BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Maestría en Ciencias de la Computación
Especialidad Ing. En Sistemas Inteligentes

Modelado y Simulación de Sistemas

Dr. Rangel Huerta

REPORTE:

Control de Balanceo de Pelota por Lógica Difusa

Alumno:

Ramírez Álvarez Anel

12 de Abril del 2020

1. INTRODUCCIÓN

Se realizó el balanceo de una pelota en un riel, con un eje fijo y otro móvil; a partir del cambio de velocidad de un motor unido a una hélice.

El control del sistema se realizó con el procesamiento de datos en una tarjea de desarrollo haciendo uso de Lógica Difusa. Este control se basa en la lectura de posición del sensor GP2Y0A21YK0F y del movimiento del eje móvil con un motor DC por PWM.

A continuación se desglosa todo el desarrollo y análisis del balanceo de pelota desde sus especificaciones hasta los resultados.

2. MARCO TEÓRICO

2.1 LÓGICA DIFUSA Y CONTROL DE SISTEMAS

Cotidianamente nos movemos en un mundo con definiciones ambiguas, si alguien dice “está por llover” nos interesa saber en qué medida esto es cierto y en cuanto tiempo sucederá. La toma de decisión a partir de información que no específica también es un procedimiento cotidiano, esto es el que se intenta emular con lógica difusa a partir de: la observación del entorno, la formulación de reglas lógicas y de los mecanismos de toma de decisión.

2.2.1 Base teórica

2.2.1.1 Conjuntos borrosos

Los conjuntos clásicos, tienen limitaciones, se define un universo de discurso que contiene a conjuntos cuyos bordes están bien definidos, un elemento puede o no pertenecer a cierto conjunto, algo es verdadero o falso, no se definen situaciones intermedias. Los conjuntos borrosos son una extensión de los clásicos, donde se añade una función de pertenencia, definida esta como un número real entre 0 y 1. Así se introduce el concepto de conjunto o subconjunto borroso y se lo asocia a un determinado valor lingüístico, definido por una palabra o etiqueta lingüística, donde esta es el nombre del conjunto o subconjunto. Por cada conjunto se define una función de pertenencia o membresía denominada $\mu A(x)$, indica el grado en que la variable x está incluida en el concepto representado por la etiqueta A ($0 \leq \mu A(x) \leq 1$), si esta función toma el valor 0 significa que tal valor de x no está incluido en A y si toma el valor 1 el correspondiente valor de x está absolutamente incluido en A. En la Figura I-1 se puede apreciar un ejemplo donde el conjunto velocidad (con variable x) está subdividido en 3 subconjuntos {Baja, Meda, Alta}, con sus respectivas funciones de membresía {µBaja(x) µMedia(x) µAlta(x)}

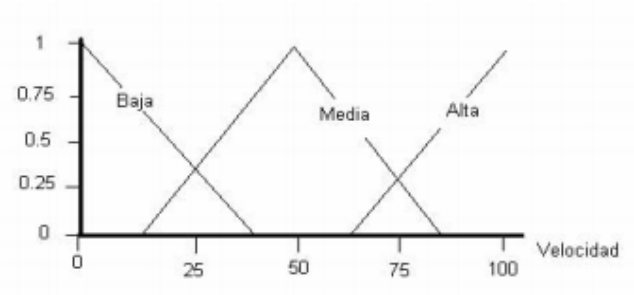


Figura 1. Ejemplo de subconjuntos borrosos para el conjunto velocidad.

Definición. Sea X una colección de objetos, expresados en forma genérica por x. Entonces, un conjunto difuso A en X, se define como un conjunto de pares ordenados

$$A = \{(x, \mu A(x))/x \in X\}$$

Donde μA(x) es una función de pertenencia cuya etiqueta es A y su dominio es x.

2.2.1.2 Funciones de membresía

Las funciones de membresía representan el grado de pertenencia de un elemento a un subconjunto definido por una etiqueta.

Existe una gran variedad de formas para las funciones de membresía, las más comunes son del tipo trapezoidal, triangular, gaussiana y sigmoidal.

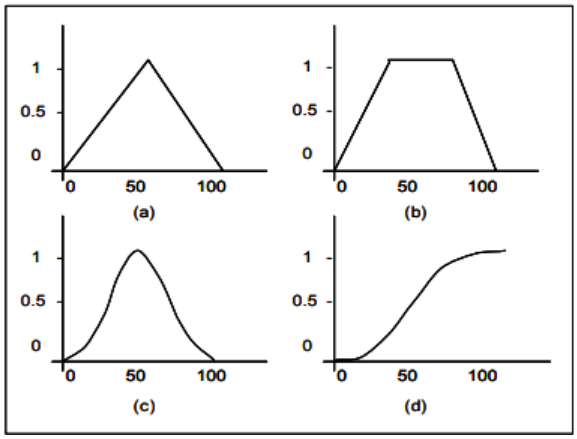


Figura 2. Algunas de las funciones características más habituales: a) triangular, b) trapezoidal, c) gaussiana y d) sigmoidal.

2.2.1.3 Operaciones borrosas

A los subconjuntos se les puede aplicar determinados operadores o bien se puede realizar operaciones entre ellos. Al aplicar un operador sobre un solo conjunto se obtendrá otro conjunto, lo mismo sucede cuando se realiza una operación entre conjuntos.

Las operaciones lógicas se utilizan en controladores y modelos difusos, son necesarias en la evaluación del antecedente de reglas (y otras etapas) que más adelante veremos.

Se definen a continuación 3 operaciones básicas a realizar sobre conjuntos, estas operaciones son complemento, unión e intersección. Sean las etiquetas A y B las que identifican a dos conjuntos borrosos asociados a una variable lingüística x, las operaciones se definen como:

- Complemento $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$
- Unión. Operador lógico de Zadeh (max) $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$
- Intersección. Operador lógico AND de Zadeh(min) $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$

Hay muchas definiciones para las operaciones lógicas, algunas otras definiciones que normalmente también se utilizan son:

- Operador lógico AND del producto $\mu_{A \cap B}(x) = \mu_A(x) \times \mu_B(x)$
- Operador lógico OR de Lukasiewicz $\mu_{A \cup B}(x) = \max[\mu_A(x) + \mu_B(x), 1]$

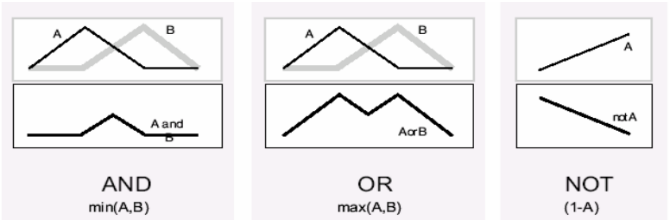


Figura 3. Operaciones lógicas en conjuntos difusos.

2.2.1.4 Fuzzificación

El control difuso siempre involucra este proceso de Fuzzificación, esta operación se realiza en todo instante de tiempo, es la puerta de entrada al sistema de inferencia difusa. Es un procedimiento matemático en el que se convierte un elemento del universo de discurso (variable medida del proceso) en un valor en cada función de membresía a las cuales pertenece.

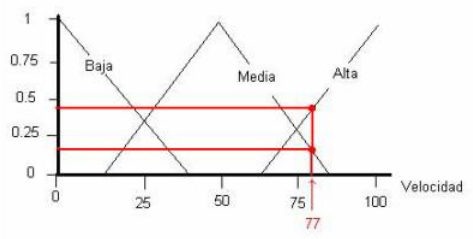


Figura 4. Ejemplo de Fuzzificación de una variable.

Para comprender mejor veamos la Figura 4 que arroja los siguientes datos:

μAlta(77)=0.45

μMedia(77)=0.20

μBaja(77)=0.00

El valor de velocidad igual a 77 pertenece a dos conjuntos con distintos grados en cada uno.

A partir de ahora y durante el resto de las operaciones en el interior del corazón fuzzy estos datos (0.45, 0.20 y 0.00, son valores de las funciones de membresía) representarán a las variables sensados del proceso. A tales datos les llamaremos μ en sentido genérico para diferenciarlos de otras funciones de membresía. μA(x) = μ.

2.2.1.5 Reglas borrosas

Los controladores difusos usan reglas, estas combinan uno o más conjuntos borrosos de entrada llamados **antecedentes o premisas** y le asocian un conjunto borroso de salida llamado **consecuente o consecuencia**. Involucran a conjuntos difusos, lógica difusa e inferencia difusa. A estas reglas se les llama **reglas borrosas o difusas o fuzzy rules**. Son afirmaciones del tipo SI-ENTONCES. Los conjuntos borrosos del antecedente se asocian mediante operaciones lógicas borrosas AND, OR, etc.

Las reglas borrosas son proposiciones que permiten expresar el conocimiento que se dispone sobre la relación entre antecedentes y consecuentes. Para expresar este conocimiento de manera completa normalmente se precisan varias reglas, que se agrupan formando lo que se conoce como **basa de reglas**, es decir, la edición de esta base determina cual será el comportamiento del controlador difuso y es aquí donde se emula el conocimiento o experiencia del operario y la correspondiente estrategia de control.

La base de reglas suele representarse por tablas. Esta es clara en el caso de 2 variables de entrada y una de salida. En la medida que la cantidad de variables lingüísticas crece, también lo hará la tabla, y más difícil se hará su edición.

Junto a cada regla puede estar asociado un valor entre cero y uno que pesa a tal regla, esto puede ser importante cuando una regla tiene menor fuerza que otras de la base de reglas.

Existe una gran variedad de tipos de reglas, dos grandes grupos son los que en general se emplean, las reglas difusas de Mamdani y las reglas difusas de Takagi-Sugeno (TS, para abreviar).

La estructura de las reglas es la misma tanto para controladores como para modelos, simplemente cambiarán las variables implementadas.

2.2.1.5.1 Reglas difusas de Mamdani

IF x1 is A AND x2 is B AND x3 is C THEN u1 is D, u2 is E

Donde x1, x2 y x3 son las variables de entrada (por ejemplo, error, derivada del error y derivada segunda del error), A, B y C son funciones de membresía de entrada (p.ej., alto, medio, bajo), u1 y u2 son las acciones de control (p.ej., apretura de válvulas) en sentido genérico son todavía variables lingüísticas (todavía no toman valores numéricos), D y E son las funciones de membresía de la salida, en general se emplean singleton por su facilidad computacional, y AND es un operador lógico difuso, podría ser otro. La primera parte de la sentencia "IF x1 is A AND x2 is B AND x3 is C" es el antecedente y la restante es el consecuente.

Un ejemplo es

IF error is Positivo Grande AND derivada del error is Positiva Baja THEN u is Positiva Chica.

Ventajas

- Es intuitivo.
- Tiene una amplia aceptación.
- Está bien adaptado a la incorporación de conocimiento y experiencia.

2.2.1.5.2 Reglas difusas de Takagi-Sugeno

IF x1 is A AND x2 is B AND x3 is C THEN u1=f(x1,x2,x3), u2=g(x1,x2,x3)

En principio es posible emplear f() y g() como funciones no lineales, pero la elección de tal función puede ser muy compleja, por lo tanto en general se emplean funciones lineales.

Ventajas

- Es computacionalmente eficiente.
- Trabaja bien con técnicas lineales (por ejemplo como lo disponible para controladores PID).
- Trabaja bien con técnicas de optimización y control adaptable.
- Tiene garantizada una superficie de control continua.
- Está bien adaptado al análisis matemático.

2.2.1.6 Inferencia borrosa

Las reglas difusas representan el conocimiento y la estrategia de control, pero cuando se asigna información específica a las variables de entrada en el antecedente, la inferencia difusa es necesaria para calcular el resultado de las variables de salida del consecuente, este resultado es en términos difusos, es decir

que se obtiene un conjunto difuso de salida de cada regla, que posteriormente junto con las demás salidas de reglas se obtendrá la salida del sistema.
Existe una gran cantidad de métodos de inferencia difusa, pero hay cuatro que generan mejores resultados en el campo del control, estos son inferencia de Mamdani por mínimos (Mamdani minimum inference), RM, la inferencia del producto de Larsen (Larsen product inference), RL, la inferencia del producto drástico (Drastic product inference) RDP y la inferencia del producto limitado (Bounded product inference), RBP. [1] [2]

Tabla I-1. Definición de los cuatro métodos de inferencia más populares.

