

Inteligencia Artificial

Guías de Laboratorio



Visión

Ser una de las 10 mejores universidades privadas del Perú al año 2020, reconocidos por nuestra excelencia académica y vocación de servicio, líderes en formación integral, con perspectiva global; promoviendo la competitividad del país.

Misión

Somos una universidad privada, innovadora y comprometida con el desarrollo del Perú, que se dedica a formar personas competentes, íntegras y emprendedoras, con visión internacional; para que se conviertan en ciudadanos responsables e impulsen el desarrollo de sus comunidades, impartiendo experiencias de aprendizaje vivificantes e inspiradoras; y generando una alta valoración mutua entre todos los grupos de interés.



NORMAS BÁSICAS DE LABORATORIO

A. INGRESO AL LABORATORIO.

1. Para acceder al laboratorio se requiere, estar matriculado en el semestre académico y/o estar desarrollando un trabajo de investigación, así mismo firmar la declaración jurada, después de la inducción dada por el equipo de trabajo del Área de Energía.
2. Deben ingresar al laboratorio puesto con la indumentaria adecuada según la práctica programada (guantes descartables, casco de seguridad, lentes de seguridad, guantes dieléctricos, guardapolvo 100% de algodón y manga larga, zapatos dieléctricos etc.), caso contrario, NO SE PERMITIRÁ EL INGRESO DEL ALUMNO AL LABORATORIO.
4. No portar accesorios personales que puedan comprender riesgos de accidentes mecánicos, químicos o por fuego, como son anillos, pulseras, collares y sombreros.
5. Evitar el cabello suelto, debe estar sujetado.
6. Revise las medidas y el equipo de seguridad en el laboratorio.

B. PERMANENCIA EN EL LABORATORIO

1. Los objetos personales o innecesarios deben guardarse en la parte baja de las mesas para tal fin.
2. Aplicar las medidas de seguridad necesaria con los equipos y materiales.
3. Verificar el estado de los equipos y materiales, ANTES Y DESPUÉS DE LA PRACTICA PROGRAMADA. En el caso de tener alguna observación sobre el estado de ellos, informar inmediatamente al docente y/o al personal del laboratorio; caso contrario se presumirá que fue causado por él y/o los manipuladores, lo que conllevará a su responsabilidad y reposición del bien.
4. Mantener sólo el material requerido para la práctica; sobre la mesa de trabajo.
5. Trabajar adecuadamente y con responsabilidad.
6. No usar los celulares dentro de las prácticas.
7. No ingerir alimentos ni bebidas en el interior del laboratorio.
8. Respetar y obedecer las señalizaciones de seguridad.
9. Evitar las distracciones durante las prácticas a desarrollarse.

C. PARA USO DE LOS EQUIPOS

1. Se atenderá de acuerdo el requerimiento presentado en forma virtual o física por el docente.
2. El uso de los equipos en su totalidad es de uso exclusivo dentro del campus universitario.



3. En el caso que amerite la salida de un equipo fuera de la universidad, se realizará con documento de autorización del docente del curso y en coordinación respectivas con el Área de Control Patrimonial de la Universidad.
4. Los equipos serán entregados al jefe de cada grupo previa entrega de su Carnet Universitario actual y DNI, operativos y funcionando correctamente.
5. En el caso de descalibración o deterioro del equipo por mal manejo, los gastos de calibración y reparación corre a cuenta de todos los integrantes del grupo.
6. El estudiante que sustraiga material del laboratorio será severamente sancionado, en concordancia con el reglamento de disciplina de la Universidad.

D. **AL CONCLUIR LA PRÁCTICA**

1. Disponer de los residuos al tacho para residuos generales.
2. Dejar la mesa de trabajo limpia y ordenada.
3. Dejar las sillas ordenadas.
4. Antes de salir del laboratorio retírese el guardapolvo y demás equipo de seguridad y guárdelo en una bolsa de plástico exclusiva para este uso.
5. Devolver los equipos limpios y en las mismas condiciones que se les entrego al Personal del Área de Energía.
6. En el laboratorio no se permitirá el almacenamiento de objeto alguno que no corresponda con los fines y objetivos académicos del mismo, y de encontrarse será retirado por el personal de mantenimiento.

E. **DE LOS MATERIALES Y EQUIPOS DETERIORADOS**

1. En caso que el alumno deteriore algún material y/o equipo, que impidan su buen estado y funcionamiento, POR MALA UTILIZACIÓN DEL MISMO; se registrara los datos del alumno responsable, quien tiene un plazo de 48 horas para la reposición del material y/o equipo, de las mismas características o superior, del bien deteriorado.
2. En el caso que se incumpla lo anterior, el alumno o alumnos firmaran un formato de autorización de recargo a su cuenta personal; el mismo que debe hacer efectivo en caja de la universidad.



Índice

| | |
|---|----|
| VISIÓN | 2 |
| MISIÓN | 2 |
| NORMAS BÁSICAS DE LABORATORIO | 3 |
| ÍNDICE | 5 |
| PRIMERA UNIDAD | |
| Guía de práctica N° 1: Introducción a MATLAB | 6 |
| Guía de práctica N° 2: Vectores y matrices en MATLAB | 9 |
| Guía de práctica N° 3: Gráficos en MATLAB | 12 |
| SEGUNDA UNIDAD | |
| Guía de práctica N° 4: Neurona Mc Culloch - Pits | 15 |
| Guía de práctica N° 5: Separabilidad lineal | 18 |
| Guía de práctica N° 6: Perceptrón de una capa | 21 |
| TERCERA UNIDAD | |
| Guía de práctica N° 7: Reconocimiento de números | 24 |
| Guía de práctica N° 8: Función de pertenencia triangular | 27 |
| Guía de práctica N° 9: Función de pertenencia trapezoidal | 30 |
| CUARTA UNIDAD | |
| Guía de práctica N° 10: Función de pertenencia Gaussiana y sigmooidal | 33 |
| Guía de práctica N° 11: Diseño de un controlador difuso (1) | 36 |
| Guía de práctica N° 12: Diseño de un controlador difuso (2) | 40 |



Guía de práctica N° 1

Introducción a MATLAB

Sección : AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

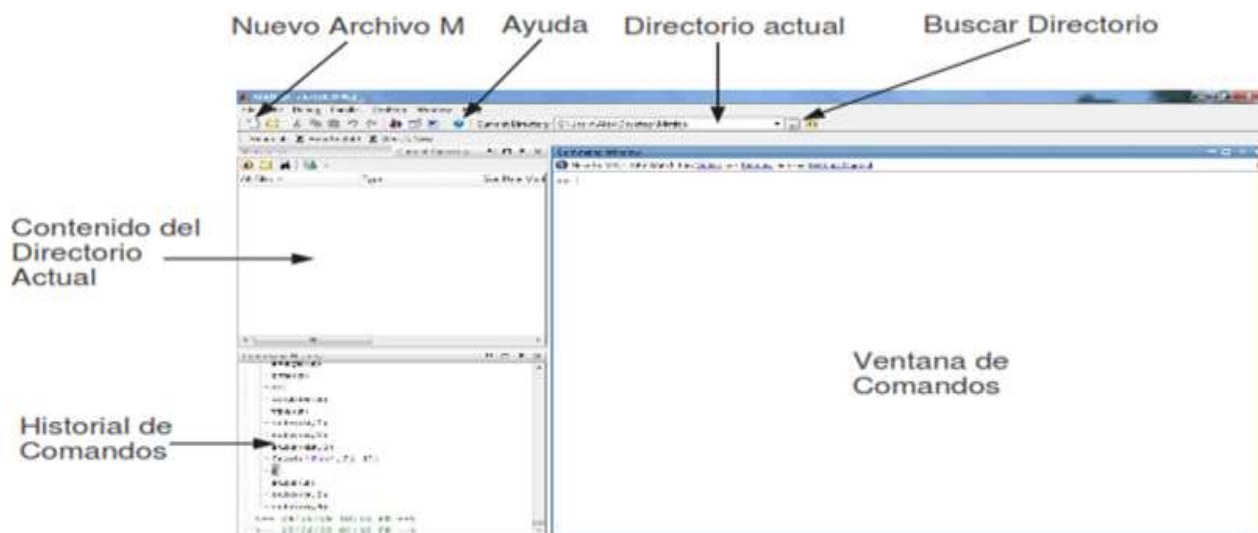
1. Objetivos

- Identificar los comandos básicos de MATLAB.
- Escribir programas básicos en MATLAB.

2. Fundamento Teórico

MATLAB es el nombre abreviado de "MATrix LABoratory". Es un programa para realizar cálculos numéricos con vectores y matrices. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones.

En la ventana de comandos se realiza todo tipo de operaciones. Los comandos se escriben a continuación del prompt (>>). Para realizar un programa es preferible crear un archivo **m** independiente. MATLAB es sensible a las mayúsculas y minúsculas.





