Chapitre VI: Introduction a Logique Floue

1. Introduction

Qu'est ce que la logique floue ?

- raisonnement humain basé sur des données imprécises ou incomplètes.
- ordinateur basé sur des données exactes.

Historique

- 1965, naissance du concept flou avec le Pr. Zedeh Lofti (Californie)
  - « Un contrôleur électromécanique doté d'un raisonnement humain serait plus performant qu'un contrôleur classique. »
  - théorie des « sous-ensembles flous ».
- 1973, Zadeh introduit la notion de variables linguistiques.
- 1974, Mamdani (Londres) réalise un contrôleur flou pour moteur à vapeur.
- 1987, explosion du flou au Japon et qui atteint son apogée en 1990.
- Aujourd'hui, une vaste gamme de nouveaux produits ont une étiquette « produit flou » (Fuzzy).

Applications

Principaux domaines : Automatisation, Instrumentation,
Conception/Jugement, Traitement de l'information, …
Champions : Japonais (80% du marché mondial).

Exemple :

- Metro de Sundae 1987 et celui de Tokyo au Japon
- Machine a laver Intelligente
- Photocopieur Intelligent
- etc…

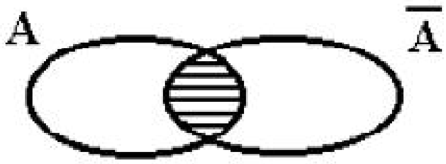
Pourquoi au Japon ?

En occident, on à un esprit fort cartésien et le concept du flou viole un des fondements du raisonnement classique : Principe du tiersexclus d'Aristote. (Syllogisme)

Prémisse 1 : Socrate est un Homme
Prémisse 2 : Tous les Hommes sont mortels
Conclusion : Socrate est mortel

Dans la théorie classique des ensembles, un objet appartient ou n'appartient pas à un ensemble.

$$A \cup \overline{A} = U$$
$$A \cap \overline{A} = \Phi$$



Exemple : U = ensemble des individus
A = ensemble des gens petits

La théorie classique ne peut résoudre certains paradoxes comme celui d'Epiménide.

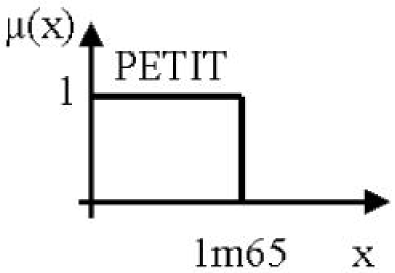
Alors qu'en logique floue, un objet peut appartenir à un ensemble et en même temps à son complément.

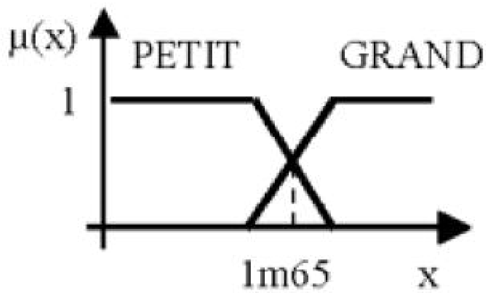
2. Théorie des sous-ensembles flous

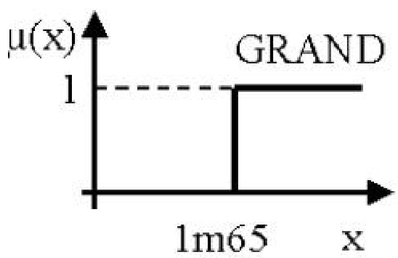
2.1.Définitions

Soit une variable x (la taille) et un univers de référence ou de discourt U (les individus).
Un sous-ensemble flou A est défini par une fonction d'appartenance $\mu_A(x)$ qui décrit le degré avec lequel l'élément x appartient à A.

