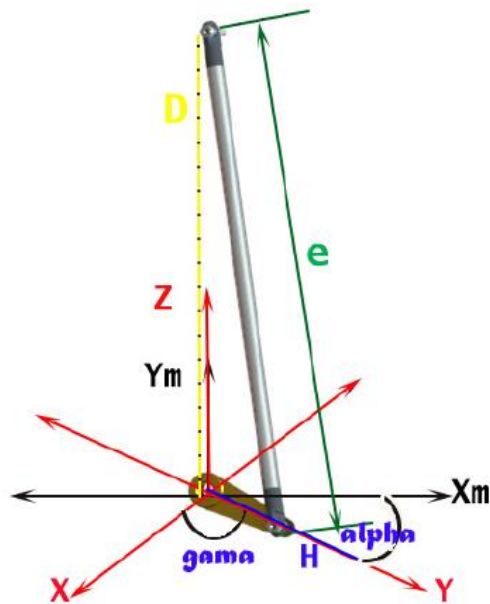
$$\vec{J}_i = R_t * \vec{J} \quad (3.8)$$

$$m_i = \vec{T} + R_t * \vec{J} \quad (3.9)$$

$$m_i = \vec{B}_i + \vec{D}_i \quad (3.10)$$

$$\vec{D}_i = \vec{T} + R_t * \vec{J} - \vec{B}_i \quad (3.11)$$

Una vez hallada la ecuación de  $\vec{D}_i$ , se debe hallar el ángulo que gire cada motor para llegar al punto deseado, como así lo determina la cinemática inversa. Para ello se analiza el siguiente sistema de referencia en cada motor:



**Figura 26** Sistema de Referencia Servomotores

Fuente: (Alejandro & Emmanuel, 2017)

Donde:

- $h$  = longitud del brazo servomotor
- $H$  = vector de coordenadas del brazo
- $e$  = longitud del eslabón
- $D_i$  = distancia variable entre brazo y junta de plataforma
- $\text{Alpha}$  = ángulo brazo respecto a eje X
- $\text{Gama}$  = ángulo  $\text{alpha}$  respecto a sistema de referencia de plataforma

Por leyes trigonométricas:

$$h^2 = (X_{hp} - X_{Bi})^2 + (Y_{Hp} - Y_{Bi})^2 + (Z_{hp} - Z_{Bi})^2 \quad (3.12)$$

$$d^2 = (X_{Ji} - X_{Bi})^2 + (Y_{Ji} - Y_{Bi})^2 + (Z_{Ji} - Z_{Bi})^2 \quad (3.13)$$

$$e^2 = (X_{Ji} - X_{hp})^2 + (Y_{Ji} - Y_{hp})^2 + (Z_{Ji} - Z_{hp})^2 \quad (3.14)$$

Resolviendo la ecuación

$$d^2 - e^2 + h^2 = 2(X_{Bi}^2 + Y_{Bi}^2 + Z_{Bi}^2) + 2X_{hp}(X_{ji} - X_{Bi}) + 2Y_{hp}(Y_{ji} - Y_{Bi}) + 2(Z_{ji} - Z_{Bi}) - 2(X_{ji}X_{Bi} + Y_{ji}Y_{Bi} + Z_{ji}Z_{Bi}) \quad (3.15)$$

Las siguientes ecuaciones relacionan el ángulo del servomotor con el sistema fijo de la plataforma:

$$X_{hp} = h * \cos(\alpha) * \cos(\lambda) + X_{Bi} \quad (3.16)$$

$$Y_{hp} = h * \cos(\alpha) * \sen(\lambda) + Y_{Bi} \quad (3.17)$$

$$Z_{hp} = h * \sen(\alpha) + Z_{Bi} \quad (3.18)$$

Reemplazando en Ecuación 3.15 y simplificando:

$$d^2 - e^2 + h^2 = 2h * (\cos(\alpha) * \cos(\lambda) (X_{ji} - X_{Bi}) + \cos(\alpha) * \sen(\lambda)(Y_{ji} - Y_{bi}) + \sen(\alpha)(Z_{ji} - Z_{bi})) \quad (3.19)$$

Mediante la identidad trigonométrica:

$$c * \sen(\alpha + \omega) = a * \sen(\alpha) + b * \cos(\alpha) \quad (3.20)$$

De donde:

$$c = \sqrt{a^2 + b^2} \quad (3.21)$$

$$\tan(\omega) = \frac{b}{a} \quad (3.22)$$

Reemplazando en la Ecuación 3.19

$$d^2 - e^2 + h^2 = \sqrt{a^2 + b^2} * \sen(\alpha + \omega) \quad (3.23)$$

Por lo que:

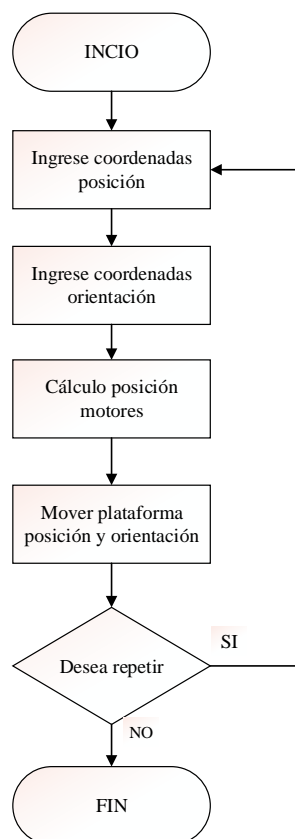
$$\alpha = \sen^{-1}\left(\frac{d^2 - e^2 + h^2}{\sqrt{a^2 + b^2}}\right) - \tan^{-1}\left(\frac{b}{a}\right) \quad (3.24)$$

Donde  $\alpha$  es el ángulo de giro del servomotor. Se debe tomaren cuenta la posición de cada motor y los valores límites de pulso, los cuales son:

$$pulso\ motor\ 1,3,5 = 2000 - \frac{\alpha * 1008}{90} \quad (3.25)$$

$$pulso\ motor\ 2,4,6 = 992 - \frac{\alpha * 1008}{90} \quad (3.26)$$

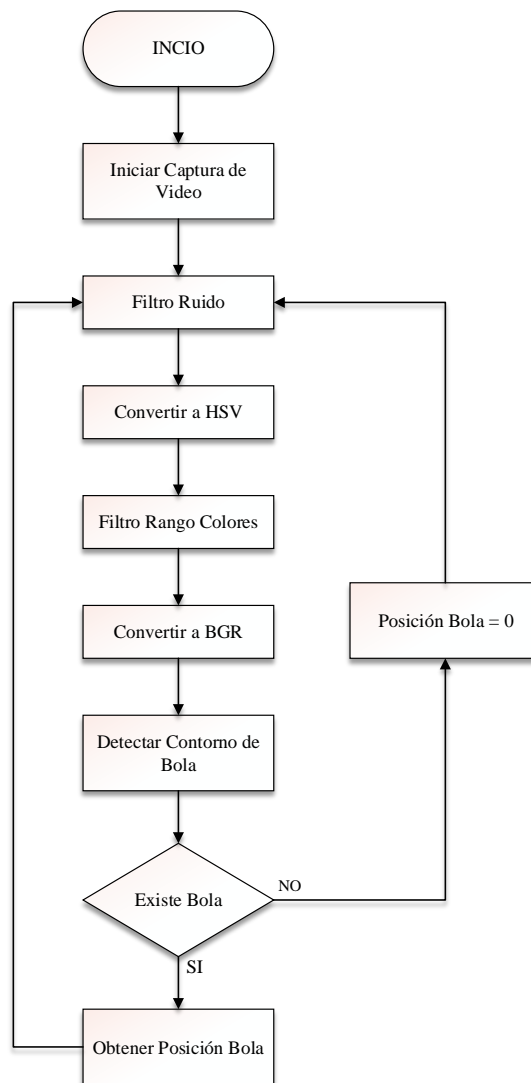
El programa Cinemática\_ejes.py (Anexo E), permite el ingreso de coordenadas por teclado, para mover la plataforma a la posición y orientación requerida. El algoritmo se muestra a continuación:



**Figura 27** Algoritmo cinemática

### 3.2 Estrategia de detección de objeto por medio de visión artificial.

El programa desarrollado para la detección de la bola ha sido escrito en Python (Anexo A), junto con la librería de OpenCV, la cual brinda herramientas útiles para el procesamiento de imagen en tiempo real. El algoritmo se muestra a continuación:

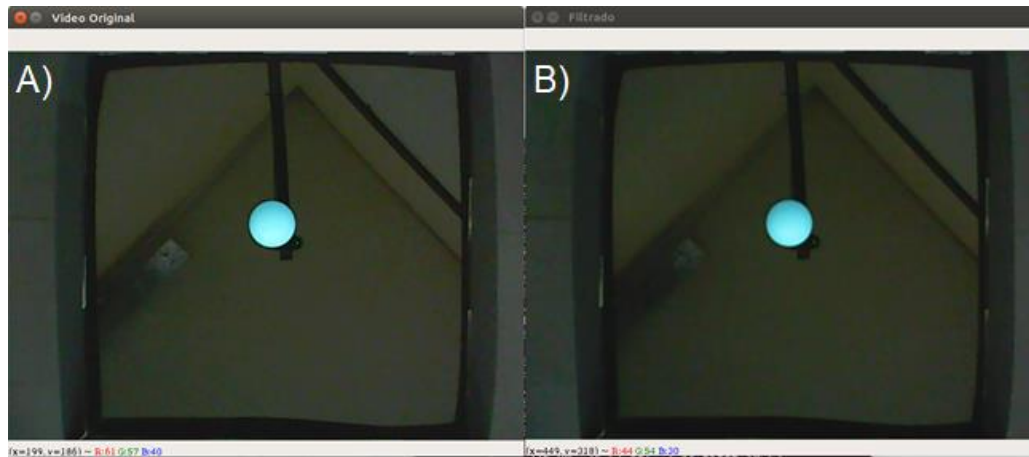


**Figura 28** Algoritmo visión artificial

A continuación, se explicará cada parte del algoritmo de visión:

### 3.2.1 Filtro Ruido

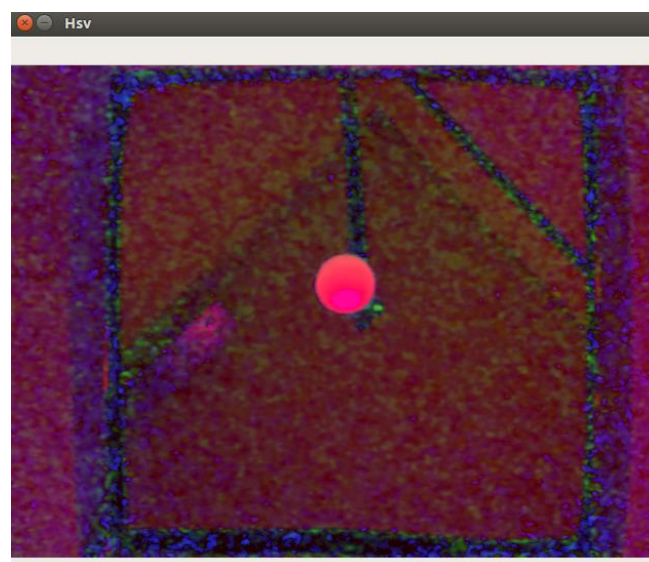
Se aplica un filtro Gaussiano para eliminar el ruido y suavizar la imagen, lo que facilita el procesamiento de la misma. Se observa la diferencia en la Figura 29



*Figura 29* Filtro Gaussiano a) sin filtro b) con filtro

### 3.2.2 Conversión a HSV

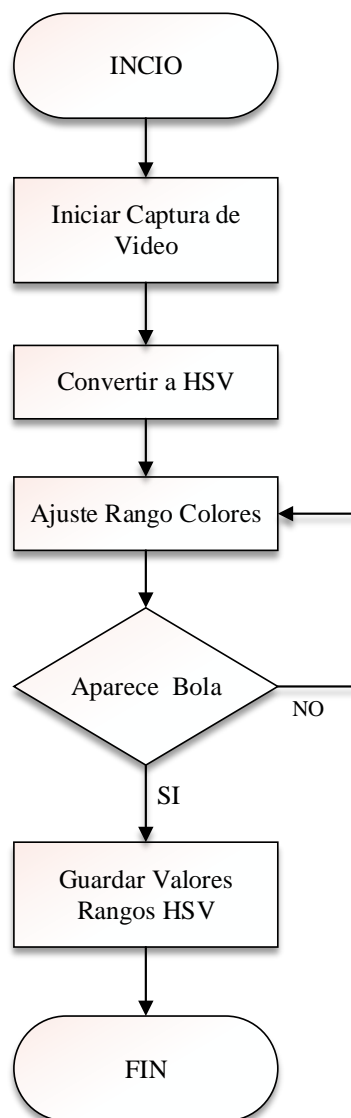
El reflejo en RGB puede ocasionar problemas al momento de procesar una imagen, introduciendo cierto ruido y por ello, malas medidas, por eso es recomendable trabajar en el espacio de color HSV (J. Christensen, Humble, & Mattinson, 2017).



*Figura 30* Transformación al espacio HSV

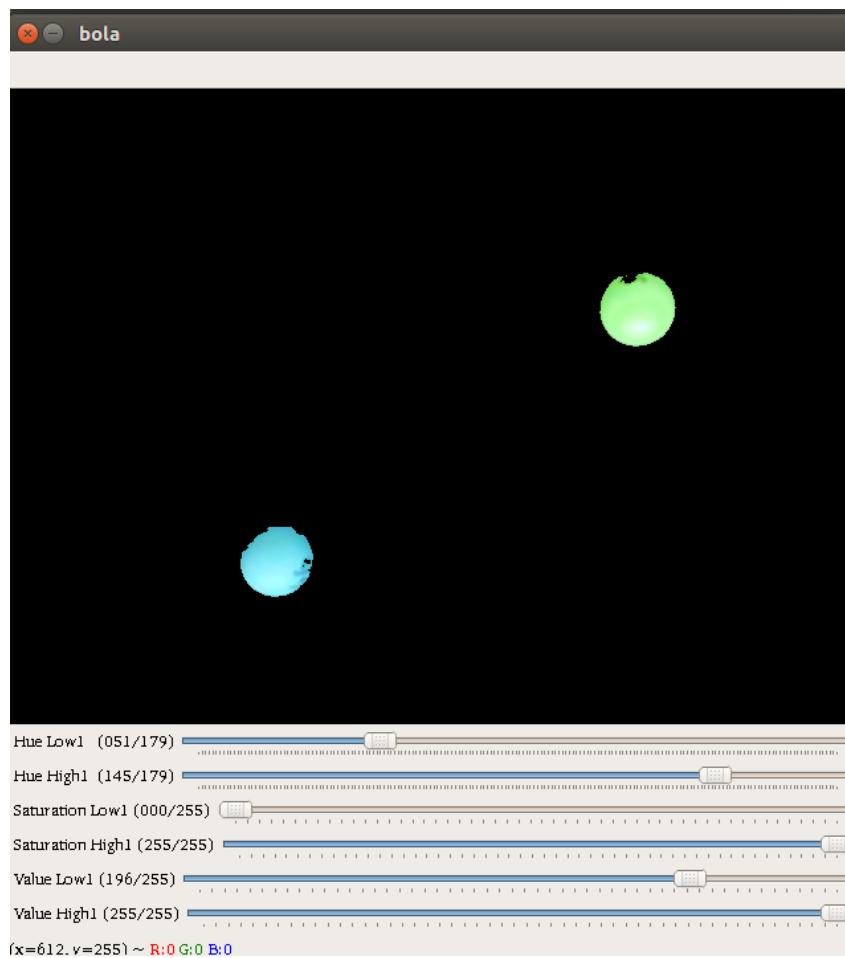
### 3.2.3 Filtro Rango de Colores

Se debe elegir dos rangos entre los cuales se va a detectar la pelota. Esto influye al momento de utilizar una u otra pelota de distinto color. Mediante el programa que se desarrolló llamado RangoColores.py (Anexo B) se puede probar el rango en tiempo real con diferentes pelotas hasta obtener el adecuado. El algoritmo del programa desarrollado se muestra a continuación:



*Figura 31* Algoritmo rango de colores

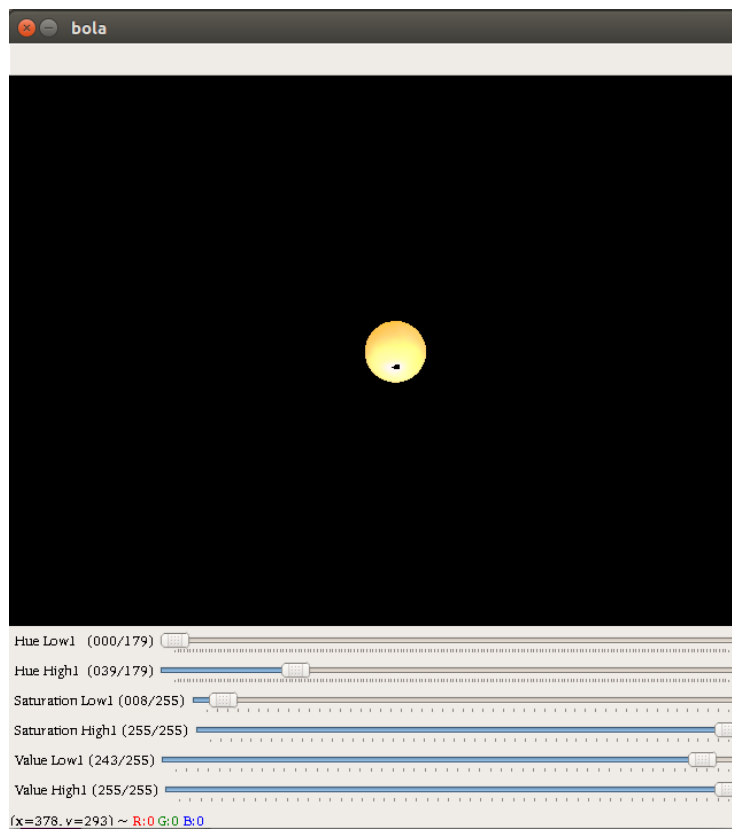
El resultado del programa, cuando se encuentran las tres bolas Verde, Azul y Tomate en la plataforma, para el rango de verde y azul es:



**Figura 32** Rango de colores para Verde y Azul

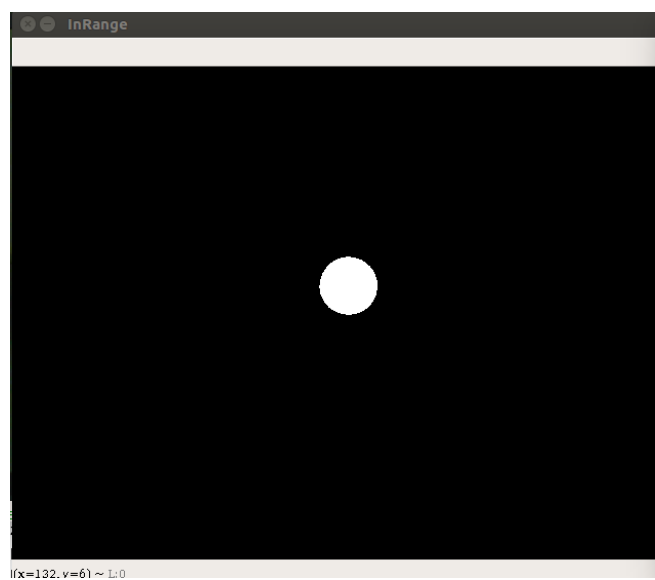
Mientras que en la Figura 33 se muestra un rango para el color tomate:





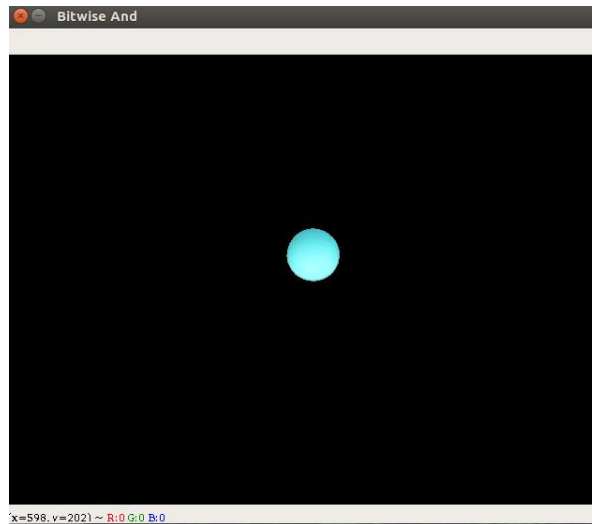
**Figura 33** Rango de Colores para Tomate

La función *In\_Range* de OpenCV, detecta un pixel dentro de un rango de colores previamente establecido, su resultado se muestra en la siguiente figura:



**Figura 34** Función *In Range* de OpenCV

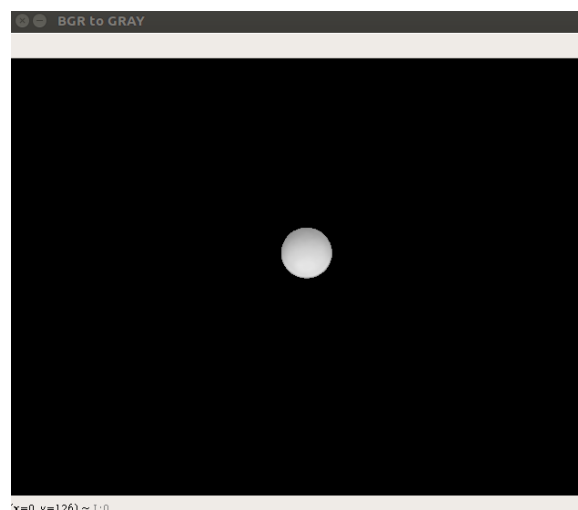
Mediante la función *Bitwise\_and* de OpenCv, se obtiene una imagen binaria donde se escribe un 1 si es el parte del color seleccionado y cero si no lo es, el resultado se muestra a continuación:



*Figura 35* Función *Bitwise\_and* perteneciente a OpenCV

### 3.2.4 Conversión a Escala de Grises

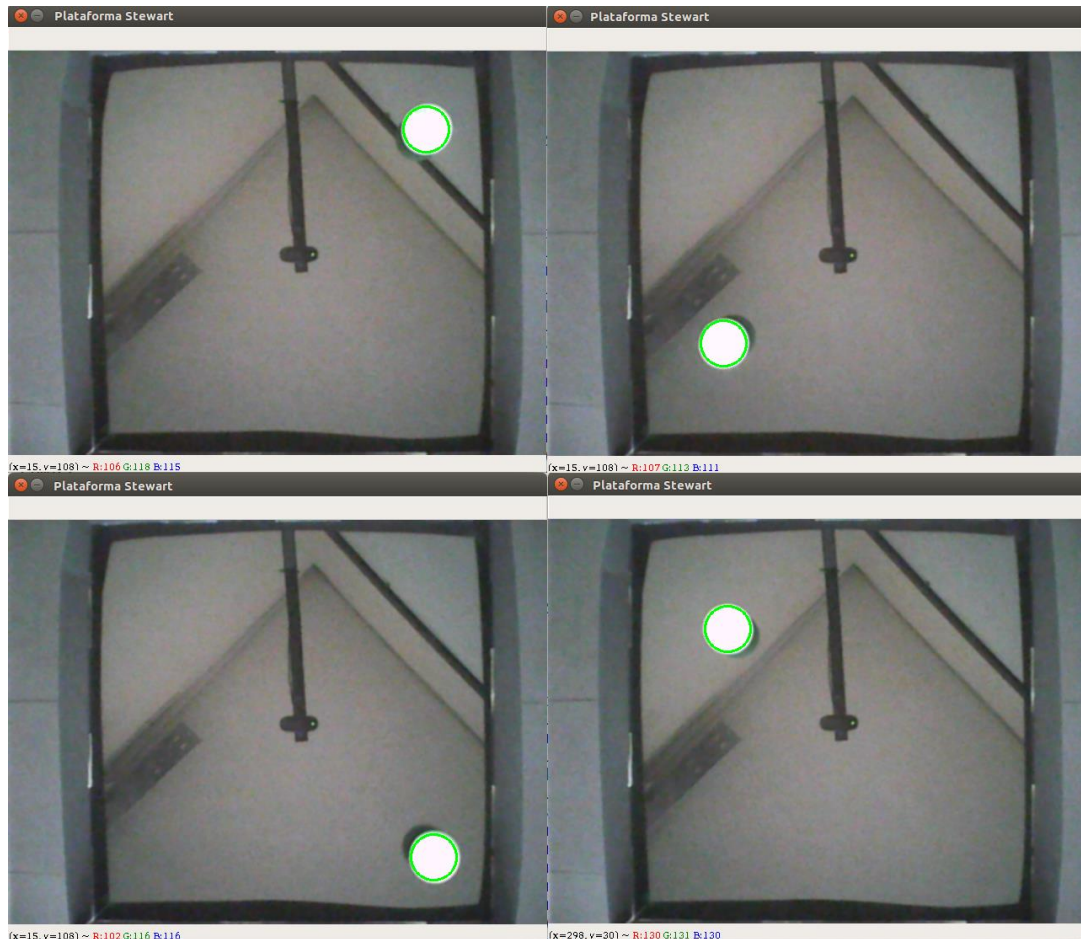
Una vez filtrado el color, se transforma a escala de grises para que sea aún más rápido el proceso de detección de contornos.



*Figura 36* Conversión a Escala de Grises

### 3.2.5 Detección Contorno

OpenCV ofrece la función *findContours* la cual devuelve el centro y radio de una circunferencia en la imagen, para esto se aplica previamente los pasos anteriores con el fin de filtrar cualquier posible ruido y mostrar la bola solamente.



*Figura 37* Detección de contorno de la bola

### 3.2.6 Obtención Posición de la Bola

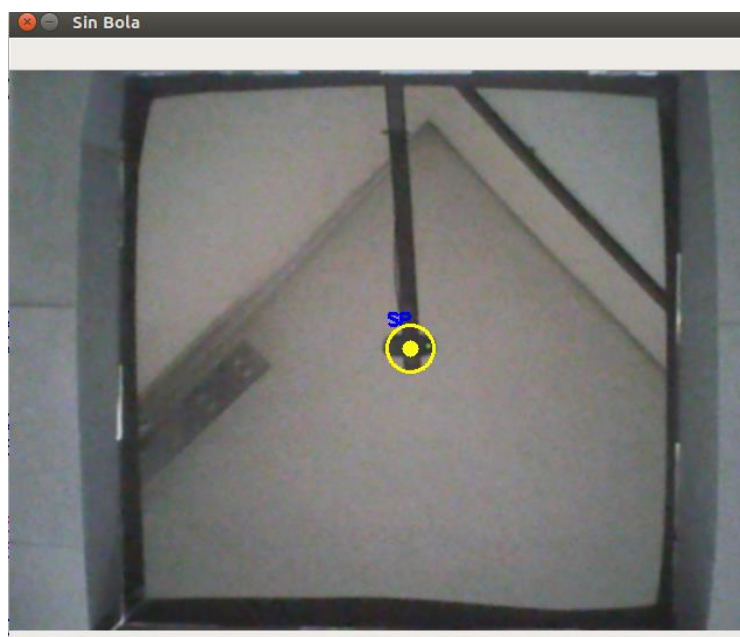
Una vez obtenido el centro y radio de la bola, se procede matemáticamente a hallar la distancia de la bola al centro de la plataforma, o punto de equilibrio deseado (punto de referencia). Posterior a ello, se procede a dibujar el contorno, el centro y una línea desde el punto de referencia hasta el centro de la bola, como se muestra en la Figura 38.



*Figura 38* Obtención posición de la bola

### 3.2.7 Bola inexistente

Cuando no exista bola sobre la plataforma, el robot se posiciona en modo home:  $0^\circ$  Roll y  $0^\circ$  Pitch. El caso se presenta cuando la bola es retirada de la plataforma para ser lanzada nuevamente o cuando simplemente se encienda la plataforma y no haya sido colocada ninguna bola. Se muestra a continuación:



*Figura 39* Bola inexistente en plataforma

### 3.2.8 Filtro de Kalman

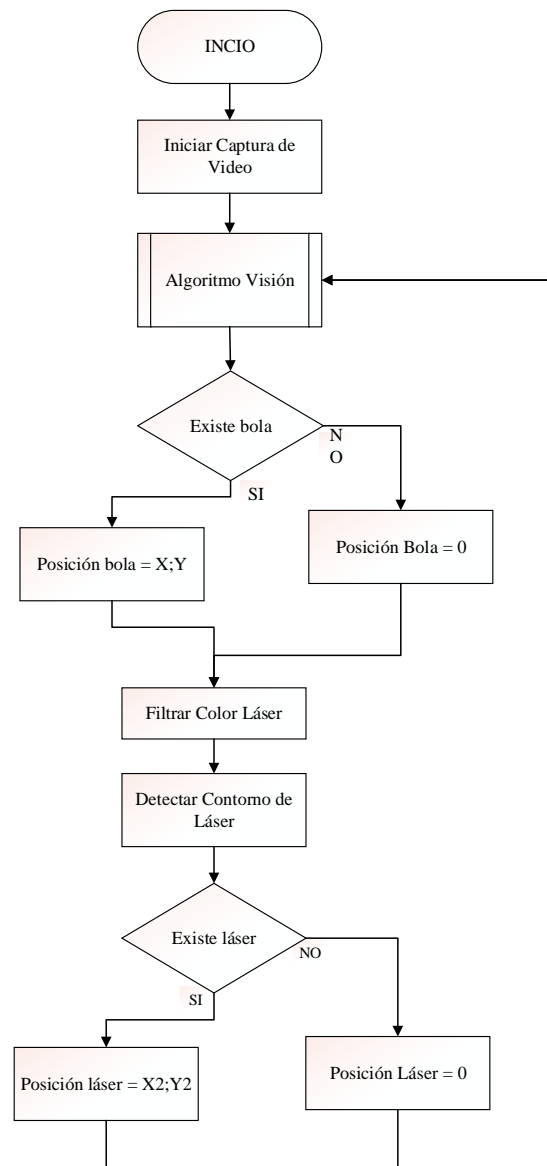
El Filtro de Kalman tiene la capacidad de predecir la posición de un objeto en movimiento, y éste ha sido implementado en la plataforma, a manera de experimento para realizar ciertas pruebas de control con este parámetro como entrada. Lo que se gana es tiempo al tomar como referencia la posición que tendrá la bola y no la que tiene en un instante de tiempo. El resultado del filtro se muestra en la Figura 40 donde el círculo verde es el de la bola misma, y el amarillo es el calculado con el filtro de Kalman.



*Figura 40* Filtro de Kalman

### 3.2.9 Detección Láser Rojo como Punto de referencia

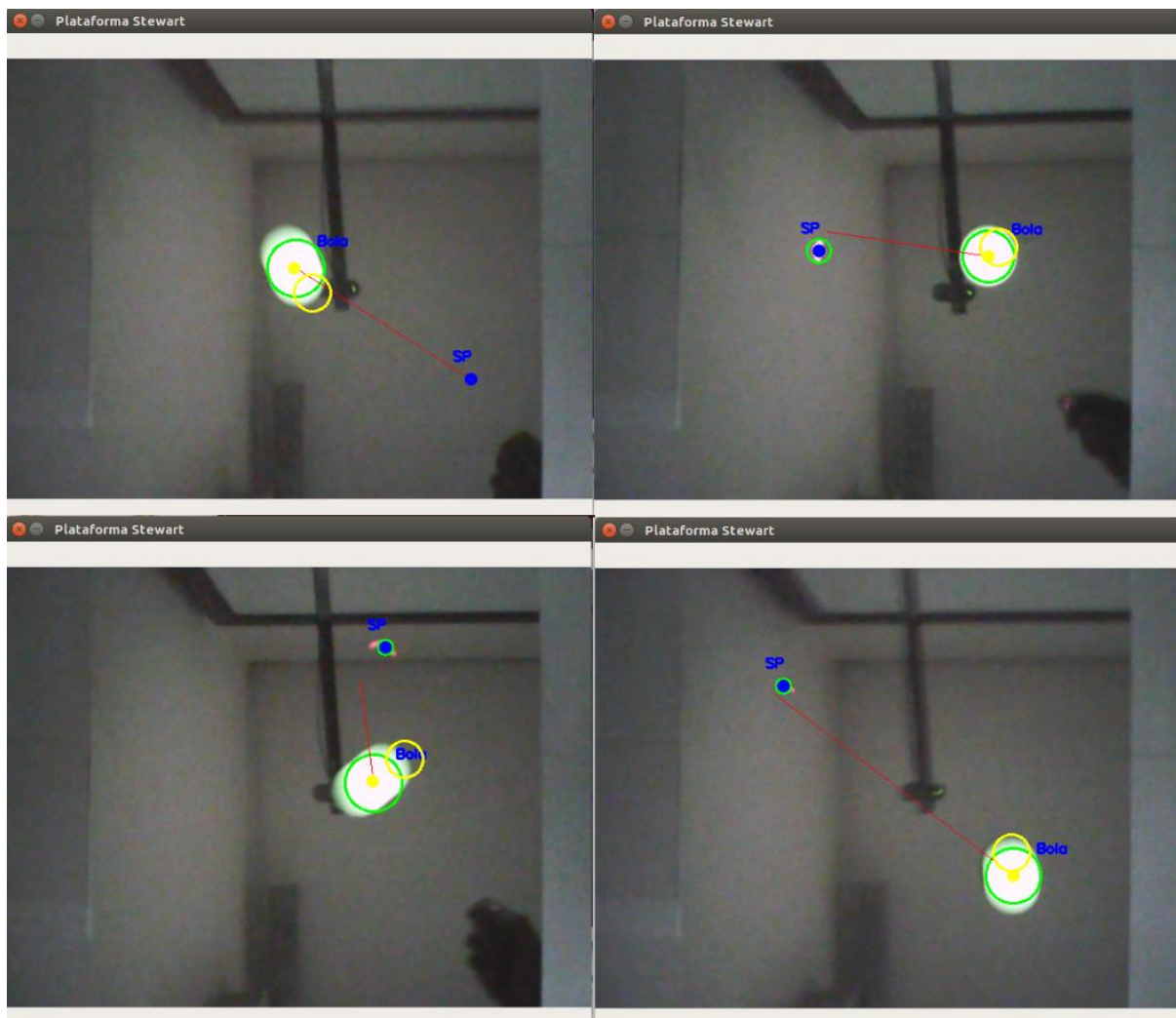
Se implementó el código para detectar un segundo color (rojo) al mismo tiempo que se detecta la bola (Anexo C), con el objetivo de variar el punto de referencia mediante el uso de un puntero laser a cualquier punto de la plataforma. El algoritmo del desarrollado se muestra a continuación:



**Figura 41** Algoritmo detección láser

El resultado del programa implementado en el robot se observa en la Figura 42





*Figura 42* Detección láser rojo

### 3.3 Control basado en lógica difusa.

Varios fueron los intentos por diseñar el controlador difuso, sobre todo por la sintonización de las funciones de pertenencia, esto es quizá, la parte más importante al momento de diseñar el controlador.

#### 3.3.1 Entradas y Salidas

Para controlar la bola en plataforma se tiene las siguientes entradas y salidas con sus respectivas unidades.

Entradas:

- Posición de la bola (píxeles).
- Velocidad de la bola (píxel por segundo).

Salidas:

- Ángulo de la plataforma (grados).

### **3.3.2 Conjuntos difusos y funciones de pertenencia**

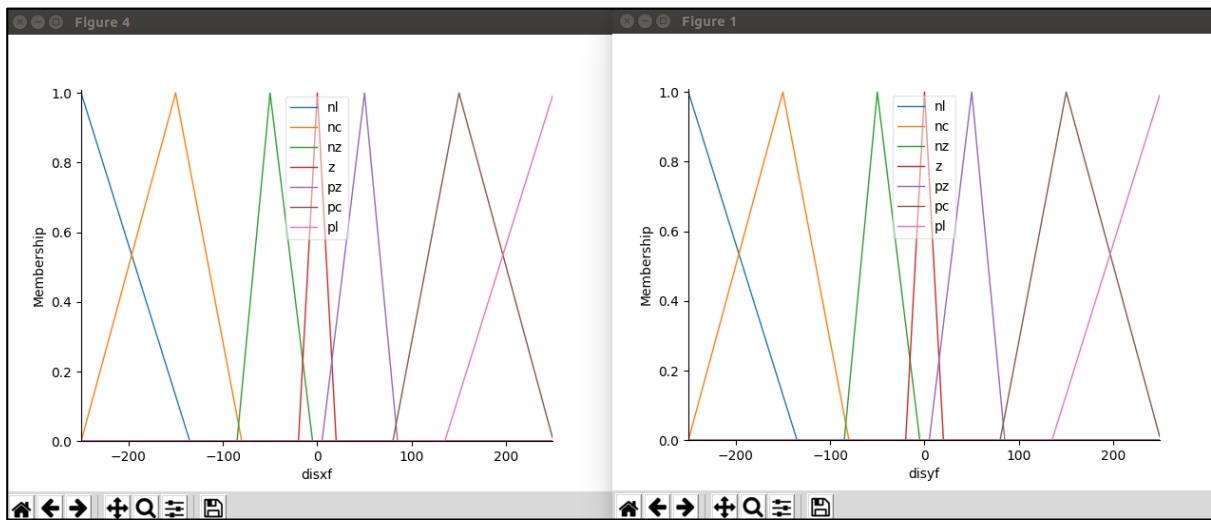
Los conjuntos difusos con sus funciones de pertenencia fueron las siguientes, tomando en cuenta el tamaño de la plataforma, la máxima velocidad de la bola y la máxima inclinación angular de la plataforma.

