

## PLF      PROCESSEURS FLOUS

### TABEE DES MATIERES

1. <i>But de l'exposé</i>	3
2. <i>Introduction</i>	3
3. <i>Bref historique</i>	4
4. <i>Définitions</i>	4
4.1 <i>Processeur</i>	4
4.2 <i>Logique floue</i>	4
4.3 <i>Processeur flou</i>	4
5. <i>Principes de la logique floue</i>	5
5.1 <i>Introduction</i>	5
5.2 <i>Variables floues</i>	5
5.3 <i>Règles d'inférence</i>	7
6. <i>Variables floues</i>	8
6.1 <i>Introduction</i>	8
6.2 <i>Fonctions d'appartenance</i>	9
6.3 <i>Intervalles flous</i>	9
6.4 <i>Cas particulier : grandeur de sortie</i>	10
7. <i>Inférences et opérateurs</i>	11
7.1 <i>Règles d'inférences</i>	11
7.2 <i>Opérateurs</i>	11
8. <i>Combinaison des règles et défuzzification</i>	11
8.1 <i>Introduction</i>	11
8.2 <i>Combinaison des règles</i>	12
8.3 <i>Défuzzification</i>	12
8.4 <i>Défuzzification par calcul du centre de gravité</i>	13
8.5 <i>Défuzzification par calcul du maximum</i>	13
9. <i>Application au réglage</i>	14
9.1 <i>Introduction</i>	14
9.2 <i>Contrôleur flou</i>	15

<b>10. Exemple</b>	<b>15</b>
<b>10.1 Description du problème</b>	<b>15</b>
<b>10.2 Fuzzification de la température externe</b>	<b>15</b>
<b>10.3 Fuzzification de la température interne</b>	<b>16</b>
<b>10.4 Fuzzification de la puissance</b>	<b>17</b>
<b>10.5 Règles d'inférences</b>	<b>18</b>
<b>10.6 Choix des opérateurs</b>	<b>18</b>
<b>10.7 Choix du type de défuzzification</b>	<b>18</b>
<b>10.8 Exemple de calcul</b>	<b>18</b>
<b>11. Processeurs flous</b>	<b>20</b>
<b>11.1 Définition et structure</b>	<b>20</b>
<b>11.2 Solutions possibles</b>	<b>20</b>
<b>12. Processeurs flous digitaux</b>	<b>20</b>
<b>12.1 Introduction</b>	<b>20</b>
<b>12.2 Exemples de réalisations</b>	<b>21</b>
<b>13. Processeurs flous analogiques</b>	<b>22</b>
<b>13.1 Introduction</b>	<b>22</b>
<b>13.2 Exemple de réalisation : version à miroirs de courants</b>	<b>22</b>
<b>13.3 Exemple de réalisation : version MOS en résistances variables</b>	<b>24</b>
<b>14. Conclusions</b>	<b>25</b>
<b>15. Sources</b>	<b>25</b>

## 1. BUT DE L'EXPOSE

Le but de cet exposé est de définir et de présenter ce que l'on entend par « processeur flou ». Pour ce faire, on présente tout d'abord les principes de la logique floue et ses applications dans le domaine du réglage puis on décrit différentes solutions pour un organe de commande (processeur) basé sur les principes du flou.

## 2. INTRODUCTION

Depuis quelques années déjà, on trouve sur le marché des appareils de grande consommation (appareils de photos, vidéo, ...) qui sont présentés comme faisant intervenir un réglage par logique floue ou « *fuzzy-logic* » ou encore « *fuzzy-control* ». Au delà de l'argument publicitaire évident, il est intéressant de comprendre ce nouveau concept de réglage et de l'appliquer à certains types de problèmes de réglage rencontrés par l'ingénieur.

Le principe du réglage par logique floue part du constat suivant: dans les problèmes de régulation auxquels il est confronté, l'homme ne suit pas, à l'image de ses inventions, un modèle mathématique fait de valeurs numériques et d'équations. Au contraire il utilise des termes tel que « un peu trop chaud, aller beaucoup plus vite, freiner à fond, etc... » ainsi que ses propres connaissances qu'il a dans le domaine. Ces connaissances sont, le plus souvent, acquises de façon empirique. Le principe du réglage par la logique floue s'approche de la démarche humaine dans le sens que les variables traitées ne sont pas des variables logiques (au sens de la logique binaire par exemple) mais des variables *linguistiques*, proches du langage humain de tous les jours. De plus, ces variables linguistiques sont traitées à l'aide de règles qui font référence à une certaine connaissance du comportement du système à régler. Sur la base de ce principe, différentes réalisations ont vu le jour et, actuellement, on trouve deux types d'approche pour le réglage par logique floue. Dans l'une de ces approches, les règles sont appliquées aux variables à l'aide d'une approche numérique par le biais d'un microprocesseur spécialisé ou non ou d'un ordinateur. Dans l'autre approche, les règles sont appliquées aux variables de façon analogique. Ces deux approches permettent de développer des organes de contrôle par le flou que l'on désigne par *processeur digital flou* ou par *processeur analogique flou*.

### 3. BREF HISTORIQUE

Les quelques points de repères historiques suivants permettent de situer dans le temps le développement de la logique floue et ses applications au réglage:

- 1965 Le Prof. L. A. Zadeh de l'Université de Berkeley (Californie) pose les bases théoriques de la logique floue.
- 1973 L. A. Zadeh propose d'appliquer la logique floue aux problèmes de réglage.
- 1974 Première application du réglage par la logique floue appliquée à une turbine à vapeur. Suivie en 1980 par une application sur un four à ciment et en 1983 sur un épurateur d'eau.
- 1985 Premiers produits industriels (Japon) utilisant le principe de la logique floue appliqué à des problèmes de réglage et de commande. Développement de processeurs dédiés à des applications de réglage par la logique floue.

### 4. DEFINITIONS

#### 4.1 *Processeur*

On appelle *processeur* un « organe capable d'assurer le traitement complet d'une série d'informations » (Larousse).

#### 4.2 *Logique floue*

« Logique qui substitue à la logique binaire une logique fondée sur des variables pouvant prendre, outre les valeurs « vrai » ou « faux », les valeurs intermédiaires « vrai » ou « faux » avec une certaine probabilité. » (citation: cf. supra)

#### 4.3 *Processeur flou*

Processeur capable de traiter des informations en faisant intervenir les principes de la logique floue.