Théorie Classique	Théorie Floue
$\mu(x) = \begin{cases} 1 & si\ x \in A \\ 0 & \text{sinon} \end{cases}$	$\mu(x) \in [0,1]$



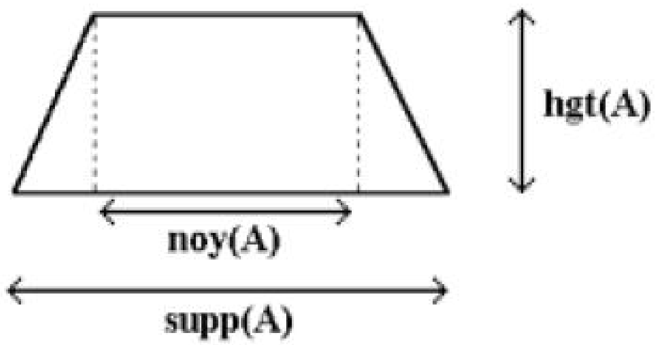




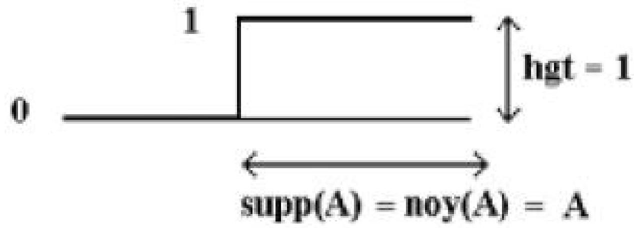
On parle de variable linguistique taille et de valeurs linguistiques petit et grand.

2.2. Autres Définitions :

Soit une fonction d'appartenance $\mu_A(x)$ de forme trapézoïdale, on peut définir sa hauteur, son support et son noyau comme suit :



Un sous-ensemble classique est un sous-ensemble flou particulier avec:



ATTENTION : $\mu_A(x) \neq P\ (x \in A)$

La théorie des probabilités est différente de la théorie de la logique floue bien que toutes deux décrivent une notion de doute, d'incertain et ce à l'aide de nombres compris entre 0 et 1.

Il existe une différence essentielle :

Probabilités : $P(A \cap \bar{A}) = P(\Phi) = 0$ et $P(A \cup \bar{A}) = P(U) = 1$
Logique Floue : $P(A \cap \bar{A})$ pas nécessairement $= P(\Phi)$
et $P(A \cup \bar{A})$ pas nécessairement $= P(U)$

Violation du Principe du tiers-exclus.

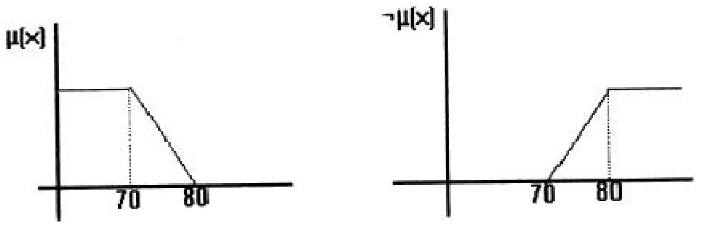
2.3. Opérateurs en Logique Floue

Généralisation des opérateurs négation, intersection et union de la théorie classique des ensembles.

L'opérateur NON (complément)

→ $\overline{A} = \{x \mid x \notin A\}$

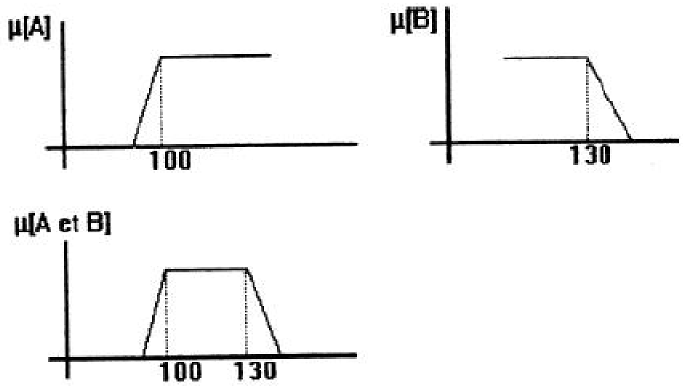
→ représenté par la fonction $\boxed{non(\mu_A(x)) = \mu_{\overline{A}}(x) = 1 - \mu_A(x)}$