Método de inferencia	Definición
Mamdani minimum inference, R_M	$\min(\mu, \mu_W(z)), \forall z$
Larsen product inference, R_L	$\mu \times \mu_W(z), \forall z$
Drastic product inference, R_{DP}	$\begin{cases} \mu & para\ \mu_W(z)=1 \\ \mu_W(z) & para\ \mu=1 \\ 0 & para\ \mu<1\ y\ \mu_W(z)<1 \end{cases}$
Bounded product inference, R_{BP}	$\max(\mu+\mu_w(z)-1,0)$

Donde μw es la función de pertenencia del conjunto de salida w. En el caso particular en que el conjunto difuso de salida del consecuente es singleton o sigmoidal, todos los métodos de inferencia tienen el mismo resultado, y este se obtiene como el singleton pesado por el valor μ obtenido del antecedente.

2.2.1.6.1 Agregado

Cuando se evalúan las reglas se obtienen tantos conjuntos difusos como reglas existan, para defusificar es necesario agrupar estos conjuntos, a esta etapa se le llama agregado y existen varios criterios para realizar este paso. Un criterio muy empleado es el de agrupar los conjuntos inferidos mediante la operación max.

2.2.1.7 Defusificación

La defusificación (defuzzyfication) es un proceso matemático usado para convertir un conjunto difuso en un número real. El sistema de inferencia difusa obtiene una conclusión a partir de la información de la entrada, pero es en términos difusos. Esta conclusión o salida difusa es obtenida por la etapa de inferencia borrosa, esta genera un conjunto borroso pero el dato de salida del sistema debe ser un número real y debe ser representativo de todo el conjunto obtenido en la etapa de agregado, es por eso que existen diferentes métodos de defusificación y arrojan resultados distintos, el “más común y ampliamente usado” es el centroide. Con el método de defusificación del centroide se transforma la salida difusa en un número real el cual es la coordenada equis (x) del centro de gravedad de tal conjunto difuso de salida.

$$y_d = \frac{\int_S y\mu_Y(y)dy}{\int_S \mu_Y(y)dy} \qquad \textbf{(1)}$$

Donde *μY* es la función de pertenencia del conjunto de salida *Y*, cuya variable de salida es *y*. *S* es el dominio o rango de integración. Este método en realidad trae una carga computacional importante, por lo que se emplean en general otros esquemas con menor carga. Uno de los defusificadores más usados es el centro de área (COA, *center of area*) también llamado de altura, el centro de gravedad es aproximado por el centro de gravedad de un arreglo de "masas puntuales", las cuales son el centro de gravedad de cada conjunto de salida correspondiente a cada regla, con "masa" igual al grado de pertenencia en ese punto de su centro de gravedad. Si se le llama *δl* al centro de gravedad del conjunto difuso de salida *Bl* de la *l-ésima* regla, el centro de gravedad queda determinado por:

$$y_d = \frac{\sum_{l=1}^{R} \delta_l \mu_{B_l(\delta l)}}{\sum_{l=1}^{R} \mu_{B_l(\delta l)}} \qquad \textbf{(2)}$$

Donde R es el número de reglas.

El concepto del centro de gravedad es en muchos casos el punto de partida para la obtención de distintos métodos de defusificación.[3]

2.2.1.7 Toma de decisión

En la sección anterior se presentaron los conceptos básicos de un sistema de inferencia o de toma de decisión. Se verá ahora de manera resumida y en forma gráfica los pasos que son llevados a cabo para la toma de decisión en este sistema de inferencia. En forma genérica el esquema de toma de decisión es el siguiente.

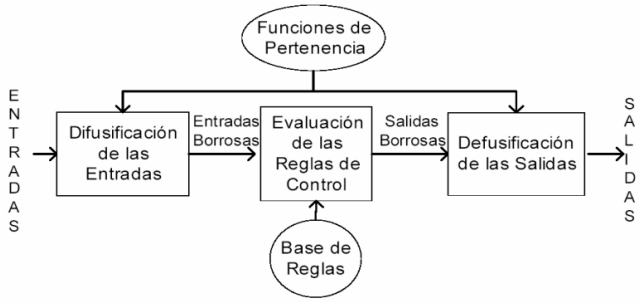


Figura 5. Sistema de inferencia difusa.

Resumen del proceso de inferencia difusa

Diagrama de flujo	Comentario
Dato de entrada	Dato proveniente del sensor que mide la variable del proceso, puede presentar ruido y desviaciones con respecto al valor real.
Fusificación	Se convierte un número en valores correspondientes a las funciones de membresía a la cual pertenece.
Evaluación de reglas	Las reglas definen la estrategia de control o conocimiento, se realizan operaciones entre los conjuntos.
Inferencia	Se determina el conjunto de salida de cada regla.
Agregado	Se obtiene la función de membresía de la variable de salida a partir de alguna operación entre todos los conjuntos de salida de la etapa de inferencia.
Defusificación	Definida la función de membresía de la salida, se determina cual dato es el más representativo del conjunto de salida total.
Dato de salida	Es la variable que tomará el actuador para modificar el estado del proceso.

3. METODOLOGÍA

3.1 Diagrama de desarrollo

A continuación se muestra el diagrama en bloques del sistema, Figura 5. El primer bloque es la entrada del Sensor de Distancia GP2Y0A21YK0F, después tenemos la entrada a la tarjeta ya que esta ya cuenta con el convertidor Analógico a Digital (A/D), posteriormente las operaciones de control difuso grabados en la memoria de la tarjeta (PIC18F4550), después de procesada la señal y hacer la toma de decisión pasa al convertidor Digital a Analógico, igualmente incluido en la tarjeta; cuya salida corresponde al PWM que es la entrada al motor, que balanceara la pelota.

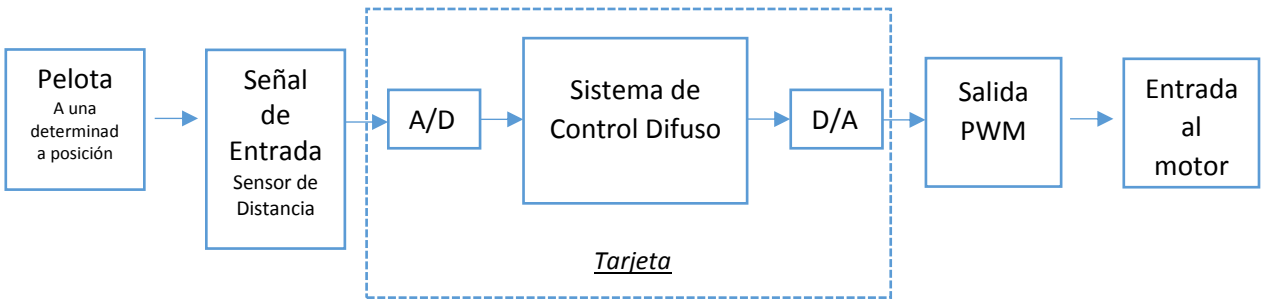


Figura 5. Diagrama en bloques del sistema.

3.1.2 SEÑALES DE ENTRADA

Inicialmente por el diagrama general del proyecto, la pelota se encontrará a la mayor distancia del sensor, ver Figura 3. Y conforme se valla elevando el eje por ayuda del motor, la distancia ira reduciéndose, pero ya que el objetivo es el balanceo la pelota deberá permanecer en el centro del eje, ver figura 7.

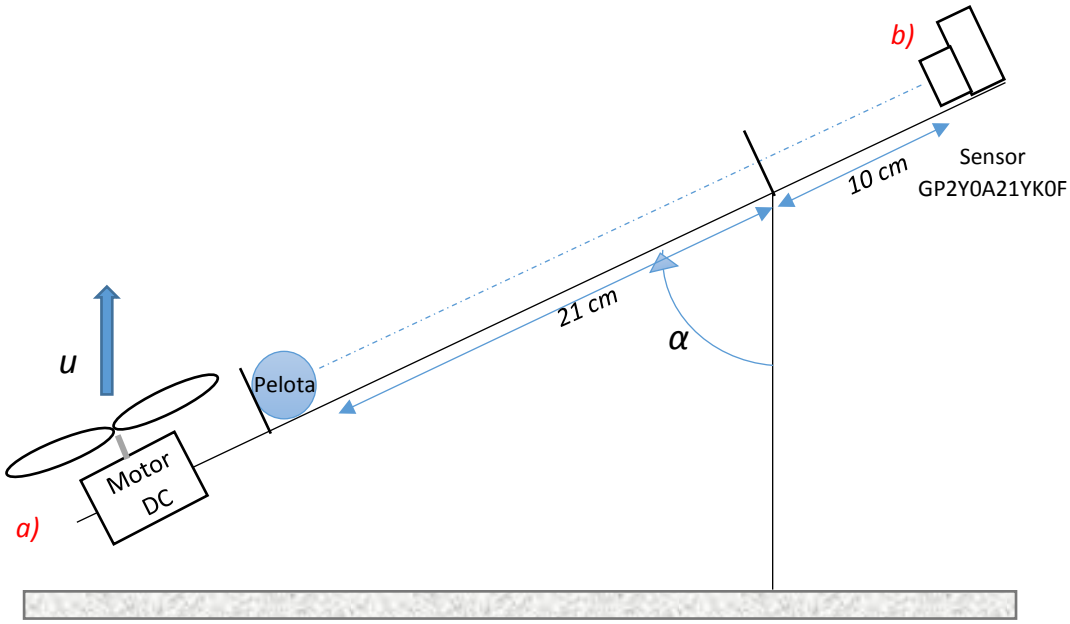


Figura 6. Diagrama esquemático de la maqueta para el balanceo de pelota.

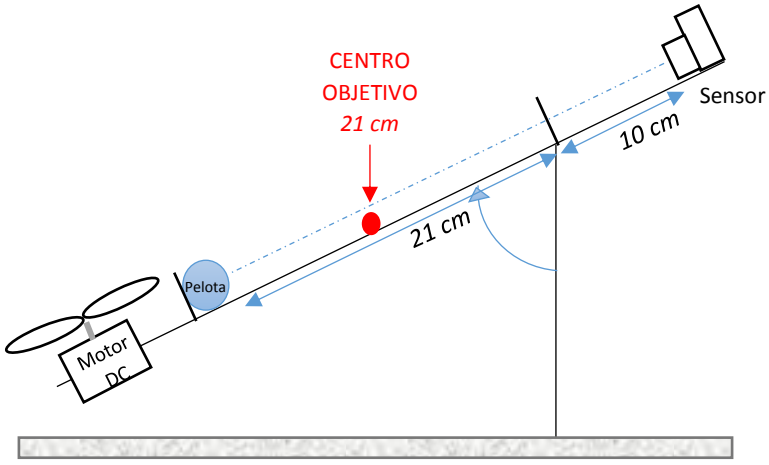


Figura 7. Objetivo del balanceo.

Notar que el sensor está a 10 cm del tope del movimiento posible de la pelota, esto se debe al mismo funcionamiento del sensor que en su ficha técnica y por ende en la práctica su medición comienza a partir de los 10cm a los 80 cm. Pero ya que en la implementación sólo arrojaba datos correctos de los 10cm a los 32 cm se usaron esos rangos en el proyecto.

Con lo que se trabajó con una magnitud de 22 cm, menos el espacio ocupado por la pelota, 1cm, dieron 21 cm de rango para el procesamiento de datos en el control difuso.

3.1.2.1 SENSOR GP2Y0A21YK0F

GP2Y0A21YK0F es una unidad de sensor de medición de distancia, compuesto de una combinación integrada de PSD (detector sensible a la posición), IRED (emisión infrarroja diodo) y circuito de procesamiento de señal.

La variedad de la reflectividad del objeto, el temperatura ambiental y la duración de funcionamiento no son influenciados fácilmente a la detección de distancia por adoptar el método de triangulación.

Este dispositivo emite el voltaje correspondiente al distancia de detección Entonces este sensor también se puede usar como un sensor de proximidad.

Figura 8. Sensor Infrarrojo de distancia.