Distancia para X e Y:

La nomenclatura elegida para este conjunto es la siguiente:

- NL: negativo lejos
- NC: negativo cerca
- NZ: negativo cero
- Z: cero
- PZ: positivo cero
- PC: positivo cerca
- PL: positivo lejos





**Figura 43** Funciones de pertenencia Distancia

Los valores se observan a continuación:

**Tabla 6**

*Conjunto Difuso Distancia*

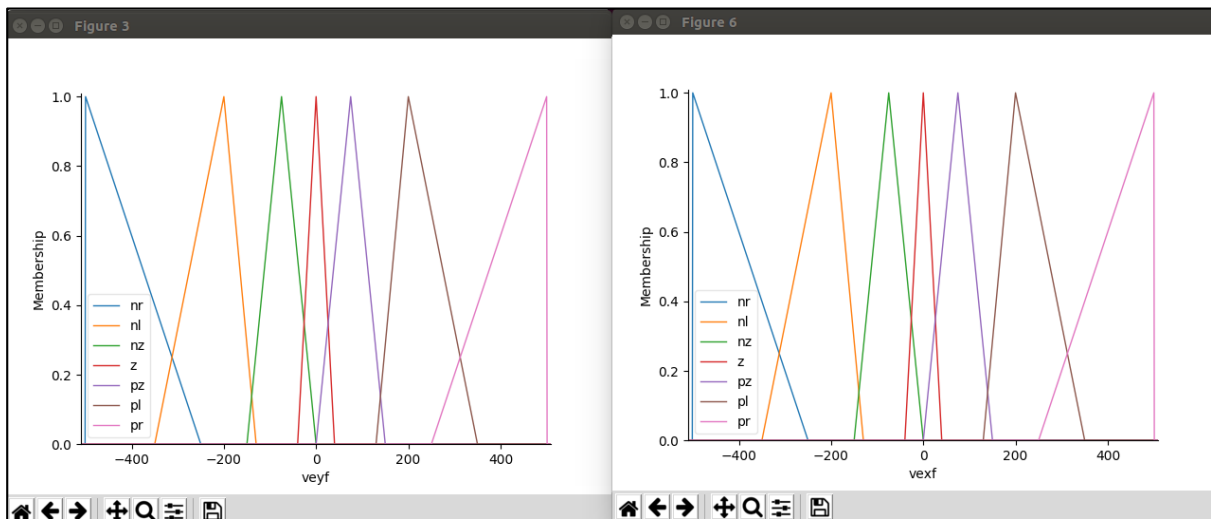
		Valor (pixel)		
		min	medio	max
<b>Conjunto Difuso Variable Distancia</b>	<b>nl</b>	-250	-135	250
	<b>nc</b>	-150	-80	250
	<b>nz</b>	-85	-5	250
	<b>z</b>	-20	20	250
	<b>pz</b>	5	85	250
	<b>pc</b>	80	250	250
	<b>pl</b>	135	250	250

Velocidad para X e Y:

La nomenclatura elegida para este conjunto es la siguiente:

- NR: negativo rápido
- NL: negativo lento
- NZ: negativo cero
- Z: cero
- PZ: positivo cero
- PL: positivo lento

- PR: positivo rápido



**Figura 44** Funciones de pertenencia Velocidad

Los valores se observan a continuación:

**Tabla 7**

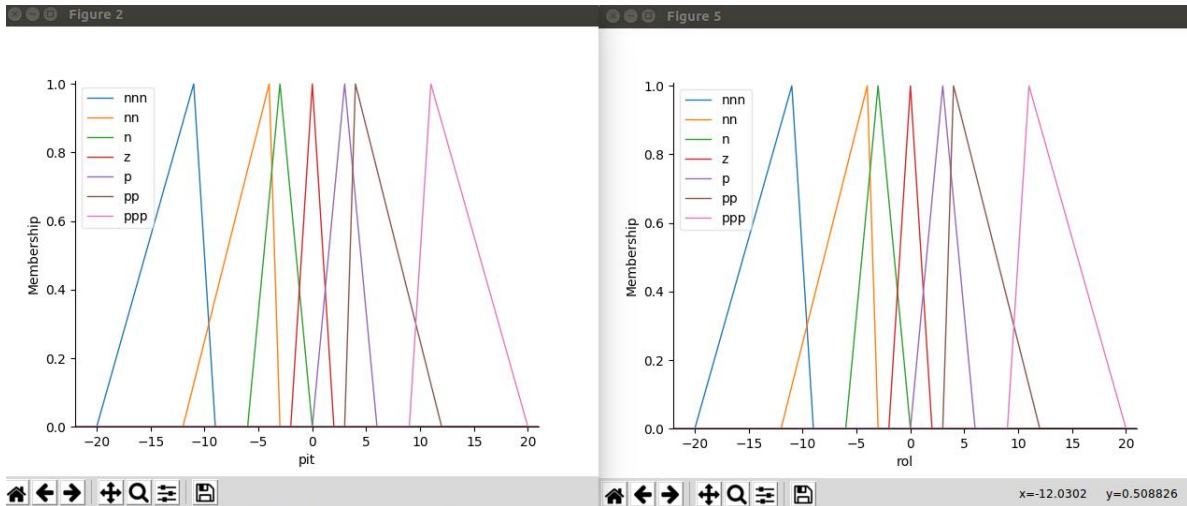
*Conjunto Difuso Velocidad*

		Valor (pixel/s)		
		min	medio	max
<b>Conjunto Difuso Variable Velocidad</b>	<b>nr</b>	-500	-250	500
	<b>nl</b>	-200	-130	350
	<b>nz</b>	-75	0	150
	<b>z</b>	-40	0	40
	<b>pz</b>	0	75	150
	<b>pl</b>	130	200	350
	<b>pr</b>	250	500	500

Ángulo para X y Y:

La nomenclatura elegida para este conjunto es la siguiente:

- NNN: negativo negativo negativo
- NN: negativo negativo
- N: negativo
- Z: cero
- P: positivo
- PP: positivo positivo
- PPP: positivo positivo positivo



**Figura 45** Funciones de pertenencia Ángulos

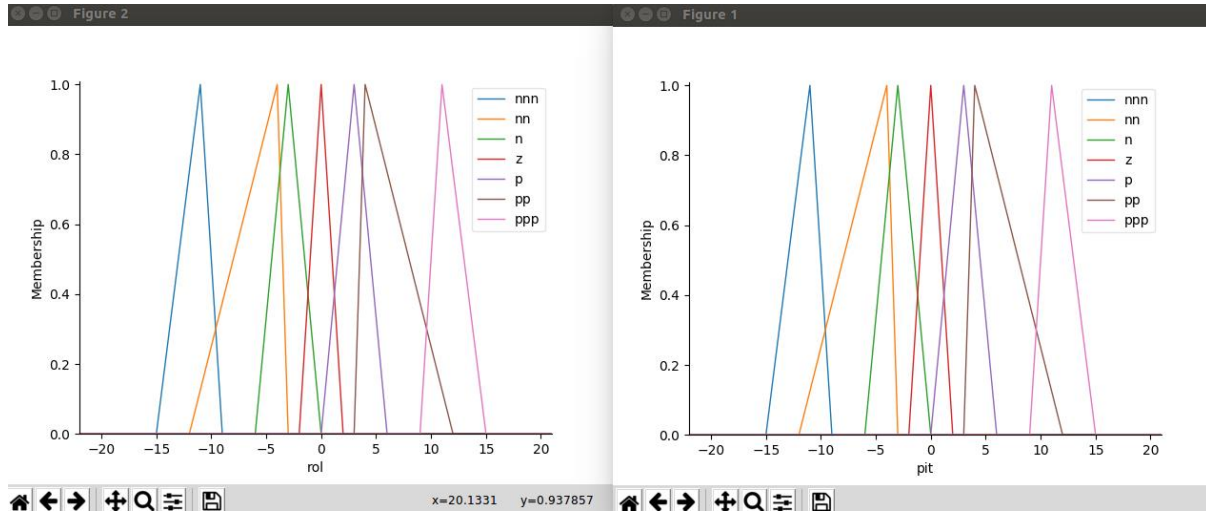
Los valores se observan a continuación:

**Tabla 8**  
*Conjunto Difuso Ángulos*

	Valor (grados)		
	min	medio	max
<b>nnn</b>	-20	-11	-9
<b>nn</b>	-12	-4	-3
<b>n</b>	-6	-3	0
<b>z</b>	-2	0	2
<b>p</b>	0	3	6
<b>pp</b>	3	4	12
<b>ppp</b>	9	11	20

Para el programa desarrollado que permite la variación del punto de referencia mediante láser (anexo H), se realizó una variación en el conjunto de ángulos, para suavizar la acción de

control de la plataforma, debido a que, si el punto de referencia se ubica cercano al filo, la bola se va a caer, por ello se sacrifica el tiempo de estabilización y llegada al punto, pero se asegura que no va a botar la bola fuera de la misma. Se observa en Figura 46:



**Figura 46** Variación funciones de pertenencia Ángulos

Los valores se observan a continuación:

**Tabla 9**

*Variación Conjunto Difuso Ángulos*

		Valor (grados)		
		min	medio	max
<b>Conjunto Difuso Variable Ángulos</b>	<b>nnn</b>	-15	-11	-9
	<b>nn</b>	-12	-4	-3
	<b>n</b>	-6	-3	0
	<b>z</b>	-2	0	2
	<b>p</b>	0	3	6
	<b>pp</b>	3	4	12
	<b>ppp</b>	9	11	15

Las funciones de pertenencia fueron triangulares, con ellas se obtuvieron los mejores resultados. Las funciones tipo Gaussianas no son recomendadas para este sistema.

### 3.3.3 Reglas difusas

Las reglas difusas se diseñaron en base al conocimiento del sistema y su dinámica, mediante observación y varias pruebas de movimiento articular. Se muestran en la Tabla 10:

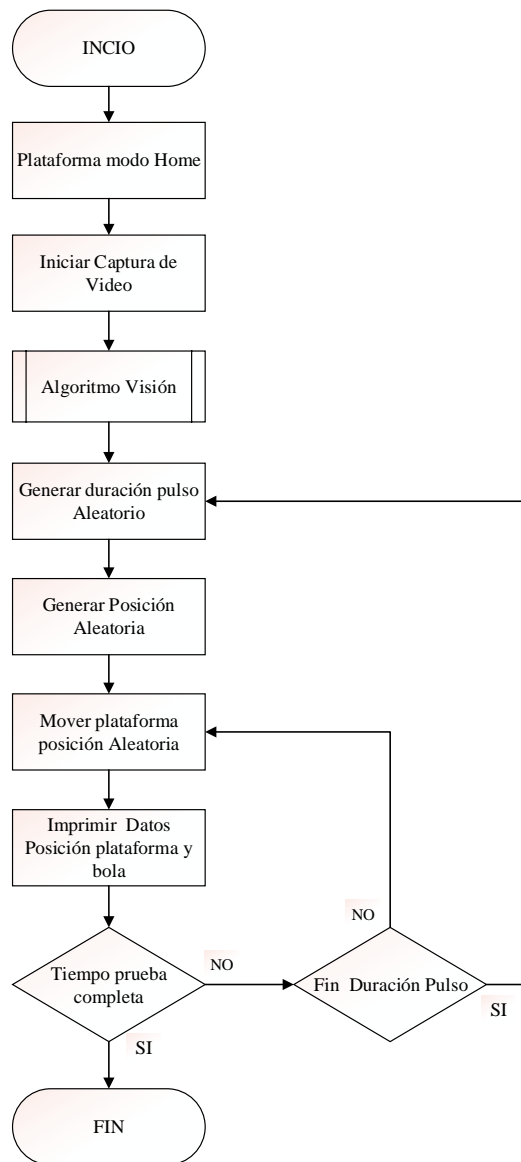
**Tabla 10**  
*Reglas difusas*

		Velocidad X e Y (pixel/s)						
		<b>nr</b>	<b>nl</b>	<b>nz</b>	<b>z</b>	<b>pz</b>	<b>pl</b>	<b>pr</b>
<b>Distancia X e Y (pixel)</b>	<b>nl</b>	ppp	ppp	ppp	pp	n	n	nn
	<b>nc</b>	ppp	ppp	pp	p	z	z	n
	<b>nz</b>	ppp	pp	p	z	z	n	nn
	<b>z</b>	ppp	pp	p	z	n	nn	nnn
	<b>pz</b>	pp	p	z	z	n	nn	nnn
	<b>pc</b>	p	z	z	n	nn	nnn	nnn
	<b>pl</b>	pp	p	p	nn	nnn	nnn	nnn

## 3.4 Control basado en redes neuronales.

### 3.4.1 Obtención de datos

Previo a aplicar el algoritmo de control neuronal es necesario obtener datos del sistema de entrada y salida, en este caso la distancia de la bola al punto de referencia y la posición angular de la plataforma. Para ello se desarrolló un programa que mueve a la plataforma aleatoriamente (Anexo D). El algoritmo del programa se presenta a continuación:



**Figura 47** Algoritmo toma de datos

La mínima variación que se puede dar al sistema es de 0.1s, debido al tiempo que tarda en procesar la imagen y obtener la posición de la bola. Por ello este tiempo es tomado en cuenta para la mínima frecuencia de entrada válida al sistema. La bola tarda no más de 1s en cruzar de lado a lado la plataforma por lo que ésta es tomada en cuenta como la frecuencia mayor.

Con esto, se envían órdenes de movimiento angular aleatorios a la plataforma que van entre los 0.1s hasta 1s con el fin de que la bola se mueva por toda la plataforma, cubriendo los 4 filos y también para que varíe su velocidad de movimiento.

En la siguiente tabla se resume los parámetros del experimento realizado para la toma de datos utilizados:

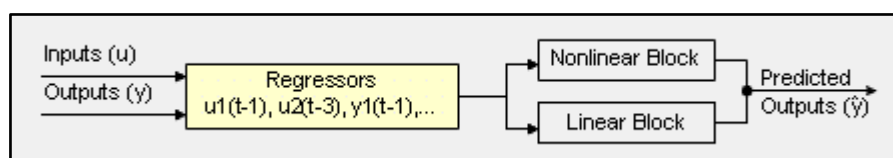
**Tabla 11**  
*Parámetros para experimento de toma de datos*

Experimento	Toma de datos del robot
Cantidad de datos aleatorios	3000
Datos de Entrada	Ángulo de la plataforma en grados
Datos de Salida	Posición de la bola en pixeles
Frecuencia de muestreo	0.1s
Pulso mínimo y máximo de entrada	0.1s – 1s
Tiempo total de experimento	5 minutos

### 3.4.2 Modelo dinámico lineal de auto regresión con variables exógenas ARX

Se plantea el modelo ARX para modelar el robot Stewart con la aplicación bola en plataforma, probando distintos grados en sus polinomios y cantidad de regresores para buscar el resultado que mejor se aproxime al modelo.

Mediante el software Matlab se realiza el modelamiento. En la siguiente figura se observa la estructura ARX:



**Figura 48** Estructura ARX de Matlab

Primero se debe cargar datos de entrada y de salida, ya sea mediante el *toolbox* de identificación de Matlab o por código. Para este trabajo se lo ha hecho por código:

```

>> p=iddata(roll,x,0.1)

p =

Time domain data set with 3000 samples.
Sample time: 0.1 seconds

Outputs    Unit (if specified)
  y1

Inputs     Unit (if specified)
  u1

```

**Figura 49** Ingreso datos entrada salida para identificación

Donde roll representa los datos de ángulos de la plataforma como entrada y x representa la distancia de la bola al punto de referencia como salida.

Luego de ello se procede a identificar con el modelo ARX, con distintos parámetros en sus grados y cantidad de regresores. Se observa en la siguiente figura:

<pre> &gt;&gt; m1=arx(p,[2 2 1])  m1 = Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t) A(z) = 1 - 0.7616 z^-1 + 0.07438 z^-2  B(z) = -0.06451 z^-1 + 0.05947 z^-2  Sample time: 0.1 seconds  Parameterization:   Polynomial orders: na=2 nb=2 nk=1   Number of free coefficients: 4 Status: Estimated using ARX on time domain data "p". Fit to estimation data: 29.52% (prediction focus) FPE: 66.03, MSE: 65.77 </pre>	<pre> &gt;&gt; m2=arx(p,[4 4 3])  m2 = Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t) A(z) = 1 - 0.7568 z^-1 + 0.03739 z^-2 + 0.04616 z^-3 + 0.008571 z^-4  B(z) = 0.2153 z^-3 - 0.4754 z^-4 + 0.248 z^-5 + 0.02011 z^-6  Sample time: 0.1 seconds  Parameterization:   Polynomial orders: na=4 nb=4 nk=3   Number of free coefficients: 8 Status: Estimated using ARX on time domain data "p". Fit to estimation data: 29.66% (prediction focus) FPE: 66.02, MSE: 65.5 </pre>
--	---

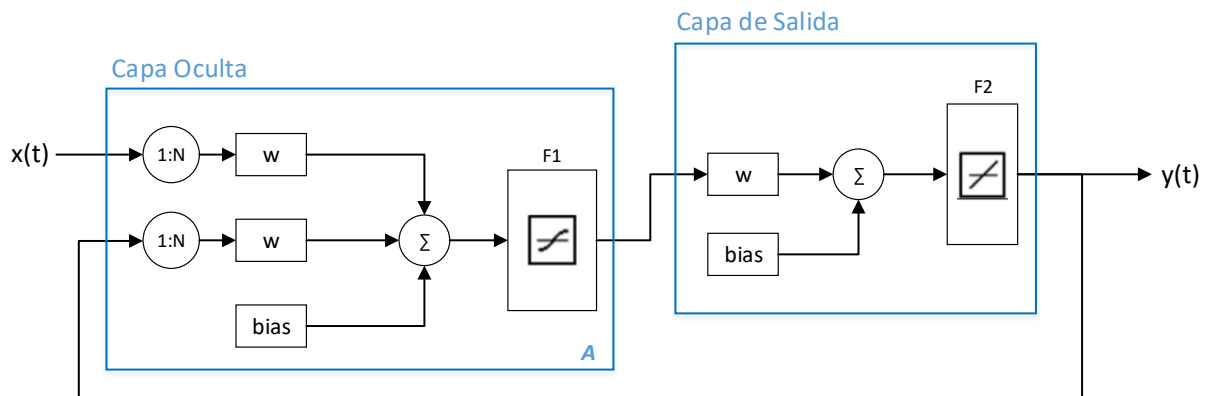
**Figura 50** Identificación ARX

### 3.4.3 Modelo dinámico no lineal de auto regresión con variables exógenas NARX

Se plantea el modelo NARX para modelar el robot Stewart con la aplicación bola en plataforma probando distintas cantidades de neuronas y regresores para encontrar el resultado



que mejor se aproxime al modelo. La estructura del modelo en Matlab se muestra a continuación:



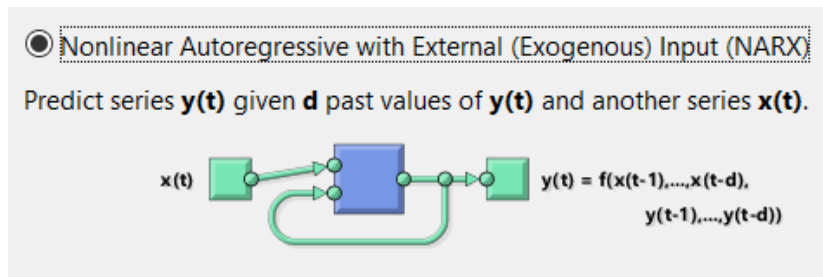
**Figura 51** Estructura NARX

Donde:

- $x(t)$  = entrada de datos
- 1:N = número de regresores
- W = pesos a calcular
- Bias = valor de ajuste *offset*
- F1 = función de activación Sigmoidea
- F2 = función de activación Lineal
- A = número de neuronas en capa oculta
- $y(t)$  = salida de la planta

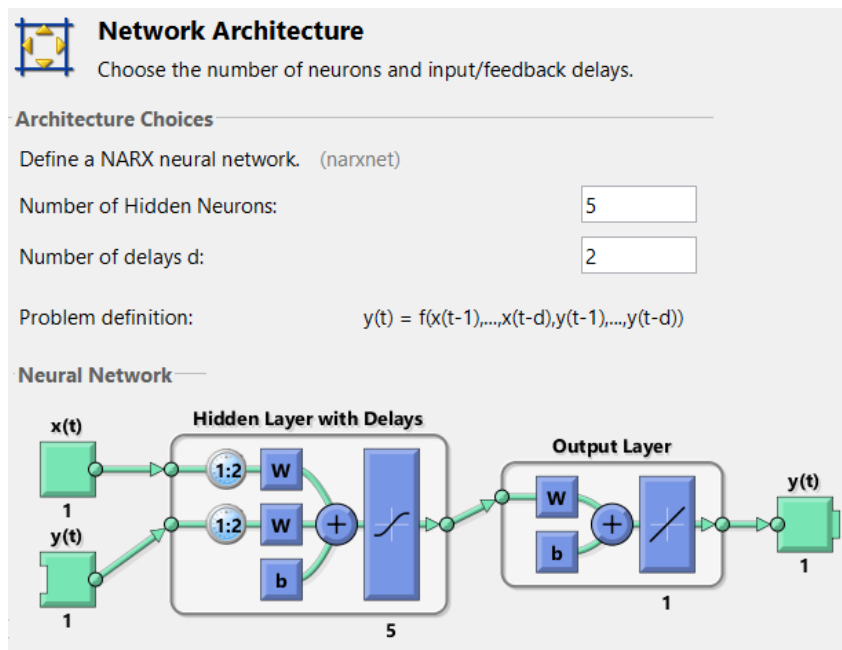
De igual manera se debe cargar los datos de entrada y salida a Matlab, mediante código o por medio de la sección “Variable” con las herramientas de importación de datos o carga de variables.