L'opérateur ET (intersection)

→ $A \cap B = \{x \mid x \in A \wedge x \in B\}$

→ représenté par la fonction $\boxed{\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x))}$



Appellation	ET	OU	NON
Zadeh	min(x,y)	max(x,y)	1-x
Probalistique	xy	x+y-xy	1-x
Lukasiewicz	max(x+y-1,0)	min(x+y,1)	1-x
Hanacher(ß>0)	xy/(ß+(1-ß)(x+y-xy))	(x+y+xy-(1-ß)xy) /(1-(1-ß)xy)	1-x
Weber	x si y=1,y si x=1 0 sinon	x si y=0,y si x=0 1 sinon	1-x

Implication Floue

Pour représenter l'implication « *Si* $x \in A$, *alors y B* » il existe encore une fois plusieurs fonctions d'appartenances (voir tableau dans les notes du cours). Celle qui est le plus utilisée est la fonction

$$\boxed{\mu_{A/B}(x,y) = \min(\mu_A(x), \mu_B(y))}$$

2.4. Déffuzification :

Le moteur d'inférence fournit une fonction d'appartenance résultante µ**RES(xR)** pour la variable de sortie **xR**.
Il s'agit donc d'une information floue.
Etant donné que l'organe de commande nécessite **un signal de commande ucm précis à son entrée**, il faut prévoir une transformation de cette information floue en une information déterminée. Cette information est appelée **défuzzification**.

La **défuzzification** ou concrétisation, consiste donc à combiner ces coefficients avec les sous-ensembles de sortie, pour les convertir en un ou plusieurs signaux de commande. C'est l'opération inverse de la pondération.

Il existe plusieurs méthodes pour calculer la valeur représentative d'un ensemble de sortie, dont les principales sont : Défuzzifications basées sur le centre de gravité des ensembles. Soit par les méthodes des min / max ou somme / produit.

La méthode de **défuzzification la plus utilisée** est celle de la détermination du centre de gravité de la fonction d'appartenance résultante µ**RES(xR)**. Dans ce contexte il suffit de calculer l'abscisse **x*R**. L'abscisse du centre de gravité peut être déterminée à l'aide de la relation générale suivante :

$$CdG = X^*_R = \frac{\int_{-1}^{1} x_R \mu_{RES}(x_R) dx_R}{\int_{-1}^{1} \mu_{RES}(x_R) dx_R}$$

Remarque :
L'intégrale du dénominateur donne la surface, tandis que l'intégrale du numérateur correspond au moment de la surface.

Calcul du centre de gravité lors de la méthode d'inférence somme / produit.

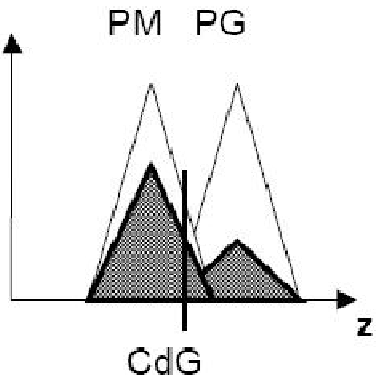
Cette méthode est la plus utilisée car son temps de calcul est court. Principe :

Les coefficients issus du moteur d'inférence sont utilisés pour multiplier les fonctions d'appartenance des sous-ensembles de sortie. La valeur de la sortie correspondra au centre de gravité de tous ces ensembles pris individuellement.

Remarque :
La position du centre de gravité de chaque sous-ensemble n'a pas été modifiée par le produit, d'où l'avantage d'un calcul simple du centre de gravité global.

Le calcul du centre de gravité peut être ramené au calcul suivant:

$$x^*_R = \frac{\sum_{i=1}^{m} \mu_{Ci} x^*_i S_i}{\sum \mu_{Ci} S_i}$$



μ_{Ci} : coeff de modification à appliquer au sous-ensemble de sortie i.
Si : surface du sous-ensemble i
X*i : centre de gravité du sous-ensemble de sortie i.