3.1.2.2 FUNCIONAMINETO DEL SENSOR

Se basa en que la salida del sensor lanza una luz infrarroja que a la hora de rebotar con un objeto, Figura 9 (de aquí la necesidad de que sea un objeto opaco, es decir, no transparente) retorna al mismo sensor, la relación de tiempo transcurrido desde la salida a su regreso nos da una relación de distancia, ver Figura 10.

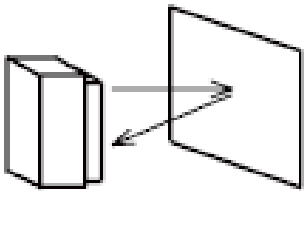


Figura 9. Medición de distancia.

Debido a esto se puede tener una fórmula lineal de distancia en función del voltaje, ver Figura 10, aquí se puede ver la razón por la que la medición correcta comenzaría a partir de 10cm, para una misma fórmula ya que en ese rango la pendiente es la misma.

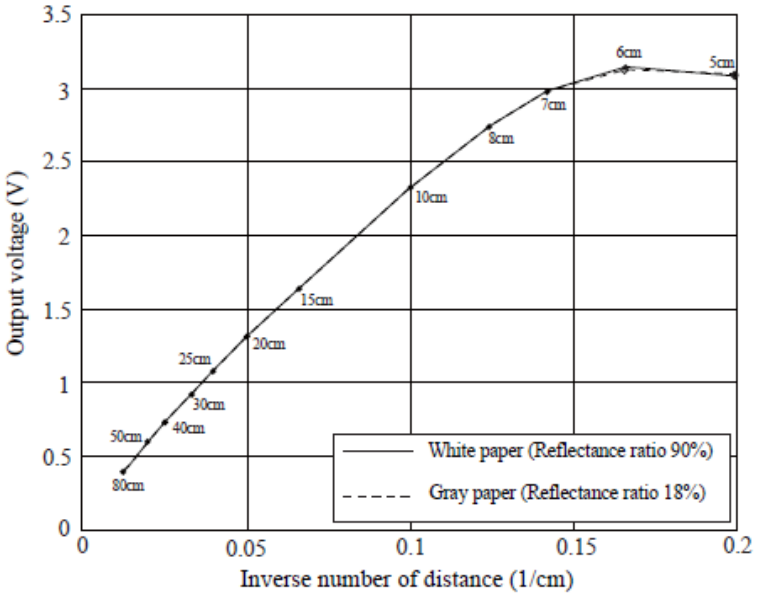


Figura 10. Caracterización del sensor.

Requerimientos del sensor

Entrada: Objeto opaco.

Salida: Voltaje.

3.1.3 PROCESAMIENTO DE LA SEÑAL: Control Lógico Difuso del sistema

En el control del sistema se requiere una variación de Voltaje por PWM para el movimiento de la pelota, esta variación se hará mediante Lógica Difusa.

Este control se realizó con dos Sigmoidales y una Gaussiana, como se ve en la Figura 11. Donde a mayor distancia entra a la Sigmoidal de Mayor Voltaje, en cuanto la pelota se encuentre en el centro entrará en la Gaussiana teniendo un Voltaje medio y en cuanto la pelota se encuentre a menor distancia del sensor entra en la segunda Sigmoide de Bajo Voltaje, para así hacer bajar el eje y caiga la pelota, de esta forma entra en un control de posición de Lógica Difusa.

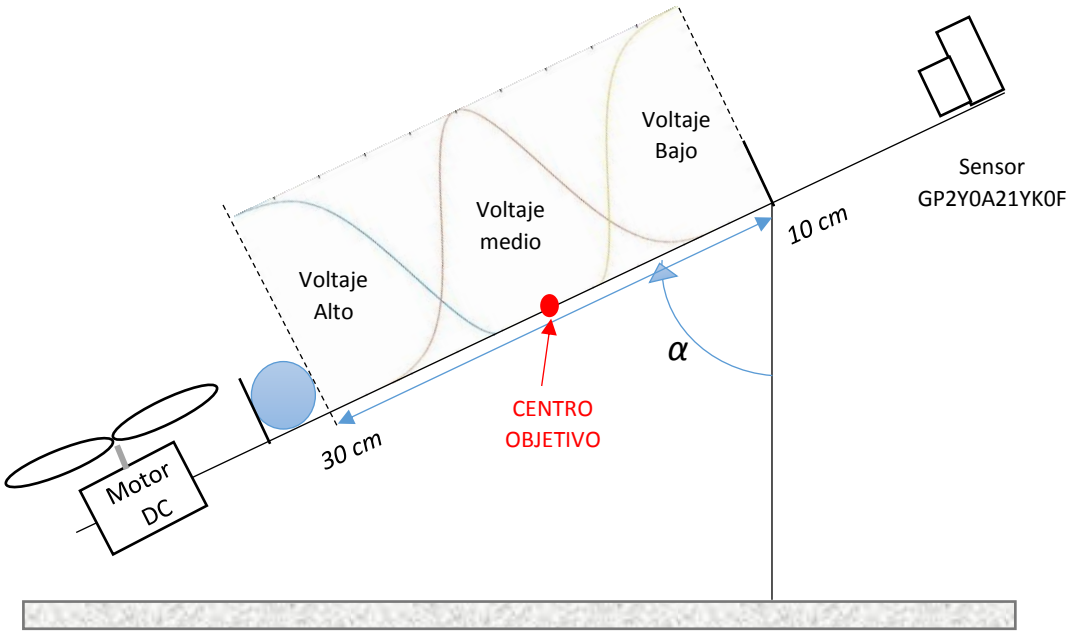


Figura 11. Modelo Difuso requerido en el Control del Sistema.

3.1.3.1 SIMULACIÓN DEL CONTROL DIFUSO EN MATLAB PARA EL SISTEMA

Las siguientes graficas son las implementadas en el control del sistema.

Código en Matlab:

```
%función e (entrada sensor)
val=10:0.01:30;
a=18;b=0.1;
res=(1-exp(-b*(a-val).^2)).*(val<a)+0.*(val>=a);
figure(1),plot(val,res),hold on

val=10:0.01:30;
a=20;b=0.1;
res=exp(-b*(val-a).^2);
figure(1),plot(val,res),hold on

val=10:0.01:30;
a=22;b=0.1;
res=(1-exp(-b*(val-a).^2)).*(val>a)+0.*(val<=a);
figure(1),plot(val,res),hold on,title('e')

%función e barra (complementario)
val=-10:0.01:0;
a=-2;b=0.1;
res=(1-exp(-b*(a-val).^2)).*(val<a)+0.*(val>=a);
figure(2),plot(val,res),hold on,title('e negado')

val=-10:0.01:10;
```

```
a=0;b=0.1;
res=exp(-b*(val-a).^2);
figure(2),plot(val,res),hold on,title('e negado')

val=0:0.01:10;
a=2;b=0.1;
res=(1-exp(-b*(val-a).^2)).*(val>a)+0.*(val<=a);
figure(2),plot(val,res),hold on,title('e negado')

%función hecho
val=10:0.01:30;
a=20;b=0.051;
res=exp(-b*(val-a).^2);
figure(3),plot(val,res),hold on,title('Hecho')
```

3.1.3.2 Resultados (Gráficas del Control) :

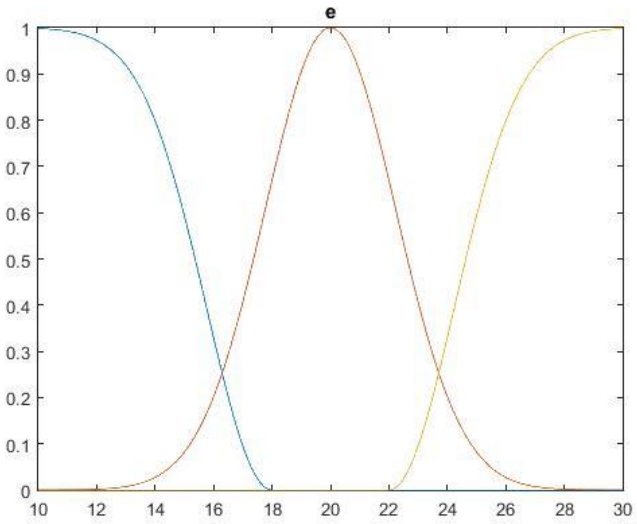


Figura I.I Gráfica e

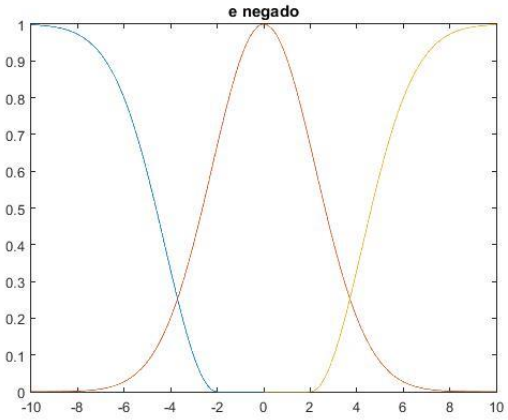


Figura I.II e complemento

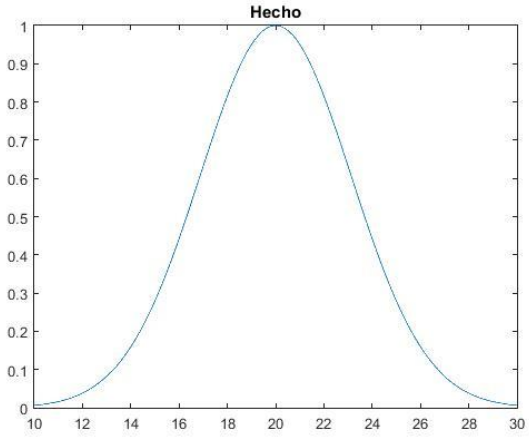


Figura I.III Hecho

3.1.3.3 Análisis

Como se mencionó en el marco teórico para aplicar control difuso no sólo necesitamos las funciones de pertenencia, **Fifura I.I** Sino que también aplicar **defuzzificación,** aquí se usó el centro de área (COA, *center of area*), esta usa el centro de gravedad o de masa para ver el grado de pertenencia o valor lógico de la variable, basado en la **fórmula 2.** Esta fórmula necesita de de la gráfica de la **Figura I.II** y la **Figura I.III**

3.1.3 SALIDA DEL SISTEMA

Cómo salida del sistema tenemos el Voltaje de alimentación del motor DC, por una señal de Ancho de Pulso (PWM), esta salida se obtiene directamente de la fórmula de defuzzificación ya que los CENTROS DE GRAVEDAD serán los valores de variación en los que rondará el PWM, ver Figura 12.

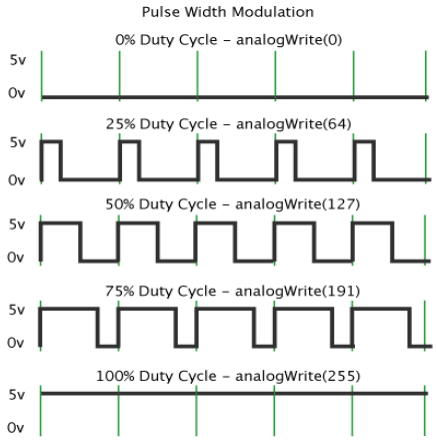


Figura 12. Modulación por Ancho de Pulso PWM.