3. Equipos

3.1. Equipos

| Ítem | Equipo | Característica | Cantidad |
|------|--------|------------------------|------------------|
| 1 | PC | Programa MATLAB/Octave | 1 por estudiante |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión.

5. Procedimientos:

5.1. Realizar las siguientes operaciones elementales en la ventana de comandos

- a. `>> 2 + 8`
- b. `>> 2 ^ 4`
- c. `>> 2*3^2 + (4 - 8)*3`
- d. `>> 2/3 + 5 - 8`
- e. `>> 2/3 + 5/3`
- f. `>> rat(2/3 + 5/3)`
- g. `>> sqrt(64)`
- h. `>> format long`
`>> pi`

5.2. Ejecutar las funciones aritméticas y trigonométricas

- a. `>> factor(44)` % factores primos de 44
- b. `>> primes(24)` % números primos hasta 24
- c. `>> isprime(15)` % determinar si un número es primo
- d. `>> gcd(12, 14)` % máximo común divisor de 12 y 14
- e. `>> lcm(12,4)` % mínimo común múltiplo de 12 y 4
- e. `>> factorial(5)` % factorial de 5
- f. `>> sin(pi/2)` % seno de pi/2
- g. `>> tan(pi/4)` % tan de pi/4

5.3. Uso de variables

- a. `>> x = 42;`
`>> a = 5;`
`>> x`
`>> x = 23;`
`>> x + a`
- b. `>> x = 5;`
`>> x = x + 2;`
`>> x`
- c. `>> b = 3;`
`>> c = 4;`
`>> c < b`

5.4. Comunicación con el usuario

- a. `>> disp('Curso de Matlab');`
- b. `>> x = 43;`
`>> disp(x);`
- c. `>> a = 56;`
`>> fprintf('El valor de a es %d \n', a);`
- d. `>> a = 56;`
`>> b = 25.5;`
`>> fprintf('Una es %d y la otra es %f ', a,b);`
- e. `>> n = input('Ingresar un numero');`
- f. `>> r = input('Ingresar su nombre', 's');`



5.5. Escribir programa, utilizando el editor de matlab, para calcular el área del círculo.

```
% AREA DEL CIRCULO
% Entrada de datos
r = input(' leer radio del circulo : ');
% Proceso
A = pi*r*r;
% Mostrar la salida
fprintf(' El área es %f\n ', A);
```

6. Resultados

6.1.
.....
.....
.....

6.2.
.....
.....
.....

6.3.
.....
.....
.....

7. Conclusiones

7.1.....
.....

7.2.....
.....

7.3.....
.....

8. Sugerencias y /o recomendaciones

8.1.....
.....

8.2.....
.....

8.3.....
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Báez, L. y Cervantes, V. (2012). *MATLAB con aplicaciones a la ingeniería* (2ª ed.). Edit. Alfaomega,
- Hanselman, D. y Littlefield (1996). *MATLAB* (4ª ed.). Edit. Prentice Hall.
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- <https://es.mathworks.com>



Guía de práctica N° 2

Vectores y matrices en MATLAB

Sección : ...AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos

- Conocer los comandos de creación de vectores y matrices en MATLAB
- Conocer los comandos para operaciones con vectores y matrices en MATLAB

2. Fundamento Teórico

MATLAB maneja en forma matricial todas las variables establecidas por el usuario como las predefinidas dentro de MATLAB. Los elementos de una matriz pueden ser números, letras, matrices, etc. Por ejemplo, una tabla de números forma una matriz, así como también los pixeles de una imagen.

Vector fila o columna

>> A = [2 0 -3 5 -8], B = [2; 0; -3; 5; -8]

A = [2 0 -3 5 -8] Vector fila

B = $\begin{bmatrix} 2 \\ 0 \\ -3 \\ 5 \\ -8 \end{bmatrix}$ Vector columna

Matriz

>> A = [2 3 -2 5; 4 0 3 6; 6 -1 -4 -7]

A = $\begin{pmatrix} 2 & 3 & -2 & 5 \\ 4 & 0 & 3 & 6 \\ 6 & -1 & -4 & -7 \end{pmatrix}$ } Filas = 3 \Leftrightarrow m = 3

} Columnas = 4 \Leftrightarrow n = 4

A es de orden 3 x 4, o bien **A** $_{3 \times 4}$

Operaciones con matrices



Son la suma, la resta, el producto escalar por una matriz, el producto y la división. Al realizar estas operaciones es importante verificar que las matrices satisfagan las condiciones establecidas en la definición de la operación respectiva.

```
>> A + B
>> A - B
>> A*B
>> A/B
```

Potencia de una matriz

Si p es un escalar A^p es la matriz elevada al escalar p

```
>> A = [2 5; 4 9];
>> p = 2;
>> A ^ p
ans =
    24 55
    44 101
```

Operaciones punto

También existen las operaciones punto multiplicación y punto división. Los resultados son operaciones elemento a elemento de las matrices.

```
>> A = [2 5; 4 9];
>> p = 2;
>> A .^ p
ans =
     4 25
    16 81
```

3. Equipos

3.1. Equipos

| Ítem | Equipo | Característica | Cantidad |
|------|--------|--------------------------------------|------------------|
| 1 | PC | Instalado con programa MATLAB/Octave | 1 por estudiante |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión.

5. Procedimiento:

5.1 Creación de vectores, matrices y ubicar elementos

- a. >> A = [1 2 3] % vector fila
- b. >> B = [1; 2; 3] % vector columna
- c. >> C = [1 2 4; 5 7 8] % matriz de 2 filas y 3 columnas
- d. >> D = [1:1:10] % vector fila de 10 elementos 1,2,...10
- e. >> E = [4:15] % vector fila de 11 elementos 4,5,...15
- >> E(1) % mostrar primer elemento de vector E
- f. Definir el vector V = (2, 3, 4,20) mostrar tercera y quinta componente.

5.2 Operaciones con vectores

- a. Dado los vectores v = (1, 0, -2, 5) y w = (1, 2, 3, 6) y c = 5, calcular: v + w, v - w, c*w
- b. >> C = [1 2 4; 5 7 8]
- >> D = [1 2; 4 5; 7 8]
- >> E = C*D %producto de matrices
- c. >> [1 2 3 4 5].^4 % operación punto
- d. >> v = [2 4 34 -2 3 8 9 0];
- >> max(v) % elemento mayor de v
- >> min(v) % elemento menor de v



```
>> sum(v)           % suma de elementos de v
>> prod(v)          % producto de elementos de v
>> mean(v)          % media aritmética de v
```

5.3 Polinomios

Se tienen los polinomios $p_1(x) = x^4+x^3-x^2+4x+8$ y $p_2(x) = x^3-2x^2+4x+8$

```
>> p1 = [1 1 -1 4 8]      % definir p1(x)
>> p2 = [1 -2 4 8]       % definir p2(x)
>> polyval(p1,2)         % evaluar p1(2)
>> polyval(p2,0)         % evaluar p2(0)
>> roots(p1)             % raíces de p1
>> conv(p1,p2)           % multiplicar p1(x) y p2(x)
>> [c,r] = deconv(p1,p2) % dividir p1(x) entre p2(x)
```

6. Resultados

- 6.1.
.....
.....
.....
- 6.2.
.....
.....
.....
- 6.3.
.....
.....
.....

7. Conclusiones

- 7.1.....
.....
- 7.2.....
.....
- 7.3.....
.....

8. Sugerencias y/o recomendaciones

- 8.1.....
.....
- 8.2.....
.....
- 8.3.....
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Báez, L. y Cervantes, V. (2012). *MATLAB con aplicaciones a la ingeniería* (2ª ed.). Edit. Alfaomega,
- Hanselman, D. y Littlefield (1996). *MATLAB* (4ª ed.). Edit. Prentice Hall.
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- <https://es.mathworks.com>



Guía de práctica N° 3

Gráficos en MATLAB

Sección :Al1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos:

- 1.1. Conocer los comandos de MATLAB para graficar en 2D
- 1.2. Conocer los comandos de opciones gráficas de MATLAB

2. Fundamento Teórico

Una de las características más importantes de MATLAB es la visualización.

Gráficos en 2D

La instrucción básica es `plot(x,y)` donde `x` es un vector de puntos igualmente espaciados que se puede generar con:

```
>> x = linspace(x1, x2, n)
```

Donde `n` es número de puntos equidistantes, `x1` es el primer punto y `x2` es el último punto.

