

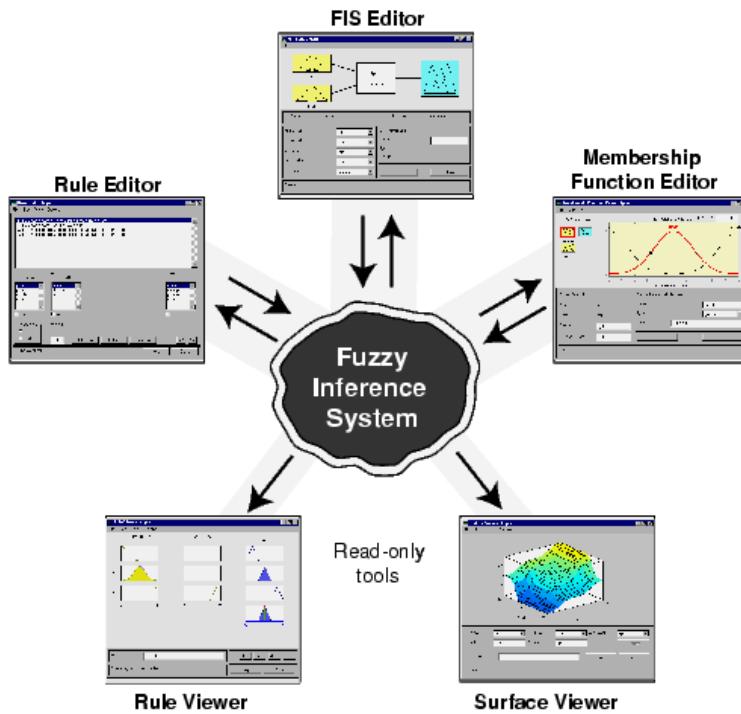


TP-1_Logique Floue

1. Présentation du Toolbox fuzzy logic

La toolbox fuzzy logic de Matlab possède un éditeur qui permet de créer des systèmes d'inférences floues : des FIS (pour fuzzy inference system). Cette boîte à outils permet de générer des fichiers « .fis », qui correspondent à des systèmes d'inférences floues.

Cette boîte à outils possède 3 éditeurs (de fis, de règles et de fonctions d'appartenance) qui permettent de saisir l'ensemble des données du FIS ainsi que 2 interfaces graphiques qui permettent de visualiser les inférences directement sur la base de règles, ainsi que des surfaces de contrôle.



2. Implémentation du système d'inférence flou sous Matlab

L'implémentation du système d'inférence flou sous Matlab, peut être réalisée par deux

méthodes différentes, soit à l'aide des commandes de la boite à outils Matlab (Fuzzy logic) ou par son interface graphique.

2.1. Utilisation de l'interface graphique de la boîte à outils «Fuzzy logic»

On commence par taper **fuzzy** à l'interface de commande de MATLAB : >> ce qui lance l'éditeur de contrôleur flou : Fis Editor.

2.1.1. Opérations de fuzzification

Etape 1

La commande **Edit** permettra de choisir le nombre d'entrées-sorties (3 entrées et 1 sortie dans notre application). On pourra renommer ces différentes variables.

Etape 2

Double-cliquer sur chaque variable d'entrée ou de sortie pour définir les valeurs linguistiques et la forme des fonctions d'appartenance. Pour cela utiliser dans le nouveau menu **Edit** la commande **Add MFs**, ce qui signifie ajouter des fonctions d'appartenance. On pourra choisir des fonctions triangulaires [**trimf**], trapézoïdales [**trapmf**], etc...

Etape 3

Double-cliquer sur le bloc central (mandani) pour entrer les règles d'inférences choisies.

Etape 4

Pour visualiser le comportement du système flou ainsi créé, on pourra utiliser dans le menu **View**, la commande **View rules**. Il est alors possible de fixer les valeurs des variables d'entrées, de voir quelles sont les règles d'inférences activées et de lire la valeur de la moyenne correspondante.

Pour visualiser la surface de décision floue, on pourra utiliser toujours dans le menu **View**, la commande **View surface**.

2.1.2. Sauvegarde du système dans l'espace de travail Matlab

Pour que le Fis ainsi créé soit disponible dans un modèle simulink, il faut qu'il soit sauvegardé dans l'espace de travail. Pour cela dans le menu **File**, utiliser l'option **Save to workspace**. Ainsi, son contenu sera mis dans un fichier de type « *.fis » (Notation.fis par exemple) auquel il sera fait appel dans simulink.

2.2. Utilisation des commandes de la boîte à outils «Fuzzy logic»

2.2.1. Fuzzification des variables d'entrée et de sortie

La création d'un système flou se fait à l'aide de la commande `newfis` qui accepte jusqu'à 7 arguments.

Dans la fenêtre de Matlab, tapez `help newfis`.

- Définition des différentes variables d'entrée et de sortie par la commande **addvar**.
- Les différentes fonctions d'appartenance associées aux différentes variables sont spécifiées par la commande **addmf**.
- Tracer les fonctions d'appartenance des valeurs d'entrée et de sortie du système à l'aide de la commande **plotmf**.