3.1.3.1 Funcionamiento del PWM y CENTROS DE GRAVEDAD

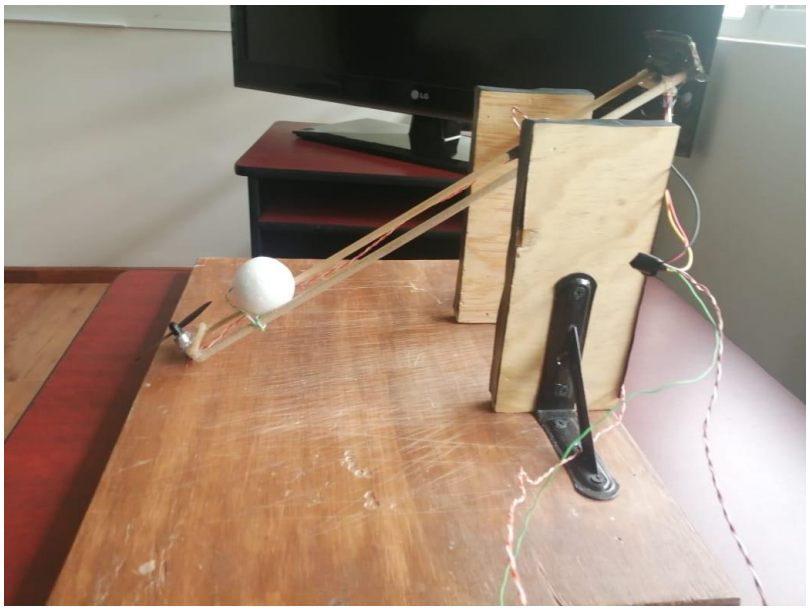
Ya que la señal PWM está en función del porcentaje de ciclo de trabajo, es decir a mayor área mayor voltaje y a menor área menor voltaje, ver Figura 12. Basándose en que si mandamos un valor de 0 tenemos 0V y 255 son 5v. Con esto los centros de gravedad elegidos por medio de la caracterización del sistema rondan entre:

- **C1** = 245;//248;//220;//115;//63;//75;
- **C2** = 252;//252;//230;//140;//97;//120;
- **C3** = 256;//256;//247;//205;//125;//163;

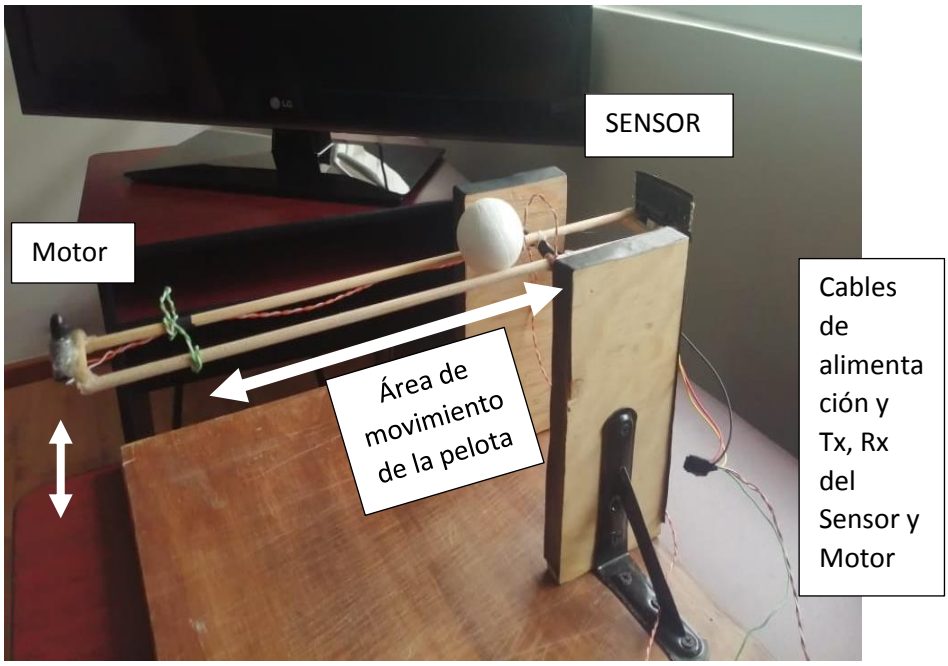
Estos centros de gravedad son los usados en la programación para las distintas pruebas hechas en el sistema, comentadas en la sección de Resultados.

4. IMPLEMENTACIÓN

Se armó un prototipo de la Figura 6, como se muestra en la siguiente imagen.



a)



b)

Figura 13. Maqueta del Balanceo de Pelota.

A continuación se muestra el sistema completo, que incluye la fuente de alimentación para el motor con su parte de potencia en la primer proto y en la segunda proto la tarjeta que contiene un PIC18F4550 junto con una LCD, y los convertidores A/D y D/A:

Figura 14. Sistema completo.

Con el sistema de adquisición, que incluye el Modulo RS232 para la adquisición de datos para así poder ser guardadas y procesadas en la Cumputadora:

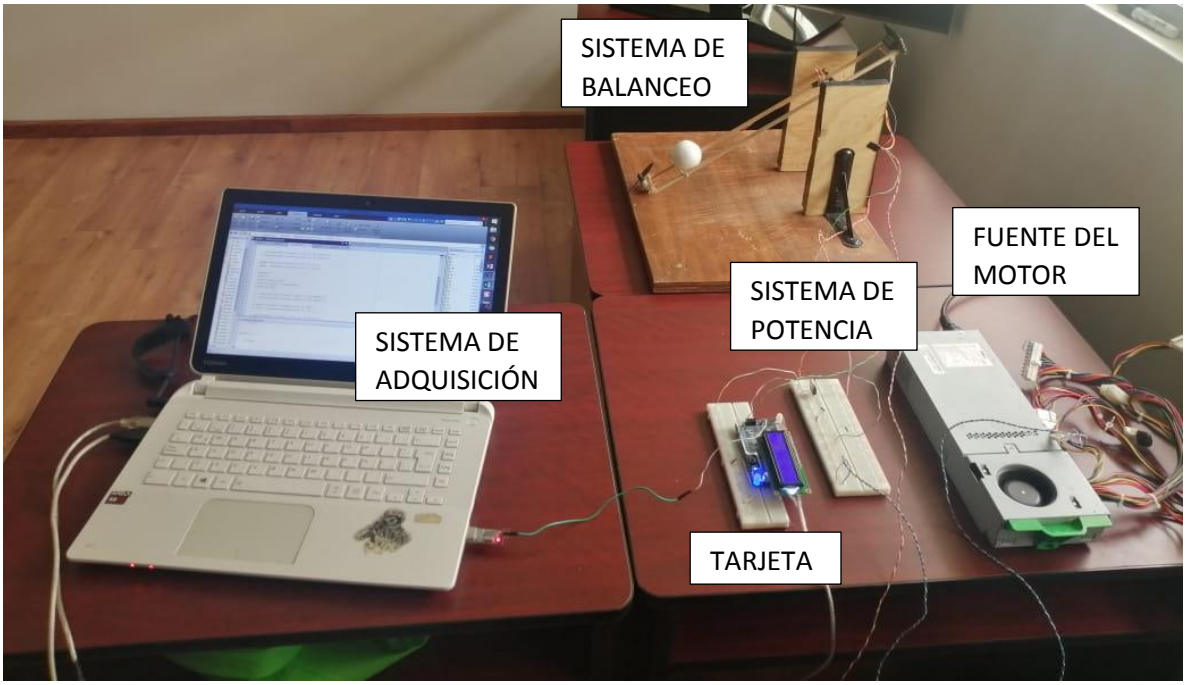


Figura 15. Sistema completo junto con la parte de adquisición de datos.

4.1 ESPECIFICACIONES

4.1.1 SISTEMA DE BALANCEO

Para la implementación primero se montó el Sistema de Balanceo **Figura 13**, este requiere principalmente de 2 Pasos:

1. Fijar el sensor de posición; alimentarlo a 5 V, y conectar la señal de salida al A/D de la tarjeta para la lectura de datos (lectura de la posición de la pelota).
2. Fijar el motor; conectarlo al sistema de Potencia.

4.1.2. SISTEMA DE POTENCIA

El sistema de potencia es usado para alimentar el motor, esto es necesario ya que la potencia de la señal de control arrojada por la Tarjeta (el PWM) no bastaría para levantar el eje, además de que podríamos quemar la tarjeta, por lo que hay que alimentarla con una Fuente externa y a mayor Potencia.

Este sistema es muy sencillo consta de una resistencia de 330ohms y de un transistor de potencia tipo TIP41C de una fuente que proporcione 12 V, ver **Figura**

16, además debe ser aterrizarlo a tierra común, es decir que todos los dispositivos deben estar aterrizados en la misma tierra.

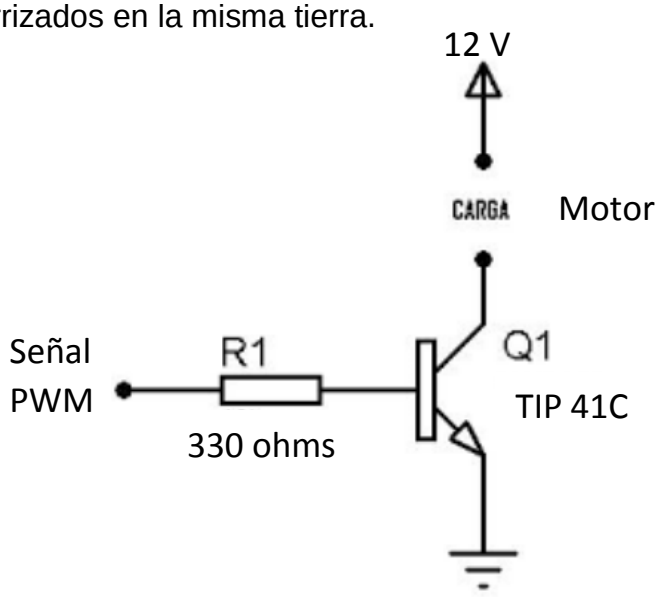


Figura 16. Sistema de Potencia.

Nota: el Motor se conecta en serie, tal y como se ve en la anterior imagen.

4.1.3. TARJETA

Se usó una tarjeta de desarrollo de Intesc, específicamente la µvva (Miuvva), esta trabaja con un microcontrolador de 8bits de Microchip. µvva cuenta con un microcontrolador PIC18F4550 y esta equipada con su propio programador/depurador, que le permitirá al desarrollador probar fácilmente su Firmware de cualquier compilador, ver **Figura 17**.

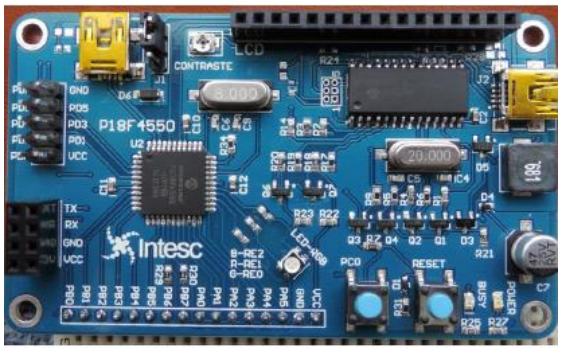


Figura 17. Tarjeta de desarrollo Miuva.

La tarjeta incluye muchos componentes electrónicos entre ellos un convertidor Analógico a Digital un PIC18F4550 y el Convertidor Digital a Analogico que son los requeridos para este proyecto, ver **figura 5**.

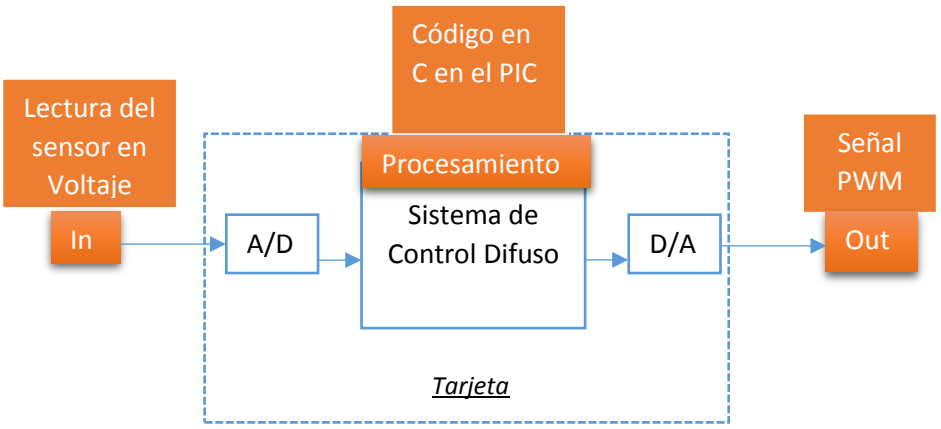


Figura 18. Diagrama de la tarjeta - Sistema.