El vector `y = f(x)` es de la misma dimensión del vector `x`

Ejemplo:

```

>> x = linspace(0,2*pi,100);      % x es un vector de 100 puntos de 0 a 2*pi
>> y = sin(x)                    % función a graficar
>> plot (x,y)                   % graficar
>> xlabel('x-radianes');        % etiqueta en el eje de abscisas
>> ylabel('f(x)');              % etiqueta en el eje de ordenadas
>> title('Sen(x)')              % título de la gráfica
>> grid on                       % activa la cuadrícula en la gráfica

```

Opciones de gráficas

Existen varias opciones mediante las cuales se puede proporcionar más información en una gráfica:

| Código de color | Color | Codigo de color | Color |
|-----------------|----------|-----------------|--------|
| y | Amarillo | g | Verde |
| m | magenta | b | azul |
| c | Cyan | w | blanco |
| r | Rojo | k | negro |

| Código de marcador | Marcador | Código de estilo de línea | Estilo de línea |
|--------------------|----------|---------------------------|-----------------|
| . | Punto | - | Sólida |



| | | | |
|---|-----------|----|-------------------|
| ○ | Círculo | : | Línea de puntos |
| x | X | -. | Línea y punto |
| * | asterisco | -- | Línea discontinua |

Ejemplo: Graficar $f(x) = x \sin x$

```
>> x = -10:0.04:10;
>> y = sin(x).*x;
>> plot(x,y,'—');
```

Modificación de escala

Es posible modificar el máximo y el mínimo de las coordenadas

```
axis([xmin, xmax, ymin, ymax]);
```

Subplot

Una ventana gráfica se puede dividir en m particiones horizontales y n particiones verticales.

```
subplot(m,n,i) % i es el número secuencial de la gráfica
```

Ejemplo:

```
>> X = -1.5:0.05:1.5;
>> Y1 = sin(x); y2 = x.^2; y3 = tan(x); y4 = sin(x).*x;
>> subplot(2,2,1), plot(x,y1,'r');
>> subplot(2,2,2), plot(x,y2,'g');
>> subplot(2,2,3), plot(x,y3,'y');
>> subplot(2,2,4), plot(x,y4,'c');
```

3. Equipos

3.1. Equipos

| Ítem | Equipo | Característica | Cantidad |
|------|--------|---|---------------------|
| 1 | PC | Instalado con programa MATLAB/Octave | 1 por estudiante |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión.

5. Procedimiento:

5.1. Gráficos en 2D en una ventana

- a.

```
>> x = 0:0.05:2*pi;
>> y = sin(x);
>> plot(x,y)
```
- b.

```
>> x = 0:0.05:2*pi;
>> y = cos(x);
>> plot(x,y)
>> title('FUNCION COSENO');
>> xlabel('eje X')
>> ylabel('eje Y')
>> grid on
```
- c.

```
>> x = -2*pi:0.05:2*pi;
>> y = sin(x);
>> plot(x,y,'x');
>> xlabel('X'); ylabel('Y');
>> axis([-8,10,-30,30]);
```



```
d. x=0:0.05:2*pi;  
>> y = sin(x);  
>> z = cos(x);  
>> plot(x,y,x,z)
```

5.2. Gráficos en 2D en más de una ventana

```
a. >> x = -2:0.05:2;  
>> y1 = x.^2; y2 = x.^3; y3= cot(x); y4=cos(x);  
>> subplot(2,2,1);plot(x,y1,'r');  
>> subplot(2,2,2);plot(x,y2,'g');  
>> subplot(2,2,3);plot(x,y3,'y');  
>> subplot(2,2,4);plot(x,y4,'c');
```

6. Resultados

6.1.
.....
.....
.....

6.2.
.....
.....
.....

6.3.
.....
.....
.....

7. Conclusiones

7.1.
.....

7.2.
.....

7.3.
.....

8. Sugerencias y/o recomendaciones

8.1.
.....

8.2.
.....

8.3.
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Báez, L. y Cervantes, V. (2012). *MATLAB con aplicaciones a la ingeniería* (2ª ed.). Edit. Alfaomega,
- Hanselman, D. y Littlefield (1996). *MATLAB* (4ª ed.). Edit. Prentice Hall.
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- <https://es.mathworks.com>



Guía de práctica N° 4

Neurona McCulloch-Pits

Sección : ...AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos:

- 1.1. Conocer modelo neuronal McCulloch-Pits
- 1.2. Implementar aplicación de la neurona McCulloch-Pits

2. Fundamento Teórico

Modelo básico de la neurona

Propuesto por McCulloch y Pitts en 1943. Calcula la suma ponderada de sus entradas producidas por otras unidades, y da como salida uno (1) si ésta se encuentra por encima de un valor denominado umbral, o cero (0) si está por debajo. La ecuación que gobierna el funcionamiento de dicho modelo de neurona es la siguiente:

$$n_i(t+1) = f(\sum_j (w_{ij} * n_j(t)) - u_i)$$

donde:

w_{ij} es el peso de la conexión entre la neurona i y la neurona j

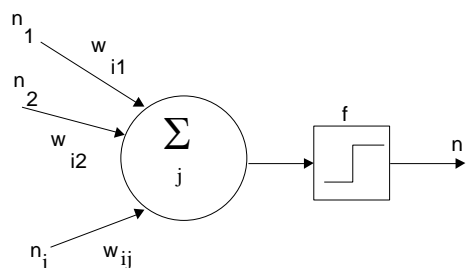
$n_j(t)$ es la salida producida por la neurona j

u_i es el umbral de la neurona j

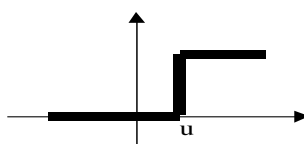
$f(x)$ = función umbral, 0 si $x < u_j$ de lo contrario 1.



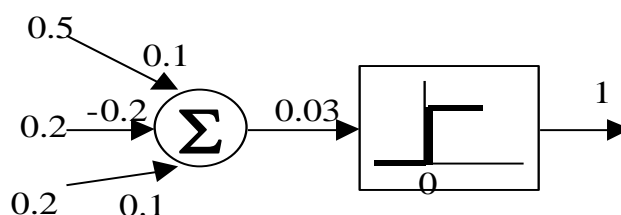
En la figura se presenta esquemáticamente una neurona.



La función de umbral tiene la forma de la figura



En la siguiente figura se puede apreciar el funcionamiento de una neurona.



3. Equipos y materiales

3.1. Equipos

| Ítem | Equipo | Característica | Cantidad |
|------|------------------------|---------------------|----------|
| 1 | Multímetro | Digital | 1 |
| 2 | Fuente de alimentación | Regulada de 0 a 30V | 1 |

3.2. Materiales

| Ítem | Material | Característica | Cantidad |
|------|-----------------------|---|----------|
| 1 | Protoboard | Estándar | 1 |
| 2 | Circuitos integrados | LM741(2), LM293(1) | 3 |
| 3 | Resistores | De carbón. Valores comerciales entre 10 Ω y 1 MΩ, ¼ W | 20 |
| 4 | Leds | Colores: rojo, verde, amarillo, azul | 4 |
| 5 | Cables con conectores | Colores: negro, rojo, anaranjado, amarillo, verde | 20 |



4. Indicaciones/instrucciones:

- 4.1. Los estudiantes forman grupos de hasta 4 integrantes y solicitan el equipo (3.1.)
- 4.2. Los estudiantes anotan las mediciones experimentales y comparan con los valores teóricos.
- 4.3. Cada grupo presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1. Problema de regresión lineal. Cinco bebés de 2, 3, 5, 7 y 8 meses, pesan, respectivamente, 1.4, 2.0, 3.2, 4.2 y 4.4 kilogramos. ¿Cuál será el peso aproximado de un bebé de seis meses?. Dibujar, en una hoja milimetrada, puntos iniciales y recta aproximada.

$$y = b_0 + b_1x, \text{ donde } b_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \quad b_0 = \bar{y} - b_1\bar{x}$$

5.2. Dibujar e implementar neurona básica con el amplificador operacional LM741 y el comparador LM293 correspondiente a 5.1.

6. Resultados

- 6.1.....
.....
.....
- 6.2.....
.....
.....
- 6.3.....
.....
.....

7. Conclusiones

- 7.1.....
.....
- 7.2.....
.....
- 7.3.....
.....

8. Sugerencias y/o recomendaciones

- 8.1.....
.....
- 8.2.....
.....
- 8.3.....
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega,
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>

Guía de práctica N° 5

Separabilidad lineal

Sección : ...AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos

- 1.1. Identificar problemas de clasificación linealmente separables
- 1.2. Implementar el perceptrón que clasifica dos grupos de objetos

2. Fundamento Teórico

Separabilidad lineal

En 1969 Minski y Papert mostraron que el perceptrón es capaz de resolver problemas de clasificación linealmente separables, es decir, el perceptrón es capaz de clasificar un grupo de patrones de otros si existe un conjunto de hiperplanos que definan las regiones clasificadoras.

