Chapitre Deux

LA LOGIQUE FLOUE

LA LOGIQUE FLOUE

2.1. Introduction

La conception et la mise en œuvre de systèmes d'informations avancées, de logiciels d'aide à la décision, de postes de supervisions de procédés industriels (avec les problèmes de communications homme machine qui les accompagnent) se caractérisent par la constante manipulation d'informations dont beaucoup sont subjectives, imprécises, vagues, incertaines. Mener à bien cette intégration de l'homme dans son environnement informatique, exploiter correctement les savoirs, automatiser certaines taches, tout cela requiert la formalisation et la mécanisation de méthodes humaines pour de raisonnement empiriques ou naturel, la rationalisation de procédures de choix.

De façon générale, on est confronté à la nécessité de modéliser la connaissance, problématique qui se trouve en rupture avec la tradition des sciences dites objectives, lesquelles se préoccupent essentiellement de la modélisation de l'univers physique.

Les problèmes de représentation et d'utilisation des connaissances sont au centre d'une discipline scientifique relativement nouvelle et en tout cas controversée, qu'on appelle l'intelligence artificielle. Cette discipline a eu un impact limité, jusqu'a une date récente, sur les applications industrielles, parce qu'elle a mis l'accent, de façon exclusive, sur le traitement symbolique de la connaissance, par opposition à la modélisation numérique utilisée traditionnellement dans les sciences de l'ingénieur. Plus récemment, on a assisté à un retour du numérique dans ces problèmes d'intelligence artificielle, avec les réseaux neuro-mimétiques et la logique floue. Alors que les réseaux neuro-mimétiques proposent une approche implicite de type «boite noir» de la représentation des connaissances, très analogue à la démarche de l'identification des systèmes en automatique, la logique floue est plus conforme à l'intelligence artificielle symbolique, qui met en avant la notion de raisonnement, et ou les connaissances sont codées explicitement. Néanmoins, la logique floue permet de faire le lien entre modélisation numérique et modélisation symbolique, ce qui a permis des développements industriels spectaculaires à partir d'algorithmes très simple de traduction de connaissances symboliques en entité numérique et inversement. La théorie des ensembles flous a également donné naissance à un traitement original de l'incertitude, fondée sur l'idée d'ordre, et qui permet de formaliser le traitement de l'ignorance partielle et de l'inconsistance dans les systèmes d'informations avancés. Les ensembles flous ont également eu un impact sur les techniques de classification automatique, et ont contribué à un certains renouvellement des approches existantes de l'aide à la décision.

2.1.1. Exemple introductif

Afin de mieux appréhender la problématique envisagé, nous proposons, avant l'étude rigoureuse des divers aspects relatifs à la commande floue, d'examiner l'exemple qui suit dont le but est de lui donner une idée de la commande floue en vue de lui permettre de mieux comprendre l'intérêt pratique des développement mathématiques présentés dans la suite de ce volume.[11]

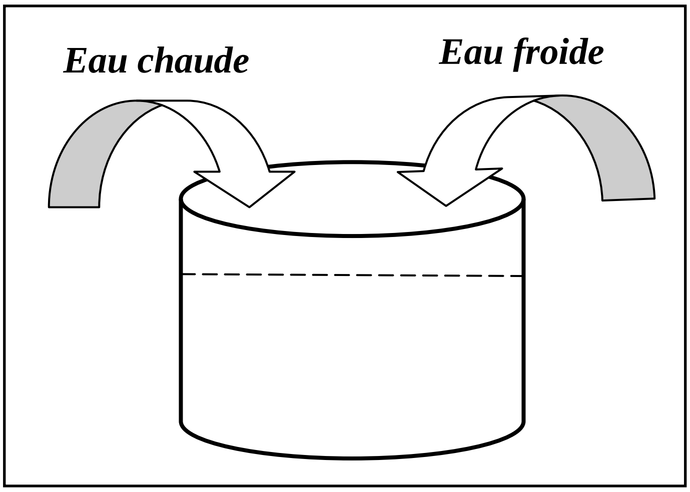


Figure 2.1– Exemple introductif

Considérons une personne désirant compléter le niveau d'un réservoir contenant de l'eau à une température donnée de façon à emplir ce réservoir d'eau à une température souhaitée T à l'aide d'un mitigeur. Dans un premier temps, nous considérons trois températures possibles de l'eau d'alimentation : froide TF, chaude TC et tiède TT. L'eau du réservoir pourra être appréciée comme froide, tiède ou chaude avec une certaine marge d'incertitude si on ne dispose pas d'appareils de mesure de température.

La commande du mitigeur sera simple :

- Si l'eau du réservoir est froide, mettre de l'eau chaude.
- Si l'eau du réservoir est tiède, mettre de l'eau tiède.
- Si l'eau du réservoir est chaude, mettre de l'eau froide.

Sans appareils de mesure précis, on a peut-être donné une information plus nuancée, telle que « tiède et plutôt chaude » et également définir des intervalles de température pour lesquels il n'y a pas d'incertitude comme « l'eau est nettement chaude » ou « réellement froide » et des zones pour lesquelles on peut hésiter.

Dans ce cas, on peut définir une quantification, éventuelle grossière, du degrés d'appartenance µ(T) à l'une des classes « tiède », « chaud » et «froid ». la valeur 1 correspond à une appartenance 100% et 0 à la non appartenance (Figure 2.2).