2.5. Exemples en logique classique

Soit un ensemble de référence U = {a,b,c,d,e,f,g} .
Soit A et B deux sous-ensembles de U :

	a	b	c	d	e	f	g
A	0	1	0	1	1	0	1
B	1	1	0	0	1	0	1

	a	b	c	d	e	f	g
$\overline{A}$	1	0	1	0	0	1	0

	a	b	c	d	e	f	g
$A \cap B$	0	1	0	0	1	0	1

	a	b	c	d	e	f	g
$A \cup B$	1	1	0	1	1	0	1

$A \cap \overline{A}$	0	0	0	0	0	0	0
	a	b	c	d	e	f	g

→ $A \cap \overline{A} = \Phi$

$A \cup \overline{A}$	1	1	1	1	1	1	1
	a	b	c	d	e	f	g

→ $A \cup \overline{A} = U$

2.6. Exemples en logique floue

Soit un ensemble de référence U = {a,b,c,d,e,f,g} .
Soit C et D deux sous-ensembles de U :

C	0.4	0.8	1	0.8	0.2	0.5	0.1
D	0	0.5	0.3	0.9	1	0.7	0
	a	b	c	d	e	f	g

$\overline{C}$	0.6	0.2	0	0.2	0.8	0.5	0.9
	a	b	c	d	e	f	g

$C \cap D$	0	0.5	0.3	0.8	0.2	0.5	0
	a	b	c	d	e	f	g

$C \cup D$	0.4	0.8	1	0.9	1	0.7	0.1
	a	b	c	d	e	f	g

$C \cap \overline{C}$	0.4	0.2	1	0.2	0.2	0.5	0.1
	a	b	c	d	e	f	g

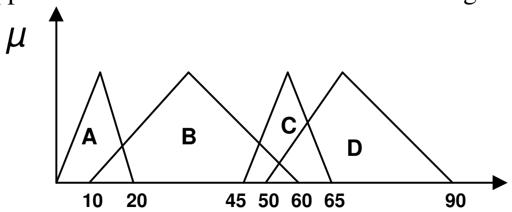
→ $C \cap C \neq \Phi$

$C \cup \overline{C}$	0.6	0.8	0	0.8	0.8	0.5	0.9
	a	b	c	d	c	f	g

→ $C \cup \overline{C} \neq U$

2.7. Exemple Floue 2 :

Supposons les sous ensembles floues de la Figure ci-dessous.



$\mu_A(x_1) = 0.67 \quad \mu_B(y_1) = 0.22 \quad \mu_D(y_1) = 0.33$

Supposons:

$$\mu_A(x_1) = 0.67 \quad \mu_B(y_1) = 0.22 \quad \mu_D(y_1) = 0.33$$

Trouver par la méthode SOMME-PRODUIT le résultat donné par x_1, x_2 dans le système logique suivant.

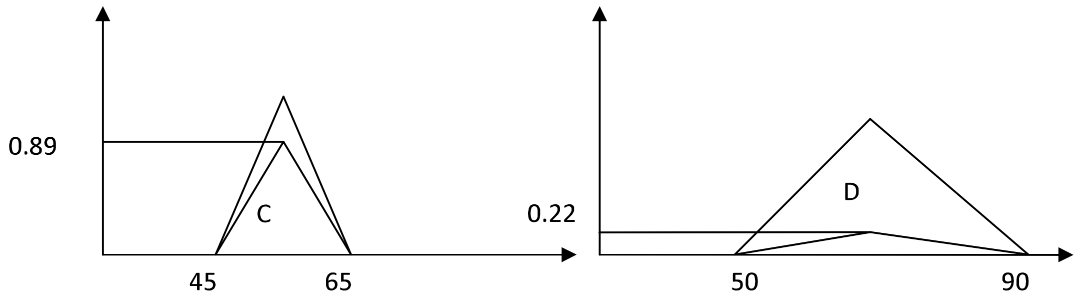
$$Si(x \in A, et, y \in D) \to x_{r2} \in D$$
$$ou$$
$$Si(x \in A, ou, y \in B) \to x_{r1} \in C$$

Si les surfaces de C et D sont 2 et 3, et leur centres de gravités x_C*=50, x_D=*70 respectivement. Deffuziffier le résultat obtenue dans 2 par la méthode du centre de gravitée.