2.2.2. Edition des règles floues

Pour un système flou possédant m entrées et n sorties, l'ensemble des règles floues est défini par une matrice de règles possédant autant de lignes que d'ensembles flous de chacune des entrées et $(m+n+2)$ colonnes.

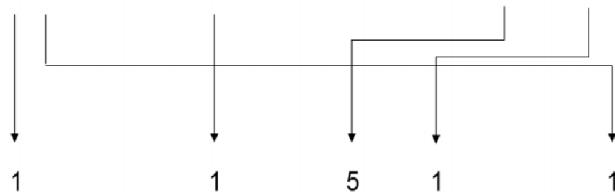
L'exemple d'une règle, d'un système flou à deux entrées et une sortie, est rappelée ci-après.

Si l'erreur est négative (1^{er} ensemble nommé N) ET la dérivée de l'erreur est négative (1^{er} ensemble nommé N) ALORS la commande est GP (5^{ème} ensemble nommé GP). Cette règle est pondérée par 1.

Le dernier chiffre symbolise l'opérateur liant les clauses des 2 prémisses (1 pour ET, 2 pour OU).

Une fois la matrice des chiffres définie, on procède à l'édition des règles, l'édition se fait par la commande **addrule**.

Si Erreur est N ET d_erreur est N ALORS commande est GP (1)



- Créer la matrice de règles.
- La commande **showrule** affiche les règles de façon normalisées.
- A l'aide de la commande **getfis**, afficher les caractéristiques: des entrées, de la sortie, du système flou, du 1er ensemble flou de la 1ère entrée.

- Afficher à l'aide de la commande **showfis** les caractéristiques de l'ensemble du système flou.
- Afficher la liste des règles avec la commande **showrule**.
- Générer la surface de la variable de sortie en fonction des entrées. On utilisera pour cela la commande **surfview**.
- Représenter graphiquement, à l'aide de la commande **plotfis**, le système.

2.2.3. Défuzzification

La commande **ruleview** affiche la fenêtre Rule Viewer dans laquelle on peut observer la défuzzification par la méthode max-min.

A l'aide de la souris on peut choisir des valeurs quelconques pour chacune des entrées et observer la fonction d'appartenance de la variable de sortie obtenue par la méthode d'inférence max -min. Par défaut, la défuzzification est réalisée par la méthode du calcul du centre de gravité.

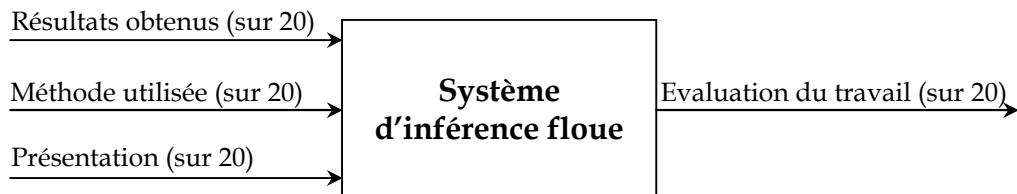
2.3.3. Test du système d'inférence flou

Le test du système d'inférence flou peut être obtenu par la commande **evalfis**.

3. Application

3.1. Système de notation floue

L'objectif de cette application est de réaliser un système de notation floue qui permet d'évaluer un mini-projet donné aux étudiants à partir des résultats obtenus, de la méthode utilisée et de la présentation de ce mini-projet comme illustrer ci-dessous.



▪ Choix des entrées/sorties

3 entrées : Résultats; Méthodes, Présentation.

1 sortie : Évaluation

▪ Univers des discours

[0..20] pour chacune des E/S

▪ Classes d'appartenances:

Résultats ∈ { Médiocre ; Moyen ; Excellent }
Méthode ∈ { Médiocre ; Moyen ; Excellent }
Présentation ∈ { Médiocre ; Moyen ; Excellent }
Evaluation ∈ { Médiocre ; Mauvais ; Moyen ; Bon ; Excellent }

▪ Types de fonctions d'appartenances:

Résultats : " Gaussienne "
Méthode : " Triangulaire "
Présentation : " Triangulaire "
Evaluation : " Triangulaire "

▪ Base de règles:

1. Si (Résultats est excellent) Alors (Evaluation est excellent)
2. Si (Résultats est moyen) Alors (Evaluation est moyen)
3. Si (Résultats est médiocre) Alors (Evaluation est médiocre)
4. Si (Résultats est moyen) Et (Méthodes est médiocre) Alors (Evaluation est mauvais)
5. Si (Résultats est moyen) Et (Méthodes est excellent) Alors (Evaluation est bon)
6. Si (Résultats est médiocre) Et (Méthodes est moyen) Alors (Evaluation est mauvais)
7. Si (Résultats est excellent) Et (Méthodes est excellent) Et (Présentation est excellent) Alors (Evaluation est excellent)
8. Si (Résultats est médiocre) Et (Méthodes est excellent) Alors (Evaluation est moyen)
9. Si (Résultats est excellent) Et (Méthodes est médiocre) Alors (Evaluation est moyen)

Chargé du module
Pr. T. Hacib