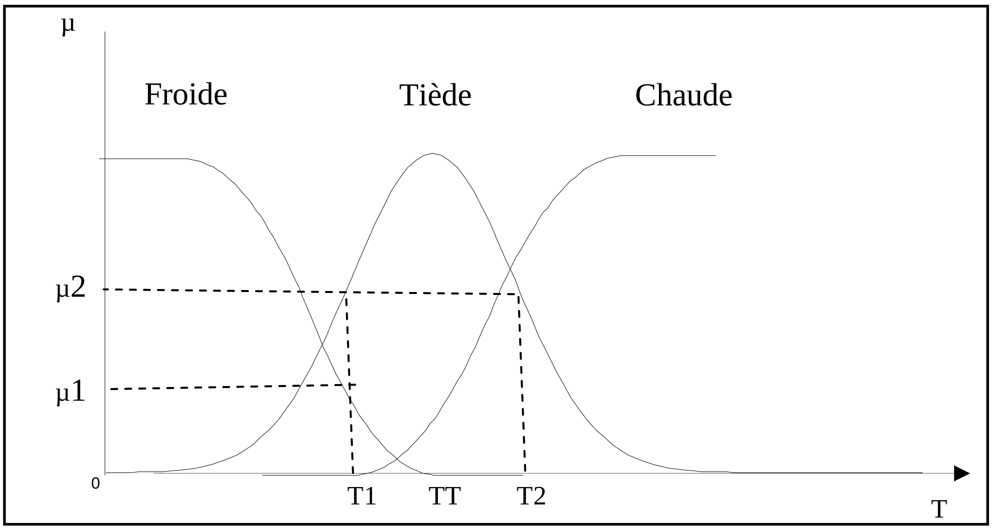


Figure 2.2– Fonction d'appartenance, variable et terme

Dans l'exemple considéré :

- Pour T=TT, l'eau est tiède ;
- Pour T<T1, l'eau est froide ;
- Pour T>T2, l'eau est chaude ;
- Pour T $\in$] T1, TT], on hésite entre froid et tiède.
- Pour T $\in$] TT, T2], on hésite entre tiède et chaud.

Un autre choix de répartition en classes aurait pu être défini suivant notre sensibilité, de même le nombre de classes aurait pu être différent. Cette phase de répartition en classe de façon à pouvoir par la suite associer à chaque classe une décision ou commande donnée : «si l'eau du réservoir est froide ajouter de l'eau chaude » correspond au principe de base de la détermination d'une commande floue. Pour simplifier nous dirons que la « fuzzification » est « l'opération » qui à une valeur donner à la variable associé un sous ensemble flou particulier.

Sa mise en œuvre conduit «l'expert » à proposer « une répartition en classes » des «caractéristiques» des propriétés considérées.

Si l'on est maintenant capable de faire une mesure, même imprécise de la valeur μR de la température de l'eau du réservoir, il faut pouvoir en déduire une valeur de la température d'alimentation. Par exemple, on peut prendre la décision qui correspond à la classe d'appartenance la plus probable.

2.1.2. Historique

Nous venons de voir que la logique floue sert à représenter des connaissances incertaines et imprécises. La commande floue sert à prendre une décision même si l'on ne peut pas estimer les entrées/sorties qu'à partir de prédicats vagues ou lorsque ses entrée/sorties sont entachées d'erreurs que l'on peut évaluer que grossièrement.

On conçoit l'intérêt de faire entrer l'approche floue dans la régulation ou l'asservissement des processus industriels, pour lesquels les informations disponibles sont souvent imprécises, incertaines et parfois qualitatives, dans des boucles de régulation parfois incomplètes. Le savoir faire de l'opérateur , constitué entre autres souvent des règles simples, lui permet de conduire chaque machine plus correctement parfois qu'un algorithme classique. Les prémisses de la logique floue sont apparues avant les années 1940, avec les premières approches, par des chercheurs américains, du concept d'incertitude. Il a fallu attendre 1965, pour que le concept de sous ensemble floue soit proposé par ***L. A. Zadeh***, automaticien de réputation internationale, professeur à l'université de Berkeley en Californie, qui a contribué à la modélisation de phénomène sous forme floue, en vue de pallier les limitations dues aux incertitudes des modèles classiques à équation différentielle [7]. En 1974, ***M. Mamdani*** expérimentait la théorie énoncée par ***Zadeh*** sur une chaudière à vapeur, matériel dont on connaît la complexité, introduisant ainsi la commande floue dans la régulation d'un processus industriel. Plusieurs applications ont alors vu le jour en Europe, pour des systèmes parfois très complexes, telle la régulation de fours de cimenterie réalisée par la société F. L. Smidt-Fuller. Grâce au chercheur japonais ***M.Sugneo***, la logique floue était introduite au Japon dès 1985[8]. Les sociétés japonaises compirent l'avantage à la fois technique et commercial de la logique floue:

- facilité d'implantation;
- solution de problèmes multivariables complexes;
- robustesse vis à vis des incertitudes;
- possibilité d'intégration du savoir de l'expert.

2.1.3. Le concept de la logique floue

Ce rapport permet de considérer des classes d'objets dont les frontières ne sont pas clairement déterminées, par l'introduction d'une fonction caractéristique (fonctions d'appartenance des objets à la classe) prenant des valeurs courantes entre 0 et1, contrairement aux ensembles «booléens», dont la fonction caractéristique ne prend que deux valeurs possibles 0 et 1.

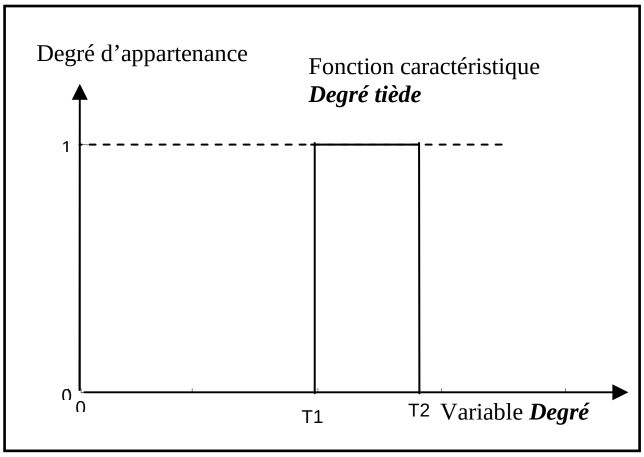
Ces ensembles flous ont le grand avantage de constituer une représentation mathématiques de labels linguistiques largement utilisés dans l'expression de connaissances expertes, qualitatives et manipulées dans le raisonnement approché qui sera fait à partir de cette connaissance. Ils apparaissent donc comme un moyen de réaliser l'interface entre l'information numérique (quantitative) et l'information symbolique (linguistique, qualitative).