Por ejemplo la función AND (figura a) se puede ver como un problema de clasificación donde si las dos entradas son uno (1) pertenece a una clase (1) y en otro caso pertenece a la otra clase (0). Es evidente que la función AND es linealmente separable, esto quiere decir que se puede encontrar un perceptrón. Por el contrario, la función XOR (figura b) no es linealmente separable por lo tanto no existe un perceptrón.

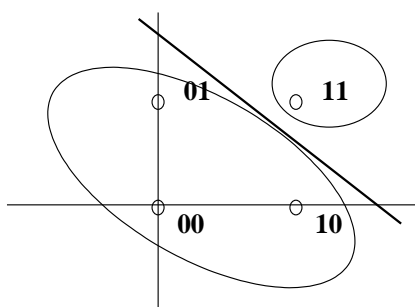


Figura a. Separabilidad de la función AND.

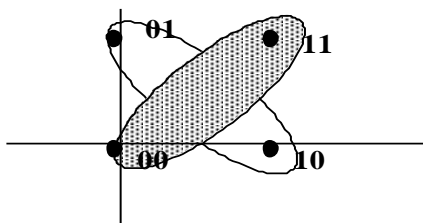


Figura b. No separabilidad de la función XOR.

3. Equipos y materiales

3.1. Equipos

| Ítem | Equipo | Característica | Cantidad |
|------|------------------------|---------------------|------------------|
| 1 | Multímetro | Digital | 1 |
| 2 | Fuente de alimentación | Regulada de 0 a 30V | 1 |
| 3 | Computadora | Programa PROTEUS | 1 por estudiante |

3.2. Materiales

| Ítem | Material | Característica | Cantidad |
|------|-----------------------|---|----------|
| 1 | Protoboard | Estándar | 1 |
| 2 | Circuitos integrados | LM741(2), LM293(1) | 3 |
| 3 | Resistores | De carbón. Valores comerciales entre 10 Ω y 1 M Ω , 1/4 W | 20 |
| 4 | Leds | Colores: rojo, verde, amarillo, azul | 4 |
| 5 | Cables con conectores | Colores: negro, rojo, anaranjado, amarillo, verde | 20 |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes forman grupos de hasta 4 integrantes y solicitan el equipo (3.1.)
- 4.2. Los estudiantes anotan las mediciones experimentales y comparan con los valores teóricos.
- 4.3. Cada grupo presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1. Separación lineal de 2 variables

Se desea construir un sistema automático que coloque las manzanas en un depósito verde y las piñas en depósito amarillo. Utilizamos un sensor para determinar el peso del fruto (p_1) y una cámara para determinar el color del fruto (p_2):

| | PIÑAS | | | MANZANAS | | |
|-----------|-------|-----|-----|----------|------|------|
| P1(peso) | 1.5 | 0.9 | 2.1 | 0.2 | 0.4 | 0.3 |
| P2(color) | -0.3 | 0.5 | 0.2 | -0.9 | -0.6 | -0.4 |

- a. Ubicar puntos en plano p_1p_2
- b. Dibujar recta de separación lineal y escribir ecuación correspondiente.
- c. Dibujar neurona en base a amplificador operacional y comparador
- d. Simular neurona en programa PROTEUS
- e. Implementar neurona en protoboard

5.2. Separación lineal de 4 variables

Un bibliotecario tiene una base de datos de 10000 libros. Necesita clasificarlos en 4 conjuntos mediante dos criterios, peso y frecuencia de uso del libro. Diseñar una red neuronal perceptrón para que automáticamente clasifique los libros. Considerar el siguiente patrón de prueba.

| | | | | |
|--|----------------|----------------|----------------|----------------|
| | LIBROS LIGEROS | LIBROS LIGEROS | LIBROS PESADOS | LIBROS PESADOS |
|--|----------------|----------------|----------------|----------------|



| | POCO USADOS | | MUY USADOS | | POCO USADOS | | MUY USADOS | |
|-----------------------|-------------|-----|------------|----|-------------|---|------------|-----|
| P1(peso) | 0.7 | 1.5 | 2 | 1 | 4 | 2 | 3.6 | 4.5 |
| P2(frecuencia) | 3 | 5 | 9 | 11 | 0 | 1 | 7 | 6 |

- a. Ubicar puntos en plano p₁p₂
- b. Dibujar rectas de separación lineal y escribir ecuaciones correspondientes.
- c. Dibujar neuronas en base a amplificador operacional y comparador
- d. Simular neuronas en programa PROTEUS

6. Resultados

6.1.
.....
.....
.....

6.2.
.....
.....
.....

6.3.
.....
.....
.....

7. Conclusiones

7.1.
.....

7.2.
.....

7.3.
.....

8. Sugerencias y /o recomendaciones

8.1.....
.....

8.2.....
.....

8.3.....
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega.
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Redes neuronales. [Consultado el 20 de enero del 2017]. Disponible en:
https://www.youtube.com/watch?v=14tU9B4ReII&list=PLIyIZGa1sAZo_eY8PpuTxfLsja_iyytSE

Guía de práctica N° 6

Perceptrón de una capa

Sección :AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

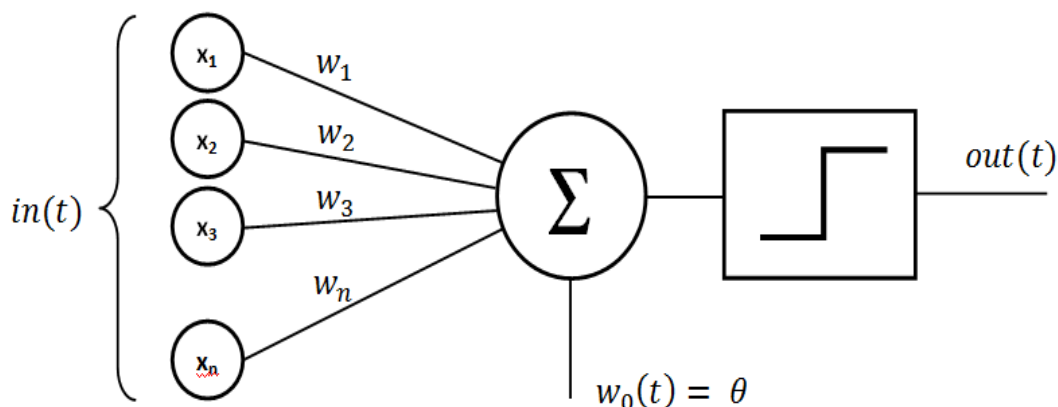
1. Objetivos:

- 1.1. Entrenar e implementar un perceptrón AND
- 1.2. Entrenar e implementar un perceptrón OR
- 1.3. Entrenar e implementar un perceptrón XOR

2. Fundamento Teórico

El perceptrón

Es una red neuronal feedforward que consta de dos capas, una de entrada y una de salida. En sus inicios se consideró que podía resolver todo tipo de problemas, sin embargo no es posible resolver el problema XOR con este modelo.



Regla de aprendizaje



- a. Inicialización de los pesos: Inicialmente se asignan valores aleatorios a cada uno de los pesos de w_i de las conexiones, coeficiente de aprendizaje(alfa), n(números de vueltas)
- b. Presentación de patrón de entrenamiento
- c. Número de vueltas
- d. Número de componentes de patrón de entrenamiento
- e. Cálculo de la salida actual $OUT=f(\sum_i (w_i * x_i))$, siendo f la función de transferencia escalón.
- f. Cálculo del error
- g. Actualización de w_i
- h. Repetir paso 4
- i. Repetir paso 3
- j. Finalizar

3. Equipos, Materiales

3.1. Equipos

| Ítem | Equipo | Característica | Cantidad |
|------|------------------------|------------------------|------------------------|
| 1 | Multímetro | Digital | 1 |
| 2 | Fuente de alimentación | Regulada de 0 a 30V | 1 |
| 3 | Computadora | Programa MATLAB/Octave | 1 por estudiante |

3.2. Materiales

| Ítem | Material | Característica | Cantidad |
|------|-----------------------|---|----------|
| 1 | Protoboard | Estándar | 1 |
| 2 | Circuitos integrados | LM741(2), LM293(1) | 3 |
| 3 | Resistores | De carbón. Valores comerciales entre 10 Ω y 1 M Ω , 1/4 W | 20 |
| 4 | Leds | Colores: rojo, verde, amarillo, azul | 4 |
| 5 | Cables con conectores | Colores: negro, rojo, anaranjado, amarillo, verde | 20 |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes forman grupos de hasta 4 integrantes y solicitan el equipo (3.1.)
- 4.2. Los estudiantes anotan las mediciones experimentales y comparan con los valores teóricos.
- 4.3. Cada grupo presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento

5.1. Escribir programa en MATLAB para entrenar compuerta AND y luego implementar con amplificadores operacionales y comparador.

```
x = [0 0 1; 0 1 1; 1 0 1; 1 1 1];
tand = [0 0 0 1];
w = [1 1 3];
alfa = 0.5;
for vueltas = 1:8
    for q = 1:4
        neta = x(q,:)*w';
        e = tand(q) - hardlim(neta);
        w = w + alfa*e*x(q,:);
    end
end
```

5.2. Escribir programa en MATLAB para entrenar compuerta OR y luego implementar con amplificadores operacionales y comparador.