## 5. PRINCIPES DE LA LOGIQUE FLOUE

### 5.1 *Introduction*

La logique floue est une branche des mathématiques et, à ce titre, toute une série de notions fondamentales sont développées. Ces notions permettent de justifier et de démontrer certains principes de base. Dans ce qui suit on ne retiendra que les éléments indispensables à la compréhension du principe du réglage par la logique floue. Ces éléments sont :

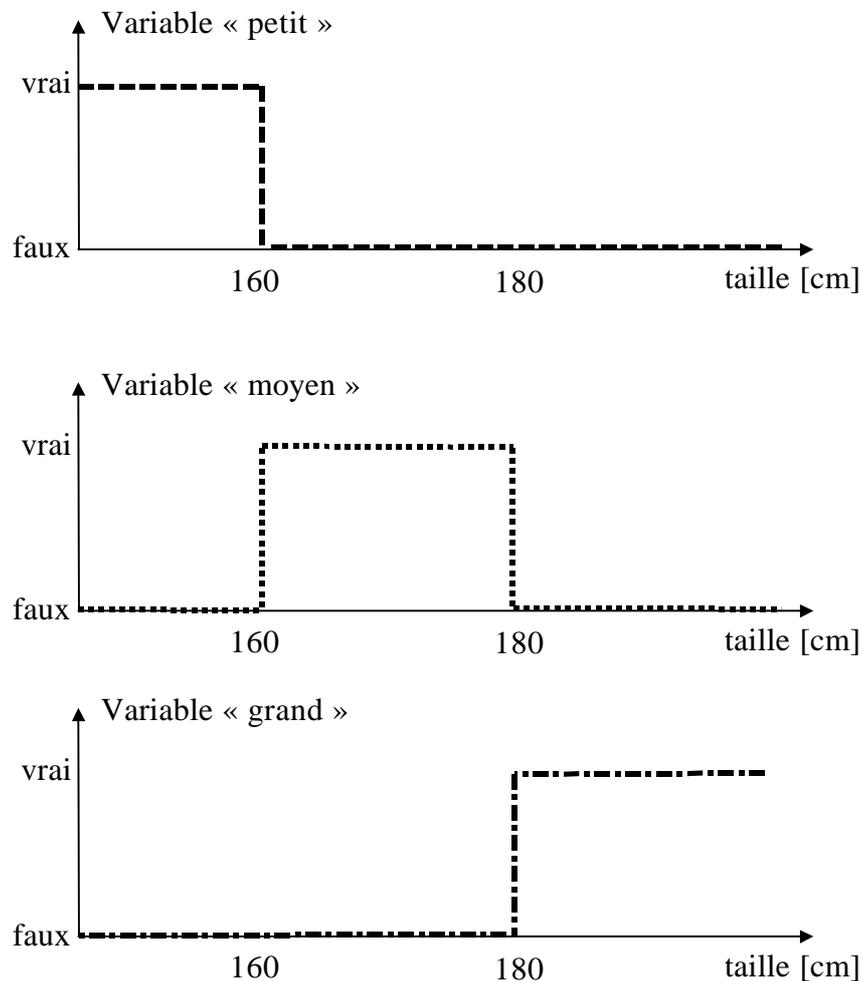
- les variables floues
- les règles d'inférences

### 5.2 *Variables floues*

Contrairement aux variables binaires qui sont définies par les deux états « vrai » ou « faux », les variables floues présentent toute une gradation entre la valeur « vrai » et la valeur « faux ».

L'exemple qui suit permet de mieux saisir la distinction qui existe entre les variables binaires et les variables floues :

Si l'on désire classer un groupe d'individu par leur taille en définissant la catégorie des petits par une taille en dessous de 160 cm, la catégorie des moyens par une taille comprise entre 160 cm et 180 cm et la catégorie des grand par une taille supérieure à 180 cm, la logique binaire donne la représentation de la figure 1 pour les trois variables « petit », « moyen » et « grand »



**Fig. 1**

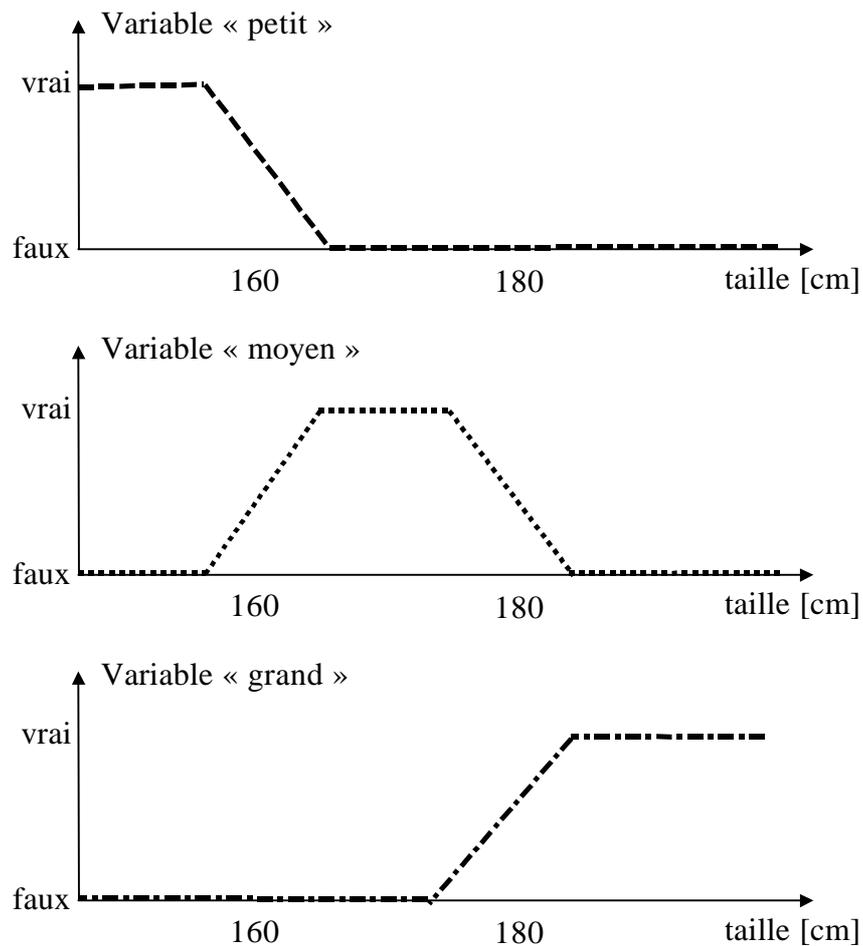
Deux remarques s'imposent au sujet de cette représentation :

D'une part, on préfère représenter l'état de la variable à l'aide de son degré de vérité en associant la valeur 1 (degré de vérité de 100%) à la valeur « vrai » et le degré de vérité nul à la valeur « faux ».