2.2. Les Bases de la commande flou

Nous avons vu à travers l'exemple introductif l'intérêt et le principe de la commande et de la logique floues. Cette partie du cite va nous permettre de formaliser mathématiquement ces concepts. On verra tout d'abord comment fuzzifier (c'est à dire passer d'une variable réelle à une variable floue), puis comment construire un schéma de commande par la définition de règles d'inférence.

2.2.1. Les fonctions d'appartenances

Un ensemble flou est définit par sa fonction d'appartenance, qui correspondant à la notion de la fonction caractéristique en logique classique. Supposons que nous voulions définir l'ensemble des personnes de taille moyenne. En logique classique, nous conviendrons par exemple que les personnes de taille moyenne sont celles dont la taille est comprise entre 1.60m et 1.80m.la fonction caractéristique de l'ensemble donne « 0 » pour les tailles hors de l'intervalle [1.6m ;1.8m] et « 1 » dans cet intervalle. L'ensemble flou des personnes de taille moyenne sera définit par une fonction d'appartenance qui diffère d'une fonction caractéristique par le fait qu'elle peut prendre n'importe quelle valeur dans l'intervalle [0,1] a chaque taille possible correspondra un degré d'appartenance à l'ensemble des tailles moyennes compris entre 0 et 1.



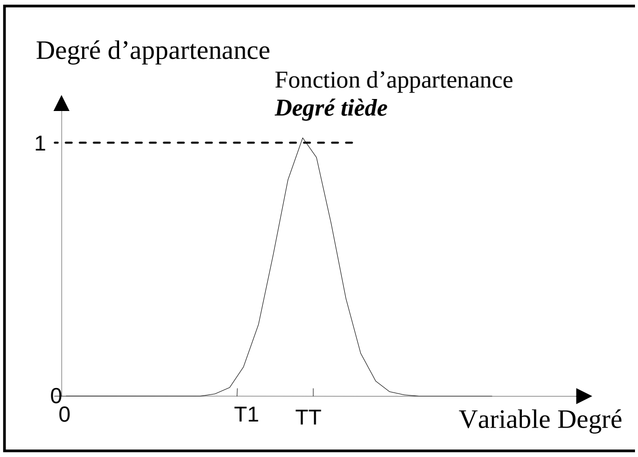


Figure 2.3– Fonction caractéristique **Figure 2.4–** Fonction d'appartenance

Plusieurs ensembles flous peuvent être définis sur le même variable, par exemple les ensembles ***taille petite***, ***taille moyenne***, ***taille grande***

Cet exemple montre la gradualité que permet d'introduire la logique floue. Une personne de 1.60m appartient à m'ensemble ***taille moyenne*** avec un degré de 0.7 et à l'ensemble ***taille petite*** avec un degré de 0.3. En logique classique le passage de petite à moyenne serait brusque [9].

Le variable (par exemple introductif) ainsi que les termes « tiède », « chaud » et «froid ». définis par les fonctions d'appartenance portent les noms de variable linguistique et de termes linguistiques.

Définition : Un ensemble flou A dans X est défini par une fonction d'appartenance qui associe à chaque élément x de X, le degré $\mu_A(x)$, compris entre 0 et 1 avec lequel x appartient à A [10]. Donc un ensemble flou peut être représenté par un ensemble de paires ordonnées :

$$A = \{(x, \mu_A(x)) / x \in X\} \tag{2.1}$$

Tel que : $\mu_A(x)$ La fonction d'appartenance d'un variable x.

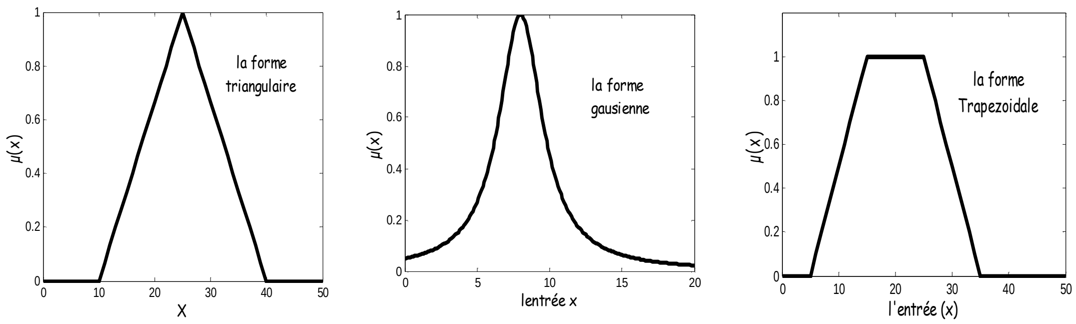
X: est appelé l'univers de discours il peut contenir des valeurs continues ou discrètes.

Notation:

Si X est discret :
$$A = \sum \mu_A(x)/x \tag{2.2}$$

Si X est continu :
$$A = \int \mu_A(x)/x \tag{2.3}$$

Tel que : X est L'univers de discours ou l'intervalle de variation de variable x



Forme triangulaire Forme gaussienne Forme trapézoïdale

Figure 2.5– Les différentes formes des fonctions d'appartenances

2.2.2. Les opérateurs de la logique flous

Il est important de savoir composer entre les divers prédicats et leurs fonctions d'appartenance comme dans l'exemple «l'air est froid et le vent est fort» ou dans «si l'air est froid ou si le vent est fort» il faut fermer la porte. Il apparaît deux types de composition ET et OU auxquels il faut ajouter la négation. Notons x et y les variables linguistiques caractérisant la température de l'air et la force du vent et $\mu_A(x)$, $\mu_B(y)$, $\mu_E(z)$, $\mu_O(z)$, $\mu_C(z)$ avec z={x,y}, les fonctions d'appartenance associées aux propriétés respectives «l'air est froid», «le vent est fort», «l'air est froid et le vent est fort», «l'air est froid ou le vent est fort», «l'air n'est pas froid» [12].

a) Opérateur NON:

La propriété «l'air n'est pas froid» peut être caractérisée de façon évidente par la fonction d'appartenance

$$\mu_C(z)=1-\mu_A(x) \tag{2.4}$$

La figure suivante met en évidence cette relation:

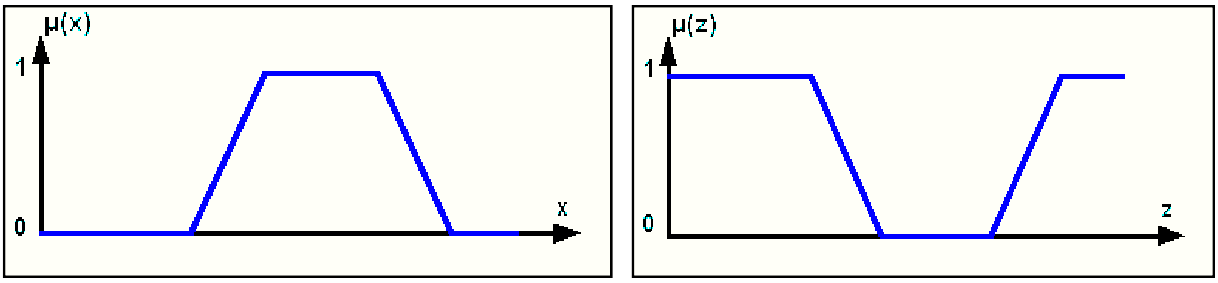


Figure 2.6– Opérateur NON

A noter qu'il s'agit de l'opérateur NON, appelée aussi «complément», «négation» ou «inverse».

b) Opérateur ET:

La solution la plus simple et la plus utilisée pour caractériser la satisfaction simultanée de deux propriétés est de poser

$$\mu_E(z)=\mu_A \text{ et } B(z)= \min\{\mu_A(x),\mu_B(y)\}. \quad (2.5)$$

On parle alors d'opérateur minimum.

Cette opération est représentée à suivante Comme on le voit, il est possible que la fonction d'appartenance résultante $\mu_E(z)$ n'atteigne pas la valeur 1.

On peut facilement vérifier que l'opérateur minimum est commutatif, c'est à dire qu'il est possible d'invertir $\mu_A(x)$ et $\mu_B(y)$ sans que la résultat change.

Cet opérateur peut être appliqué à plus de deux ensembles. Dans ce cas s'applique le théorème d'associativité.

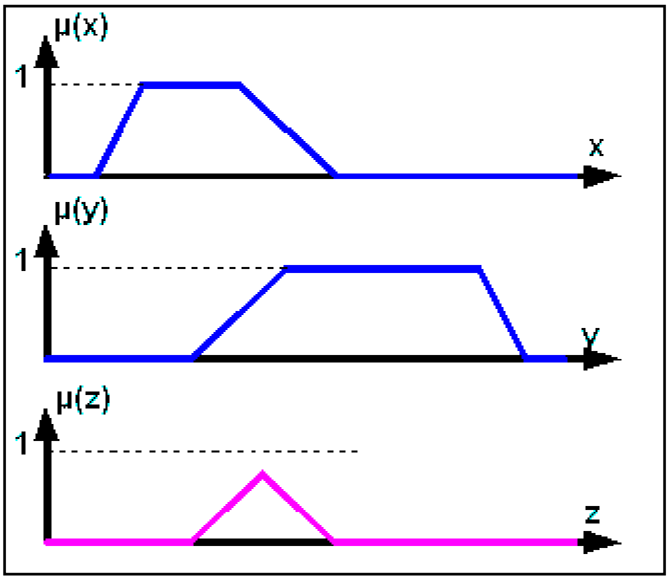


Figure 2.7– Opérateur ET

A noter qu'il s'agit de l'opérateur ET, appelée aussi «intersection».

c) Opérateur OU:

La réalisation de l'opérateur ou au niveau de la logique floue se fait en général par la formation du maximum, appliquée aux fonctions d'appartenance $\mu_A(x)$ et $\mu_B(y)$ des deux ensembles A et B. On a donc l'opérateur maximum.

$$\mu_O(z)=\mu_A \text{ ou } B(z)=\max\{\mu_A(x),\mu_B(y)\} \quad (2.6)$$

La figure 2.8 montre cette opération. A noter qu'il est possible que la fonction d'appartenance résultante $\mu_O(z)$ atteigne deux fois la valeur 1.

Evidemment, l'opérateur maximum est aussi commutatif et associatif.

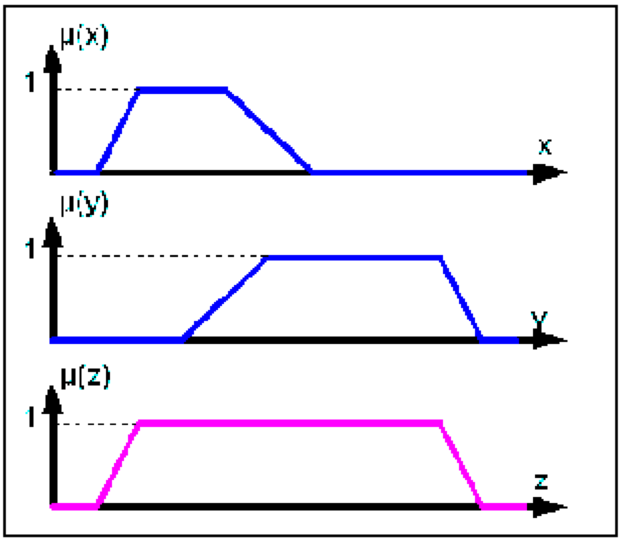


Figure 2.8– Opérateur OU

Dans ce paragraphe, on a introduit la formation du minimum et du maximum pour réaliser les opérateurs ET et OU. Dans la plupart des cas, ces opérateurs donnent des résultats convenables, surtout pour le réglage et la commande par logique floue. Cependant, dans certaines circonstances, il peut être judicieux d'utiliser d'autres opérateurs, soit pour simplifier le traitement numérique, soit pour mieux tenir compte des opérations floues.

d) Opérateurs ET et OU, réalisés par opérateurs arithmétiques

Souvent, l'opérateur ET est réalisé par la formation du produit appliqué aux fonctions d'appartenance, selon la relation

$$\mu_E(z)=\mu_A \text{ et } B(z)=\mu_A(x).\mu_B(y) \quad (2.7)$$

Il s'agit de l'opérateur produit.