5.3. Escribir programa en MATLAB para entrenar compuerta XOR y luego implementar con amplificadores operacionales y comparador.



6. Resultados

6.1.
.....
.....
.....

6.2.
.....
.....
.....

6.3.
.....
.....
.....

7. Conclusiones

7.1.
.....

7.2.
.....

7.3.
.....

8. Sugerencias y /o recomendaciones

8.1.
.....

8.2.
.....

8.3.
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega,
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Redes neuronales. [Consultado el 20 de enero del 2017]. Disponible en:
https://www.youtube.com/watch?v=14tU9B4ReII&list=PLIylZGa1sAZo_eY8PpuTxfLsja_ijytSE

Guía de práctica N° 7

Reconocimiento de números

Sección :AI1076.....Docente: *Escribir el nombre del docente*

Fecha:/...../..... Duración: 90 minutos

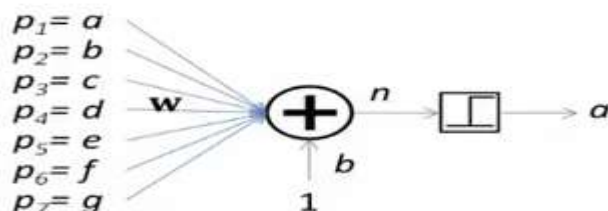
Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos:

- 1.1. Reconocimiento de números pares mediante un perceptrón
- 1.2. Reconocimiento de números mayores a 5 mediante un perceptrón
- 1.3. Reconocimiento de números primos mediante un perceptrón

2. Fundamento Teórico

- a. Dibujamos las entradas, función de activación y la salida de la neurona. Las entradas a, b, c, d, e, f, g son los segmentos del display cátodo común

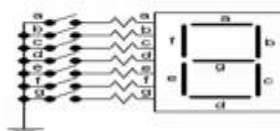


- b. Luego escribimos la tabla que expresa, con más detalle, el reconocimiento de números.

- a) Los números pares.
- b) Los números mayores a 5.
- c) Los números primos.

| # | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

| ta | tb | tc |
|----|----|----|
| 1 | 0 | 0 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 0 |



- c. Entrenamos la red utilizando programa MATLAB
- d. Implementamos la red con amplificadores operacionales y comparadores

3. Equipos y materiales

3.1. Equipos



| Ítem | Equipo | Característica | Cantidad |
|------|------------------------|------------------------|----------|
| 1 | Multímetro | Digital | 1 |
| 2 | Fuente de alimentación | Regulada de 0 a 30V | 1 |
| 3 | Computadora | Programa MATLAB/Octave | |

3.2. Materiales

| Ítem | Material | Característica | Cantidad |
|------|-----------------------|---|----------|
| 1 | Protoboard | Estándar | 1 |
| 2 | Circuitos integrados | LM741(2), LM293(1) | 3 |
| 3 | Resistores | De carbón. Valores comerciales entre 10 Ω y 1 M Ω , 1/4 W | 20 |
| 4 | Leds | Colores: rojo, verde, amarillo, azul | 4 |
| 5 | Cables con conectores | Colores: negro, rojo, anaranjado, amarillo, verde | 20 |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes forman grupos de hasta 4 integrantes y solicitan el equipo (3.1.)
- 4.2. Los estudiantes anotan las mediciones experimentales y comparan con los valores teóricos.
- 4.3. Cada grupo presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1. Escribir un programa en MATLAB para reconocimiento de números pares:

```
% abcdefg
p = [1 1 1 1 1 1 0; %0
     0 1 1 0 0 0 0; %1
     1 1 0 1 1 0 1; %2
     1 1 1 1 0 0 1; %3
     0 1 1 0 0 1 1; %4
     1 0 1 1 0 1 1; %5
     1 0 1 1 1 1 1; %6
     1 1 1 0 0 0 0; %7
     1 1 1 1 1 1 1; %8
     1 1 1 1 0 1 1]; %9

p = p';
tpar = [1 0 1 0 1 0 1 0 1 0];
w = [0.25 0.25 0.25 0.5 0.5 0.25 0.25];
b = 0.6;
for epocas = 1:2
    for q=1:10
        neto = w*p(:,q) + b
        salida = hardlim(neto);
        e(q) = tpar(q) - salida;
        w = w + e(q)*p(:,q)';
        b = b + e(q);
    end
end
```

5.2. Implementar red neuronal de reconocimiento de números pares

5.3. Escribir un programa en MATLAB para reconocimiento de números mayores a 5 y simular red neuronal en PROTEUS

5.4. Escribir un programa en MATLAB para reconocimiento de números primos y simular red neuronal en PROTEUS.



6. Resultados

6.1.....
.....
.....

6.2.....
.....
.....

6.3.....
.....
.....

7. Conclusiones

7.1.....
.....
.....

7.2.....
.....
.....

7.3.....
.....
.....

8. Sugerencias y /o recomendaciones

8.1.
.....

8.2.
.....

8.3.
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega,
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Redes neuronales. [Consultado el 20 de enero del 2017]. Disponible en: https://www.youtube.com/watch?v=14tU9B4Rell&list=PLIyIZGa1sAZo_eY8PpuTxfLsja_iyytSE

Guía de práctica N° 8

Función de pertenencia triangular

Sección :Al1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos

1.1. Identificar la función de pertenencia triangular

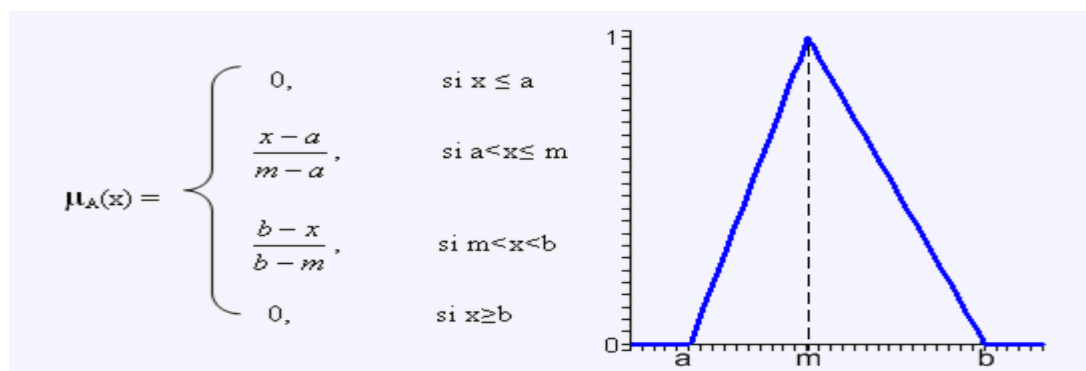
2. Fundamento Teórico

La función de pertenencia de un conjunto nos indica el grado en que cada elemento de un universo dado, pertenece a dicho conjunto. Es decir, la función de pertenencia de un conjunto A sobre un universo X será de la forma: $\mu_A: X \rightarrow [0,1]$, donde $\mu_A(x) = r$ si r es el grado en que x pertenece a A .

Las funciones de pertenencia son una forma de representar gráficamente un conjunto borroso sobre un universo. A la hora de determinar una función de pertenencia, normalmente se eligen funciones sencillas, para que los cálculos no sean complicados.

Función Triangular

Definida mediante el límite inferior a , el superior b y el valor modal m , tal que $a < m < b$.



3. Equipos, materiales



3.1. Equipos

| Item | Equipo | Característica | Cantidad |
|------|------------|------------------------|--------------------|
| 1 | Computador | Programa MATLAB/Octave | Uno por estudiante |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1. Escribir el código en MATLAB para una función de pertenencia triangular

```
%Gráfica triangular
clc, clear all;
a= 1;
b=5;
c=8;
for x=1:0.5:10
    if(x < a y(x) = 0);
        elseif and (a<=x, x<=b) y(x) = (x-a)/(b-a);
        elseif and (b<=x, x<=c) y(x) = (c-x)/(c - b);
        elseif(x > c) y(x) = 0;
    end
end
plot(x, y)
axis([-3 13 0 1])
```

5.2. Escribir el código simplificado en MATLAB para una función de pertenencia triangular

```
%Gráfica triangular
x=0:0.5:1;
y=trimf(x,[1 5 9]);
plot(x,y)
```

6. Resultados

6.1.....
.....
.....
.....