4.1.4. SISTEMA DE ADQUISICIÓN

Se usó un Módulo RS232, protocolo serial para la Recepción y Transmisión de los datos, en específico las lecturas en tiempo real del sensor, para poder procesar y analizar los datos.
Se implementó la lectura en Matlab y se exportaron 6 mil datos en Excel para poder graficarlos, realizar su estadística descriptiva y graficar su tipo de distribución de probabilidad, para averiguar así el tipo de comportamiento que se tiene a la hora de intentar controlar la posición en el Sistema.

Código en Matlab:

```
close all;
clear;
clc;
RESPALDO = 0;
n=0;
VECTORFINAL=[];
s = serial('COM8','BaudRate',9600);
fopen(s);
cnt=0;
% p=0;
while cnt<6000
%M=zeros(1,cnt+1);
LECTURA=fread(s,1);
VECTORFINAL=cat(1,VECTORFINAL,LECTURA);

pause(0.01);
cnt = cnt + 1
end

fclose(s);
M=zeros(1,cnt);
p=0;
for a=1:cnt
    M(1,a)= p;
    p=p+1;
end
plot(M,VECTORFINAL)
title('Gráfica de Balanceo de Pelota')
xlabel('No de Muestras')
```

```
ylabel('Distancia (cm)')
```

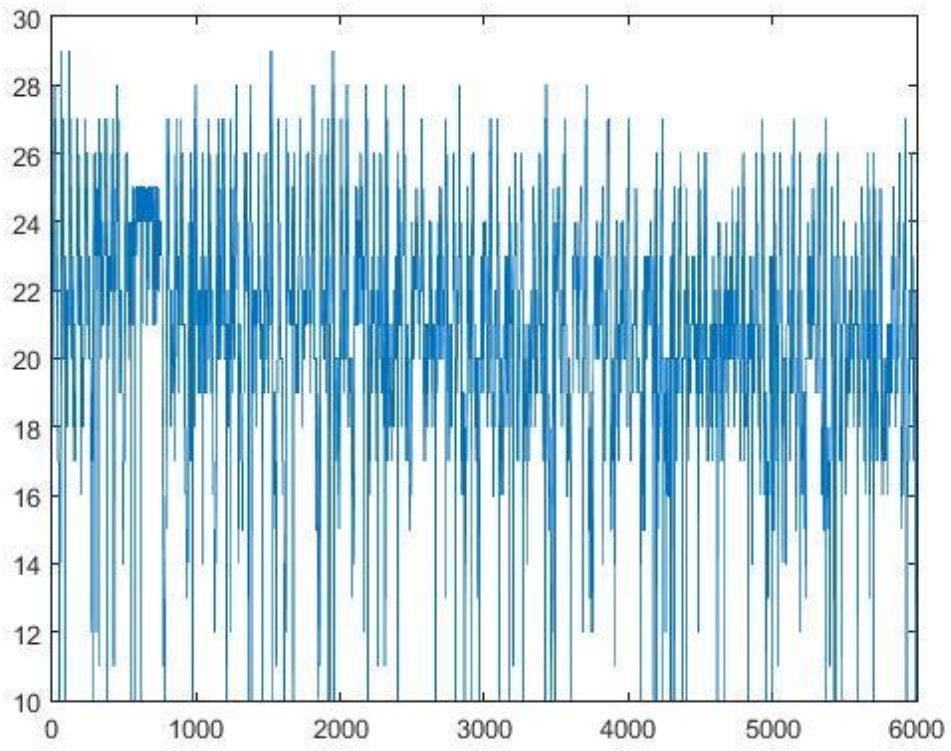
5. RESULTADOS

Los videos del Sistema se anexan junto con el reporte. A continuación se muestran las pruebas hechas con distintos Centros de Gravedad en la Lógica Difusa que arrojaran cambios en la pelota al intentar balancearla estas gráficas arrojadas en Matlab son las lecturas de posición hechas por el Sensor en Tiempo Real mientras corre el sistema. También se muestra una tabla de Estadística Descriptiva de los datos. Todas las pruebas constan de 6 mil datos (lecturas del Sensor), la recolección de estos datos tomo un tiempo de 2.30 mins por Prueba.

Prueba 1

Centros de Gravedad:

C1=245;
C2=252;
C3 = 256;



Gráfica 1. Prueba 1 de Balanceo de Pelota: No. De Muestras vs Distancia

Estadística descriptiva

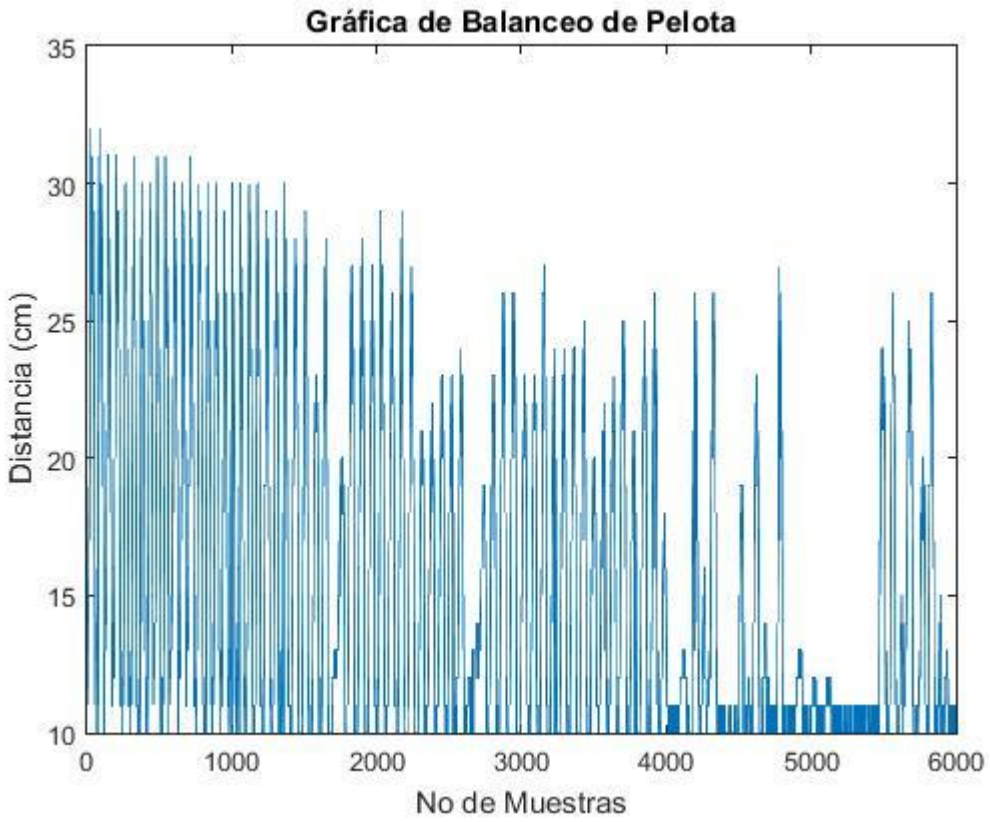
Media	**21.15766667**
Error típico	0.042381083
Mediana	**21**
Moda	21
Desviación estándar	**3.282824604**
Varianza de la muestra	10.77693738
Curtosis	1.054279235
Coeficiente de asimetría	-0.69406464
Rango	19
Mínimo	10
Máximo	29
Suma	126946
Cuenta	6000
Mayor (1)	29
Menor(1)	10
Nivel de confianza(95.0%)	0.08308216

Tabla 1. Estadística Descriptiva de Prueba 1.

Prueba 2

Centros de Gravedad:

C1=248;
C2=255;
C3 = 256;



Gráfica 2. Prueba 2 de Balanceo de Pelota: No. De Muestras vs Distancia

Estadística descriptiva	
Media	**15.499**
Error típico	0.07536542
Mediana	**13**
Moda	10
Desviación estándar	**5.83778031**
Varianza de la muestra	34.0796789
Curtosis	-0.63049481
Coeficiente de asimetría	0.8283872
Rango	22

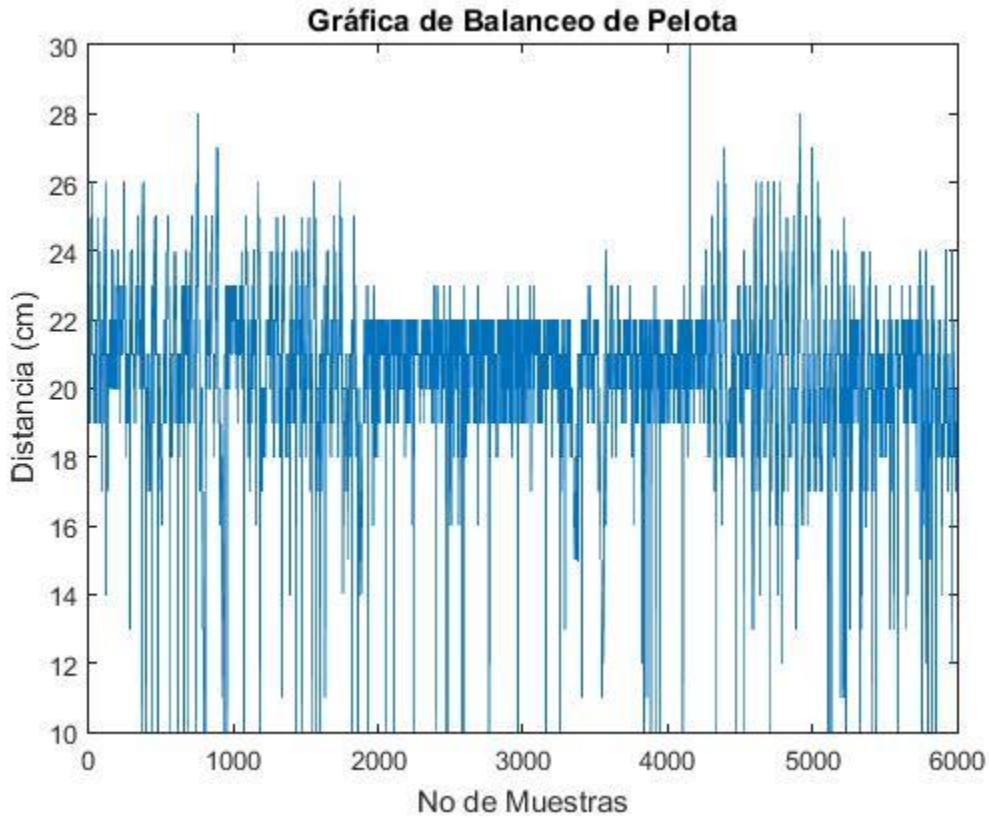
Mínimo	10
Máximo	32
Suma	92994
Cuenta	6000
Mayor (1)	32
Menor(1)	10
Nivel de confianza(95.0%)	0.14774332

Tabla 2. Estadística Descriptiva de Prueba 2.