Le résultat de cette opération est représentée à la figure 2.9. La fonction d'appartenance résultante est toujours est inférieure ou égale à 1. Elle reste donc à l'intérieur de l'intervalle défini par μ€ [0,1]. La règle de calcul (2.7) peut être étendue à plus de deux termes dans le produit lorsqu'il faut combiner trois ou plusieurs ensembles. L'opérateur produit est souvent utilisé dans le domaine de réglage et de commande par logique floue comme alternative à l'opérateur minimum.

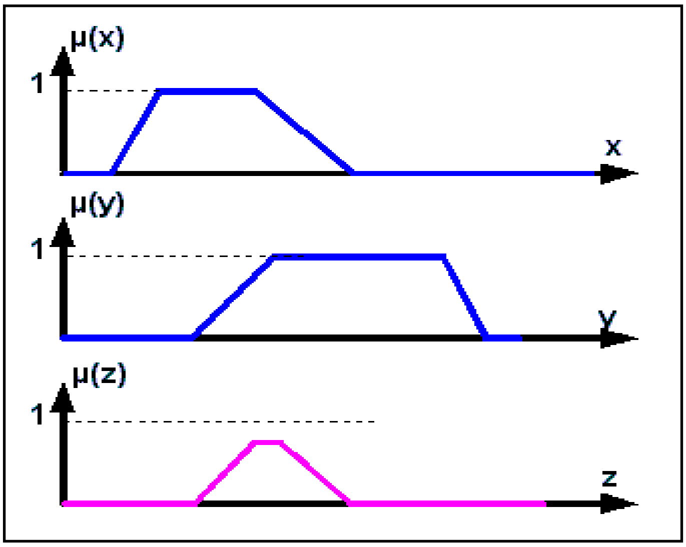


Figure 2.9– Opérateur ET réalisé par la fonction produit

Par analogie, on peut réaliser l'opérateur OU par la formation de la somme des fonctions d'appartenances ou plus précisément par la valeur moyenne, à savoir:

$$\mu_O(z)=\mu_A \text{ ou } B(z)= 1/2\ [\mu_A(x)+\mu_B(y)] \qquad (2.8)$$

On parle alors de l'opérateur somme.

La figure 2.10 montre l'effet de cet opérateur. La somme est divisé par 2. En effet, il est fort possible que la somme $[\mu_A(x)+\mu_B(y)]$ dépasse le domaine admissible [0,1]. Afin que cette somme reste dans le domaine défini, on peut l'écrêter ou la normaliser, comme effectué dans la définition 2.3.2. Dans ce cas aussi, il est possible d'étendre la règle de calcul (2.8) à plusieurs termes. il faut alors diviser la somme par le nombre de termes, afin d'obtenir une normalisation simple.

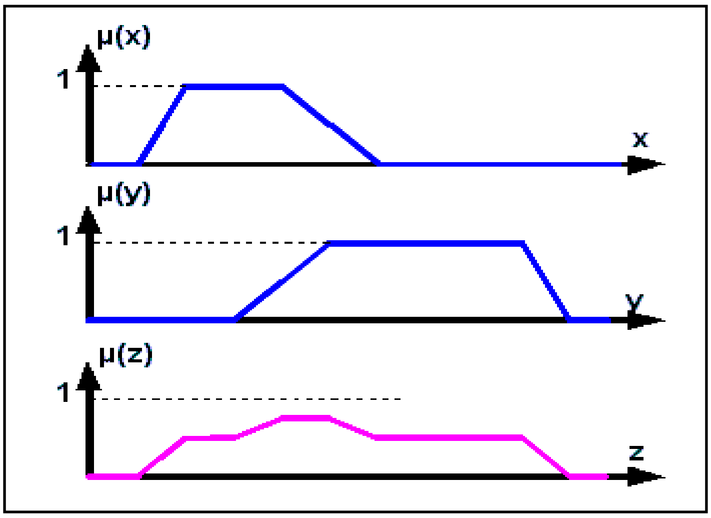


Figure 2.10– Opérateur OU réalisé par la fonction de la somme

e) Opérateurs ET flou et Ou flou

Les opérateurs Et flou et OU flou sont des opérateurs combinés entre l'opérateur minimum et la moyenne arithmétique.

L'opérateur Et flou est défini par:

$$\mu_E(z)=\mu_{A\ et\ B}(z)=ß\ \min\left[\mu_A(x),\mu_B(y)\right] + \left[(1-ß)/2\right]\left[\mu_A(x)+\mu_B(y)\right] \quad (2.9)$$

et l'opérateur Ou flou par:

$$\mu O(z)=\mu_{A\ ou\ B}(z)=ß\ \max\left[\mu_A(x),\mu_B(y)\right] + \left[(1-ß)/2\right]\left[\mu_A(x)+\mu_B(y)\right] \quad (2.10)$$

Avec le facteur *ß=1Є [0,1]*, il est possible de pondérer l'influence des deux termes. Pour *ß=1*, on aboutit respectivement à l'opérateur minimal ou maximal. Par contre, pour *ß=0*, on obtient pour les deux opérateurs la moyenne arithmétique correspondant à l'opérateur somme selon (2.8). Dans ce cas, le ET flou et le OU flou se confondent. On peut étendre les deux opérateurs ET flou et le OU flou à trois ou à plusieurs termes. La somme qui apparaît entre crochets doit alors être divisé par le nombre de termes de la somme. La figure 2.11 représente l'opérateur ET flou et montre l'influence du facteur *ß* sur l'allure de la fonction d'appartenance $\mu_E(z)$