6.2.....
.....
.....
.....

6.3.....
.....
.....
.....



7. Conclusiones

- 7.1.....
.....
- 7.2.....
.....
- 7.3.....
.....

8. Sugerencias y /o recomendaciones

- 8.1.
.....
- 8.2.
.....
- 8.3.
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega,
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Lógica difusa. [Consultado el 20 de enero del 2017]. Disponible en:
https://www.youtube.com/watch?v=14tU9B4ReII&list=PLIylZGa1sAZo_eY8PpuTxfLsja_iyytSE



Guía de práctica N° 9

Función de pertenencia trapezoidal

Sección :AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

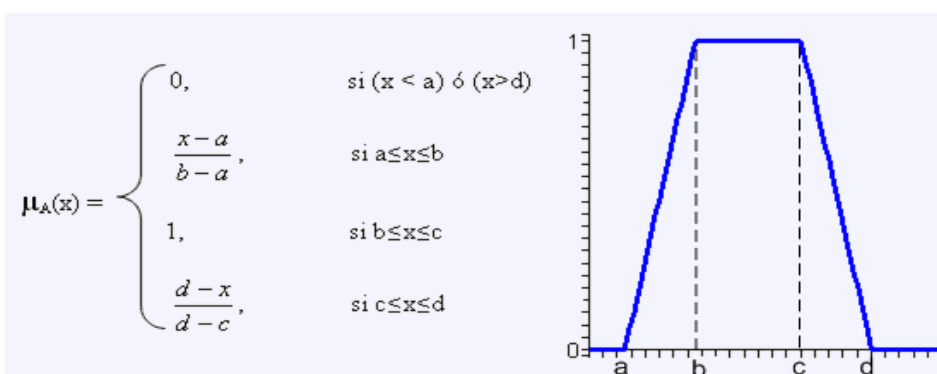
Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos

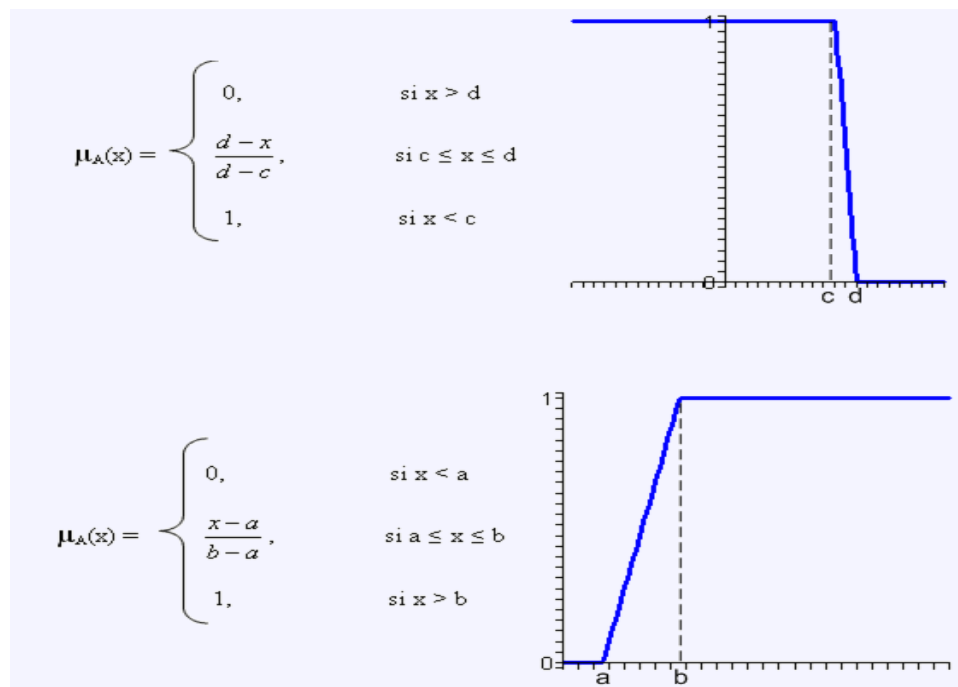
1.1. Identificar la función de pertenencia trapezoidal

2. Fundamento Teórico

Definida por sus límites inferior **a**, superior **d**, y los límites de soporte inferior **b** y superior **c**, tal que **a < b < c < d**. En este caso, si los valores de b y c son iguales, se obtiene una función triangular.



Casos especiales de estas funciones trapezoidales son aquellas en las que algunos parámetros toman valores no finitos:



3. Equipos

3.1. Equipos

| Item | Equipo | Característica | Cantidad |
|------|------------|------------------------|--------------------|
| 1 | Computador | Programa MATLAB/Octave | Uno por estudiante |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1. Escribir el código en MATLAB para la función de pertenencia trapezoidal

```

%Gráfica trapezoidal
clc, clear all;
a= 1;
b=5;
c=6;
d=8;
for x=0:0.5:10
    if(x < a) y(x) = 0;
        elseif and (a<=x, x<=b) y(x) = (x-a)/(b-a);
            elseif and (b<=x, x<=c) y(x) = 1;
                elseif and (c<=x, x<=d) y(x) = (d-x)/(d - c);
                    elseif(x > d) y(x) = 0;
end
end
plot(x,y)
axis([-3 13 0 1])

```



```
5.2. Escribir el código simplificado en MATLAB para una función de pertenencia
trapezoidal
%Gráfica trapezoidal
x=0:0.5:1;
y=trapmf(x,[1 2 5 8]);
plot(x,y)
```

6. Resultados

6.1.....
.....
.....

6.2.....
.....
.....

6.3.....
.....
.....

7. Conclusiones

7.1.....
.....

7.2.....
.....

7.3.....
.....

8. Sugerencias y /o recomendaciones

8.1.
.....

8.2.
.....

8.3.
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega,
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Lógica difusa. [Consultado el 20 de enero del 2017]. Disponible en: https://www.youtube.com/watch?v=14tU9B4ReII&list=PLlYlZGa1sAZo_eY8PpuTxfLsja_iyytSE

Guía de práctica N° 10

Funciones de pertenencia Gaussiana y sigmoidal

Sección : ...AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

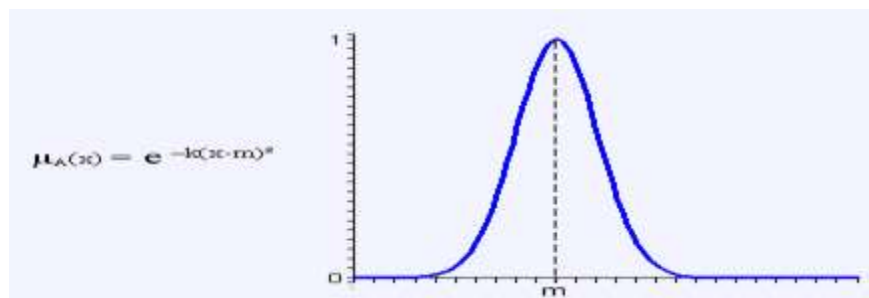
1. Objetivos:

- Identificar la función de pertenencia Gaussiana
- Identificar la función de pertenencia sigmoidal
- Identificar la función de pertenencia campana

2. Fundamento Teórico

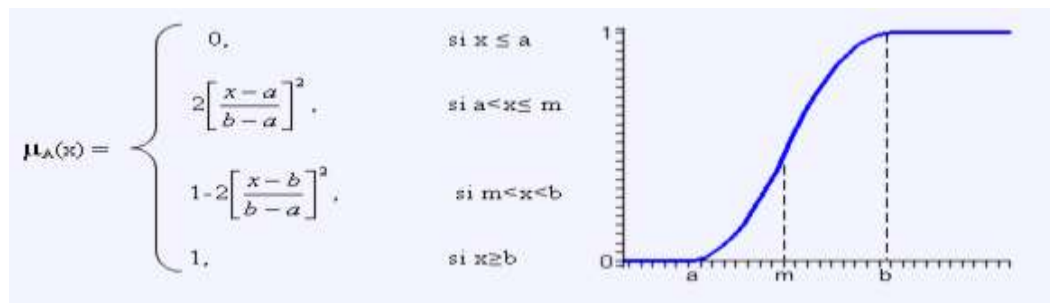
Función Gaussiana

Definida por su valor medio **m** y el parámetro **k>0**. Esta función es la típica campana de Gauss y cuanto mayor es el valor de **k**, más estrecha es dicha campana.