Prueba 3

Centros de Gravedad:

C1=245;
C2=253;
C3 = 256;



Gráfica 3. Prueba 3 de Balanceo de Pelota: No. De Muestras vs Distancia

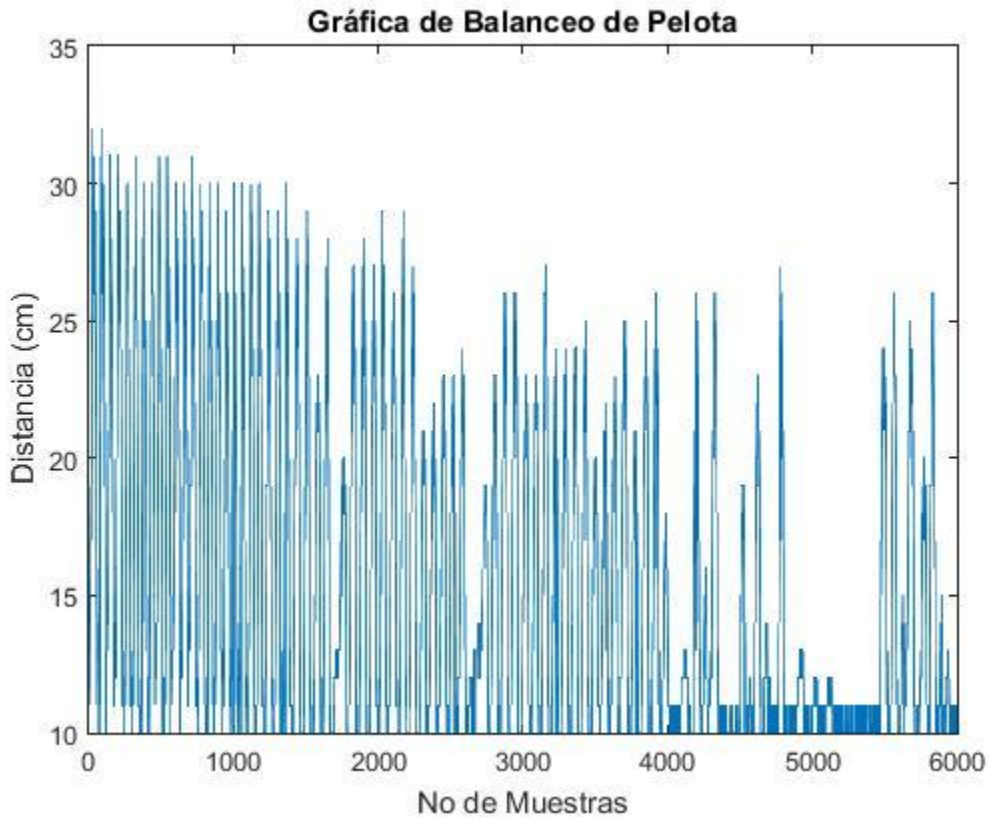
Estadística descriptiva	
Media	**20.78433333**
Error típico	0.029536306
Mediana	**21**
Moda	21
Desviación estándar	**2.287872436**
Varianza de la muestra	5.234360282
Curtosis	5.691970443
Coeficiente de asimetría	-1.542627012
Rango	20
Mínimo	10
Máximo	30
Suma	124706
Cuenta	6000
Mayor (1)	30
Menor(1)	10
Nivel de confianza(95.0%)	0.057901779

Tabla 3. Estadística Descriptiva de Prueba 3.

Prueba 4 *sensor más fijo

Centros de Gravedad:

C1=245;
C2=253;
C3 = 256;



Gráfica 4. Prueba 4 de Balanceo de Pelota: No. De Muestras vs Distancia

Estadística descriptiva	
Media	**17.6458333**
Error típico	0.0795439
Mediana	**17**
Moda	11
Desviación estándar	**6.16144408**
Varianza de la muestra	37.9633932
Curtosis	-1.22933607
Coeficiente de asimetría	0.34847004
Rango	20

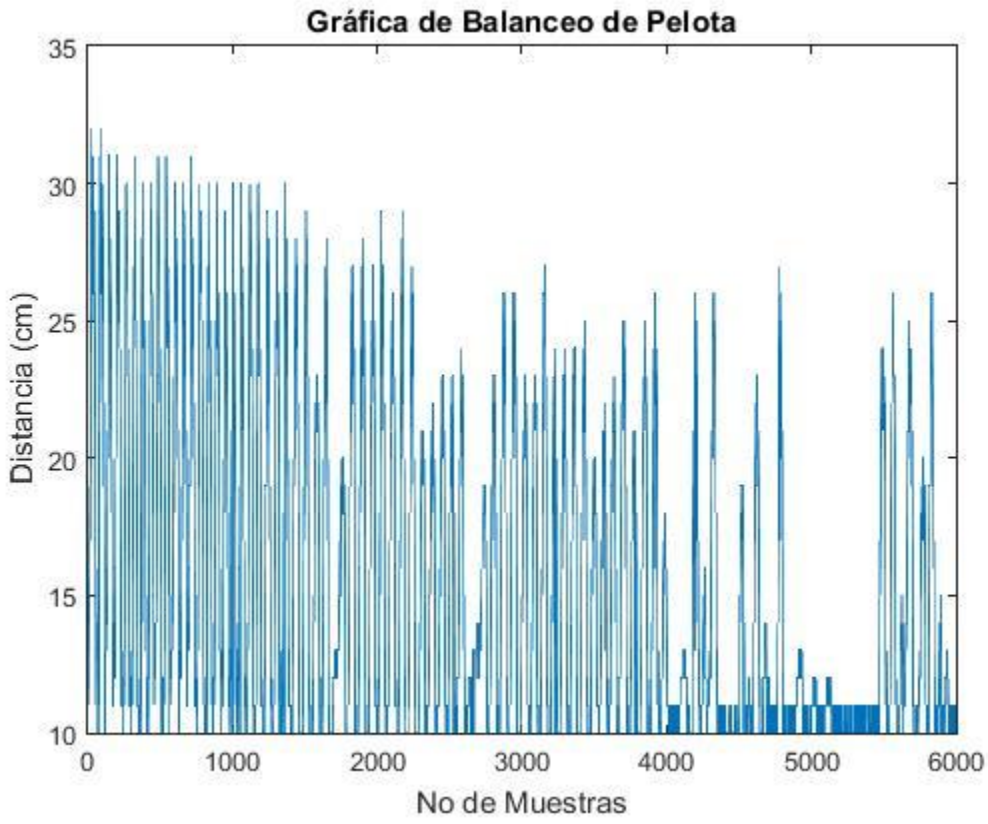
Mínimo	10
Máximo	30
Suma	105875
Cuenta	6000
Mayor (1)	30
Menor(1)	10
Nivel de confianza(95.0%)	0.15593464

Tabla 4. Estadística Descriptiva de Prueba 4.

Prueba 5

Centros de Gravedad:

C1=248;
C2=255;
C3 = 256;



Gráfica 5. Prueba 5 de Balanceo de Pelota: No. De Muestras vs Distancia

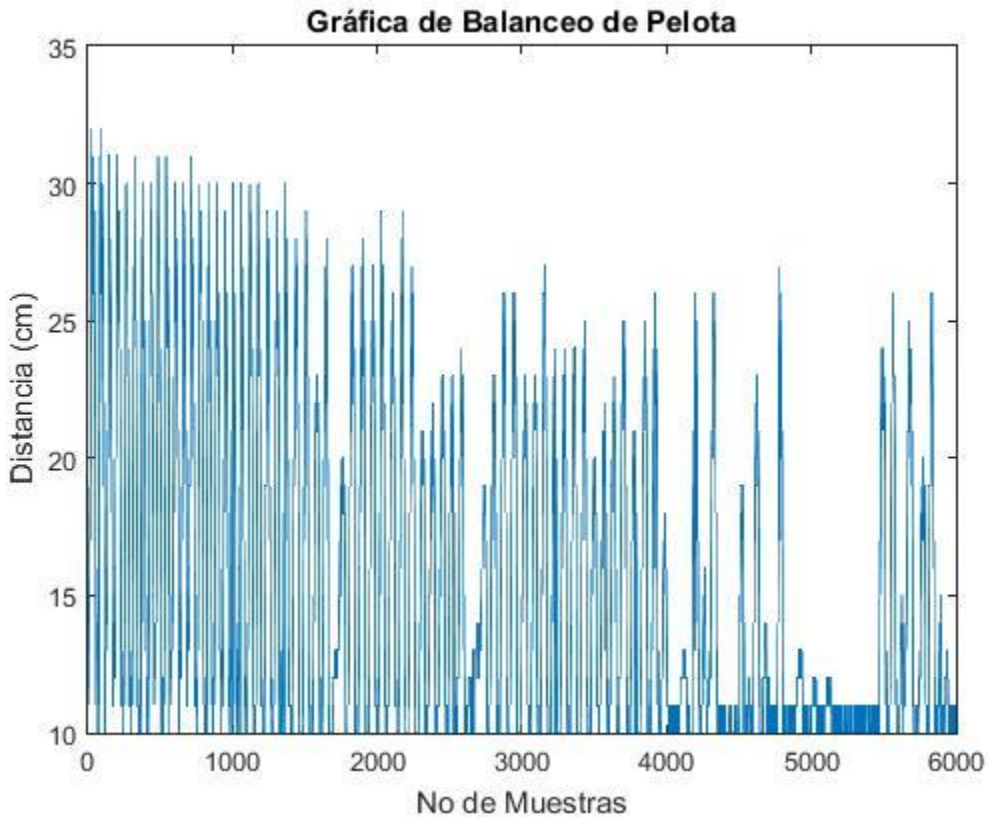
Estadística descriptiva	
Media	**17.5916667**
Error típico	0.05061427
Mediana	**19**
Moda	21
Desviación estándar	**3.92056446**
Varianza de la muestra	15.3708257
Curtosis	-1.18290605
Coeficiente de asimetría	-0.54371473
Rango	19
Mínimo	10
Máximo	29
Suma	105550
Cuenta	6000
Mayor (1)	29
Menor(1)	10
Nivel de confianza(95.0%)	0.09922216

Tabla 5. Estadística Descriptiva de Prueba 5.

Prueba 6

Centros de Gravedad:

C1=250;
C2=255;
C3 = 256;



Gráfica 6. Prueba 6 de Balanceo de Pelota: No. De Muestras vs Distancia

Estadística descriptiva	
Media	**17.936**
Error típico	0.04621034
Mediana	**20**
Moda	20
Desviación estándar	**3.57943749**
Varianza de la muestra	12.8123727
Curtosis	-0.56451652
Coeficiente de asimetría	-0.94696587
Rango	12

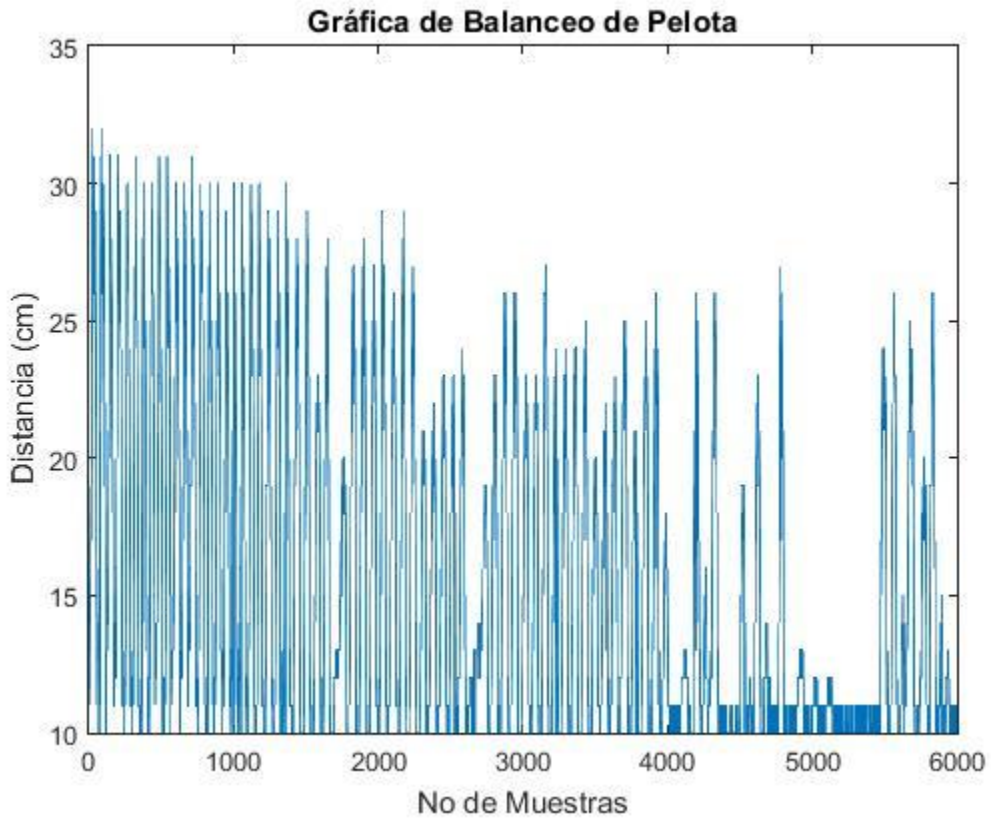
Mínimo	10
Máximo	22
Suma	107616
Cuenta	6000
Mayor (1)	22
Menor(1)	10
Nivel de confianza(95.0%)	0.09058888

Tabla 6. Estadística Descriptiva de Prueba 6.