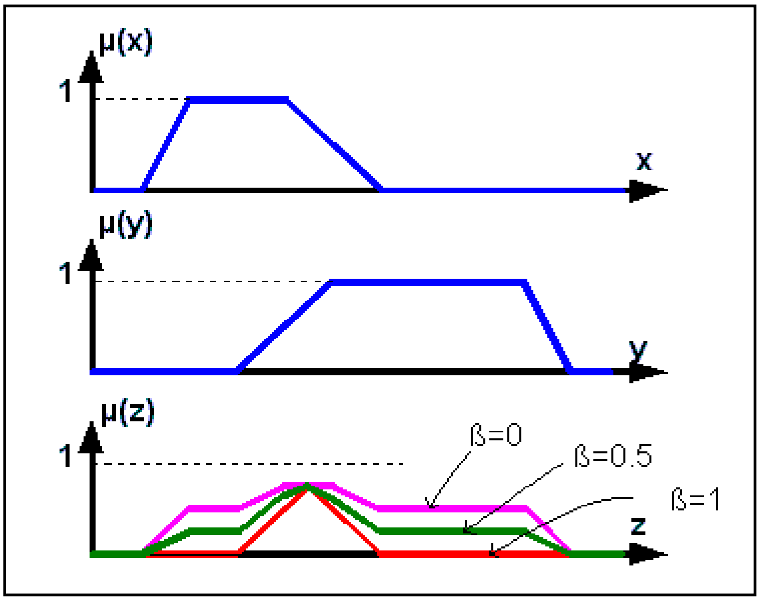


Figure 2.11– Opérateur ET flou réalisé par la relation (2.9)

L'influence du facteur ß sur la fonction d'appartenance résultante pour l'opérateur OU flou est mise en évidence par la figure 2.3.4

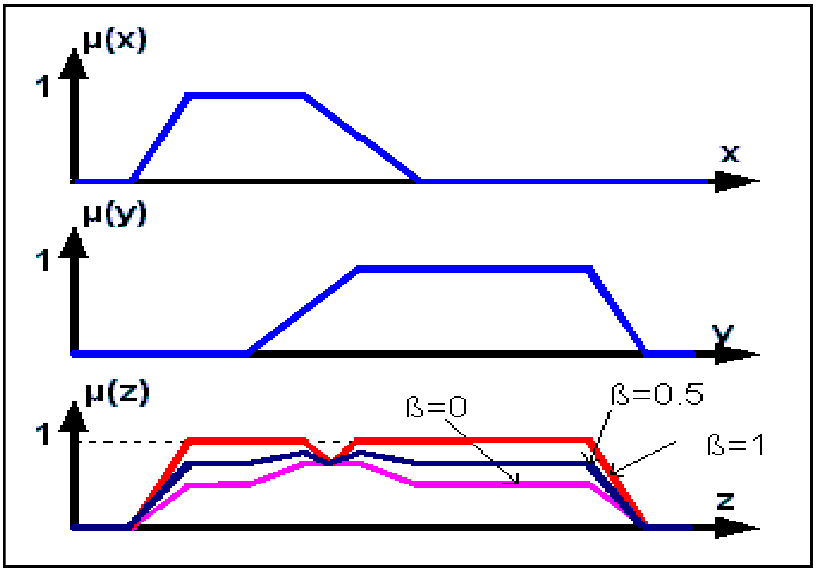


Figure 2.12– Opérateur OU flou réalisé par la relation (2.10)

f) Opérateurs min-max et opérateur *ß*

L'opérateur min-max est défini par la combinaison des opérateurs minimum et maximum, selon

$$\mu(z)=ß \min [\mu A(x),\mu B(y)] + (1-ß) \max [\mu A(x),\mu B(y)] \qquad (2.11)$$

Le facteur *ß*Є *[0,1]*, permet de pondérer les deux opérateurs.

Pour *ß=1*, on obtient l'opérateur ET, réalisé par la formation du minimum, tandis que pour *ß=0*, on aboutit à l'opérateur OU, réalisé par la formation du maximum.

Par contre, *ß=0,5* conduit à l'opérateur Ou, réalisé par la formation de la somme.

La figure 2.13 montre de l'effet de l'opérateur min-max en fonction du facteur *ß*. On constate bien la grande variation de l'allure de la fonction d'appartenance résultante

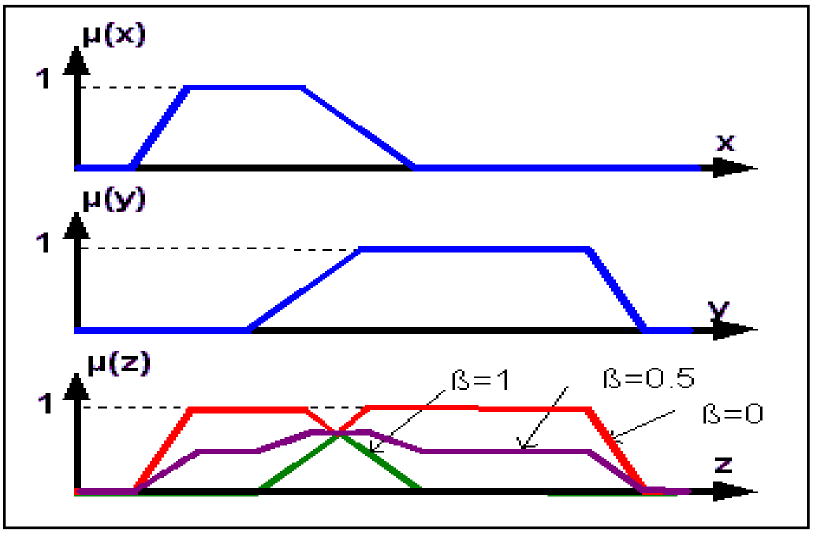


Figure 2.13– Opérateur min-max réalisé par la relation (2.11)

2.2.3. Univers de discours et classes d'appartenance

La notion d'univers de discours se conçoit aisément: reprenons notre concept de température: l'utilisateur pourra décrire la variable «température» par un certain nombre de mots: par exemple «chaud», «froid», «tiède», ou «très chaud», «assez chaud», «tiède», «assez froid», «très froid». Pour chacun de ces prédicats, on pourra donner une fonction d'appartenance. L'univers de discours sera considéré comme le domaine de fonctionnement du processus.