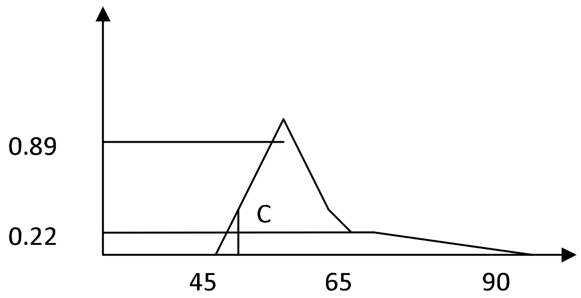
Solution :

Somme/produit: ET=Somme, Ou=Produit, →=Produit.

$Si(x \in A, ou, y \in B) \to x_{r1} \in C \qquad ou \qquad Si(x \in A, et, y \in D) \to x_{r2} \in D$



Le résultat final est donne par la somme des deux sous ensemble modifiés.



Deffuzification :

$$CdG = X^* = \frac{\sum_{i=1}^{m} \mu_{Ci} x^*_i \; S_i}{\sum \mu_{Ci} S_i} = \frac{0.89 * 2 * 50 + 0.221 * 3 * 70}{0.89 * 2 + 0.221 * 3} = 55.4278$$

3. Application avec *Matlab*

3.1. Principe

Le calcul en logique floue nécessite, outre la licence de base *Matlab*, l'achat et l'installation du *Fuzzy Control Toolbox*. Celui-ci met à disposition de l'utilisateur deux outils fondamentaux:

- Un guide de construction du régulateur par logique floue.
- Un bloc fonctionnel à intégrer dans un schéma de régulation *Simulink*.

3.2. Conception du régulateur
3.3.

On accède à la fenêtre principale du «FIS Editor» par un instruction sur la ligne de commande *Matlab*:

» fuzzy

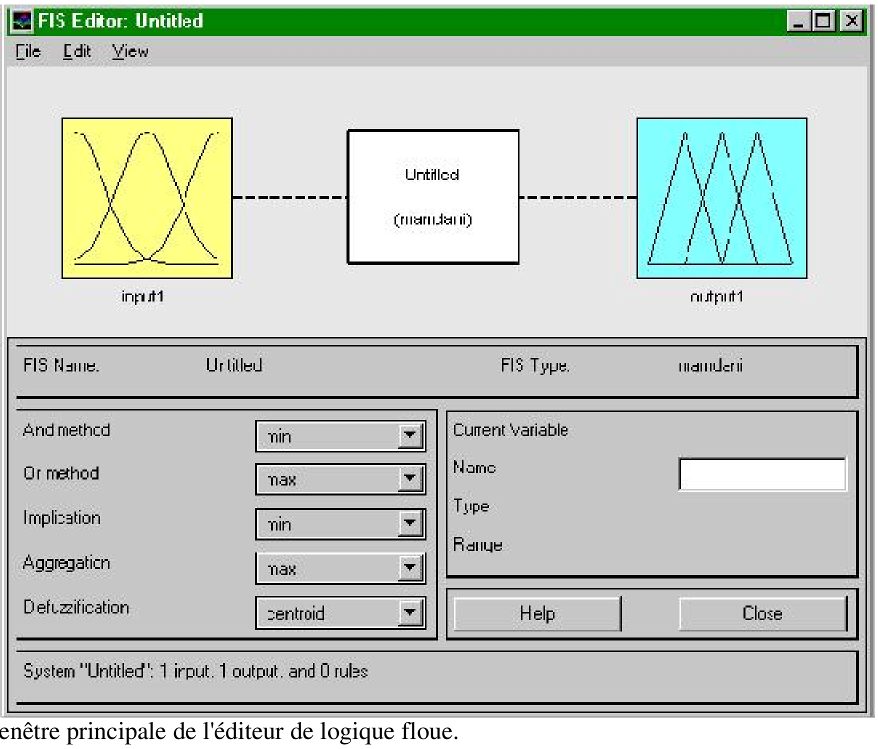


Fig. 4.3.1 : Fenêtre principale de l'éditeur de logique floue.