Función sigmoidal

Definida por sus límites inferior **a**, superior **b** y el valor **m** o punto de inflexión, tales que **a<m<b**. El crecimiento es más lento cuanto mayor sea la distancia **a-b**. Para el caso concreto de **m=(a+b)/2**, que es lo usual, se obtiene la siguiente gráfica



3. Equipos, materiales

3.1. Equipos



| Item | Equipo | Característica | Cantidad |
|------|------------|------------------------|----------------|
| 1 | Computador | Programa MATLAB/Octave | Uno por alumno |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1 Escribir programa en MATLAB

% función de pertenencia Gaussiana

```
x=-10:0.1:10;
m = 5;
mf1=gaussmf(x, [0.5,m]);
mf2=gaussmf(x, [1, m]);
mf1=gaussmf(x, [2, m]);
subplot(311);
plot(x,mf1);
subplot(312);
plot(x,mf2);
subplot(313);
plot(x,mf3);
```

5.2. Escribir programa en MATLAB

% función de pertenencia sigmoidal

```
x=0:0.1:12;
mf1=sigmf(x, [2, 4]);
mf2=sigmf(x, [2, 6]);
mf1=sigmf(x, [2, 8]);
subplot(311);
plot(x,mf1);
subplot(312);
plot(x,mf2);
subplot(313);
plot(x,mf3);
```

5.3. Escribir programa en MATLAB

% función de pertenencia campana

```
x=0:0.1:10;
%cambio de a
b=2;c=0;
mf1 = gbellmf(x,[2, b, c]);
mf2 = gbellmf(x,[4, b, c]);
mf3 = gbellmf(x,[6, b, c]);
subplot(311);
plot(x,mf1,x,mf2,x,mf3);
title(' (a) Cambio de a ');
axis([-inf inf 0 1.2]);
%cambio de b
a=5;c=0;
mf1 = gbellmf(x,[a, 1, c]);
mf2 = gbellmf(x,[a, 2, c]);
mf3 = gbellmf(x,[a, 4, c]);
subplot(312);
plot(x,mf1,x,mf2,x,mf3);
```



```
title(' (b) Cambio de b ');  
axis([-inf inf 0 1.2]);  
a=5;b=2;  
%cambio de b  
mf1 = gbellmf(x,[a, b, -5]);  
mf2 = gbellmf(x,[a, b, 0]);  
mf3 = gbellmf(x,[a, b, 5]);  
subplot(313);  
plot(x,mf1,x,mf2,x,mf3);  
title(' (c) Cambio de c ');  
axis([-inf inf 0 1.2]);
```

6. Resultados

- 6.1.
.....
.....
- 6.2.
.....
.....
- 6.3.
.....
.....

7. Conclusiones

- 7.1.....
.....
- 7.2.....
.....
- 7.3.....
.....

8. Sugerencias y/o recomendaciones

- 8.1.....
.....
- 8.2.....
.....
- 8.3.....
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega,
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Lógica difusa. [Consultado el 20 de enero del 2017]. Disponible en:
https://www.youtube.com/watch?v=14tU9B4Rell&list=PLlyZGa1sAZo_eY8PpuTxfLsja_iyytSE

Guía de práctica N° 11

Diseño de un controlador difuso (1)

Sección :Al1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

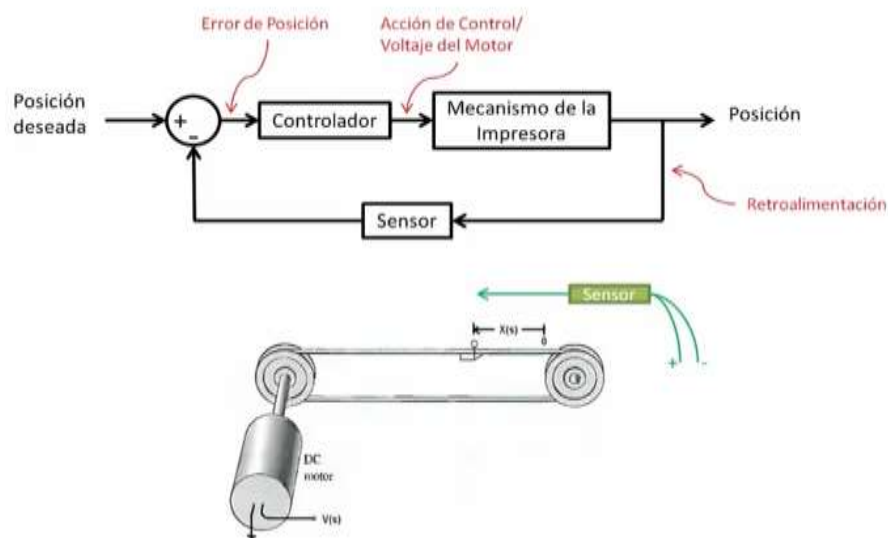
Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivos:

- Identificar los componentes de un sistema de control realimentado
- Identificar las entradas y salida del controlador difuso
- Escribir programa en MATLAB para las funciones de pertenencia de las variables de entrada y salida del controlador difuso

2. Fundamento Teórico

Para ejemplificar, consideremos un sistema de control realimentado que regule la posición del cartucho en una impresora de inyección de tinta.



Mediante el sensor ubicaremos la posición del cartucho. La comparamos con la posición deseada y generamos un error de posición, que ingresa al controlador difuso.

Controlador difuso



Pre-procesamiento: Acondiciona las señal antes de procesarla.

Fusificación: Convierte la magnitud de la señal de entrada en magnitud difusa, obteniendo el valor de pertenencia que tiene en cada uno de los valores lingüísticos.

Reglas de control: Conjunto de reglas lingüísticas SI-ENTONCES que utilizaremos para controlar el sistema

Defusificación: Proceso inverso a la fusificación. Convierte un conjunto difuso en una cantidad certera, producto de la inferencia.

Pos-procesamiento: Genera la señal de control para el controlador

3. Equipos, materiales

3.1. Equipos

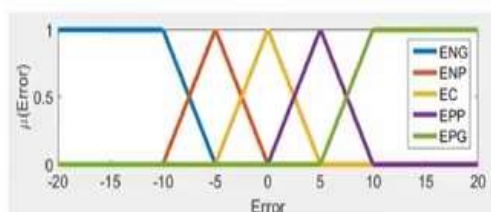
| Ítem | Equipo | Característica | Cantidad |
|------|------------|------------------------|----------------|
| 1 | Computador | Programa MATLAB/Octave | Uno por alumno |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1. Escribir un programa en MATLAB para definir las variables lingüísticas de la señal de entrada(Error) del controlador difuso:



$$M(ENG) = trapmf(e; -20, -20, -10, -5)$$

$$M(ENP) = trimf(e; -10, -5, 0)$$

$$M(EC) = trimf(e; -5, 0, 5)$$

$$M(EPP) = trimf(e; 0, 5, 10)$$

$$M(EPG) = trapmf(e; 5, 10, 20, 20)$$

Donde:

- ENG : Error negativo grande
- ENP : Error negativo pequeño
- EC : Error cero
- EPP : Error positivo pequeño
- EPG : Error positivo grande

```
% Error de posición
e = -20:0.01:20;
ENG = trapmf(e, [-20 -20 -10 -5]);
ENP = trimf(e, [-10 -5 0]);
EC = trimf(e, [-5 0 5]);
EPP = trimf(e, [0 5 10]);
```

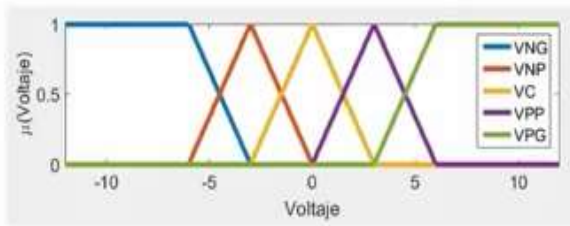


```

EPG = trapmf(e,[5 10 20 20]);
plot(e, ENG,e,ENP,e,EC,e,EPP,e,EPG,'LineWidth',5);
set(gca,'FontSize',18),legend('ENG','ENP','EC','EPP','EPG');
xlabel('Error'),ylabel('μ(error)');

```

5.2. Escribir un programa en MATLAB para definir las variables lingüísticas de la señal de salida (Voltaje) del controlador difuso:



```

M(VNG) = trapmf(e; -12, -12, -6, -3)
M(VNP) = trimf(e; -6, -3, 0)
M(VC) = trimf(e; -3, 0, 3)
M(VPP) = trimf(e; 0, 3, 6)
M(VPG) = trapmf(e; 3, 6, 12, 12)

```

Donde:

- VNG : Voltaje negativo grande
- VNP : Voltaje negativo pequeño
- VC : Voltaje cero
- VPP : Voltaje positivo pequeño
- VPG : Voltaje positivo grande

```

% Acción de control
v = -20:0.01:20;
VNG = trapmf(v,[-12 -12 -6 -3]);
VNP = trimf(v,[-6 -3 0]);
VC = trimf(v,[-3 0 3]);
VPP = trimf(v,[0 3 6]);
VPG = trapmf(v,[3 6 12 12]);
plot(v, VNG,v,VNP,v,VC,v,VPP,v,VPG,'LineWidth',5);
set(gca,'FontSize',18),legend('VNG','VNP','VC','VPP','VPG');
xlabel('Voltaje '),ylabel('μ(voltaje)');

```

6. Resultados

- 6.1.
- 6.2.
- 6.3.

