



**LA COPIA TOTAL O PARCIAL DE CUALQUIERA DE LOS PROBLEMAS INVALIDA AL TP DE ORIGEN Y AL TP DE DESTINO, QUEDANDO AMBOS DESAPROBADOS.**

## PROBLEMAS A RESOLVER

### 1. Funciones de pertenencia:

a) Utilizando la interfaz gráfica de los sistemas de inferencia, definir por separado las variables que se indican:

- 1° variable (V1): alcance (-10,+10). Tres particiones lineales hombro-triángulo-hombro con parámetros (-10, -8, -3), (-6, 1, 5), (2.5, 7, 10) respectivamente.
- 2° variable (V2): alcance (-20, 20). Cinco particiones tipo triángulo rectángulo izquierdo-trapecio-triángulo-trapecio-triángulo rectángulo derecho, igualmente espaciados con solapamiento del 30%.
- 3° variable (V3): alcance (0, 10). Cinco particiones con funciones continuas y derivables Z shape ( $zmf()$ ) - gaussiana - campana generalizada - gaussiana - S shape ( $smf()$ ). Parámetros a definir por el usuario para una distribución simétrica y un solapamiento aproximado al 25%.

b) Un sistema de lógica fuzzy debe reemplazar dos funciones lineales por tramos, con funciones continuas y derivables, de manera que el error cuadrático medio sea menor a  $10^{-2}$ . Calcular los parámetros en cada caso:

- a. función escalón (heaviside): alcance (0, 10), con flanco decreciente en 4. Reemplazar con función hombro derecho (z-shape) o función sigmoide según convenga.
- b. función trapecio (trapmf): alcance (-5, 5), con parámetros (-3, 0, 1, 4). Reemplazar con función pi o diferencia de sigmoides según convenga.

En Matlab el error cuadrático medio se calcula con la función `mse()`. ECM = Error Cuadrático Medio (en español); MSE = Mean Square Error (en inglés).

Investigar la posible utilización de la aplicación gráfica `cftool`.

### 2. Definir y escribir los códigos de Matlab que realicen las siguientes funciones. Graficar.

a) Función concentración	$f_1 = \text{concen}(x, \mu_0)$	$f_1 = \mu_0^2$
b) Función dilación	$f_2 = \text{dilac}(x, \mu_0)$	$f_2 = \mu_0^{1/2}$
c) Función intensificación	$f_3 = \text{intens}(x, \mu_0)$	$f_3 = \begin{cases} 2 * i_0 & \text{si } i_0 \in [0 ; 0.5] \\ 1 - 2 * (1 - i_0)^2 & \text{si } i_0 \in [0.5 ; 1] \end{cases}$
d) Función difuminación	$f_4 = \text{difum}(x, \mu_0)$	$f_3 = \begin{cases} \left(\frac{i_0}{2}\right)^{1/2} & \text{si } i_0 \in [0 ; 0.5] \\ 1 - \left(\frac{1 - i_0}{2}\right)^{1/2} & \text{si } i_0 \in [0.5 ; 1] \end{cases}$

$x$  = secuencia de datos de entrada en el eje de abscisas;  
 $\mu_0$  = valores de pertenencia correspondiente a los datos  $x$ .

En todos los casos implementar el *control de datos* para cada función, como por ejemplo que los argumentos sean vectores fila de igual longitud, que la función de pertenencia de entrada esté en el alcance correcto, etc. Utilizando las funciones anteriores como base, escribir por lo menos tres funciones representativas de adverbios fuzzy, darles valores en el intervalo  $[0 ; 1]$ , fuzzyficar y graficar junto a la función de base.

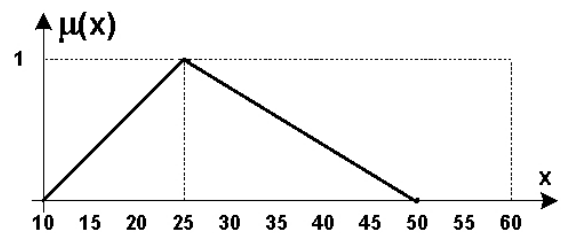
**3. Definir según un criterio coherente, las variables lingüísticas *movimiento* (particiones “lento”, “moderado”, “rápido”); y *temperamento* (particiones “calmado”, “insensible” “furioso”). Seleccionar las funciones de pertenencia para cada partición, en un rango  $[0 ; 10]$ . Luego, mostrar cómo se componen las funciones de pertenencia cuando se aplican los operadores lógicos que se indican a continuación:**

- Rápido y furioso.**
- Muy lento y algo calmado.**
- No rápido XOR insensible** (en este caso conviene descomponer la función XOR en los operadores básicos).