On y choisit premièrement le nombre d'entrées depuis la barre de menu: Edit —> Add input. On trouve alors sur la fenêtre principale du «FIS Editor» autant d'icônes qu'on veut de signaux d'entrées et sorties, et une pour les règles d'inférence. La partie inférieure permet de spécifier les méthodes d'inférences. On accède alors à une fenêtre spécifique par un doubleclic sur l'icône du signal dont on veut spécifier les fonctions d'appartenance.

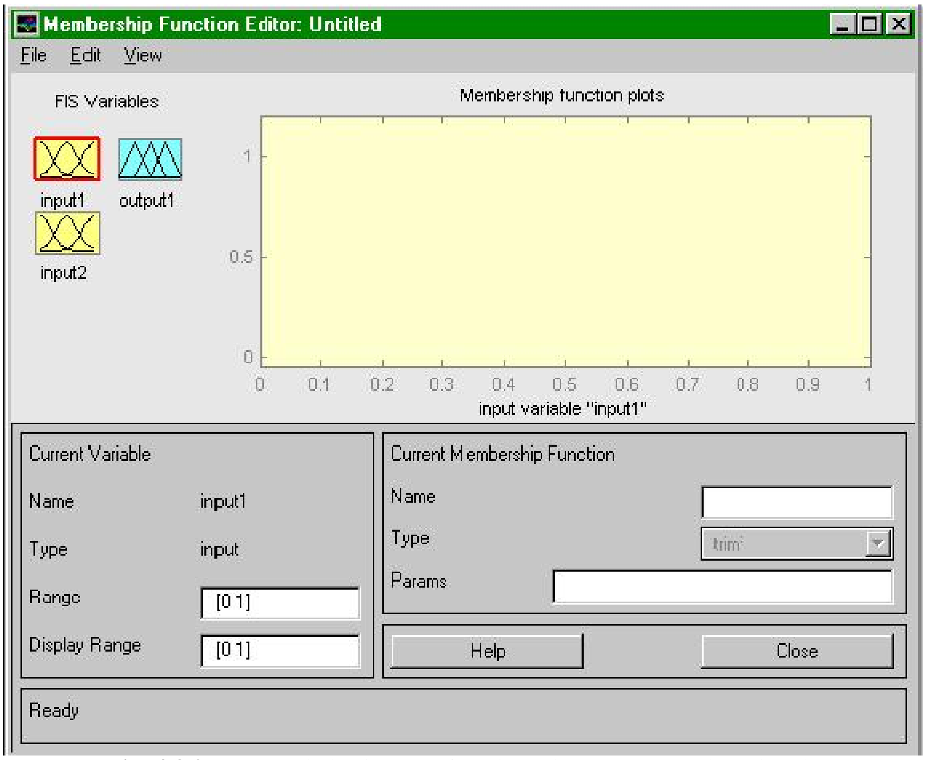


Fig. 4.3.2. Fenêtre de l'éditeur de fonctions d'appartenance d'un signal.

On doit y accomplir les actions suivantes, de préférence dans l'ordre indiqué: Choix de la gamme de variation de chaque signal, de préférence [–1; 1], ce qui implique que les signaux soient d'abord normalisés.

Choix des fonctions d'appartenance depuis la barre de menu: Edit —> Add MFs... . On accède alors à une fenêtre qui permet de choisir le nombre de fonctions et leur type. On préférera les formes simples – triangle «trimf» et trapèze «trapmf» – peu coûteuses en temps et en espace mémoire.