D'autre part, on constate que cette façon de faire est très éloignée de ce que fait l'être humain lorsqu'il résout ce genre de problème. En effet, l'homme ne fait pas naturellement une distinction franche entre « petit » et « moyen » par exemple. Il utilise des expressions du genre « plutôt petit » pour qualifier une taille légèrement inférieure à 160 cm et « plutôt moyen » pour une taille légèrement supérieure à cette valeur.

En conclusion, la logique binaire présente l'avantage de la simplicité mais est assez éloignée de la logique utilisée naturellement par l'être humain.

Si l'on représente le même problème à l'aide de la logique floue, les variables ne sont plus binaires mais présentent une infinité de valeurs possible entre le « vrai » et le « faux » (cf. figure 2).



**Fig. 2**

On constate que cette représentation est beaucoup plus proche de la façon dont l'être humain raisonne puisqu'elle permet de faire intervenir des notions telles que « plutôt petit », « assez grand »... Cette avantage se fait, évidemment, au détriment de la simplicité de la représentation.

### 5.3 Règles d'inférence

On appelle *règles d'inférence* l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système. Ces règles se présentent sous la forme :

**Si** condition 1 **et/ou** condition 2 ( **et/ou...**) **alors** action sur les sorties

L'exemple suivant, tiré de la vie quotidienne, permet d'illustrer ceci :

Lorsque l'on prend une douche, un des problèmes qui se présente est de régler la température de l'eau. La variable d'entrée du système homme-douche est la température de l'eau mesurée à l'aide de nos capteurs de température. Les variables de sorties sont les deux robinets eau chaude et eau froide. Dans la pratique, le réglage de la température se fait en utilisant notre expérience, expérience qui recouvre à la fois nos préférences et notre connaissance de l'installation sanitaire utilisée. Ce réglage se fait en utilisant des règles du genre :

- **Si** la température est très froide **alors** ouvrir à fond l'eau chaude
- **Si** la température est un peu trop froide **alors** fermer un peu l'eau froide
- **Si** la température est bonne **alors** laisser les deux robinets dans leur état
- **Si** la température est trop chaude **alors** fermer un peu l'eau chaude et ouvrir un peu l'eau froide
- etc...

En termes d'intelligence artificielle, ces règles résument en fait *l'expérience de l'expert* et elles ne sont en général pas définissables de façon unique puisque chaque individu crée ses propres règles.

## 6. VARIABLES FLOUES

### 6.1 Introduction

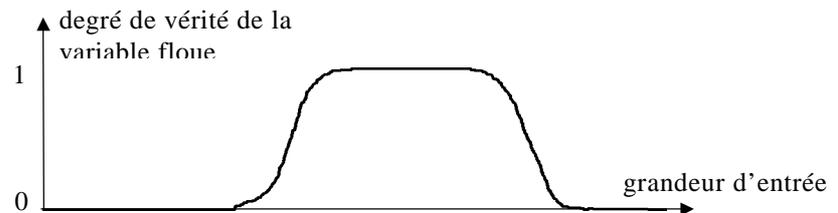
Les grandeurs utilisées dans un système de réglages sont généralement générées par des capteurs. Il est nécessaire de convertir ces grandeurs en variables floues. Pour ce faire on définit les deux notions suivantes :

- Les *fonctions d'appartenances* qui permettent de définir le degré de vérité de la variable floue en fonction de la grandeur d'entrée
- Les *intervalles flous* qui déterminent le nombre de variables floues

Dans l'exemple de la figure 2, on fait intervenir trois intervalles flous : « petit », « moyen » et « grand ». En outre chaque intervalle fait référence à une fonction d'appartenance qui permet de définir le degré de vérité de la variable floue correspondante en fonction de la taille.

### 6.2 Fonctions d'appartenance

Il s'agit d'établir une relation entre le degré de vérité de la variable floue et la grandeur d'entrée correspondante (cf. figure 3). On parle de *fuzzification* :



**Figure 3**

On peut évidemment choisir n'importe quelle forme pour les fonctions d'appartenance. Cependant, en pratique, on utilise les formes trapézoïdales (cas de la figure 2) et les formes en cloche (gaussiennes).

### 6.3 Intervalles flous

Ces intervalles définissent le nombre de variables floues associées à une grandeur d'entrée. Dans le cas du réglage, trois à cinq intervalles s'avèrent suffisants. De façon générale ils sont caractérisés à l'aide de symboles tels que ceux présentés dans le tableau 1.

Symbole	Signification
NG	négatif grand
NM	négatif moyen
EZ	environ zéro
PM	positif moyen
PG	positif grand

**Tableau 1**

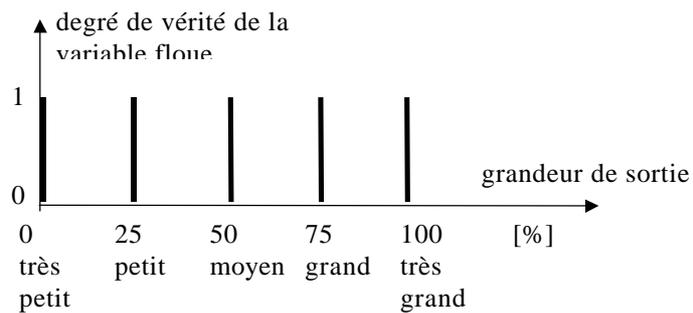
6.4 *Cas particulier : grandeur de sortie*

La grandeur de sortie peut être définie à l'aide d'un certain nombre d'intervalles flous et diverses fonctions d'appartenance. Toutefois, en pratique, cette définition peut sembler assez lourde et le concepteur (l'*expert*) peut choisir d'associer une seule valeur à chaque intervalle flou. Par exemple, pour une grandeur à cinq intervalles flous, on peut définir les valeurs suivantes (cf. tableau 2) :

Intervalle	Valeur en % du maximum
très petit	0
petit	25
moyen	50
grand	75
très grand	100

**Tableau 2**

Ce qui définit des fonctions d'appartenances en forme de raies comme illustré à la figure 4.



**Figure 4**

## 7. INFERENCE ET OPERATEURS

### 7.1 Règles d'inférences

Ces règles permettent de relier les variables floues d'entrée aux variables floues de sortie à l'aide de différents opérateurs (cf. § 5.3 et 7.2). Elles doivent être définies par le concepteur de système de réglage en fonction de son expérience (rôle d'*expert*) et mémorisées dans l'organe de commande.