*Comentar cómo resultan las funciones de pertenencia compuestas (comprimidas, expandidas, sesgadas, deformadas, etc).*

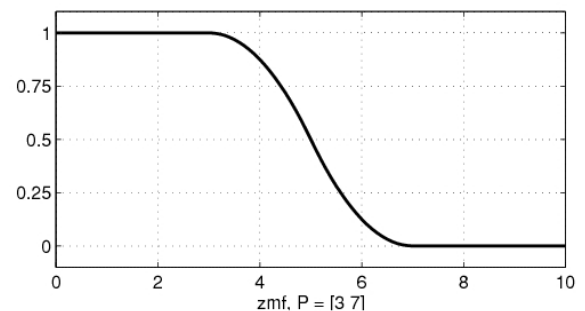
**4. Dibujar las modificaciones que sufren las funciones de pertenencia dadas, al aplicarle sucesivamente los adverbios fuzzy que se indican:**

- Función de base:  $\mu(x) \rightarrow$  Periferia  
Modificación 1:  $\mu_1(x) \rightarrow$  cercano a (Periferia)  
Modificación 2:  $\mu_2(x) \rightarrow$  ligeramente cercano en (Periferia)



- Función de base:  $\mu(x) \rightarrow$  Cerca  
Modificación 1:  $\mu_1(x) \rightarrow$  Mas o menos(Cerca)  
Modificación 2:  $\mu_2(x) \rightarrow$  Mas(Menos(Cerca))  
*Comentar los resultados, comparando ambas modificaciones.*

Ref.  $\mu(x) = \text{zmf}(x, [3, 7])$  con  $x = 0:0.1:10$



Calcular  $\mu(x)$  de las figuras, utilizando por lo menos 100 puntos. *Comentar si las heurísticas aplicadas producen modificaciones coherentes, analizando algunos puntos de referencia (por lo menos dos).*

**5. Función para cálculo de adverbios múltiples.** Escribir una función de Matlab que aplique sucesivamente hasta tres adverbios fuzzy a una secuencia de pertenencia base  $\mu_0$ , con las siguientes condiciones:

- La función debe recibir como entrada una secuencia  $[x_0, \mu_0]$ .
- Debe incorporar adicionalmente una, dos o tres etiquetas que indiquen el tipo de adverbio a aplicar. En la función se debe reconocer la cantidad de adverbios a aplicar (comando

**nargin**). Las etiquetas pueden ser por ejemplo ‘no’, ‘muy’, ‘algo’, ‘aproximadamente’, etc.

- c) Incorporar controles para asegurar que  $x_0$  y  $\mu_0$  sean de la misma longitud y que los valores de  $\mu_0$  estén en el intervalo  $[0 ; 1]$ .
- d) Incorporar un mínimo de cinco adverbios reconocibles por la función (se pueden utilizar los propuestos en clase). Puede utilizar internamente las funciones definidas en el punto anterior como funciones auxiliares (deben estar en la misma carpeta).
- e) Como argumentos de salida, la función debe generar la secuencia de base  $x_0$  (la misma que ingresa), la secuencia de pertenencia  $\mu_0$  (la misma que ingresa) y la secuencia de pertenencia modificada -muadv- por la aplicación sucesiva de los adverbios indicados en el argumento de la función. Por ejemplo:

```
[x0,mu0,muadv] = adv_multiples(x0,mu0,'no','muy','aproximadamente')
```

- f) La función debe generar además una gráfica que muestre el conjunto de base y los conjuntos modificados con diferentes colores, sobre el mismo sistema de ejes.

**6. Función de transferencia fuzzy.** Para cada uno de los conjuntos fuzzy de entrada ( $\mu_1, \mu_2, \mu_3$ ), y considerando que el conjunto fuzzy de la derecha ( $\mu_0$ ) es de salida, y con la asistencia de la GUI *fuzzy*, mostrar como resultan las funciones de transferencia. Considerar implicación de Mamdani, defuzzyficación por centroide y el trazado de la curva de inferencia. Se debe generar una base de reglas que relacione entrada y salida. Por ejemplo:

Regla i --> **Si**  $\mu_{ENTRADA}$  es  $XX_i$  **Entonces**  $\mu_{SALIDA}$  es  $XX_i$   
 $XX_1 = \text{“NG”}; XX_2 = \text{“NP”}; \dots ; XX_5 = \text{“PG”};$   
 (se correspondería con un controlador SISO tipo proporcional)

Comentar los resultados.