Les problèmes qui peuvent se poser sont:

- Combien de prédicats sont nécessaire à la commande et comment les choisir.?
- est-il nécessaire de choisir chaque prédicat?

En général, on peut donner un nombre de règles de commande important, mais l'intérêt de la commande floue est que seul un petit nombre de règles est nécessaire. Le nombre de prédicats dépend essentiellement de la manière dont l'expert peut décrire le processus et de la précision souhaitée. Par exemple, en commande floue classique, 5 prédicats (grand, très grand, moyen, petit, très petit) est un bon compromis; parfois trois peuvent suffire (dans la cas extrêmes, on peur aller jusqu'à 7). Les trois prédicats «petit», «moyen», «grand» ou «négatif», «positif», et «proche de zéro» servent à la régulation; les autres prédicats servent à la poursuite. La commande floue permet d'assurer une transition harmonieuse entre les deux modes de fonctionnement.

L'univers de discours d'une variable couvrira l'ensemble des valeurs prises par cette variable. En pratique, l'intersection entre deux prédicats consécutifs est non nulle, de façon à pouvoir exercer une pondération sur la commande. Il en résulte un *chevauchement* des variables qui doit être suffisant pour permettre une description continue des variables mais pas trop important pour limiter l'imprécision. Il est notamment préférable d'éviter que les fonctions d'appartenance de deux prédicats voisins soient simultanément égale à 1.

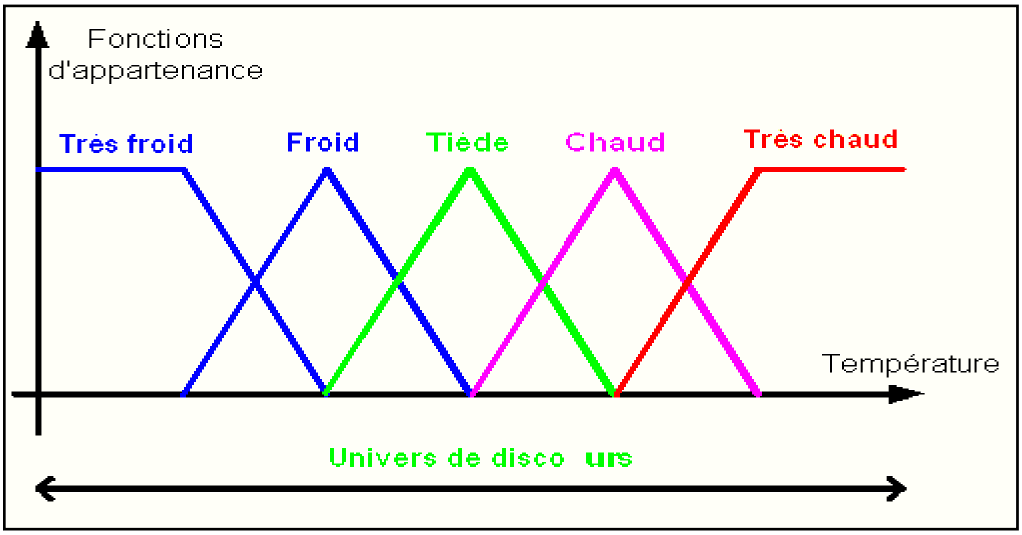


Figure 2.14– Univers de discours de l'exemple introductif

2.2.4. Schéma de la commande floue

La mise en oeuvre d'une commande floue fait apparaître trois grands modules.

Le premier module traite les entrées du système (valeurs réglant). on définit tout d'abord un univers de discours, un partitionnement de cet univers en classes pour chaque entrée, et des fonctions d'appartenance pour chacune de ces entrées (par exemple pression grande, petite, faible et changement d'écart mesure consigne de débit de matériau sortant d'une trémie très élevé, élevé, moyen, négatif, très négatif). La première étape, appelée fuzzification, consiste à attribuer à la valeur réelle de chaque entrée, au temps t, sa fonction d'appartenance à chacune des classes préalablement définies, donc à transformer l'entrée réelle en un sous ensemble floue.

Le deuxième module consiste en l'application de règles de type «si l'écart de température est grand, diminuer le débit du fuel». Ces règles vont, comme dans l'exemple introductif, permettre de passer d'un degrés d'appartenance d'une grandeur réglante au degrés d'appartenance d'une commande. Ce module est constitué d'une base de règles et d'un moteur d'inférence qui permet le calcul.

Le troisième et le dernier module décrit l'étape de défuzzification qui est la transformation inverse de la première. Il permet de passer d'un degrés d'appartenance d'une commande à la détermination de la valeur à donner à cette commande.

Le schéma de commande est donné par la figure (2.4).

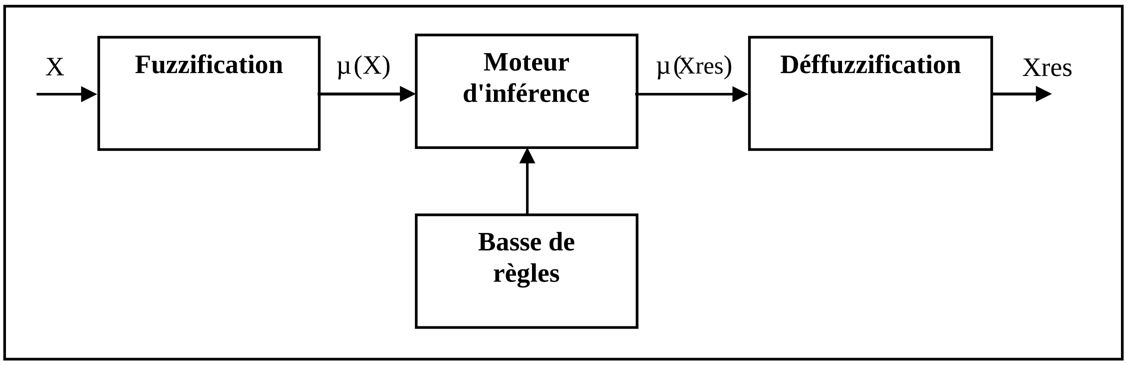


Figure 2.15– Schéma de la commande floue

x représente le vecteur des entrées, x_{RES} celui des commandes, μ(x) et μ(x_{RES}) les fonctions d'appartenances correspondantes

2.3. Structure d'un commande Floue

La structure conventionnelle d'une commande floue est présentée par figure 3.1. Elle est composée de quatre blocs distincts dont les définitions son données ci-dessous .