Una vez cargados los datos, se procede a cargar en el toolbox de “Neural Net Time Series” seleccionando el modelo NARX como se muestra a continuación



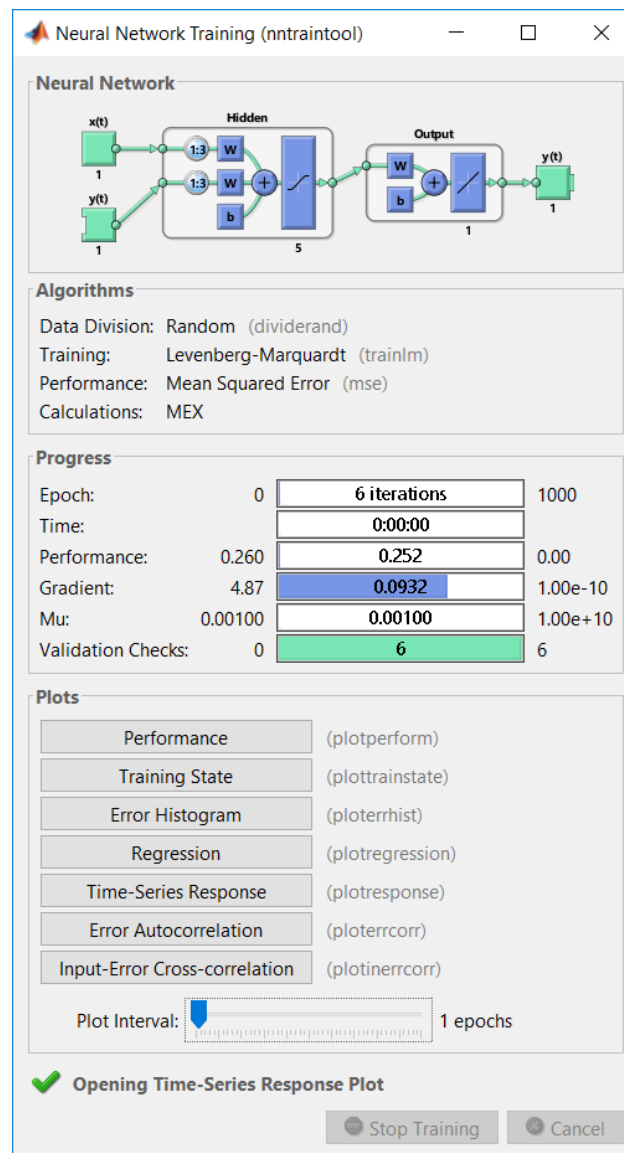
**Figura 52** Neural Net Time Series Toolbox de Matlab

Una vez seleccionado el tipo de red y las respectivas entradas y salidas, se elige el número de neuronas en la capa oculta y el número de regresores, para esta planta se han elegido 5 neuronas en la capa oculta y 2 regresores.



**Figura 53** Número de neuronas y regresores

Luego del entrenamiento de la red se obtiene los resultados del modelo de la red neuronal NARX, junto con su rendimiento, respuesta en series de tiempo, histogramas de error, etc.



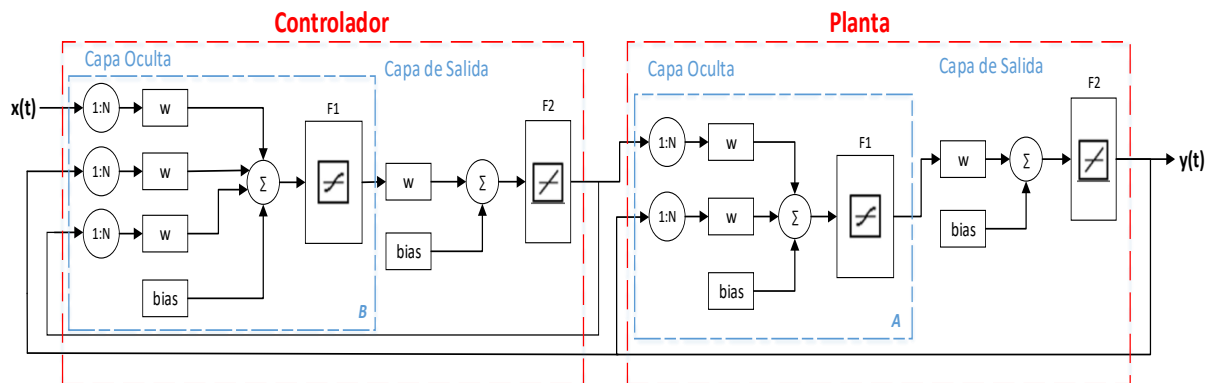
**Figura 54** Entrenamiento de red neuronal

### 3.4.4 Estrategia de control por Modelo de Referencia

Esta estrategia usa dos redes neuronales, la una para obtener el modelo de la planta, que es la que se desarrolló en la sección 3.4.3 y otra red para el controlador (Anexo I).

El modelo de referencia se lo toma de Matlab con el comando *refmodel\_dataset*, esto es un conjunto de 2000 datos de entrada y salida de un brazo robótico que sirve como modelo para controlar sistemas dinámicos no lineales.

La estructura del controlador neuronal se muestra a continuación:

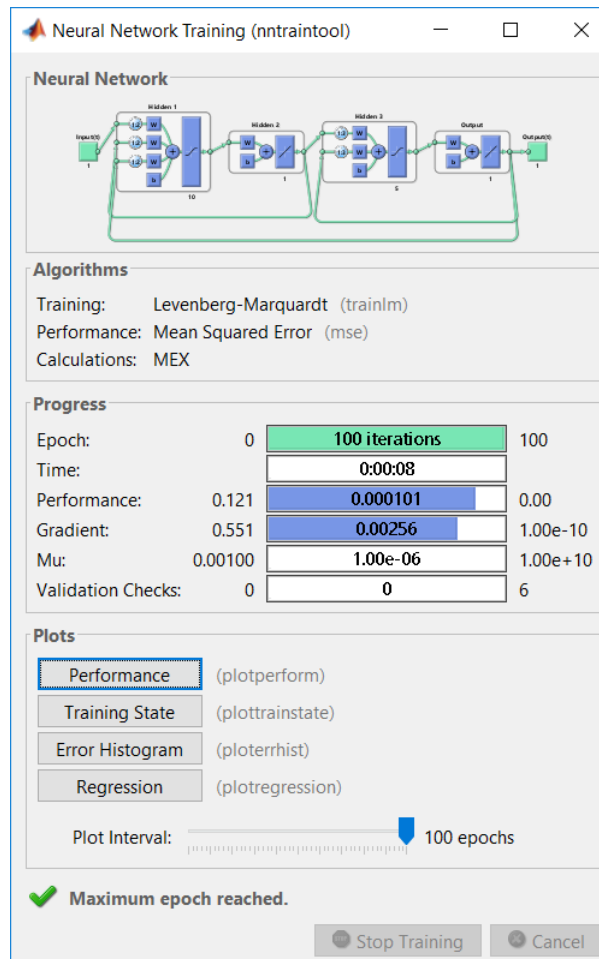


**Figura 55** Estructura controlador neuronal

Donde:

- $x(t)$  = entrada de datos
- $1:N$  = número de regresores controlador y planta
- $W$  = pesos a calcular
- Bias = valor de ajuste *offset*
- $F1$  = función de activación Sigmoidea
- $F2$  = función de activación Lineal
- $A$  = número de neuronas en capa oculta planta
- $B$  = número de neuronas en capa oculta controlador
- $y(t)$  = salida de la planta

Finalmente se procede a entrenar el controlador en conjunto con la planta modelada, mediante el *nntaintool* de Matlab, de donde se obtiene de igual manera, el rendimiento, histogramas de error, respuesta a series de tiempo, como se observa a continuación.



**Figura 56** Entrenamiento de controlador + planta

## CAPÍTULO IV

### PRUEBAS Y ANÁLISIS DE RESULTADOS

#### 4.1 Exactitud de la planta

Se realizaron 30 pruebas de orientación para 5 coordenadas, los resultados obtenidos para cada orientación en promedio fueron:

**Tabla 12**  
*Pruebas Exactitud ISO9283*

Orientaciones		1	2	3	4	5
Enviados (grados)	Roll	6	0	-2	15	5
	Pitch	-15	11	0	-8	14
	Yaw	9	0	-4	6	-13
Promedio Medidos (grados)	Roll	4,8	-1,1	-3,2	16,4	3,6
	Pitch	-	11,9	1,3	-9,1	15,2
	Yaw	16,1	8,1	-1,1	-4,9	4,7
						14,2

Aplicando la fórmula 3.1:

**Tabla 13**  
*Resultados pruebas exactitud*

Orientaciones	1	2	3	4	5	Promedio
<b>Apo</b> (grados)	1,86	1,797	1,985	2,205	2,2	2.009

La exactitud al punto de prueba es de 2.009 grados.

Esto significa que cuando se ubique la plataforma en un punto Roll, Pitch; la posición real será  $\text{Roll} \pm 2.009$  ;  $\text{Pitch} \pm 2.009$  grados. Esta exactitud es un factor que afecta a la acción de control, sin embargo, se ha podido controlar y tener buenos resultados, que serán presentados más adelante.

El peso y tipo de bola con la que se trabaja se eligió debido a este factor de exactitud. Se utilizan pelotas de ping pong que son relativamente livianas a comparación con las bolas metálicas presentadas en la sección 3.1.1 literal 5.

Si se centra a la plataforma en Roll=0 y Pitch=0, los valores reales son  $0 \pm 2.009$ . Una pelota liviana con uno o dos grados de inclinación no va a moverse, mientras que una pesada si, será más fácil para la pesada vencer su momento de inercia con una pequeña inclinación y rodar. Es por ello que se ha optado por utilizar las pelotas livianas de ping pong.

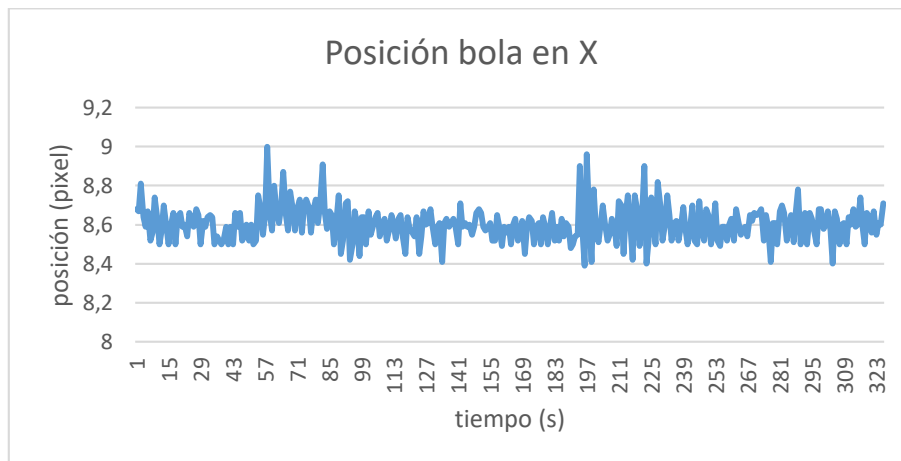
## 4.2 Algoritmo de Visión

Se calculó en promedio a cuantos FPS se captura la imagen y se obtuvo como media 12FPS como velocidad de captura. Este valor bajo se debe al tiempo que tarda el programa en detectar el contorno de la bola, junto con los filtros y transformaciones de color que se realizan previamente.

Respecto a la detección de la bola, se presenta cierto ruido al momento de detectar el centro, para lo cual se diseñó un experimento para evaluar el nivel de ruido y estos fueron los resultados:

- Experimento: la bola está quieta en el punto  $X = 8.6$  ;  $Y = 4.5$  por un tiempo de 30 segundos y se obtiene la posición mediante visión artificial.

Posición en X:

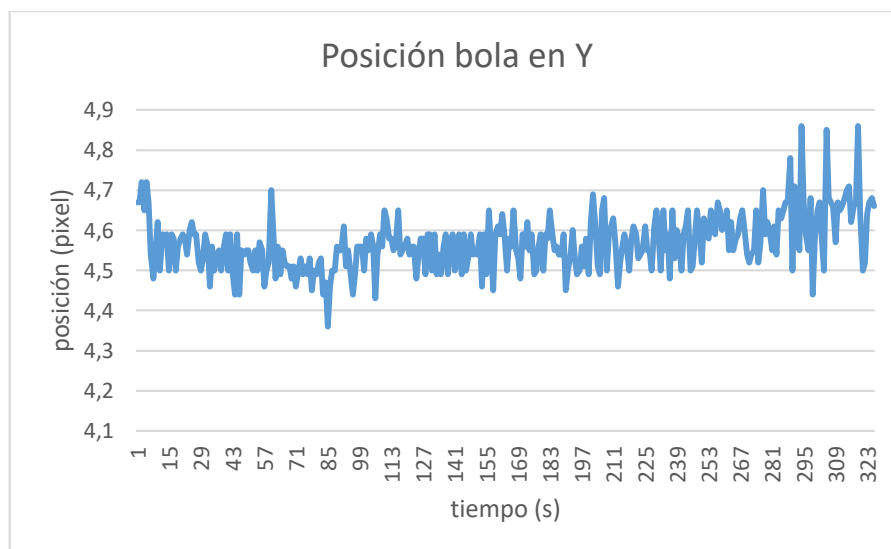


**Figura 57** Posición bola en x

Media: 8.60px

Desviación: 0.09px

Posición en Y:



**Figura 58** Posición bola en y

Media: 4.57px

Desviación: 0.07px



En ambos casos, la desviación es menor a 0.1 pixel. Si bien existe ruido en la detección de posición de la bola, este no afecta a las acciones de control, sin embargo, ha sido analizado y evaluado.

### 4.3 Controlador Difuso

A continuación, se presentan los resultados obtenidos de varias pruebas, partiendo de distintos puntos de la plataforma y con distinta velocidad de la bola. Las variables de interés, que miden el rendimiento de la acción de control son:

- Ts: tiempo que la plataforma tarda en dejar quieta la bola medido en segundos.
- Pos Final: posición final de la bola con respecto al punto de referencia medida en pixeles.
- Ess: Error en estado estable, respecto al punto de referencia medido en porcentaje. Para este cálculo se toma en cuenta el máximo error que la bola podría tener respecto al punto de referencia una vez equilibrada, para lo cual existen dos casos:
  - a) Cuando el punto de referencia está en el centro, como se observa en la línea roja de la Figura 59:



**Figura 59** Máximo error punto de referencia en centro

La fórmula aplicada:

$$Ess = \frac{|Pos\ final| * 100}{D\_max} \quad (4.1)$$

Donde:

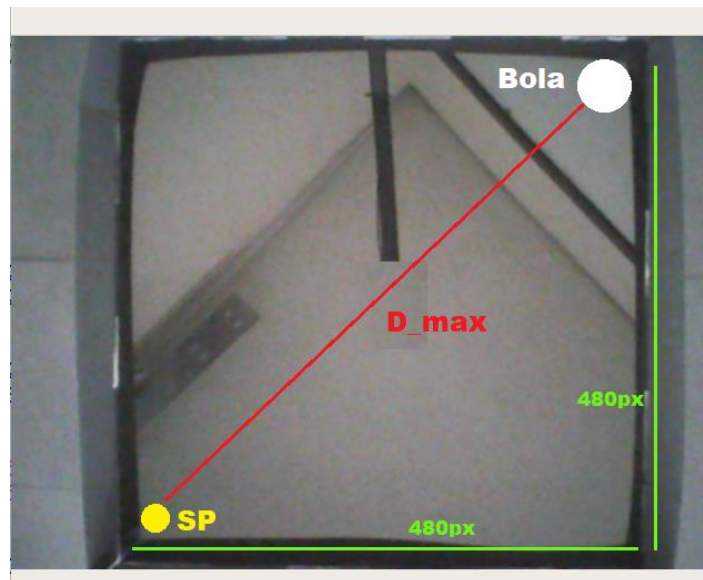
$$D_{max} = \sqrt{240^2 + 240^2} = 339.4 \quad (4.2)$$

Siendo 240x240px la mitad de ancho de la plataforma vista desde la cámara y por tanto el máximo error.