Sur la fenêtre MF, on choisit le nom et la dimension de chaque fonction d'appartenance: soit en déplaçant les points du graphique avec la souris, soit en spécifiant le vecteur dans la fenêtre «params». Pour le nom, un mnémonique – tel EZ pour "environ zéro" – vaut mieux que mf1 à mf*n* attribués par défaut. On ferme ensuite chaque fenêtre de signal.

De retour dans le du «FIS Editor», on choisit le nom du signal de préférence à input*k* attribué par défaut.

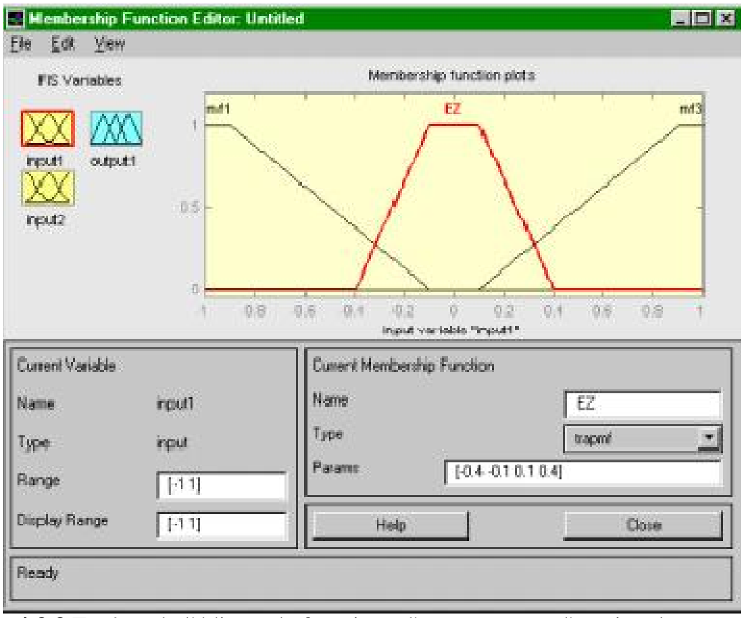


Fig. 4.3.3 Fenêtre de l'éditeur de fonctions d'appartenance d'un signal: exemple.

Lorsque tous les signaux sont spécifiés, on ouvre l'éditeur des règles d'inférence:

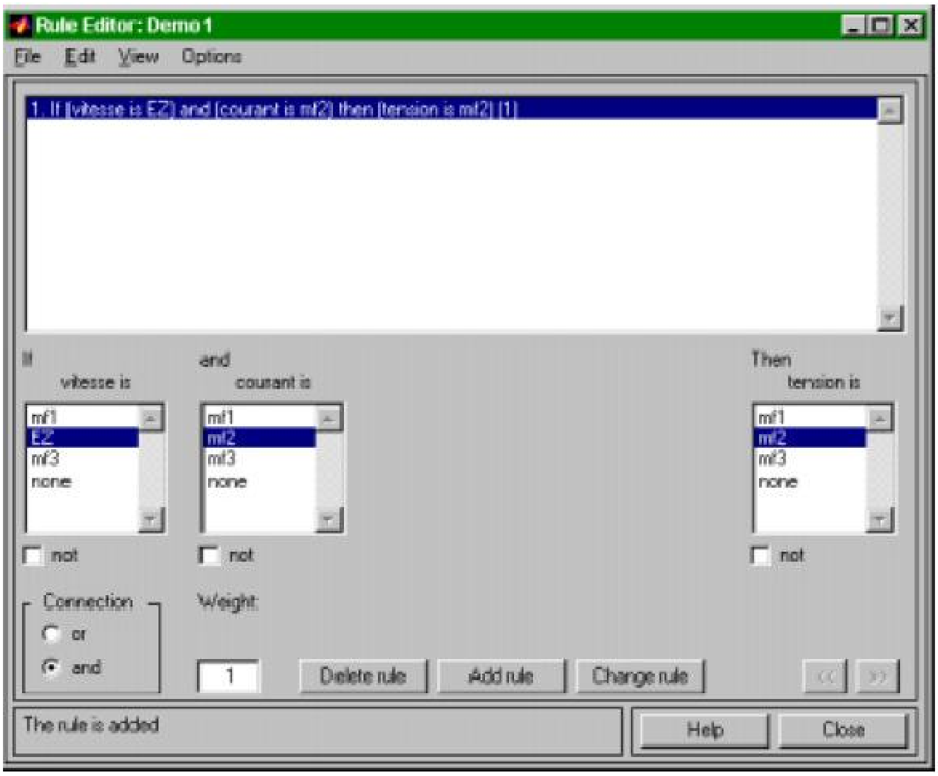


Fig. 4.3.4 Fenêtre de l'éditeur de règles d'inférence.

On termine en définissant les méthodes pour interpréter les combinaisons logiques et les règles d'inférence. On recommande ici des choix qui ne soient pas trop gourmands en temps de calcul, en donnant des résultats assez bons. Pour le ET logique (And method), on choisit la méthode produit (prod) et pour le OU logique (Or method), on choisit la méthode somme (mean).

Pour les règles d'inférence, on préfère la méthode somme-produit: Implication: prod et Aggregation: mean. Pour la défuzzyfication, on recommande la méthode du centre de gravité (centroid). On signale que la méthode somme (mean) n'est pas dans le menu, mais qu'il faut passer par la méthode Custom, dans laquelle on écrit en toute lettre «'mean'» sans oublier l'encadrement d'apostrophes.

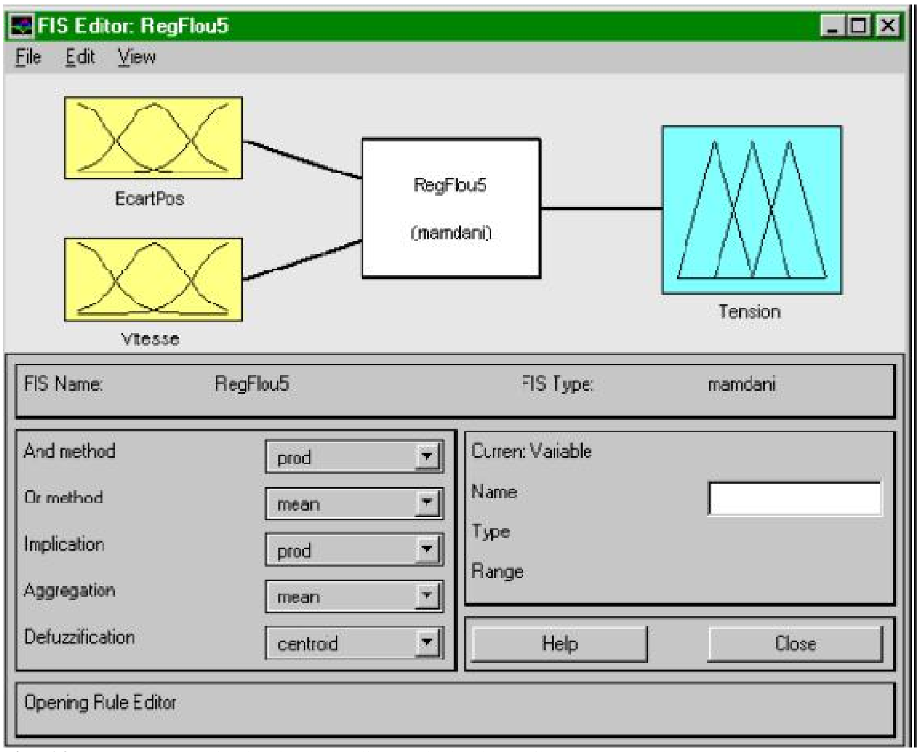


Fig. 12.B5 Fenêtre principale de l'éditeur de logique floue: exemple complété.

On ferme toutes les fenêtres secondaires de l'éditeur FIS avant d'enregistrer le régulateur sur le disque en choisissant un nom dont on se souviendra facilement. Si on veut ensuite utiliser ce régulateur dans un schéma *Simulink*, il faut encore l'enregistre dans l'espace de travail *MATLAB*: Workspace.