### 7.2 Opérateurs

Les règles d'inférences font appel aux opérateurs **et**, **ou** et **non**, qui s'appliquent aux variables floues. Dans le cas de la logique binaire ces opérateurs sont définis de façon simple et univoque. Dans le cas de la logique floue, la définition de ces opérateurs n'est plus univoque et on utilise le plus souvent les relations présentées dans le tableau 3.

Opérateur	Opération sur le degré de vérité des variables
<b>et</b>	minimum
	produit
<b>ou</b>	maximum
	valeur moyenne
<b>non</b>	complément à 1

Tableau 3

Les opérations **minimum** et **maximum** présentent l'avantage de la simplicité lors du calcul, par contre, elles privilégient l'une des deux variables. Les opérations de **produit** et **valeurs moyenne** sont plus complexes à calculer mais elles produisent un résultat qui tient compte des valeurs des deux variables.

## 8. COMBINAISON DES REGLES ET DEFUZZIFICATION

### 8.1 Introduction

Les différentes règles d'inférences (cf. § 5.3 et 7.1) produisent chacune une valeur. Ces différentes valeurs doivent être combinées afin d'obtenir la (éventuellement les) variable(s) de sortie. Ensuite la (ou les) variable(s) floue(s) de sortie doivent être converties en une grandeur de commande

(tension, couple...) afin d'être appliquée au système à régler. On appelle cette dernière étape la *défuzzification*.

## 8.2 *Combinaison des règles*

L'ensemble des règles se présente sous la forme d'une énumération du type :

**Si** condition 1 **et/ou** condition 2 ( **et/ou...**) **alors** action sur les sorties

**Si** condition 3 **et/ou** condition 4 ( **et/ou...**) **alors** action sur les sorties

**Si** condition 5 **et/ou** condition 6 ( **et/ou...**) **alors** action sur les sorties

...

La combinaison de ces différentes règles se fait à l'aide de l'opérateur **ou**. La justification du choix de l'opérateur se fonde sur la pratique du langage courant : en effet, une telle énumération est comprise dans le sens

Si... alors...

**ou**

Si... alors...

**ou**

...

bien que l'opérateur **ou** ne soit pas explicitement mentionné.

## 8.3 *Défuzzification*

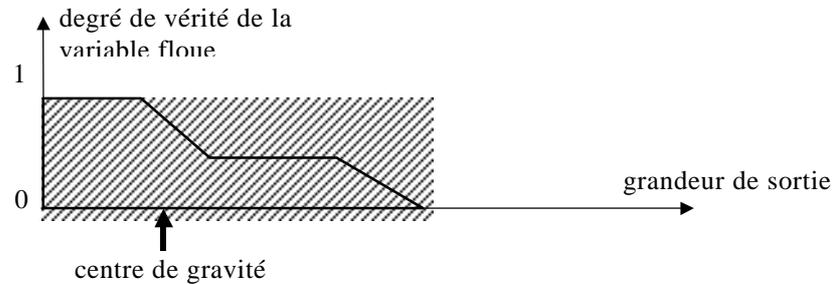
Les valeurs obtenues lors de la combinaison des règles appliquées aux intervalles flous de la variable de sortie défini une fonction d'appartenance.

Il s'agit de convertir cette information en une grandeur physique. Plusieurs façons de faire peuvent être envisagées mais, en pratique, on utilise surtout les deux méthodes suivantes :

- défuzzification par calcul du centre de gravité
- défuzzification par calcul du maximum

#### 8.4 *Défuzzification par calcul du centre de gravité*

Il s'agit de calculer le centre de gravité de la fonction d'appartenance de la variable de sortie (figure 5) :



**Fig. 5**

Le calcul du centre de gravité permet bien d'obtenir une seule valeur pour la grandeur de sortie. Son calcul est cependant relativement complexe puisqu'il nécessite le calcul d'une intégrale, ou dans le cas simple de fonctions d'appartenance en raies, d'une somme pondérée.

#### 8.5 *Défuzzification par calcul du maximum*

Il s'agit de la façon la plus simple, au point de vue du volume de calcul, pour effectuer la défuzzification. La façon de procéder diffère cependant fondamentalement du cas général exposé ci-dessus. Tout d'abord, la grandeur de sortie doit être normalisée (en pour-cent par exemple) et la définition des intervalles flous doit se résumer à une valeur : par exemple « petit » correspond à 0 et « moyen » à 0,5 (fonctions d'appartenance en forme de raies). L'opération de défuzzification consiste à prendre d'abord le minimum entre la valeur produite par la règle concernée et la valeur de la variable floue de sortie. La valeur de sortie est définie par la valeur maximale des variables floues de sortie.

L'exemple suivant permet d'en illustrer le principe :

Soit un système avec trois règles :

la règle 1 donne une sortie de type « petit » d'un degré de vérité de 0,8

la règle 2 : sortie de type « moyen » d'un degré de vérité de 0,3

la règle 3 : sortie de type « grand » d'un degré de vérité de 0,1

la valeur normalisée de l'intervalle « petit » vaut 0

la valeur normalisée de l'intervalle « moyen » vaut 0,5

la valeur normalisée de l'intervalle « grand » vaut 1

la règle 1 donne une valeur de sortie de 0 (minimum entre 0,8 et 0)  
la règle 2 donne une valeur de sortie de 0,3 (minimum entre 0,3 et 0,5)  
la règle 3 donne une valeur de sortie de 0,1 (minimum entre 0,1 et 1)  
la grandeur de sortie est déterminée par le maximum des valeurs obtenues  
et vaut 0,3 ce qui correspond à une valeur « plutôt petite ».

On constate que cette méthode est simple à appliquer mais, étant basée sur l'opérateur **maximum**, elle privilégie une seule règle parmi celles présentes.

## 9. APPLICATION AU REGLAGE

### 9.1 Introduction

Les problèmes de réglage peuvent être résolus par la logique floue en conservant une structure proche de la structure traditionnelle (PI par exemple). Cette structure est illustrée à la figure 6.