Por lo que resulta:

$$Ess = |Pos\ final| * 0.295 \quad (4.3)$$

- b) Cuando el punto de referencia es variable, como se observa en la línea roja de la Figura 60:



**Figura 60** Máximo error punto de referencia variable

La fórmula aplicada:

$$Ess = \frac{|Pos\ final| * 100}{D\_max} \quad (4.4)$$

Donde:

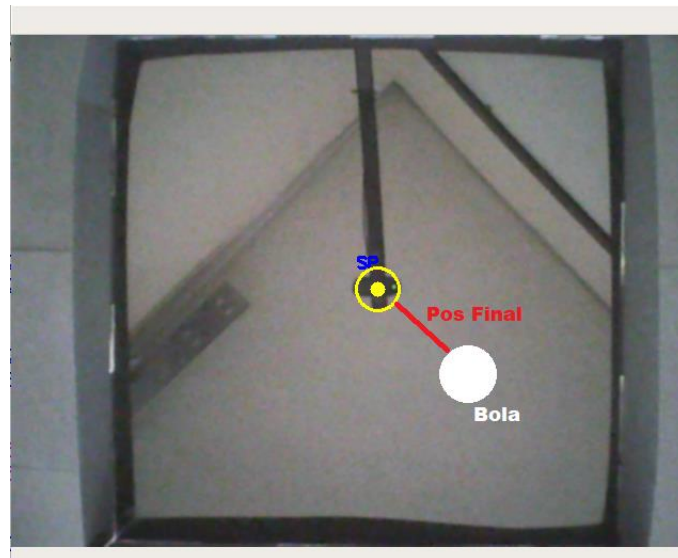
$$D_{max} = \sqrt{480^2 + 480^2} = 678.8 \quad (4.5)$$

Siendo 480x480px el ancho de la plataforma vista desde la cámara y por tanto el máximo error.

Por lo que resulta:

$$Ess = |Pos\ final| * 0.147 \quad (4.6)$$

Donde  $|Pos\ final|$  representa la distancia del punto de referencia a la bola una vez equilibrada como se observa a continuación:



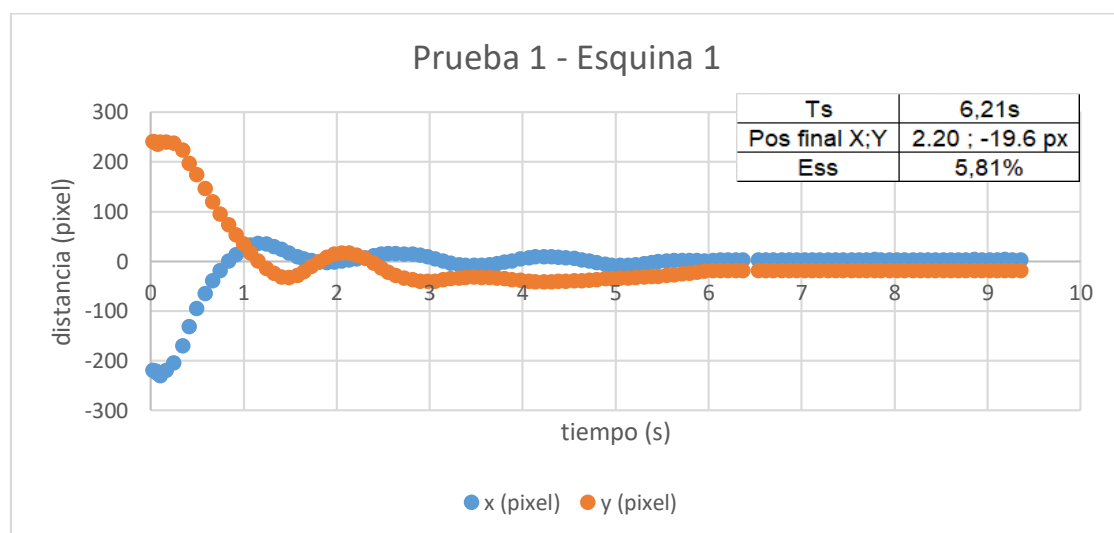
**Figura 61** Error en estado estable

#### 4.3.1 Control difuso con punto de referencia fijo en el centro

Tomando en cuenta la fórmula 4.3 para el cálculo del error, las pruebas realizadas con el programa desarrollado (anexo F) partieron de:

1. La pelota ubicada en una esquina de la plataforma

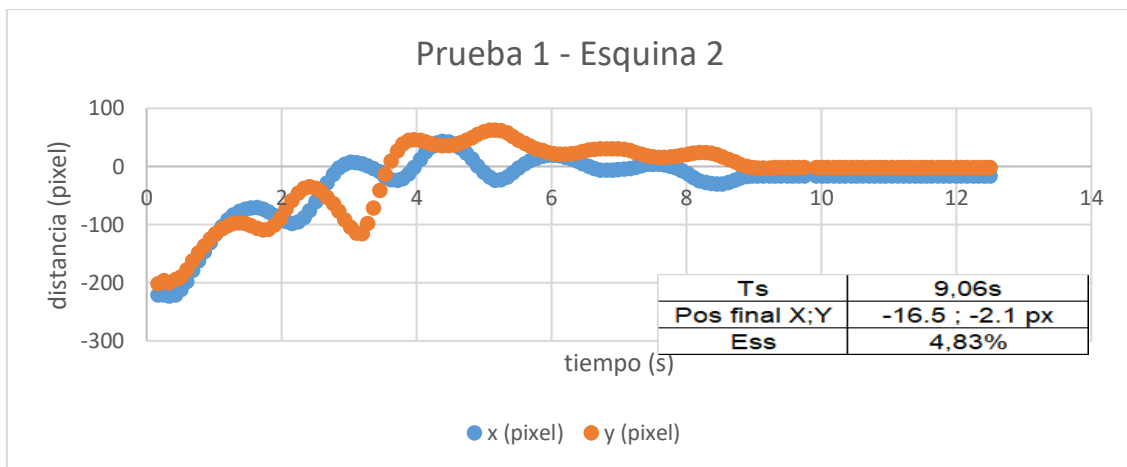
Esquina 1



**Figura 62** Prueba 1 esquina 1

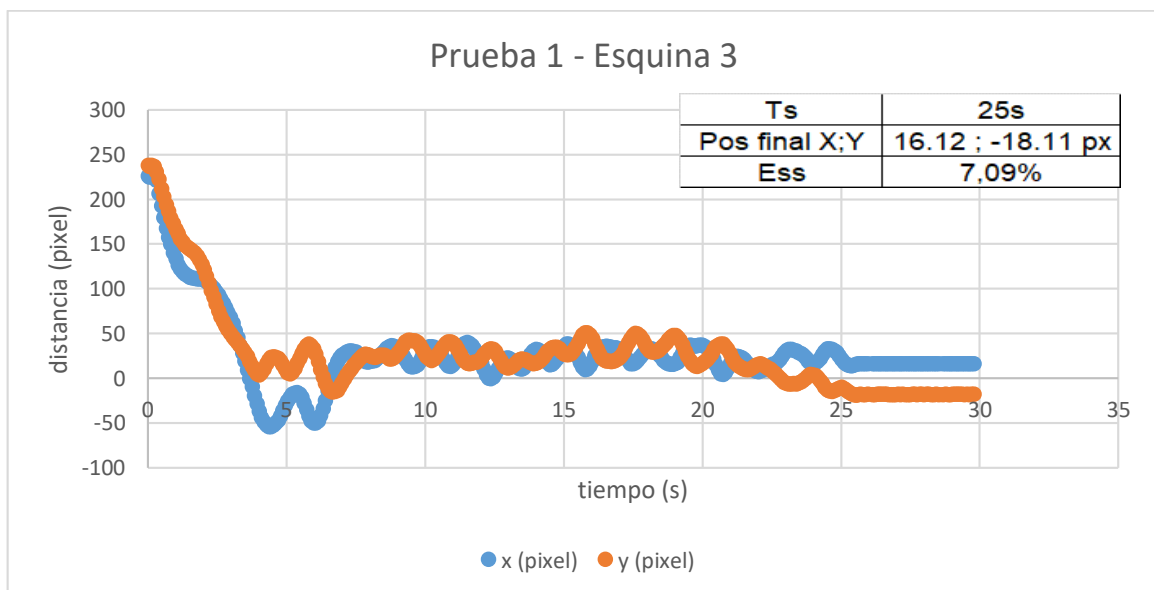
En esta prueba se obtiene el mejor resultado, con un tiempo de 6.21s y error 5.81%. Recordando el estado del arte planteado en la sección 2.4, de plataformas con tiempo de establecimiento entre 4 a 6s (Yaovaja, 2018), el rendimiento de este controlador es bastante bueno, acercándose a resultados de otros trabajos similares y recientes.

### Esquina 2



**Figura 63** Prueba 1 esquina 2

### Esquina 3

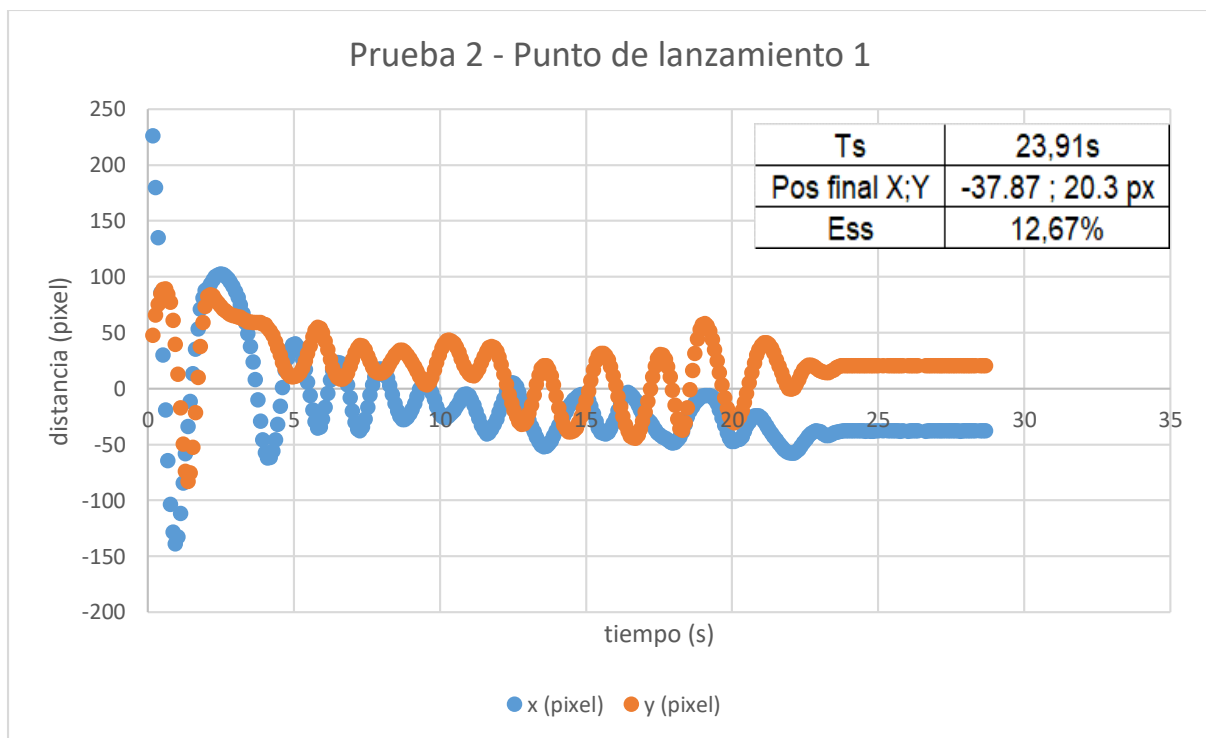


**Figura 64** Prueba 1 esquina 3

Para esta prueba, este es uno de los peores resultados obtenidos, con un tiempo de estabilización de 25s. Es importante recalcar que al ser un sistema no dinámico y debido a la estrategia de control (difusa o incierta) no siempre se van a obtener los mismos resultados. En ocasiones el rendimiento del controlador será mejor y en otras no, esto se demuestra en esta prueba 1, y se mostrará en las siguientes pruebas.

## 2. La pelota lanzada desde fuera hacia dentro de la plataforma sin rebote

### Punto de lanzamiento 1

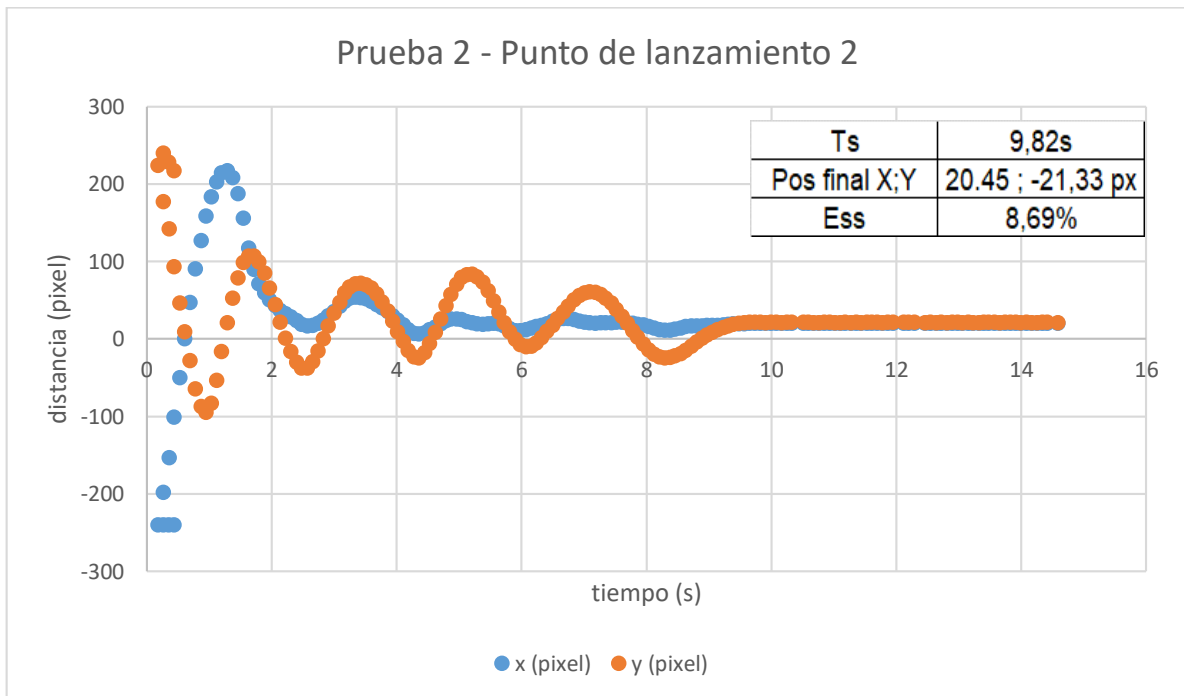


**Figura 65** Prueba 2 punto 1

En la Figura 65 al igual que en la anterior prueba en la Figura 64 se observa al sistema oscilando, y el tiempo de establecimiento es más alto que en los otros. Esto se debe a varios factores, en este caso la precisión de los motores y la velocidad de captura de video son las que más influyen en que el rendimiento del controlador baje.

Además, se debe tomar en cuenta que este sistema tiende a oscilar, debido a la dinámica del mismo. Sin embargo, los resultados no son tan malos, se equilibra la bola en menos de 30 segundos y con errores menores al 10% o 15%.

#### Punto de lanzamiento 2

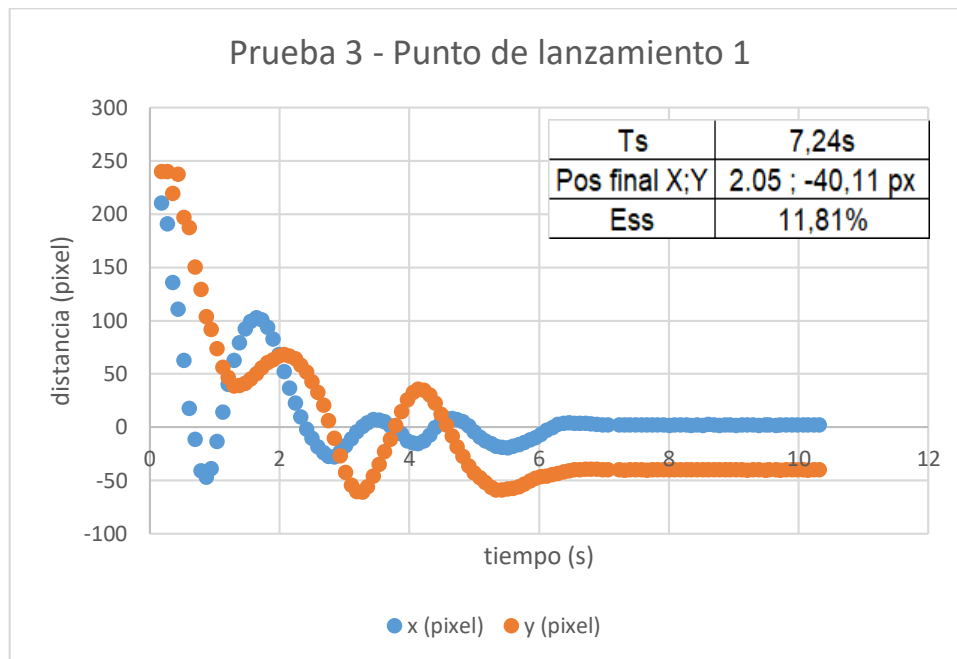


**Figura 66** Prueba 2 punto 2

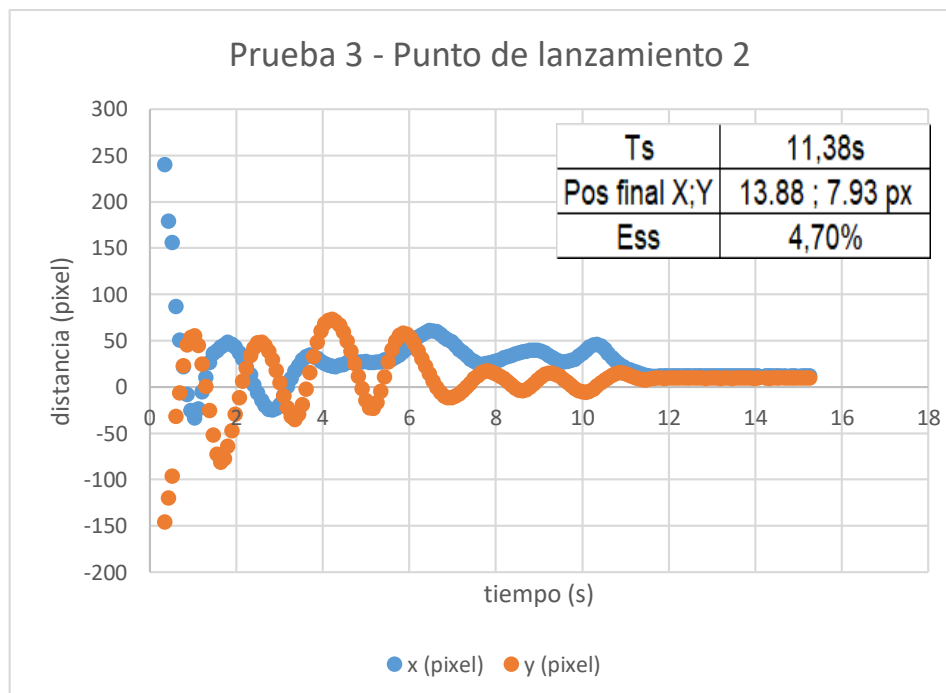
Los resultados en esta prueba son distintos al anterior, el tiempo menor a 10s y error menor a 10%. Por ello anteriormente se dijo que no siempre se obtienen los mismos resultados, por ello se trata de evidenciar todos los resultados de control de esta planta.