Prueba 7

Centros de Gravedad:

C1=251;
C2=253;
C3 = 256;



Gráfica 7. Prueba 7 de Balanceo de Pelota: No. De Muestras vs Distancia

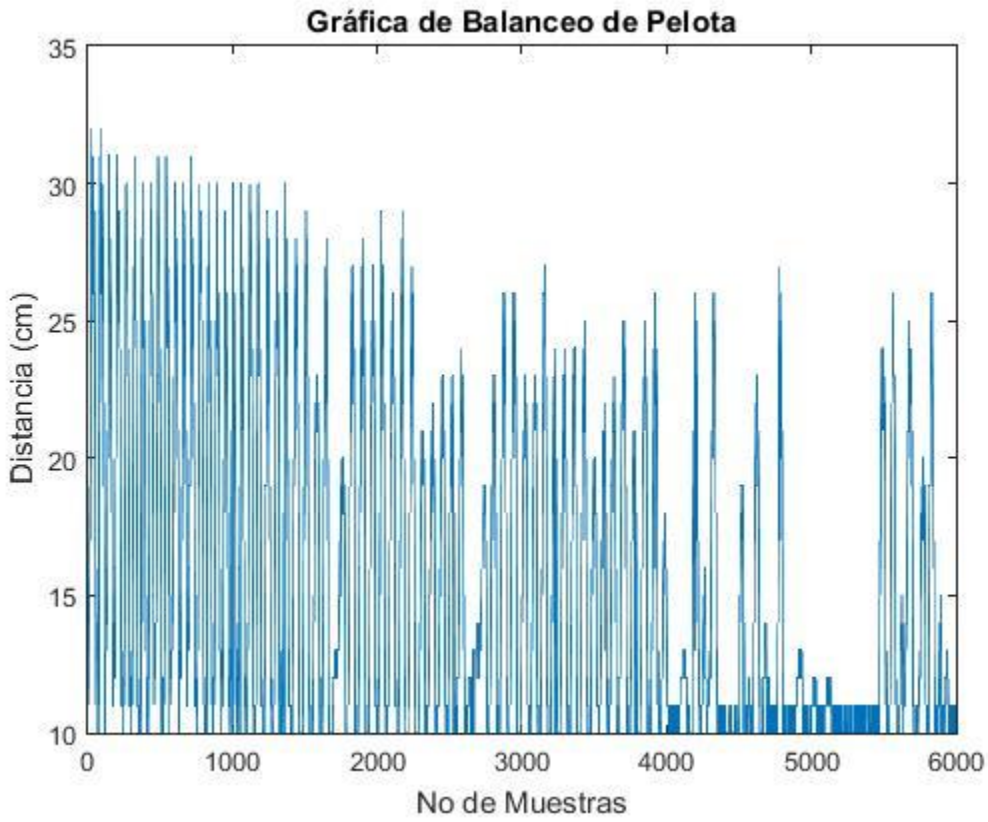
Estadística descriptiva	
Media	**15.721**
Error típico	0.07214495
Mediana	**14**
Moda	11
Desviación estándar	**5.58832389**
Varianza de la muestra	31.2293639
Curtosis	-0.67305345
Coeficiente de asimetría	0.81297521
Rango	19
Mínimo	10
Máximo	29
Suma	94326
Cuenta	6000
Mayor (1)	29
Menor(1)	10
Nivel de confianza(95.0%)	0.14143004

Tabla 7. Estadística Descriptiva de Prueba 7.

Prueba 8

Centros de Gravedad:

C1=250;
C2=253;
C3 = 256;



Gráfica 8. Prueba 8 de Balanceo de Pelota: No. De Muestras vs Distancia

Estadística descriptiva	
Media	**19.1786667**
Error típico	0.08358734
Mediana	**20**
Moda	11
Desviación estándar	**6.47464787**
Varianza de la muestra	41.9210651
Curtosis	-1.54724003
Coeficiente de asimetría	-0.08952458
Rango	29

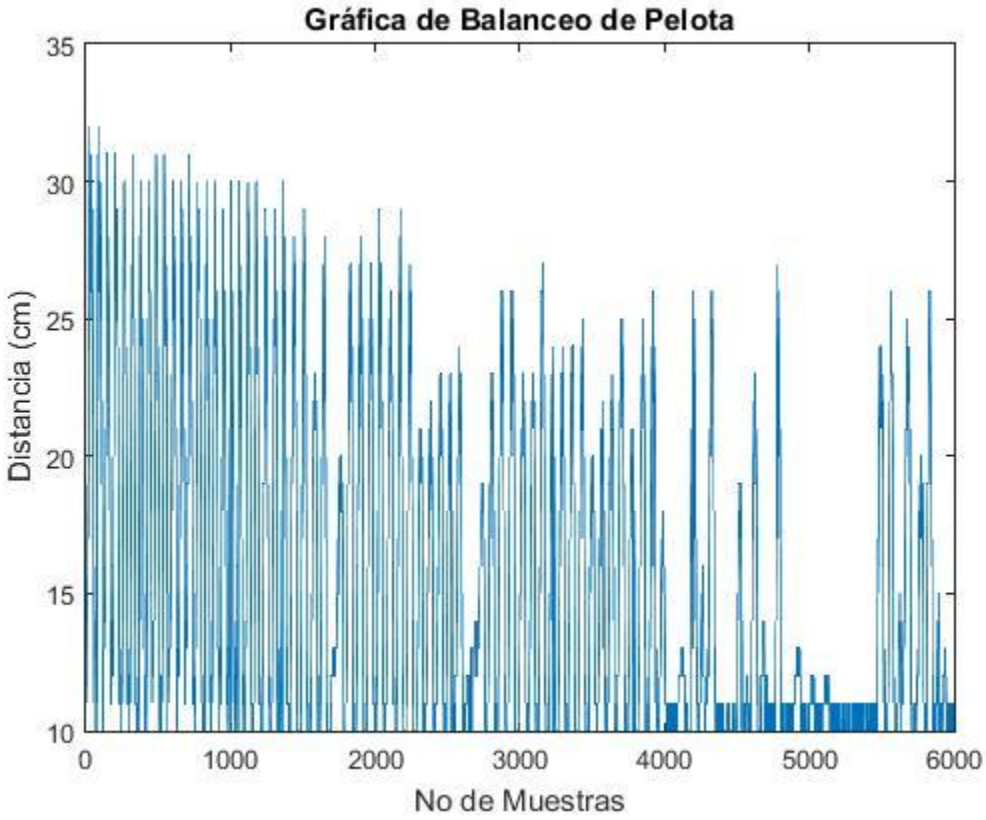
Mínimo	10
Máximo	39
Suma	115072
Cuenta	6000
Mayor (1)	39
Menor(1)	10
Nivel de confianza(95.0%)	0.16386125

Tabla 8. Estadística Descriptiva de Prueba 8.

Prueba 9

Centros de Gravedad:

C1=250;
C2=251;
C3 = 256;



Gráfica 9. Prueba 9 de Balanceo de Pelota: No. De Muestras vs Distancia

Estadística descriptiva	
Media	**18.6141667**
Error típico	0.08058377
Mediana	**19**
Moda	11
Desviación estándar	**6.24199165**
Varianza de la muestra	38.9624597
Curtosis	-1.4693461
Coeficiente de asimetría	0.07100959
Rango	32
Mínimo	10
Máximo	42
Suma	111685
Cuenta	6000
Mayor (1)	42
Menor(1)	10
Nivel de confianza(95.0%)	0.15797315

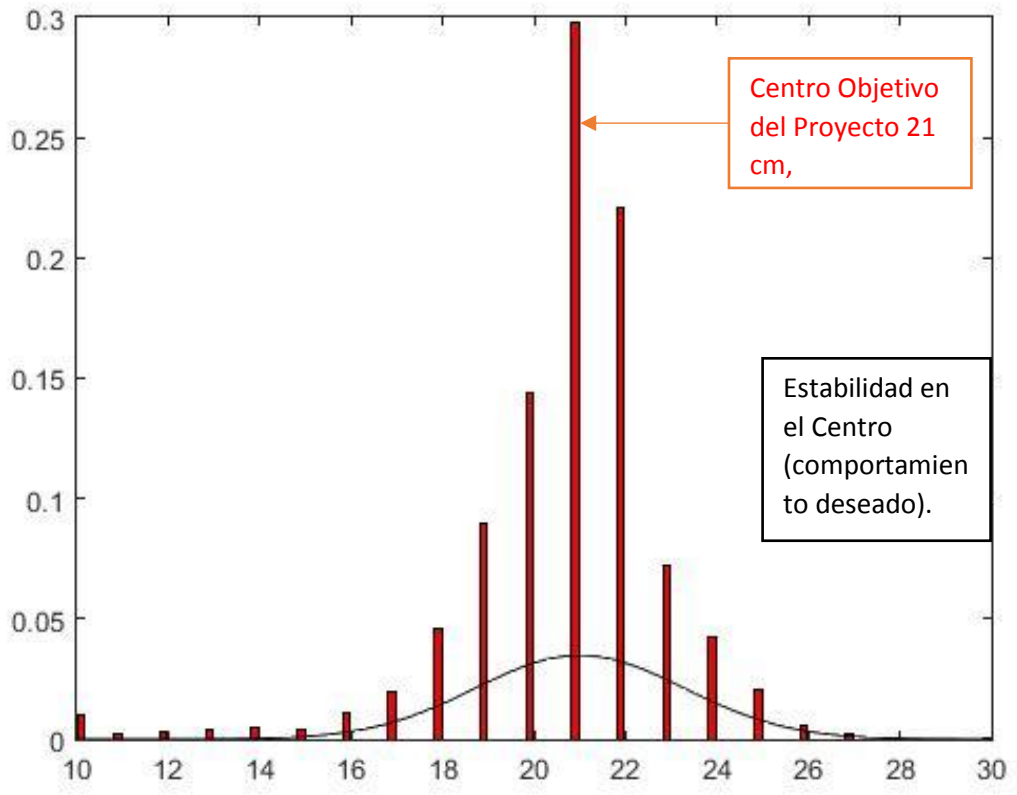
Tabla 9. Estadística Descriptiva de Prueba 9.

5.1 ANÁLISIS DE RESULTADOS

A continuación se presenta el análisis de la prueba que arrojo mejores resultados y la que arrojo peores resultados.

Como se puede ver en las Tablas de Estadística Descriptiva; la **Tabla 3 (Prueba 3)** esta fue la que arrojo mejores resultados para los Centros de Gravedad C1: 245, C2: 253 y C3: 256. Ya que tiene una menor desviación estándar de **2.287872436** y la Media más cercana a lo deseado de **20.78433333,** ver **Figura 7**, en comparación a las demás pruebas. La información que nos da la Desviación Estándar son que tanto fluctúa del valor de la media, es decir que tanto se desvía de dicho valor, con lo que sabemos que aquí tenemos una

mayor estabilidad en el sistema con centro en 20.78 cm. A continuación se muestra la distribución de probabilidad que siguieron a los datos de esta prueba.