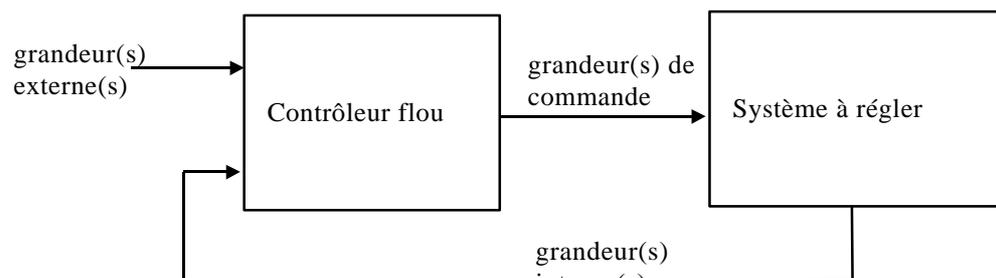
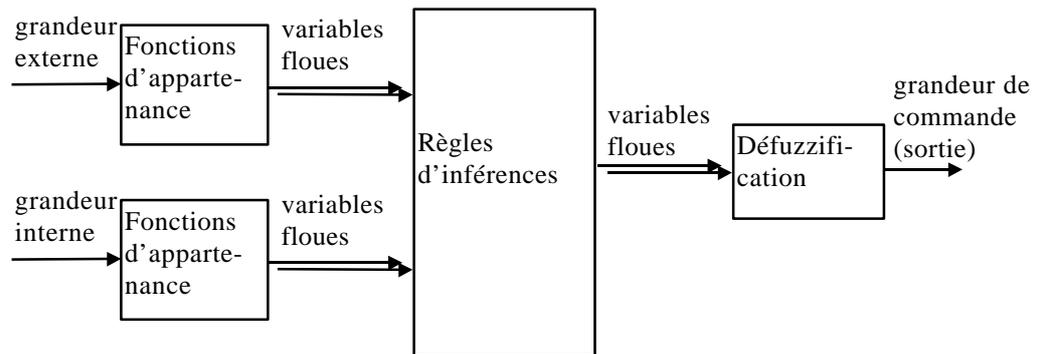


Figure 6

## 9.2 *Contrôleur flou*

La structure du contrôleur flou est illustrée à la figure 7. Par souci de simplicité, on s'est limité à un contrôleur à deux entrées.



**Figure 7**

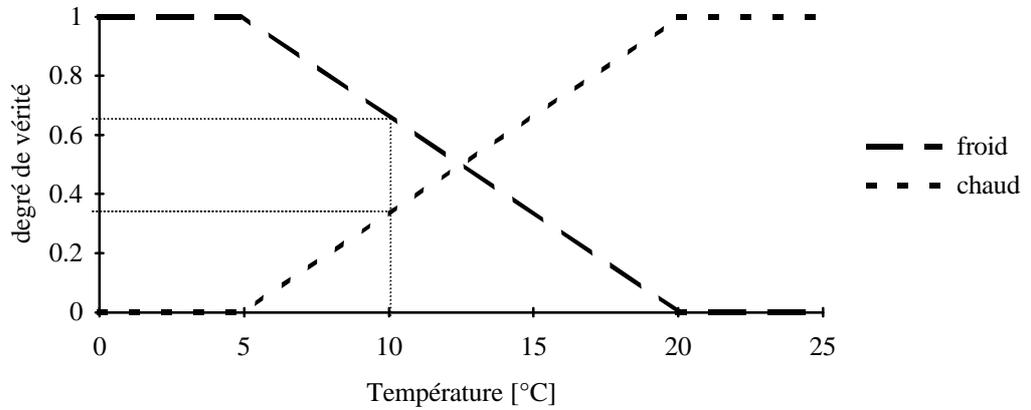
## 10. **EXEMPLE**

### 10.1 *Description du problème*

On souhaite commander l'installation de chauffage d'un immeuble à l'aide d'un contrôleur flou. On dispose de deux sondes de température : l'une à l'extérieur de l'immeuble (grandeur externe sur la figure 6) l'autre à l'intérieur (grandeur interne). Sur la base de ces deux mesures et en faisant appel aux règles d'inférence, le contrôleur flou doit régler la puissance de l'installation de chauffage.

### 10.2 *Fuzzification de la température externe*

On choisit deux intervalles flous et des fonctions d'appartenance de type trapézoïdales en définissant le « froid » comme correspondant à une température inférieure à 5 °C et le « chaud » comme étant une température supérieure à 20 °C (cf. fig. 8)

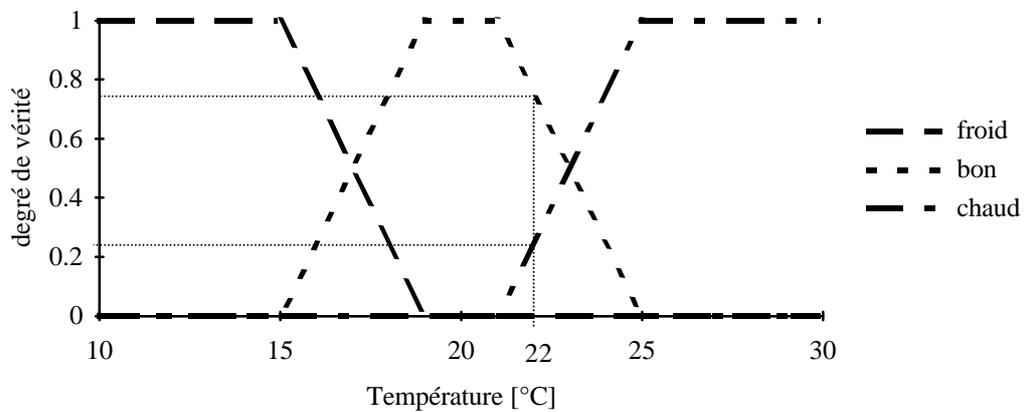


**Figure 8**

10.3

*Fuzzification de la température interne*

On choisit trois intervalles flous et des fonctions d'appartenance de type trapézoïdales en définissant le « froid » comme correspondant à une température inférieure à 15 °C, le « bon » comme étant une température comprise entre 19 °C et 21 °C et le « chaud » comme étant une température supérieure à 25 °C (cf. fig. 9)



**Figure 9**

10.4 *Fuzzification de la puissance*

On choisi quatre intervalles flous pour définir la puissance de l'installation avec des fonctions d'appartenance en forme de raies. On définit les valeurs suivantes :