7. Conclusiones

- 7.1.
- 7.2.
- 7.3.



8. Sugerencias y /o recomendaciones

8.1.....
.....

8.2.....
.....

8.3.....
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega.
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Lógica difusa. [Consultado el 20 de enero del 2017]. Disponible en:
https://www.youtube.com/watch?v=14tU9B4ReII&list=PLlylZGa1sAZo_eY8PpuTxfLsja_iyytSE

Guía de práctica N° 12

Diseño de un controlador difuso (2)

Sección : ...AI1076.....Docente: Escribir el nombre del docente

Fecha:/...../.....

Duración: 90 minutos

Instrucciones: El estudiante debe llegar antes del inicio de la práctica de laboratorio, con mandil blanco y materiales solicitados.

1. Objetivo:

- Programar fusificación e inferencia de Mamdani de controlador difuso
- Programar defusificación de controlador difuso

2. Fundamento Teórico

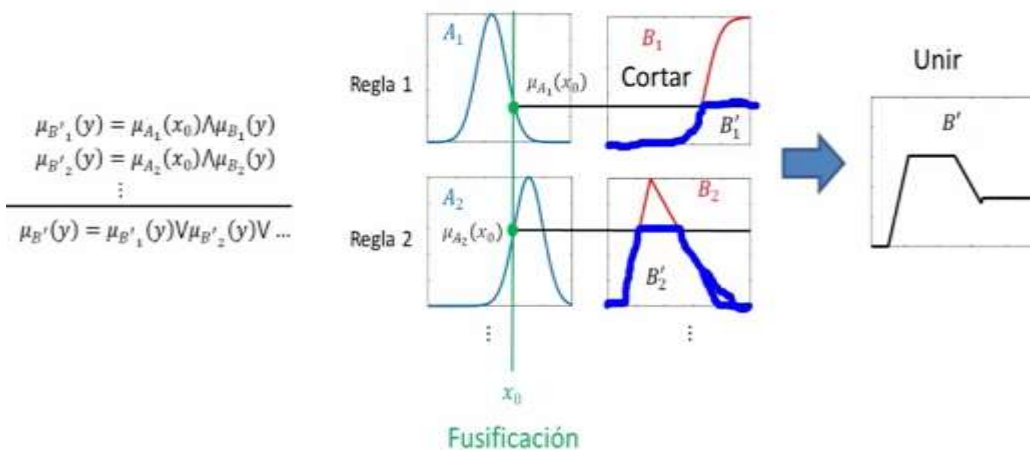
Reglas de control

Si el error es negativo grande, entonces el voltaje es negativo grande.
 Si el error es negativo pequeño, entonces el voltaje es negativo pequeño.
 Si el error es cero, entonces el voltaje es cero.
 Si el error es positivo pequeño, entonces el voltaje es positivo pequeño.
 Si el error es positivo grande, entonces el voltaje es positivo grande.

| ERROR | VOLTAJE |
|-------|---------|
| ENG | VNG |
| ENP | VNP |
| EC | VC |
| EPP | VPP |
| EPG | VPG |

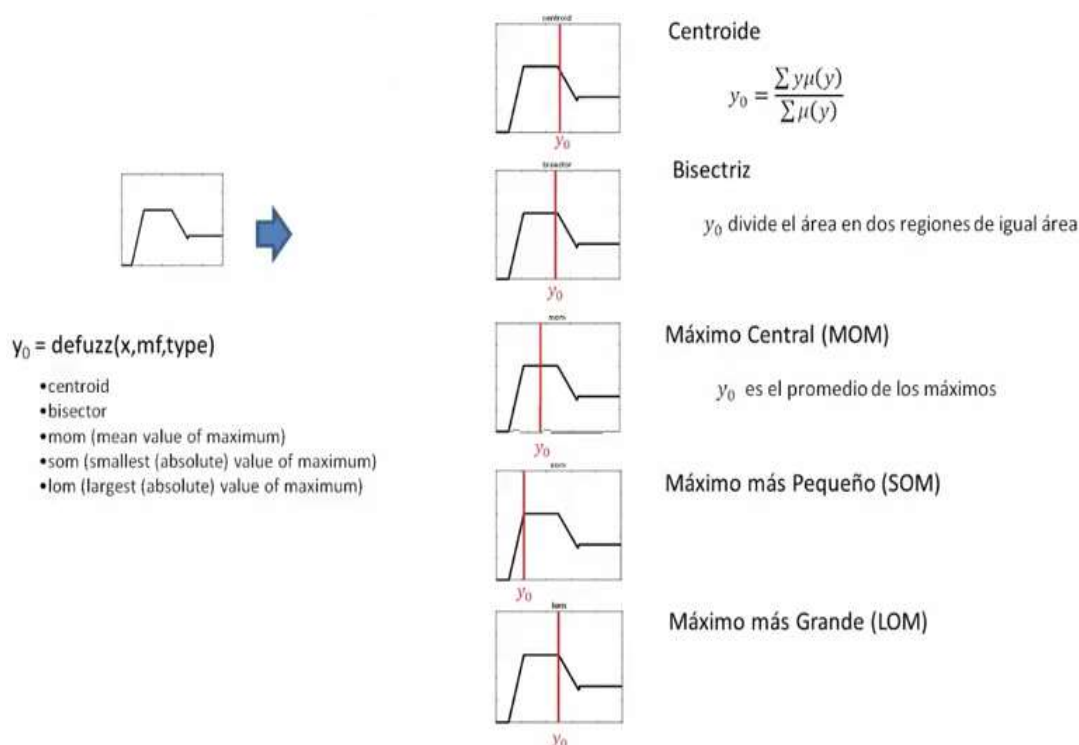
Método de inferencia

Consta de 3 pasos, mostrado en el gráfico: fusificar, cortar y unir



Defusificación

El conjunto difuso, producto de la inferencia, se expresa en una cantidad. En la figura mostramos 5 métodos.



3. Equipos, materiales

a. Equipos

| Ítem | Equipo | Característica | Cantidad |
|------|------------|------------------------|----------------|
| 1 | Computador | Programa MATLAB/Octave | Uno por alumno |

4. Indicaciones/instrucciones:

- 4.1. Los estudiantes trabajan individualmente.
- 4.2. Cada estudiante presentará un informe de la práctica de laboratorio en la próxima sesión

5. Procedimiento:

5.1. Escribir código en MATLAB para ubicar el error = 9 en función de pertenencia de entrada.

```
%Error leído, por ejemplo e0=9
e0 = 9;
n == find (e==e0);
subplot(3,1,1), hold on;
plot(e0,ENG(n),'*', e0,ENP(n),'*', e0,EC(n),'*', e0,EPP(n),'*', e0,EPG(n),'*',...
,'LineWidth',5),hold off;
```

5.2. Escribir código en MATLAB para fusificación e inferencia de Mamdani

```
% Fusificación e inferencia de Mandani
B1 = min(VNG, ENG(n));
B2 = min(VNP, ENP(n));
B3 = min(VC, EC(n));
B4 = min(VPP, EPP(n));
B5 = min(VPG, EPG(n));
B = max(B1, max(B2, max(B3, max(B4,B5)))));
subplot(3,1,2);
```



```
plot(v,B);  
set(gca, 'FontSize',18),legend('V'), axis([-12 12 0 1]);
```

5.3 Escribir código en MATLAB para defusificación.

```
y0 = defuzz(v,B,'centroid')  
hold on, plot(y0*ones(1,3),[0 0.5 1], 'r', 'LineWidth',5)
```

5.4 Utilizar la herramienta de MATLAB fuzzylogicdesigner para diseñar el controlador difuso.

6. Resultados

6.1.....
.....
.....

6.2.....
.....
.....

6.3.....
.....
.....

7. Conclusiones

7.1.....
.....
.....

7.2.....
.....
.....

7.3.....
.....
.....

8. Sugerencias y/o recomendaciones

8.1.....
.....

8.2.....
.....

8.3.....
.....

Referencias bibliográficas consultadas y/o enlaces recomendados

- Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Editorial Alfaomega,
- <http://www.ieee.org/Instituto de Ingenieros electricistas y electrónicos>
- Lógica difusa. [Consultado el 20 de enero del 2017]. Disponible en: https://www.youtube.com/watch?v=14tU9B4ReII&list=PLlYlZGa1sAZo_eY8PpuTxfLsja_iyytSE