3. La pelota lanzada desde fuera y a cierta altura para que rebote

## Punto de lanzamiento 1

**Figura 67** Prueba 3 punto 1

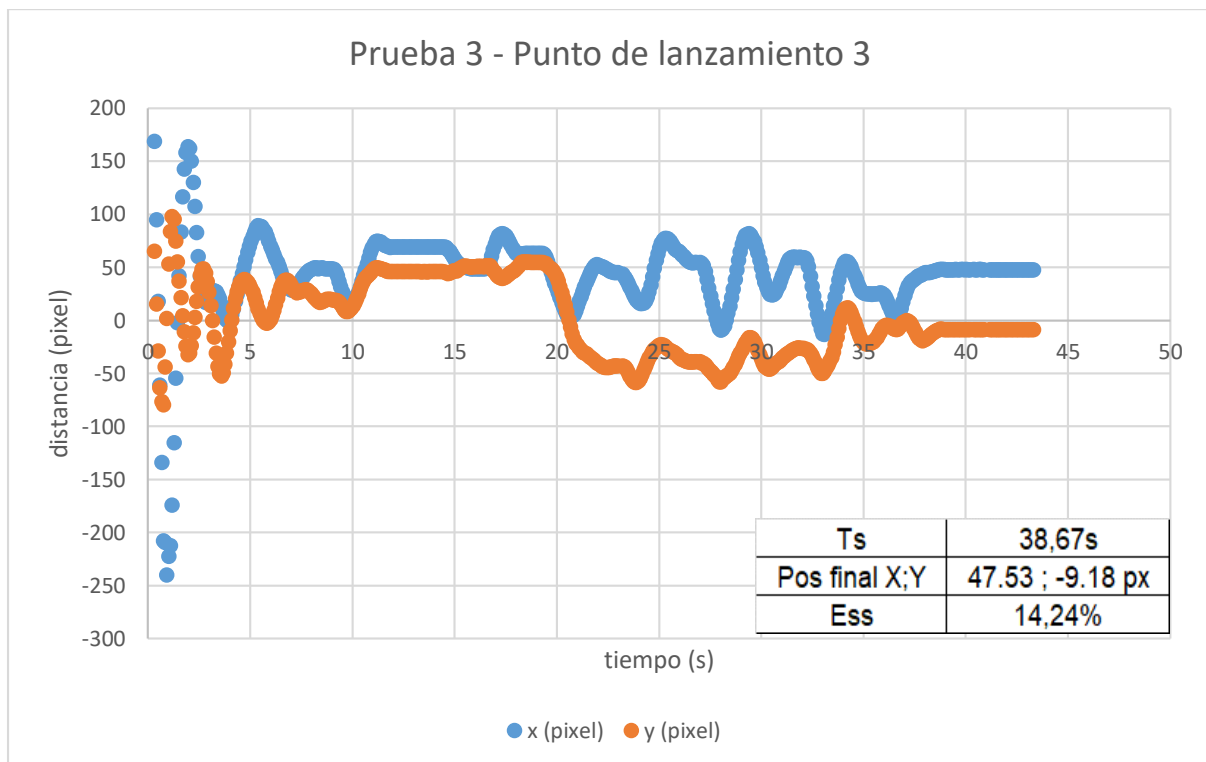
## Punto de lanzamiento 2

**Figura 68** Prueba 3 punto 2



En esta prueba 3, los datos iniciales son más dispersos debido al rebote de la bola y la acción de detectar la posición mediante la cámara, sin embargo, no es problema para la acción de control, ya que se obtienen resultados positivos como se observa en la Figura 68 que el tiempo de estabilización y error son bajos.

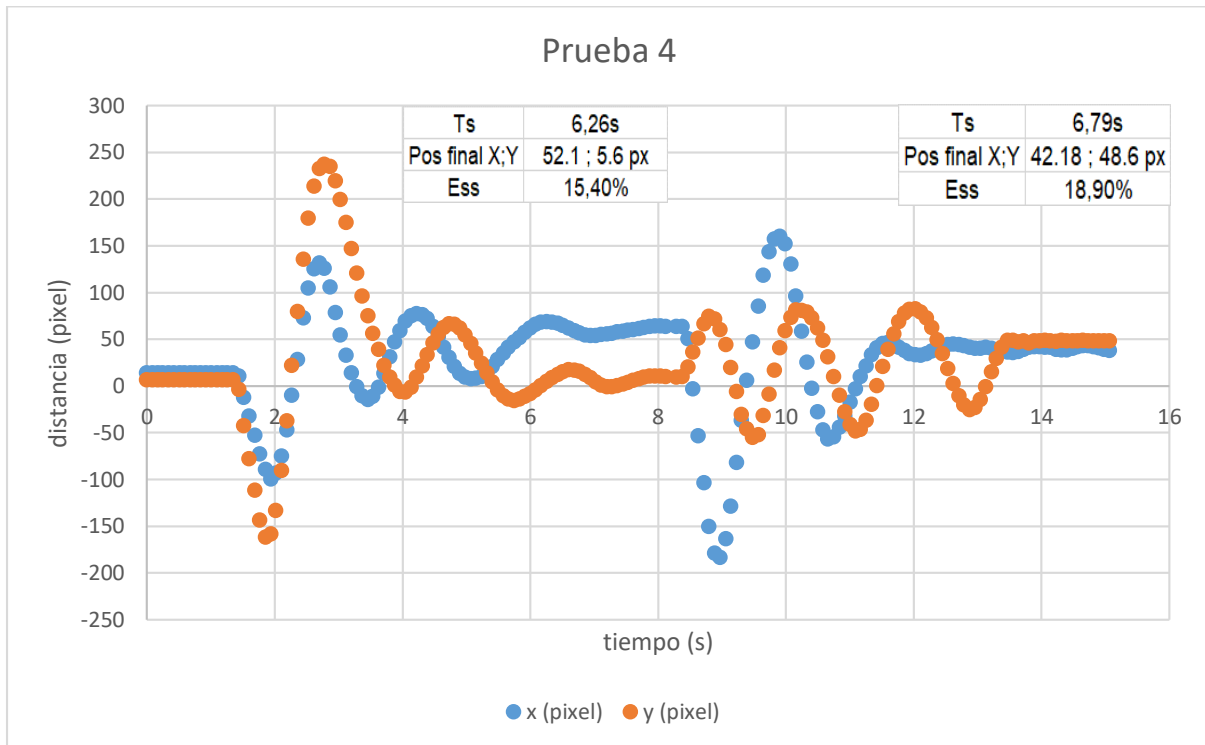
### Punto de lanzamiento 3



**Figura 69** Prueba 3 punto 3

El error es importante al momento de ajustar el controlador. En la Figura 69 en  $t=13s$  la bola está equilibrada con un error mayor al 25%, debido a esto el controlador actuó nuevamente para tratar de disminuir el error, logrando así en este caso bajarlo a 14% tomando un tiempo total de 38.6s.

4. Una vez equilibrada la pelota en el centro, moverla hacia uno de los lados



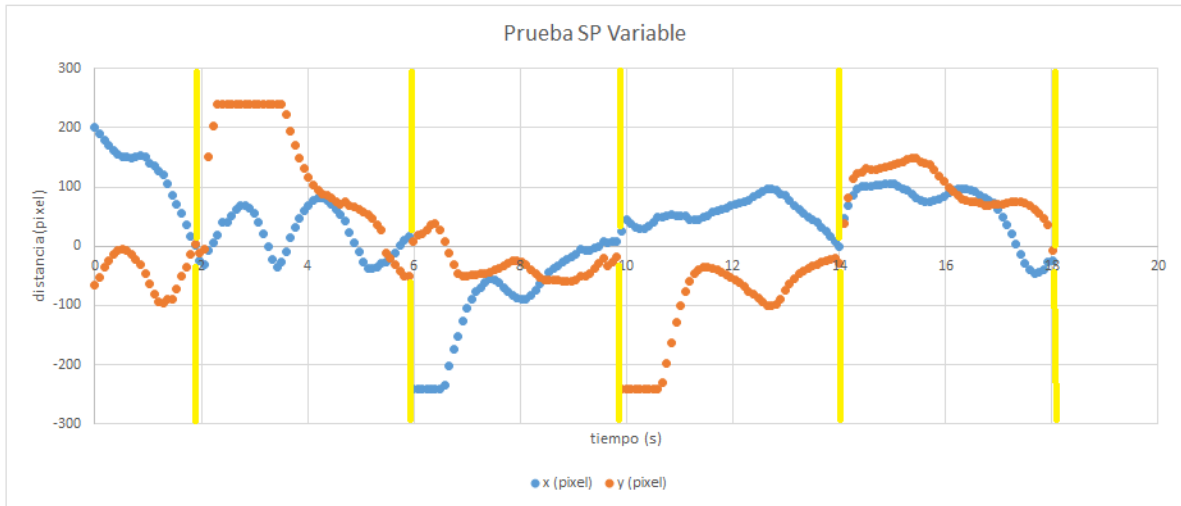
**Figura 70** Prueba 4

Para esta prueba, una vez equilibrada la bola se procede a moverla, para lo cual el primer resultado se obtiene un tiempo de estabilización de 6.26s y un error del 15.4% mientras que para el segundo movimiento se estabiliza en 6.79s con un error de 18.9%. Estas son las pruebas en las que mayor error se ha obtenido.

#### **4.3.2 Control difuso con punto de referencia variable por laser**

Para esta prueba del programa desarrollado (Anexo H), la fórmula del error se calcula diferente, tomando en cuenta la fórmula 4.6. El experimento consiste en ubicar la bola en la plataforma y una vez ahí, variar el punto de referencia a otro punto y dejar hasta que la bola alcance el punto, una vez alcanzado, variar nuevamente y así sucesivamente.

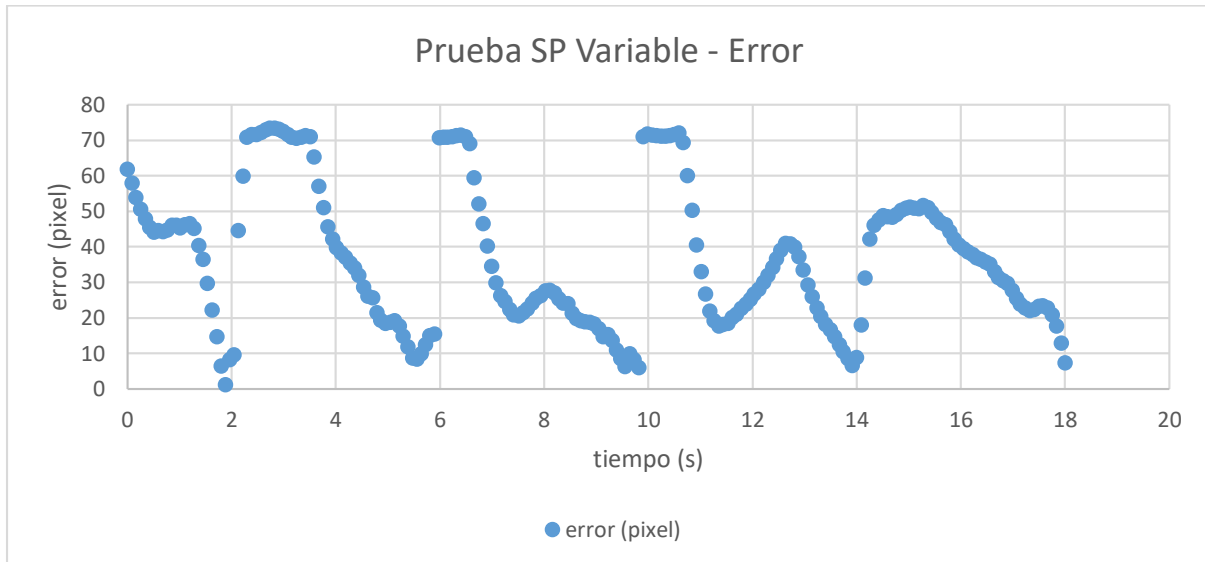
De esta manera, la bola persigue al láser tratando siempre de disminuir la distancia entre los dos. En el Figura 71, cada raya amarilla representa un nuevo punto de referencia marcado por el láser, una vez que la bola ha alcanzado el anterior.



**Figura 71** Prueba Punto de referencia variable

Se observa que el tiempo es aproximadamente de 4s que tarda en alcanzar el nuevo punto de referencia y en esta prueba el error ha sido pequeño, menor al 10%.

En la siguiente figura, se entiende mejor el experimento, donde se observa la gráfica del error y como crece y decrece conforme a cada cambio de punto de referencia.



**Figura 72** Prueba Punto de referencia variable Error

También se desarrolló un programa (Anexo G) en el que el usuario previamente ingresa por teclado las coordenadas de punto de referencia para equilibrio de la bola, tomando como referencia al punto 0,0 como centro de la plataforma y  $\pm 230\text{px}$  a cada lado para el ingreso de las mismas. El programa validará el ingreso de coordenadas, como se muestra a continuación:

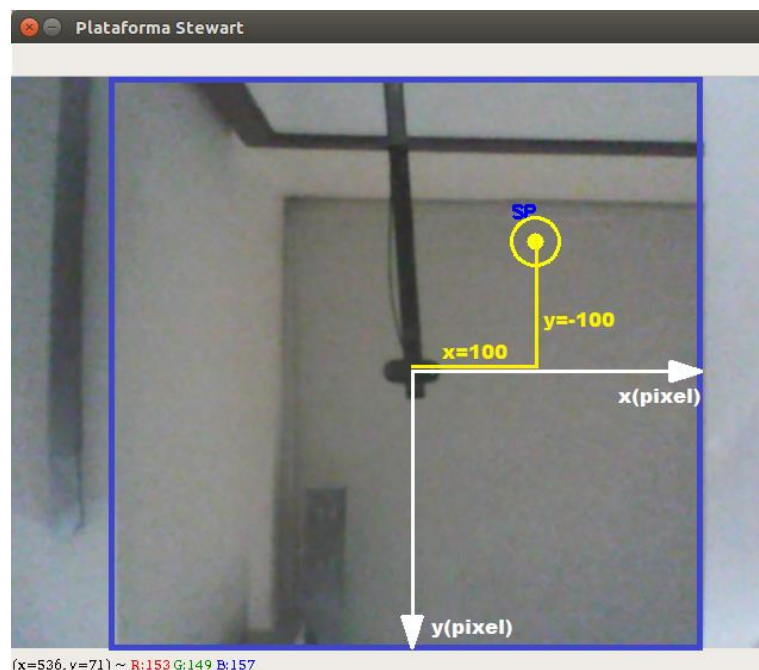
```

dxdiag-ubuntu@dxubuntu-Lenovo-ideapad-320-15A
dxdiag-ubuntu@dxubuntu-Lenovo-ideapad-320-15ABR:/
media/dxdiag-ubuntu/data/Tesis/Tesis_plataforma
stewart/Ing/Tesis/Vela$ python fuzzy_sp_var.py
Presione enter para continuar
El valor de X debe estar entre [-230,230]
Ingrese coordenada set point de X
500
El valor de X debe estar entre [-230,230]
Ingrese coordenada set point de X
100
El valor de Y debe estar entre [-230,230]
Ingrese coordenada set point de Y
-500
El valor de Y debe estar entre [-230,230]
Ingrese coordenada set point de Y
-100

```

**Figura 73** Ingreso coordenadas

Como se observa en la Figura 73, el usuario ingresó 500 para X, un valor no permitido por lo que ingresó nuevamente 100. Lo mismo para Y. El resultado del punto de referencia visto en la plataforma se muestra en la siguiente figura:



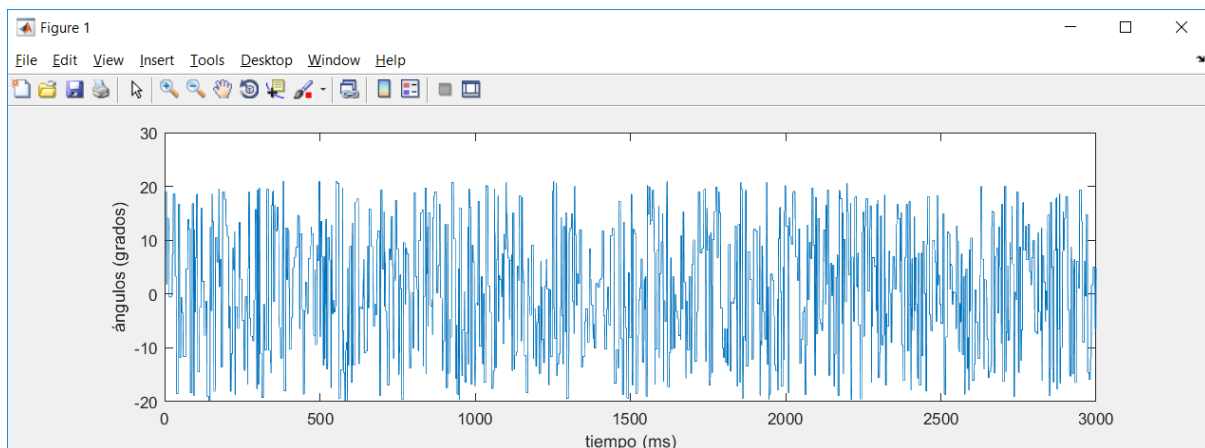
**Figura 74** Resultado ingreso coordenadas

## 4.4 Controlador Neuronal

### 4.4.1 Obtención de datos

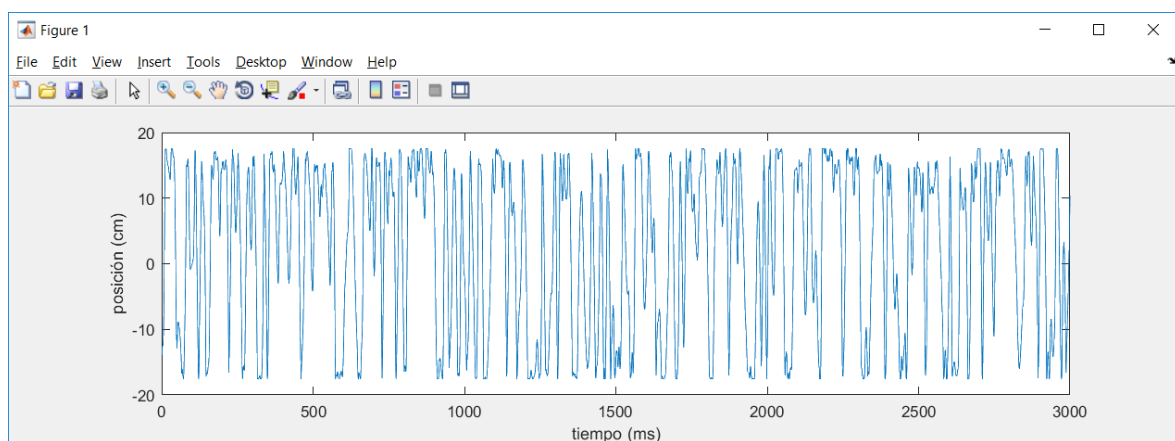
Se obtuvieron 3000 datos de entrada y salida con movimientos aleatorios, de los cuales los valores de X e Y fueron escalados de pixeles a cm, mediante mediciones realizadas en la plataforma. La relación de escalado parte de 240 pixeles equivale a 17.5cm (la mitad de la plataforma). Las razones de escalado serán explicadas más adelante.