Puissance	Valeur en %
nulle	0
faible	33
moyenne	67
maximale	100

Ce qui définit les fonctions d'appartenance illustrées à la figure 10

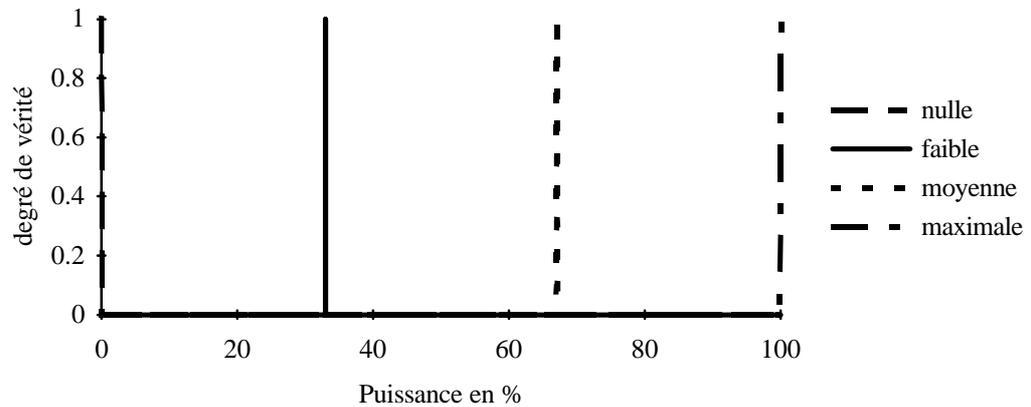


Figure 10

### 10.5 Règles d'inférences

L'expérience acquise sur l'installation de chauffage a permis de définir les six règles suivantes :

1. **Si** la température extérieure est « froide » **et** la température intérieure est « froide » **alors** mettre la puissance au « maximum »
2. **Si** la température extérieure est « froide » **et** la température intérieure est « bonne » **alors** mettre une puissance « moyenne »
3. **Si** la température extérieure est « froide » **et** la température intérieure est « chaude » **alors** mettre une puissance « faible »
4. **Si** la température extérieure est « chaude » **et** la température intérieure est « froide » **alors** mettre une puissance « moyenne »
5. **Si** la température extérieure est « chaude » **et** la température intérieure est « bonne » **alors** mettre une puissance « faible »
6. **Si** la température extérieure est « chaude » **et** la température intérieure est « chaude » **alors** mettre une puissance « nulle »

### 10.6 Choix des opérateurs

- L'opérateur **et** est réalisé par le calcul du **minimum**
- L'opérateur **ou** est réalisé par le calcul du **maximum**

### 10.7 Choix du type de défuzzification

La défuzzification se fait par le calcul du centre de gravité.

### 10.8 Exemple de calcul

Hypothèse : la température extérieure est de 10 °C et la température intérieure est de 22 °C

- Les variables floues de la température extérieure sont donc (cf. figure 8)
  - « froide » avec un degré de vérité de 0,67
  - « chaude » avec un degré de vérité de 0,33
- Les variables floues de la température intérieure sont donc (cf. figure 9)
  - « froide » avec un degré de vérité de 0
  - « bonne » avec un degré de vérité de 0,75
  - « chaude » avec un degré de vérité de 0,25

Les règles d'inférences donnent les valeurs suivantes pour les variables floues de sortie :

1. « maximum » avec un degré de vérité de 0 (minimum de 0,67 et 0)
2. « moyenne » avec un degré de vérité de 0,67 (mini. de 0,67 et 0,75)
3. « faible » avec un degré de vérité de 0,25 (mini. de 0,67 et 0,25)
4. « moyenne » avec un degré de vérité de 0 (mini. de 0,33 et 0)
5. « faible » avec un degré de vérité de 0,33 (mini. de 0,33 et 0,75)
6. « nulle » avec un degré de vérité de 0,25 (mini. de 0,33 et 0,25)

L'opérateur **ou** appliqué sur les règles qui donnent les mêmes variables floues donnent :

- « maximum » avec un degré de vérité de 0
- « moyenne » avec un degré de vérité de 0,67 (maxi. de 0,67 et 0)
- « faible » avec un degré de vérité de 0,33 (maxi. de 0,25 et 0,33)
- « nulle » avec un degré de vérité de 0,67

Le calcul du centre de gravité se calcule à l'aide de :

$$P = \frac{\sum_{i=1}^4 \mu_i \cdot P_i}{\sum_{i=1}^4 \mu_i} = \frac{0 \cdot 100 + 0,67 \cdot 67 + 0,33 \cdot 33 + 0 \cdot 0,25}{0 + 0,67 + 0,33 + 0,25} = 44,6$$

- avec P : Puissance du chauffage en %  
 $\mu_i$  : degré de vérité de la variable floue « puissance » indice i  
 $P_i$  : valeur de la variable floue « puissance » indice i

Le contrôleur flou impose donc une puissance de 44,6% sur l'installation de chauffage.

## 11. PROCESSEURS FLOUS

### 11.1 Définition et structure

Un processeur flou est un circuit ou un ensemble de circuits qui permet de commander un système à régler en utilisant les principes de la logique floue exposés ci-dessus. La structure d'un tel processeur est la suivante (figure 11) :

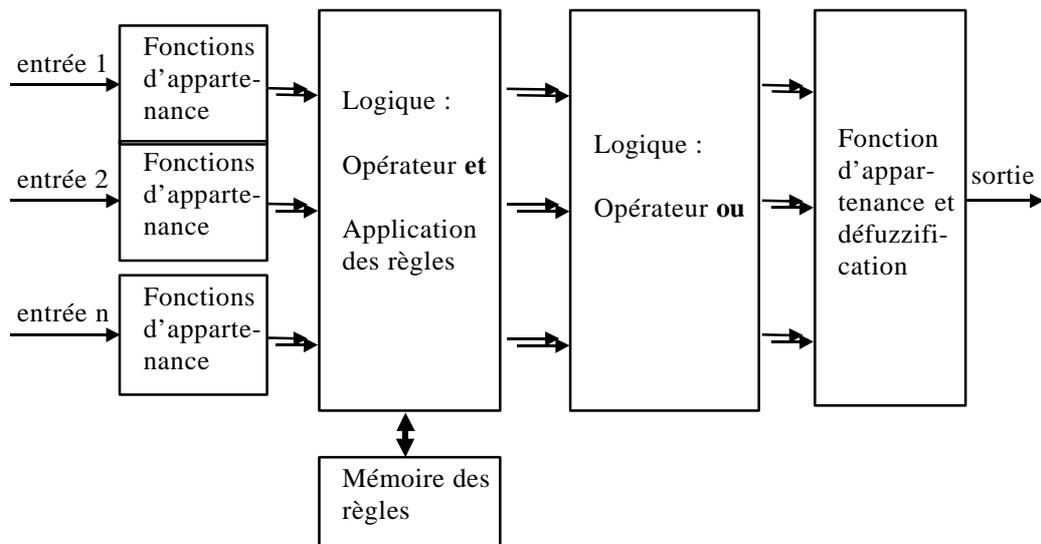


Figure 11

### 11.2 Solutions possibles

Deux solutions peuvent être envisagées pour réaliser une telle structure :