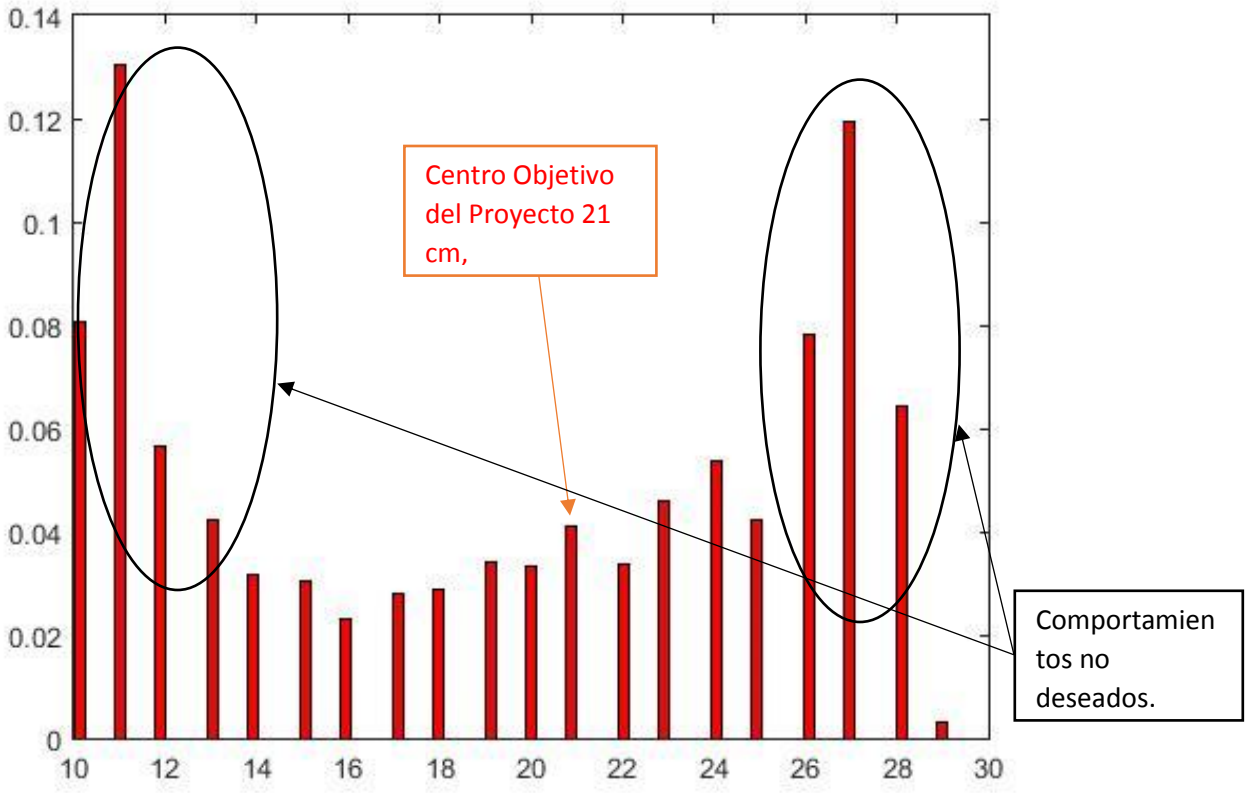
Gráfica 10. Distribución de Probabilidad de la **Prueba 3 Probabilidad (%) vs Distancia (cm).**

Con esta gráfica podemos ver claramente como el comportamiento de los datos sigue una distribución Normal o Gaussiana, ver tendencia de la línea negra, **Grafica 10**. Con lo cual podemos decir que el comportamiento es el deseado ya que los datos buscan una estabilidad en la media, notando que la mediana fue el objetivo del Proyecto en 21 cm ver **Figura 7** y Estadística descriptiva de esta Prueba. Repaso de su Estadística Descriptiva:

Estadística descriptiva	
Media	**20.78433333**
Error típico	0.029536306
Mediana	**21**

Moda	21
Desviación estándar	**2.287872436**
Varianza de la muestra	5.234360282
Curtosis	5.691970443
Coeficiente de asimetría	-1.542627012
Rango	20
Mínimo	10
Máximo	30
Suma	124706
Cuenta	6000
Mayor (1)	30
Menor(1)	10
Nivel de confianza(95.0%)	0.057901779

Análogamente también hacemos el análisis de la Prueba que arrojo peores resultados, para ver el comportamiento no deseado, esta fue la **Prueba 8.**



Gráfica 11. Distribución de Probabilidad de la **Prueba 8 Probabilidad (%) vs Distancia (cm).**

En la **Gráfica 11** podemos ver como el comportamiento de los datos se aleja mucho de la media buscada y se distribuye más en los extremos; por 11 y 27 cm. Con lo cual se tuvo un comportamiento más alejado del deseado. Repaso de su Estadística Descriptiva:

Estadística descriptiva	
Media	**19.1786667**
Error típico	0.08358734
Mediana	**20**
Moda	11
Desviación estándar	**6.47464787**
Varianza de la muestra	41.9210651

Curtosis	-1.54724003
Coeficiente de asimetría	-0.08952458
Rango	29
Mínimo	10
Máximo	39
Suma	115072
Cuenta	6000
Mayor (1)	39
Menor(1)	10
Nivel de confianza(95.0%)	0.16386125

6. CONCLUSIÓN

Se logró balancear una pelota en el centro de un riel por medio del control por Lógica Difusa. Este resultado se logó respaldar con el análisis de 6 mil datos por prueba, por el cambio de Centros de Gravedad en el Control de la Lógica Difusa en el Sistema, estos cambios se implementaron a nivel del código en C grabado en la Tarjeta de Desarrollo Miuva, ver Anexo. Realizando 8 pruebas y respaldando el mejor resultado (comportamiento deseado) tratándose de la **Prueba 3** Vs el peor resultado (el más alejado del deseado), tratándose de la **Prueba 8.** Con lo cual podemos garantizar que los resultados de este proyecto fueron los esperados.

7. BIBLIOGRAFÍA

[1] Hao Ying. “The simplest fuzzy controllers using different inference methods are different nonlinear proportional-integral controllers with variable gains”. Automática, vol 29 N° 6, pp. 1579-1589. 1993.

[2] Ambalal V. Patel, “Transformation Functions for Trapezoidal Membership Functions”, International Journal of Computational Cognition http://www.YangSky.com/yangijcc.htm Volume 2, Number 3, Pages 115–135, September 2004.

[3] Mourad Ousslaah, Hung T. Nguyen, Vladik Kreinovich. “A new derivation of centroide defuzzification”.

[4] Bonifacio Martín del Brio, Alfredo Sanz Molina. “Redes neuronales y sistemas difusos”. 2° Edición. © RA-MA Editorial. 2002.

8. ANEXO

Código en C grabado en la tarjeta. Incluye la lectura del Sensor, la Lógica difusa y la Defuzzificación que nos da el valor de PWM al sistema, además de que también tiene el código que nos da los valores del sensor impresos en la LCD.

```
//LCD
sbit LCD_RS at RD5_bit;
sbit LCD_EN at RD4_bit;
sbit LCD_D4 at RD0_bit;
sbit LCD_D5 at RD1_bit;
sbit LCD_D6 at RD2_bit;
sbit LCD_D7 at RD3_bit;

sbit LCD_RS_Direction at TRISD5_bit;
sbit LCD_EN_Direction at TRISD4_bit;
sbit LCD_D4_Direction at TRISD0_bit;
sbit LCD_D5_Direction at TRISD1_bit;
sbit LCD_D6_Direction at TRISD2_bit;
sbit LCD_D7_Direction at TRISD3_bit;

unsigned int sensor, kcentro;
char *text;
// VARIABLES
double A1, A2, A3, NEG, CEN, POS;
double C1, C2, C3;
double hecho;
int  dif,prueba;
```

```
double  K[ 3 ][ 3 ], Cs[ 3 ][ 3 ] , DD[ 3 ][ 3 ];
int i, j;
double num, den, yc;

//salida
unsigned char txt;
unsigned char txtUno;
unsigned int yc_pwm;

//Sensor
void distancia(void){
 sensor = ADC_Read(0);   //ADC 16 bits
 sensor = ADRESH;       //necesita dos registros L/H
 sensor = (sensor * 255);
 sensor = sensor+ ADRESL;

 sensor= (6050 / sensor);    //[cm] math[0-99cm]  d=1/volts;   escalando(6050)

 if(sensor >= 2) sensor =sensor - 2;   // error lineal (-2)
 if(sensor < 10) sensor = 10;
 if(sensor > 80) sensor = 80;
}

//Función de entrada  e
double FUNCIONA( double val , int op  ) {
double a,b, res;

if(op == 1 ) {
 a =  18.;    b =  0.1;   //Modificar para Ajustar
   if (val >= a)
      res = 0.;
   else
      res = 1. - exp( -b * ( ( a - val )* ( a - val)) ) ;
}
```

```
else if(op == 2){

  a =  20.; b =  .1;  //Modificar para Ajustar
  res = exp( -b*((val-a)*(val-a)) ) ;
}
else {

  a =  22.;  b =  0.1;  //Modificar para Ajustar
  if (val <= a)
    res = 0.;
  else
    res = 1. - exp( -b *((val-a)*(val-a)))   ;
}
  return res;
}

//FUNCION PARA OBTENER NEG, CEN , POS   e negada
double FUNCIONBARRA( double val , int op  ) {
double a,b,res;

if(op == 1 ) {
 a =  -2.;    b =  .1;   //Modificar para Ajustar
   if (val >= a)
      res = 0.;
   else
      res = 1. - exp( -b *(( a-val )*( a-val ))) ;
}
else if(op == 2){
  a =  0.; b =  .1;  //Modificar para Ajustar
  res = exp( -b* ((val-a )*( val-a )));
}
else {

  a =  2.;  b =  0.1;  //Modificar para Ajustar
```

```
  if (val <= a)
    res = 0;
  else
    res = 1. - exp( -b * (( val-a )*( val-a )) )   ;
}
  return res;
}
//FUNCION HECHO
double FUNCIONH ( double val ) {
double a,b,res;
  a =  20.; b =  0.051;  //Modificar para Ajustar
  res = exp( -b * (( val-a )*( val-a )) );
  return res;
}
void main(void)
{
 //ADCON1 = 0X0F;
 OSCCON = 0x66;
 ADCON0 = 0X01;     // RA0 analogica
 ADCON1 = 0X0E;
 TRISA0_bit = 1;     //entrada
 PORTA = 0x00;       //inicializada en 0
 PORTC = 0;
 TRISC = 0;

 PWM2_Init(5000);
 PWM2_Start();
 PORTB = 0;
 TRISB = 0;
 LCD_Init();
 LCD_Cmd(_LCD_CURSOR_OFF);
 LCD_Cmd(_LCD_CLEAR);

 UART1_Init(9600);
```

```
while(1){
distancia();
WordToStrWithZeros(sensor,txt);       //printf    distancia
Lcd_Out(1,1,txt);                     //(double) sensor;    //int-unsigned to double
kcentro = 19;  // Valor propuesto para k

C1 = 245;//248;//220;//115;//63;//75;
C2 = 252;//252;//230;//140;//97;//120;
C3 = 256;//256;//247;//205;//125;//163;

PORTB.F0= 1;
                        //e
A1 = FUNCIONA ( sensor, 1 );   //Valor de Pertenencia para el Conjunto A1     double
A2 = FUNCIONA ( sensor, 2 );   //Valor de Pertenencia para el Conjunto A2
A3 = FUNCIONA ( sensor, 3 );   //Valor de Pertenencia para el Conjunto A3

                      //e negada
dif = kcentro - sensor;               //La diferencia (int) entre el Valor de K y el obtenido  int-unsigned

NEG = FUNCIONBARRA( dif, 1 );   // Valores para NEG   double
CEN = FUNCIONBARRA( dif, 2 );   // Valores para CEN
POS = FUNCIONBARRA( dif, 3 );   // Valores para POS

                    //Hecho
hecho = FUNCIONH ( sensor ) ;   // Valor de Pertenencia del Hecho A'  double

//FloatToStr(hecho,txt);        //printf        hecho
//Lcd_Out(2,1,txt);

K[0][0] = 0;                  K[0][1] =(A1 < CEN)? A1: CEN;  K[0][2] = (A1 < POS)? A1: POS;  //reglas  double
K[1][0] = (A2 < NEG)? A2: NEG;  K[1][1] =(A2 < CEN)? A2: CEN;  K[1][2] = (A2 < POS)? A2: POS;
K[2][0] = (A3 < NEG)? A3: NEG;  K[2][1] =(A3 < CEN)? A3: CEN;  K[2][2] = 0 ;

Cs[0][0] = 0;            Cs[0][1] = C1;        Cs[0][2] = C1; //constantes
```

CENTROS DE GRAVEDAD

```
 Cs[1][0] = C2;//C1;
 Cs[1][1] = C2;
 Cs[1][2] = C2;//C3;
 Cs[2][0] = C3;          Cs[2][1] = C3;      Cs[2][2] = 0 ;

 for(i=0; i<3; i++ ){
   for (j=0; j<3; j++) {

     DD[i][j] = (K[i][j] < hecho)? K[i][j]: hecho;
   }
 }
 num = den = 0 ;

 for(i=0; i<3; i++ ){
   for (j=0; j<3; j++) {
    num += DD[i][j] * Cs[i][j];
    den += DD[i][j];
   }
 }
 yc=num/den;
 yc_pwm=yc; //3-100
 PWM2_Set_Duty(yc_pwm);
 IntToStr(yc_pwm,txt);          //printf
    }
}
```