Datos obtenidos en la variable Roll:



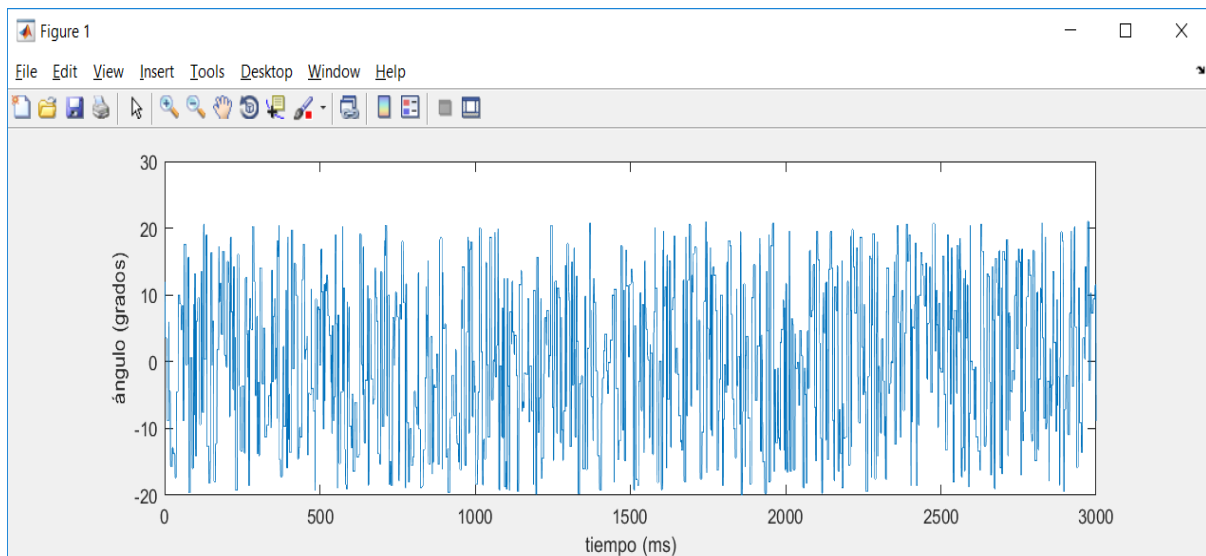
*Figura 75* Datos roll

Datos obtenidos en la variable X:



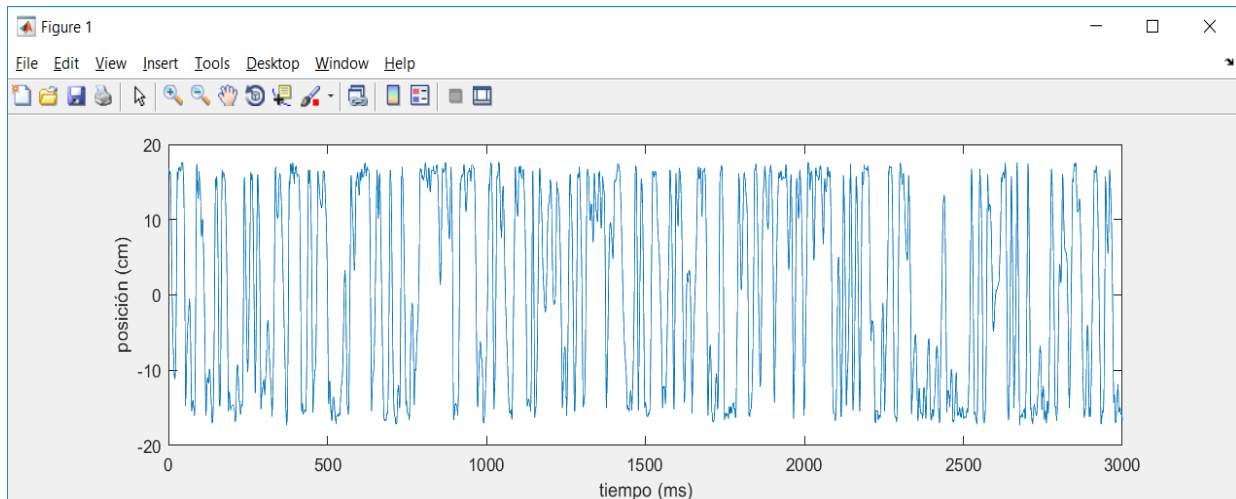
*Figura 76* Datos X

Datos obtenidos en la variable Pitch:



**Figura 77** Datos Pitch

Datos obtenidos en la variable Y:



**Figura 78** Datos de Y

En la siguiente tabla se presenta un resumen de los datos obtenidos:

**Tabla 14**  
*Datos obtenidos modelo planta*

	<b>Entradas (ángulos)</b>		<b>Salidas (cm)</b>	
	<b>Roll</b>	<b>Pitch</b>	<b>x</b>	<b>y</b>
<b>Promedio</b>	0,82	0,27	2,43	0,69
<b>Varianza</b>	132,43	136,97	153,12	156,43

Idealmente los valores medios deberían ser cero, para ello sería necesarias infinitas iteraciones.

Además, otros factores influyen como son la inclinación de la mesa de trabajo y la perfecta aleatoriedad de los datos de entrada.

Por otra parte, los valores de varianza son altos, lo que indica justamente que los datos son muy variantes unos con otros, por lo que se puede proceder a identificar.

#### 4.4.2 Identificación de la planta

##### 4.4.2.1 Modelo ARX

Con los resultados obtenidos en este modelado, se comprueba que la planta no es lineal, ya que la estructura ARX se utiliza en modelos lineales.

Para las pruebas m1 y m2 mostradas en la sección 3.4.2 se obtienen los siguientes resultados:

**Tabla 15**  
*Identificación ARX resultados*

		<b>Resultados</b>		
		<b>Ajuste</b>	<b>Error final de predicción</b>	<b>Error cuadrático medio</b>
<b>Prueba</b>	<b>m1</b>	29,52%	66,03	65,77
	<b>m2</b>	29,66%	66,02	65,5
	<b>Ideal</b>	89%	0,03	0,002

Como se observa en la tabla, los valores de un modelado con ARX ideales serían mucho mayor para el ajuste y mucho menor para ambos errores, por lo que el modelo ARX no es



aplicable al modelo dinámico de la planta y con ello se demuestra que no se trata de un modelo lineal.

#### 4.4.2.2 Modelo NARX

Al momento de ingresar los datos de entrada y salida en una red neuronal, es importante que estén correctamente escalados, esto significa que los rangos de variación deben ser similares y su diferencia no debe ser mayor. Así como se mencionó en el apartado 4.3.1, al ingresar los datos originales de entrada que varían de 20 a -20 grados y los datos de salida que varían de 240 a -240 pixeles, la red neuronal no puede encontrar solución al modelo pese al número de neuronas y regresores que se utilice. Por esto con el escalado, los datos de entrada variando de 20 a -20 grados y los de salida variando de 17.5 a -17.5cm se puede hallar una solución por medio de una red neuronal.

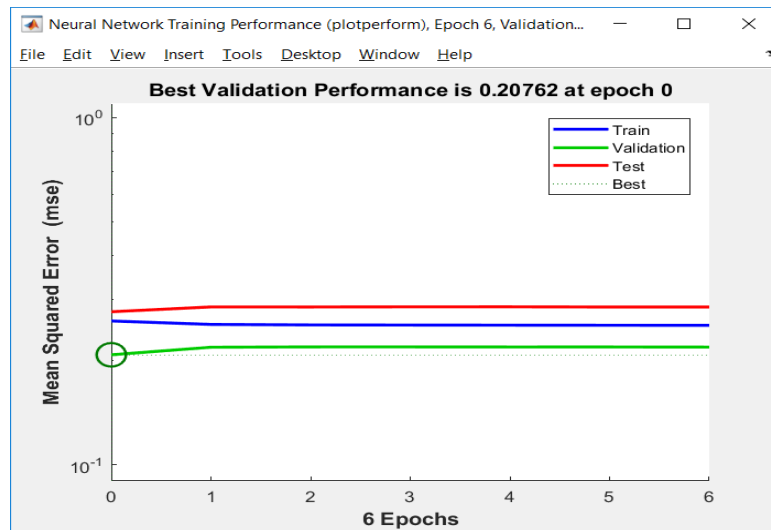
Luego de varias pruebas con distintas cantidades de neuronas y regresores, este fue el mejor resultado para el modelo no lineal dinámico de la planta:

**Tabla 16**  
*Identificación NARX resultados*

	<b>Desempeño</b>	<b>Gradiente</b>	<b>Iteraciones</b>
<b>Red Neuronal Planta</b>	0,252	0,0932	6/100

Luego de entrenar varias veces a la red, bastan 6 iteraciones (se observa en la Figura 54) para obtener un desempeño de 0.252 con un gradiente bajo. Con estos valores se prueba que el modelo se puede obtener mediante una red neuronal de 5 neuronas en su capa oculta y con 3 regresores.

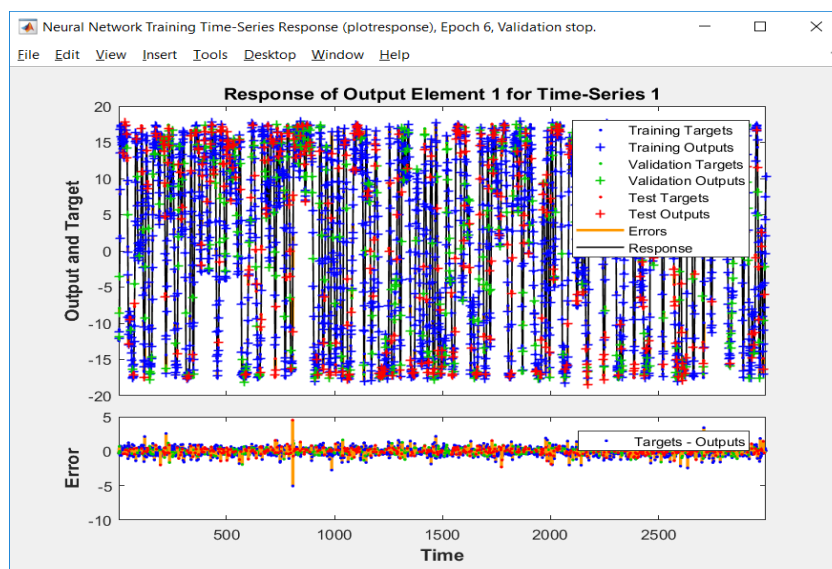
A continuación, se observa el desempeño de la red:



**Figura 79** Desempeño red neuronal planta

El menor valor se obtuvo en la primera iteración, con 0.20762. un valor bajo, lo que indica un buen desempeño de la red.

Y su respuesta en el tiempo:



**Figura 80** Respuesta de red neuronal en tiempo

#### 4.4.3 Controlador neuronal por modelo de referencia

Una vez modelada la planta, se procede a entrenar al controlador con los pesos actualizados de la planta modelada. Previamente se carga el modelo de referencia ya explicado en la sección 3.4.4.

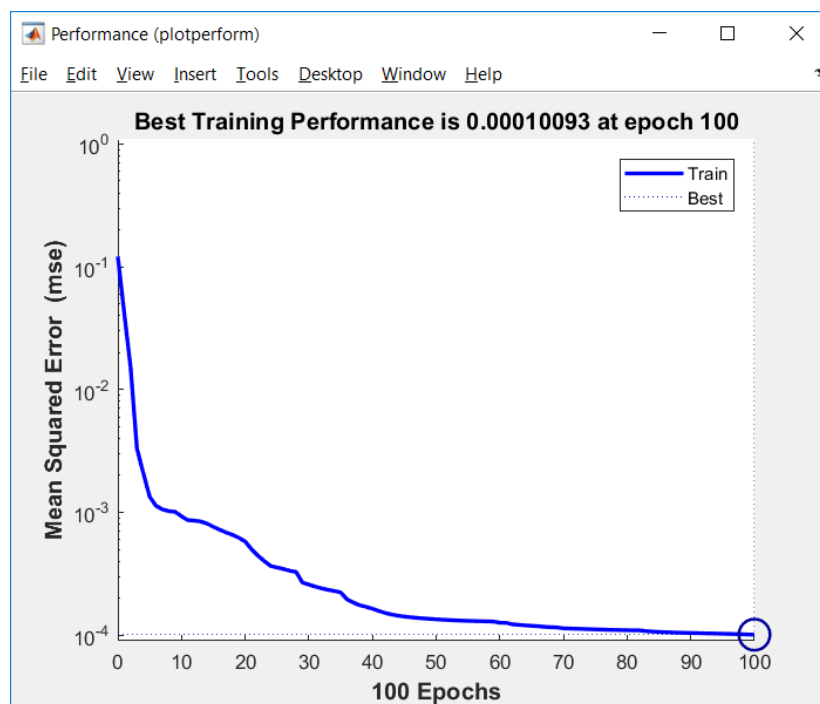
Luego de varias pruebas con distintas cantidades de neuronas y regresores, este fue el mejor resultado para el controlador, con 10 neuronas en su capa oculta y 3 regresores:

**Tabla 17**

*Entrenamiento controlador*

	Desempeño	Gradiente	Iteraciones
<b>Red Neuronal Controlador</b>	0,121	0,551	100/100

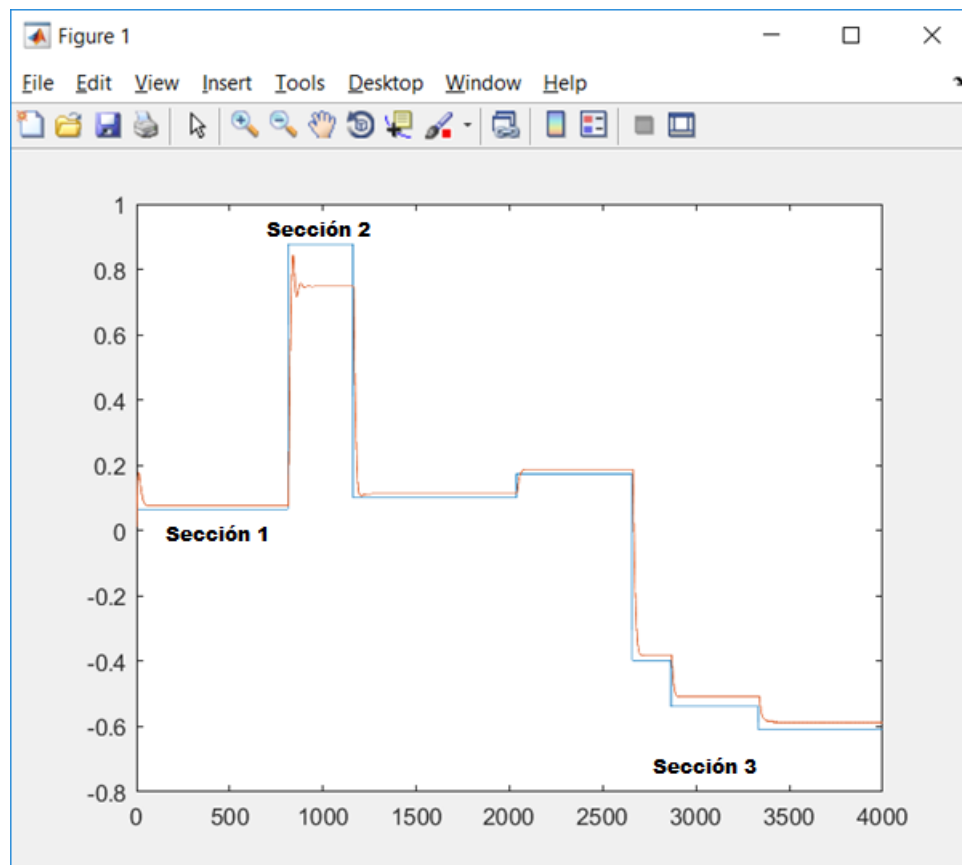
A continuación, se observa el desempeño de la red:



**Figura 81** Desempeño controlador

El menor valor se obtuvo en la iteración 100 y es de 0.0001, valor cercano a cero, lo que indica un correcto desempeño del controlador para este sistema.

Y la respuesta del controlador a cambios en el punto de referencia:

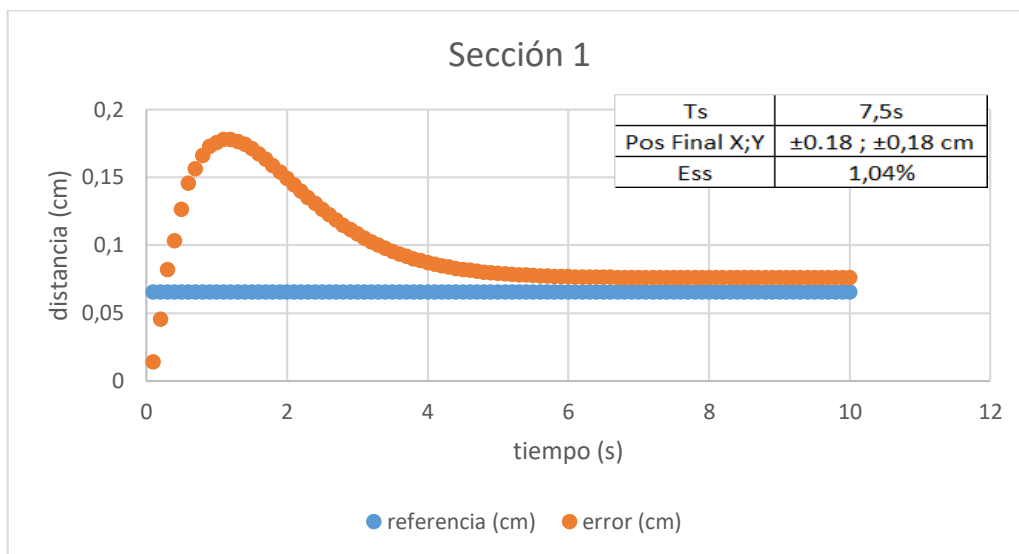


**Figura 82** Respuesta cambios de punto de referencia planta controlada

En la Figura 82 se observa la respuesta del controlador, al variar el punto de referencia de la planta, y como este tiene la capacidad de seguimiento, por lo que el controlador resulta ser ideal para este sistema.