- le traitement se fait de façon digitale et on parle de *processeur flou digital*
- le traitement se fait essentiellement de façon analogique (seule la partie de mémorisation peut être digitale), on parle alors de *processeur flou analogique*

## 12. PROCESSEURS FLOUS DIGITAUX

### 12.1 Introduction

Les fonctions de la logique floue décrites précédemment peuvent sans autre être implantées sur un système microprocesseur classique. Cependant, l'utilisateur se heurte vite au compromis rapidité-consommation. Pour s'en convaincre, il suffit de constater que la fuzzification d'une entrée nécessite au moins une multiplication par intervalle flou soit plusieurs dizaines de cycles machine. Il faut donc compter avec plus d'une centaine de cycles machine

pour uniquement la fuzzification de toutes les entrées. Le calcul des règles prend encore plus de temps. Ceci nécessite donc une machine rapide dès que le processus à régler présente des constantes de temps inférieur à la seconde. De plus, les processeurs traditionnels sont relativement mal adaptés aux problèmes de la logique floue car plusieurs de leurs fonctions ne sont tout simplement pas utilisées. Pour cette raison, on recourt le plus souvent à des processeurs spécialisés de type RISC. Ces processeurs sont souvent livrés avec un logiciel de développement qui permet à l'utilisateur de mettre en oeuvre très rapidement son système de réglage. Il ne faut cependant pas perdre de vue que ce faisant, on réalise une simulation *digitale* d'un processus qui est, par définition, de type *analogique*.

## 12.2 Exemples de réalisations

Ci-dessous on trouvera une liste non exhaustive de quelques réalisations disponibles sur le marché :

### **Inform FC 166**

Nombre d'entrées	10
Nombre de sorties	32 x 1 bit (digitales)
Nbre de règles possible	>256 000 (avec mémoire externe)
Rapidité	400 000 règles évaluées par seconde
Résolution	10 bits
Opérateurs <b>et</b> et <b>ou</b>	<b>minimum</b> et <b>maximum</b> , <b>maximum</b> et <b>produit</b>
Défuzzification	centre de gravité, centre de gravité du maximum et moyenne du maximum
Prix indicatif	200.—FS

### **Neuralogix NLX 230**

Nombre d'entrées	8
Nombre de sorties	8
Nbre de règles possible	64
Rapidité	300 règles évaluées par seconde
Résolution	8 bits
Opérateurs <b>et</b> et <b>ou</b>	<b>minimum</b> et <b>maximum</b>
Défuzzification	incrémentale
Prix indicatif	30.-- FS

### **Omron FP 3000**

Nombre d'entrées	8
Nombre de sorties	4
Nombre de règles possible	128 (maximum)
Rapidité	1 500 règles évaluées par seconde
Résolution	12 bits

Opérateurs <b>et et ou</b>	<b>minimum et maximum</b>
Défuzzification	centre de gravité et hauteur maximale
Prix indicatif	130.-- FS

#### **Togai FC 110 DFP**

Nombre d'entrées	8
Nombre de sorties	4
Nombre de règles possible	800 (environ)
Rapidité	220 000 règles évaluées par seconde
Résolution	8 bits
Opérateurs <b>et et ou</b>	<b>minimum et maximum</b>
Défuzzification	centre de gravité et hauteur maximale
Prix indicatif	100.-- FS

## **13. PROCESSEURS FLOUS ANALOGIQUES**

### *13.1 Introduction*

Pour des applications spécifiques dans lesquelles le nombre d'entrées et de sorties est faible, il peut s'avérer intéressant d'effectuer le traitement de façon essentiellement analogique. En effet, cette façon de procéder permet une solution qui nécessite un nombre de transistors nettement moins élevé que la solution digitale. Pour de petits systèmes (2 entrées, une sortie, 3 intervalles flous par variables) quelques milliers de transistors suffisent.

Pour illustrer la solution analogique, on trouvera ci-dessous une brève description de deux réalisations :

- Dans la première réalisation, les opérateurs **et et ou** sont réalisés par les opérations **minimum et maximum**. La défuzzification se fait par calcul du maximum et par calcul du centre de gravité (deux sorties à disposition). La réalisation des différentes opérations est axée sur l'utilisation de miroirs de courant.
- Dans la seconde réalisation, les opérateurs **et et ou** sont réalisés par les opérations **produit sur somme et somme**. La défuzzification se fait par calcul du centre de gravité. La réalisation des différentes opérations se fait en utilisant les transistors MOS en résistances variables.

### *13.2 Exemple de réalisation : version à miroirs de courants*

La structure est celle présentée à la figure 12. On y retrouve les principales fonctions décrites au paragraphe 11.1. Ce processeur permet de traiter deux entrées et travaille avec trois intervalles flous. Les fonctions d'appartenance sont de type trapézoïdales. Les neuf règles sont programmées à l'aide de tensions appliquées sur les entrées de programmation. Cette réalisation fait

actuellement l'objet d'un travail de microélectronique pour une classe d'ingénieurs en électronique à l'EISI.

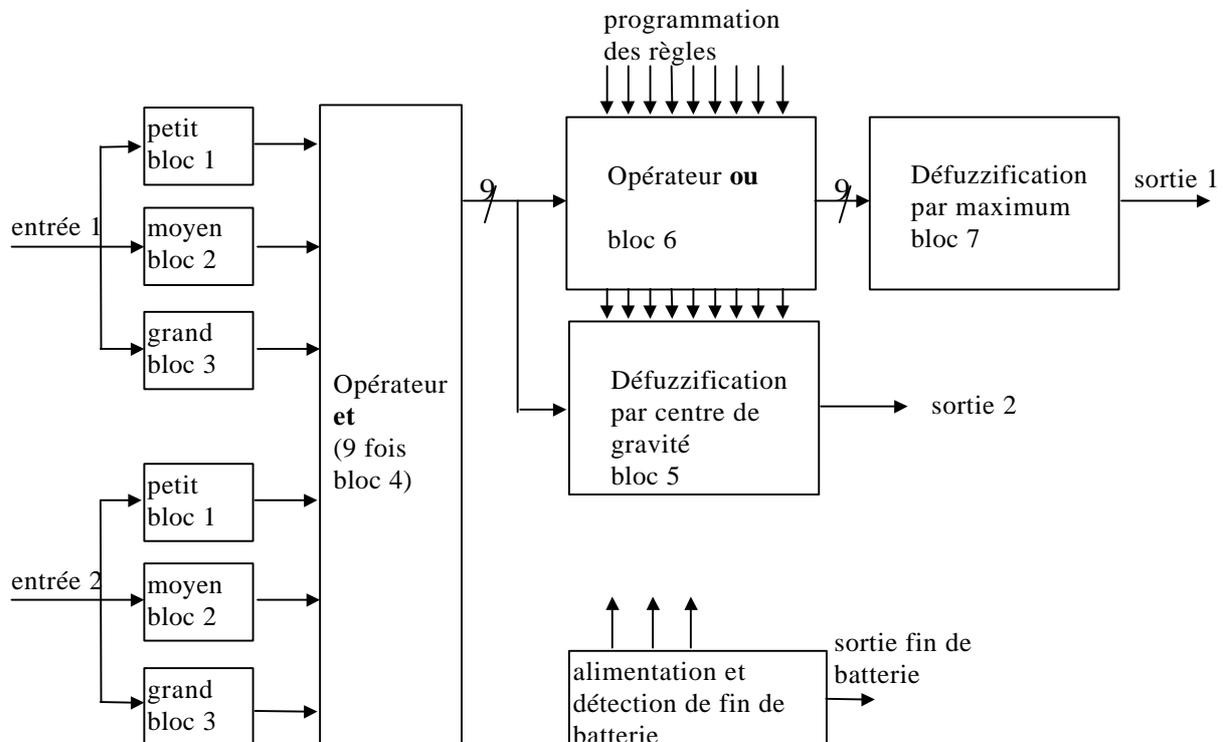


Figure 12

Les différentes opérations (fuzzification, min, max, défuzzification) sont, pour la plupart, réalisées à l'aide de miroirs de courants. L'exemple suivant illustre ce principe pour l'opérateur **et** (fonction **minimum**).

Les miroirs de courants présentent la particularité de ne fonctionner que dans un seul quadrant. On peut tirer profit cette propriété afin de réaliser des fonctions discontinues. Afin de simplifier les schémas, le miroir réalisé avec des NMOS sera symbolisé par un opérateur N et celui de type PMOS par un opérateur P (cf. figure 13). Ces opérateurs présentent les caractéristiques suivantes :

- Opérateur N
 
$$I_s = -I_e \quad \text{si } I_e > 0$$

$$I_s = 0 \quad \text{si } I_e < 0$$
- Opérateur P

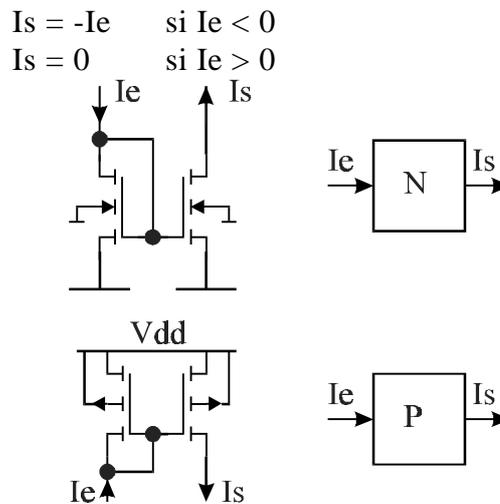


Figure 13

L'opérateur **et** est réalisé à l'aide d'une combinaison d'opérateurs P et N (cf. figure 14).

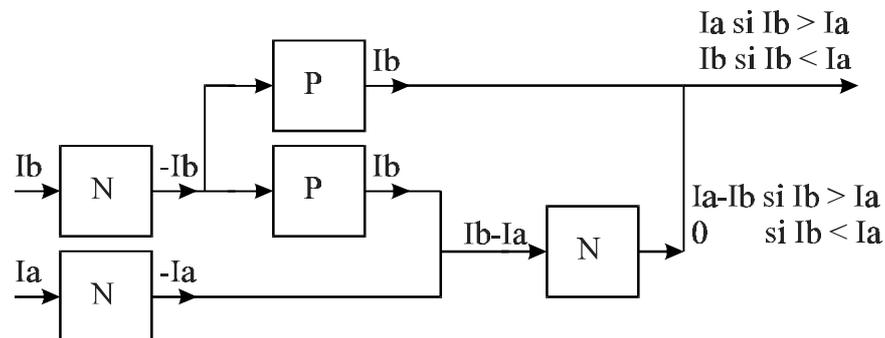


Figure 14

L'opérateur **ou**, la fuzzification et la défuzzification sont réalisés à l'aide de structures semblables.

L'ensemble utilise quelques milliers de transistors sur une base « gate-array » et la fréquence maximale de travail prévue se situe aux alentours de 10 kHz. Il s'agit donc d'une solution compacte, rapide et économique face à ce qui peut être fait en technique digitale.

### 13.3

*Exemple de réalisation : version MOS en résistances variables*

Cette réalisation faite en collaboration étroite avec le CSEM présente deux entrées codées à l'aide de 5 intervalles flous. Les fonctions d'appartenance sont en forme de cloche. Les 25 règles d'inférences sont mémorisées sous forme digitale dans le processeur. L'opérateur **et** est réalisé par la mise en

série de MOS travaillant en résistances variables. L'opération ainsi réalisée est l'inverse de la somme des inverses. L'opérateur **ou** est simplement réalisé par la somme des courants. La défuzzification s'effectue par le calcul du centre de gravité sur des fonctions d'appartenances en raies. Pour plus de détails, le lecteur voudra bien se référer aux divers articles du CSEM traitant de ce sujet.

Cette réalisation fait actuellement l'objet d'un travail de diplôme à l'EISI.

Comme dans l'exemple précédent, cette réalisation présente l'avantage de la simplicité, du faible coût et de la rapidité face à un système digital.

## 14. CONCLUSIONS

Sans grand risque de se tromper on peut affirmer que la logique floue va prendre de plus en plus de place dans le domaine du réglage. Cette affirmation est motivée par le fait que ce principe ne nécessite pas de modèle du système à régler et qu'il utilise les connaissances de l'opérateur (expert). Dans le développement, l'approche digitale est, évidemment, la plus souple à mettre en oeuvre. Son coût, son encombrement et sa consommation peuvent devenir un handicap pour certaines applications en production. Pour ces applications l'approche analogique apporte une solution intéressante.

## 15. SOURCES

1. H. Bühler Réglage par la logique floue PPR 1994
2. J. Décosterd (EICN), P. Marchal (CSEM) et J.-F. Perotto (CSEM) séminaire « logique floue » EISI 1993
3. O. Landolt, P. Marchal et E. Vittoz (CSEM) : divers articles dont certains en cours de publication
4. MSM No. 4 1992