A continuación se analizará las 3 secciones de la Figura 82 junto con su error  $E_{ss}$  y tiempo de establecimiento  $T_s$ :

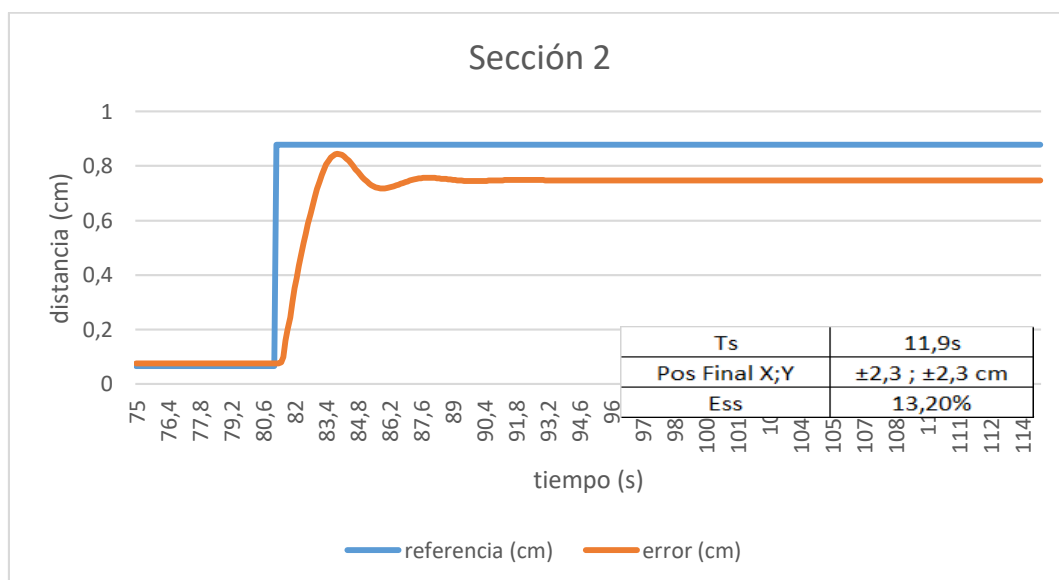
## Sección 1:



**Figura 83** Sección 1 prueba control RNA

La acción de control es buena, comparado con los resultados obtenidos por el controlador difuso en literales anteriores. El error para esta respuesta de cambio de punto de referencia es bajo: 1.04%.

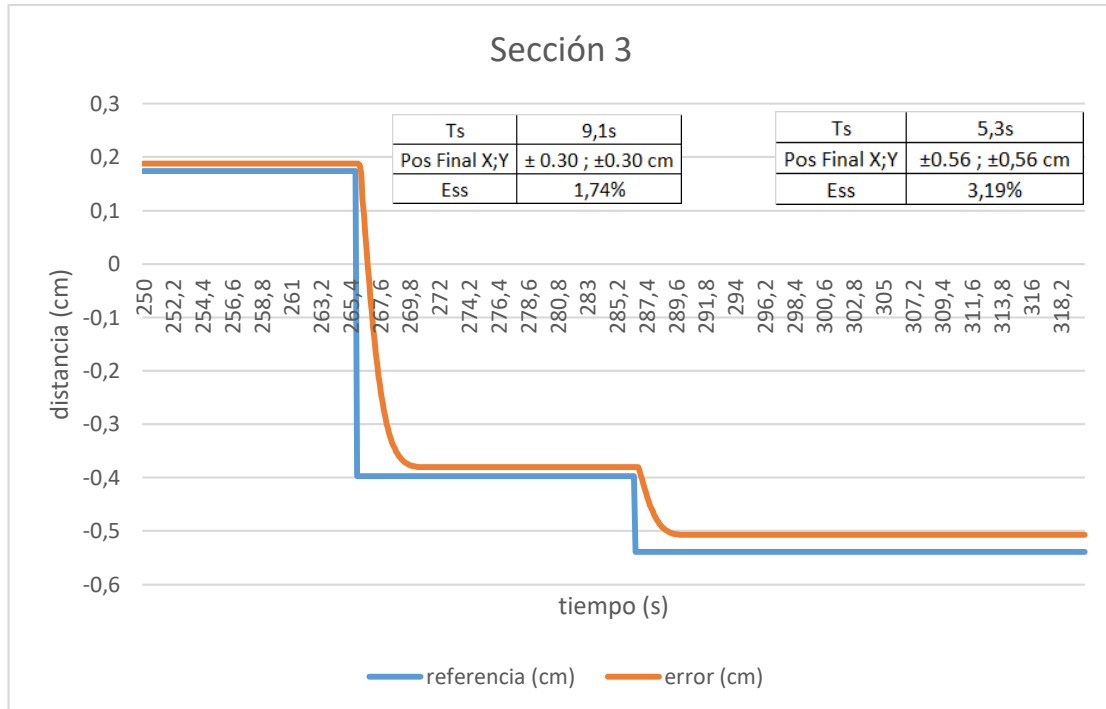
## Sección 2:



**Figura 84** Sección 2 prueba control RNA

Para este caso, el tiempo de estabilización aumentó y de igual manera el error. Como ya se ha mencionado, un sistema dinámico no lineal, va a presentar diferentes resultados para cada prueba, por tanto, el desempeño del controlador no va a ser igual.

### Sección 3:



**Figura 85** Sección 3 prueba control RNA

En esta tercera sección junto con las 2 anteriores, se prueba el buen desempeño que se ha obtenido con el control neuronal por modelo de referencia. Errores bajos y tiempos de estabilización buenos han sido resultado de estas pruebas.

Cuando se habla de redes neuronales, se debe tomar en cuenta el coste computacional que esto implica, sobre todo si el controlador tiene esta estructura (neuronal). Para este caso, las 10 neuronas y sus 3 regresores representan un costo elevado computacionalmente, y se debe tomar en cuenta al momento de ser implementado.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones

A partir del robot construido por Cristian Bueno y Alejandro Maldonado, como proyecto de fin de la carrera de Ingeniería Mecatrónica en 2017, se ha realizado los procesos necesarios para llevar a cabo el control del robot para la aplicación bola en plataforma. Ciertos cambios a nivel estructural fueron hechos previo al desarrollo y prueba de los controladores.

Se estudió e investigó estrategias de control inteligente, obteniéndose así resultados positivos en ambos casos. Se ha desarrollado estrategias de control basada en lógica difusa, mediante la cual se ha controlado la posición de la bola en el centro de la plataforma eficientemente, logrando un tiempo de estabilización de 6.21 segundos con un error de 5.81%. Se desarrolló también una estrategia de control basada en redes neuronales mediante un control adaptativo por modelo de referencia, logrando un desempeño de 0.121 y una respuesta a cambios en el punto de referencia adecuada.

Tomando la recomendación como trabajos futuros de Cristian y Alejandro, se desarrolló adecuadamente el algoritmo para variar el punto de equilibrio de la bola (centro) a cualquier lugar sobre la plataforma, sin necesidad de cambiar el método de obtención de posición de la bola o sensor, logrando así mediante el procesamiento de imagen de un láser (punto rojo), utilizarlo como nuevo punto de referencia y así poder variar en cualquier momento. De esta manera incluso se puede ver en el funcionamiento como si la bola tratara de seguir la trayectoria del láser que el usuario decida.

La dinámica de la plataforma, brinda las prestaciones para desarrollar técnicas de control inteligente de distintos tipos. Así un control difuso ha sido probado y un control neuronal adaptativo por modelo de referencia ha sido desarrollado, en ambos se han obtenido buenos resultados, comparando con el estado del arte actual para esta aplicación.

En lo que ese refiere a visión artificial, hay mucho por mejorar, sobre todo por la baja velocidad de captura con la que este proyecto ha sido implementado: 12fps. Sin embargo, se han implementado nuevas herramientas como el filtro de Kalman que predice la posición de la bola a medida que esta se desplaza por la plataforma, o como el hecho de procesar un segundo punto en tiempo real para tomarlo a este como nuevo punto de referencia.

Por último, son varios los parámetros que influyen en el desempeño del controlador, para este trabajo son dos los que afectan al robot: velocidad de captura como ya se ha mencionado, que es relativamente baja en relación al estándar de tiempo real 30fps y exactitud de posicionamiento del robot Stewart de 2.009 grados, factor que está limitado por la precisión de los servomotores utilizados en el mismo, y que afecta en el posicionamiento de la plataforma, sin embargo este valor ha sido mejorado frente al resultado de Cristian y Alejandro de 3.1 grados, gracias a los cambios en la estructura implementados en el robot.

## **5.2 Recomendaciones**

Cuando se realiza un trabajo de investigación, surgen muchas interrogantes, también nuevas ideas. Día a día nuevos y modernos controladores son desarrollados y esta plataforma, brinda la oportunidad de probarlos en ella. Se recomienda implementar nuevos controladores, inteligentes y/o adaptativos como un neuro-fuzzy, incluso un neuro-fuzzy-pid. Las técnicas de control clásicas también se pueden combinar con técnicas modernas, y obtener resultados que aporten a la ciencia en este amplio campo de control.



Mejorar la velocidad de captura es de suma importancia, con eso se puede subir el rendimiento de los controladores mucho más, evaluar el uso de la PC, GPU y también de la cámara Logitech C170 utilizada en el robot. De igual manera, si se quiere mejorar la exactitud en la orientación de la plataforma, es necesario implementar nuevos motores, de mejor precisión, frecuencia y velocidad de respuesta.

Stewart fue un científico que diseñó este robot como simulador de vuelos, por tanto, se recomienda implementar un simulador de vuelo y evaluar su desempeño. En este campo, existe mucha investigación y el estado del arte está muy avanzado. Además de otras aplicaciones y maneras de controlar al robot. Un sensor leap motion por ejemplo, es un dispositivo que se puede probar en esta planta, permitiendo el control del robot tan solo con la mano, y cierto procesamiento del sensor desde el CPU.

En lo que a identificación se refiere, se recomienda evaluar nuevos métodos de toma de datos y comparar resultados con lo que en este trabajo se han obtenido. La no linealidad del sistema brinda el medio para probar distintos algoritmos de identificación, siempre tratando de reducir el costo computacional, no solo del modelo si no del controlador a implementar.

# Bibliografía

- Alejandro, B. S. C., & Emmanuel, M. S. A. (2017). Diseño, Construcción e Implementación de una Plataforma tipo Stewart para Control de una Aplicación Bola en Plataforma a Través de Visión Artificial.
- Ali, E., & Aphiratsakun, N. (2015). AU ball on plate balancing robot. *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, (December 2015), 2031–2034. <https://doi.org/10.1109/ROBIO.2015.7419072>
- Alonso Pérez, M. A. (2009). Espacios de Color RGB , HSI y sus Generalizaciones a n - Dimensiones, 118. Retrieved from <https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/362/1/AlonsoPeMA.pdf>
- Aracil, R., Saltaron, R., & others. (2006). Robots paralelos: máquinas con una pasado para una robótica del futuro. *Revista Iberoamericana de Automática e Informática Industrial*, 3(1), 16–28. <https://doi.org/ISSN:1697-7912>
- Awtar, S., Bernard, C., Boklund, N., Master, A., Ueda, D., & Craig, K. (2002). Mechatronic design of a ball-on-plate balancing system. *Mechatronics*, 12(2), 217–228. [https://doi.org/10.1016/S0957-4158\(01\)00062-9](https://doi.org/10.1016/S0957-4158(01)00062-9)
- Bang, H., & Lee, Y. S. (2018). Implementation of a Ball and Plate Control System Using Sliding Mode Control. *IEEE Access*, 6, 32401–32408. <https://doi.org/10.1109/ACCESS.2018.2838544>
- Bdoor, S. R., Ismail, O., Roman, M. R., & Hendawi, Y. (2016). Design and implementation of a vision-based control for a ball and plate system. *2016 2nd International Conference on*

*Industrial Engineering, Applications and Manufacturing, ICIEAM 2016 - Proceedings*, 7–10. <https://doi.org/10.1109/ICIEAM.2016.7910965>

Cazalilla, J. I. (2017). Diseño E Implementación De Un Sistema Automático De Control De Calidad, Mediante La Aplicación De Visión Artificial, En El Laboratorio De Ingeniería De Producción Industrial De La Universidad De Las Américas.

Christensen, J., Humble, J., & Mattinson, J. (2017). Ball-Balancing Platform Design Documentation.

Christensen, S. (2014). Stewart-Gough Platform as a Test setup, (January). Retrieved from MSc Thesis

Craig, Kevin, S. A. (2008). Mechatronic Design of a Ball on Plate Balancing System, *12*(4), 93–114.

Craig, K. C. (2015). Diseño mecatrónico de una bola en plataforma.

Eduardo Aznar Colino, J. R. G. (2007). Conceptos y aplicaciones de la lógica borrosa - Eduardo Aznar Colino, Joaquín Royo Gracia - [tecnicaindustrial.es](http://www.tecnicaindustrial.es). Retrieved October 14, 2018, from <http://www.tecnicaindustrial.es/TIFrontal/a-625-conceptos-aplicaciones-logica-borrosa.aspx>

Garrido, S. (1999). Identificación, Estimación y Control de Sistemas No-lineales mediante RGO. *Journal of Endocrinological Investigation*, *40*(4), 367–376. <https://doi.org/10.1007/s40618-016-0569-7>

Garzón Mancera, O. L., & Garzón Melo, Y. (2016). Diseño e implementación de una plataforma bola y plato, para la aplicación de técnicas de control PID y LQR.

Gobierno de España, & Union Europea. (2012). Aplicación práctica de la visión artificial en el

control de procesos industriales.

González Álvarez, N. X., & Reinoso Mendoza, E. P. (2011). Estudio, diseño y construcción de una plataforma robótica didáctica tipo Stewart aplicada al estudio de controladores difusos. Retrieved from <http://dspace.ups.edu.ec/bitstream/123456789/1921/14/UPS-CT002354.pdf>

Gordillo, R., & Ayala, P. (2013). Control Adaptativo por el Método de Gain Scheduling aplicado a una L<sup>1</sup>. *ESPE*, 1–5.

Hilera, J. R., & Martínez, V. J. (2000). “Redes neuronales artificiales: fundamentos modelos y aplicaciones.” (April 2017).

International Organization for Standardization. (1998). Manipulating industrial robots - Performance criteria and related test methods (ISO 9283:1998). *Une En Iso 9283:1998*. <https://doi.org/ISO 9283:1998>

Kassem, A., Haddad, H., & Albitar, C. (2015). Comparison Between Different Methods of Control of Ball and Plate System with 6DOF Stewart Platform. *IFAC-PapersOnLine*, 48(11), 47–52. <https://doi.org/10.1016/j.ifacol.2015.09.158>

Kouro, S., & Musalem, R. (2002). Control mediante lógica difusa. *Técnicas Modernas Automaticas*, (1-7)-7, 1–7. Retrieved from <http://www2.elo.utfsm.cl/~elo377/documentos/Fuzzy.pdf>

Kuncan, M., & Ertunç, H. M. (2016). Fuzzy Logic Based Ball on Plate Balancing System Real Time Control by Image Processing, *10*(1), 14–18.

La Izquierda Diario. (n.d.). Karel Capek: Papá Robot. Retrieved October 10, 2018, from <https://www.laizquierdadiario.com/Karel-Capek-Papa-Robot>

Larrañaga, P., & Iñaki, I. (n.d.). Redes Neuronales, 1–19.

Matich, D. J. (2001). Redes Neuronales: Conceptos Básicos y Aplicaciones. *Historia*, 55. Retrieved from <ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf>

Matsuo, T., Tsuruta, K., & Suemitsu, H. (1999). Fuzzy Adaptive Identification Method Based on Riccati Equation and Its Application to Ball-Plate Control System PI-, 162–167.

Mejia, I. (n.d.). Perceptron simple. Retrieved from [http://www.academia.edu/7568072/Ejemplo\\_perceptron\\_simple](http://www.academia.edu/7568072/Ejemplo_perceptron_simple)

Nikolas, I. C. H. A. (2016). Diseño, Modelación Y Simulación De Un Sistema De Impresión 3D Basado En Sistemas De Cadena Cinemática Cerrada Con 4 Grados De Libertad Tipo Delta.

Ogata, K. (2013). *Ingeniería de Control Moderna. Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>

OpenCV Team. (2018). OpenCV library. Retrieved October 19, 2018, from <https://opencv.org/>

Ramón, P. M. (1999). Evolución Histórica de la Ingeniería de Control.

Rodríguez Rubio, F., & López Sánchez, M. J. (1996). Control Adaptativo y Robusto. *Control Adaptativo y Robusto*, (May), 381. <https://doi.org/10.6092/1590-8577/1979>

Sánchez-Martín, F. M., Jiménez Schlegl, P., Millán Rodríguez, F., Salvador Bayarri, J., Monllau Font, V., Palou Redorta, J., & Villavicencio Mavrigh, H. (2007). Historia de la robótica: De Arquitas de Tarento al robot da Vinci. (Parte II). *Actas Urologicas Espanolas*, 31(3), 185–196. [https://doi.org/10.1016/S0210-4806\(07\)73624-0](https://doi.org/10.1016/S0210-4806(07)73624-0)

Sanz, R. (1990). Arquitectura de Control Inteligente de Procesos, 31–39.

Satya Mallick. (2017). Object Tracking using OpenCV. Retrieved October 19, 2018, from <https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/>

Sedano, J., & Ramón, J. (2005). Introducción a la identificación de sistemas. *Tecnica Industrial*, 256, 31–36.

SENPLADES. (2017). Plan nacional de desarrollo 2017-2021 “Toda una Vida.” *Senplades*, 1–148.

Serrano J, A., Soria, E., & Martín, J. D. (2009). Redes Neuronales Artificiales, 1–141. [https://doi.org/10.1016/S0210-5691\(05\)74198-X](https://doi.org/10.1016/S0210-5691(05)74198-X)

Shih, C.-L., Hsu, J.-H., & Chang, C.-J. (2017). Visual Feedback Balance Control of a Robot Manipulator and Ball-Beam System. *Journal of Computer and Communications*, 05(09), 8–18. <https://doi.org/10.4236/jcc.2017.59002>

Tomás Arredondo Vidal. (2014). Introducción a la Lógica Difusa. Retrieved from <http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/Introduccion a la Logica Difusa.pdf>

Universidad de Jaén. (2006). Detección de bordes en una imagen, 1–21. Retrieved from [http://www4.ujaen.es/~satorres/practicas/practica3\\_vc.pdf](http://www4.ujaen.es/~satorres/practicas/practica3_vc.pdf)

Víctor R. González. (2002). Robots industriales: definición y clasificación. Retrieved October 10, 2018, from [http://platea.pntic.mec.es/vgonzale/cyr\\_0204/ctrl\\_rob/robotica/industrial.htm](http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/industrial.htm)

Vidal, T. A. (2012). Introducción a las Redes Neuronales. Retrieved from <http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/Introduccion a las redes>

neuronales.pdf

- Villacís, C., Navarrete, M., Rodríguez, I., Romero, F., Escobar, L., Fuertes, W., ... Burns, C. (2018). Real-time flight simulator construction with a network for training pilots using mechatronics and cyber-physical system approaches. *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017*, 238–247. <https://doi.org/10.1109/ICPCSI.2017.8392169>
- Vivas Albán, O. A. (1989). Robótica Paralela: Aplicaciones Industriales, Modelado Y Control. *Ai*, 10. Retrieved from [http://www.unicauca.edu.co/ai/publicaciones/ISAShow\\_Vivas.pdf](http://www.unicauca.edu.co/ai/publicaciones/ISAShow_Vivas.pdf)
- Yaovaja, K. (2018). Ball Balancing on a Stewart Platform using Fuzzy Supervisory PID Visual Servo Control. *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 170–175.
- Zacapoaxtla, D. E. (2013). Reseña de control clásico vs control moderno, (1), 1–20.