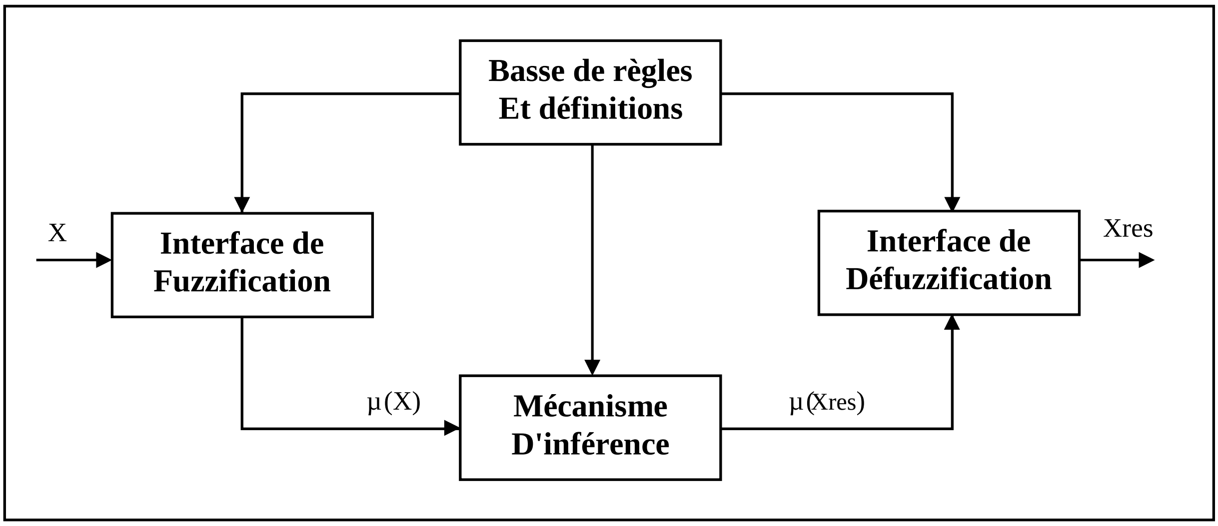


Figure 2.16– Structure d'une commande floue

2.3.1. Bases de règles et définitions

On regroupe dans ce bloc, d'existence virtuelle, l'ensemble des définitions utilisées dans la commande floue (univers de discours, partitions floue, choix des opérateurs...), ainsi que la base de règles «Si...alors...» de la stratégie de commande de l'expert.

<u>Partition floue:</u>

La création et l'utilisation d'une base de règles nécessitent l'existence, pour chaque univers de discours considéré, de sous ensemble floues particuliers. La définition de ces sous-ensembles floues fait l'objet de la partition floue.

La partition floue d'un univers de discours U consiste à définir n sous-ensembles flous Fi de façon à recouvrir U. C'est à dire que pour tout élément x de U, il faut assurer une appartenance minimale ϵ à l'union de Fi.

$$\bigcup F_{\phi} \supseteq U_{\delta} = \{x \in U; \mu_{U_{\delta}}(x) = \varepsilon\} \qquad (2.12)$$

La condition (2.12) se traduit au niveau des fonctions d'appartenance par la condition

$$\forall x \in U; \mu_{F_{\phi}}(x) \vee \ldots \vee \mu_{F_{\varphi}}(x) \geq \varepsilon \qquad (2.13)$$

ou v est un opérateur d'union (appelé également "ou"). Comme la fonction max minore toutes les fonctions utilisées comme opérateurs d'union, pour assurer une partition floue de niveau ϵ, il faut et il suffit que tout élément x de U possède un degrés d'appartenance à l'union des Fi avec pour opérateur d'union la fonction max, supérieure ou égal à ε.

L'ensemble de sous-ensembles flous à définir dans une partition d'un univers de discours est fixé par l'expert. Plus ce nombre est important et plus on définit de classes sur cet univers, ce qui permet d'augmenter la sensibilité de la commande floues.

Bases de règles

Une fois la partition des univers de discours réalisée, il est possible de définir la base de règle. Celle-ci caractérise les relations entre les classes d'événements possibles en entrée et les commandes correspondantes.

Par conséquent, si l'on considère n univers de discours U_i pour les prémisses des règles floues et si pour chaque univers U_i on définit une partition en mi sous ensembles flous, le nombre maximum de règles r_{max} est de:

$$r_{\max} = \prod_{i=1}^{n} m_i \qquad (2.14)$$

Le nombre de règles définis par l'expert peut-être inférieure à r_{max}. C'est le cas, en particulier, s'il existe des configurations des sous ensembles flous impossibles à obtenir pour le système. Par exemple, dans le cas d'un freinage automatique des véhicules, la configuration «Vitesse importante ET Distance de l'obstacle nulle» n'est pas considérée.

De plus le nombre de sous ensemble flous définissant la partition de l'univers de discours de la commande n'est pas forcément égale au nombre de règles. En effet, il est possible de considérer des configurations différentes aboutissants à la même conclusion. Par exemple, dans le cas de la commande automatique de freinage, les cas «Vitesse petite et Distance de l'obstacle faible» et «vitesse moyenne Et Distance de l'obstacle importante» amènent à la même conclusion «Freinage faible».

Enfin, on peut remarquer qu'une augmentation de la sensibilité de la commande floue obtenue par une partition plus fine des univers de discours des prémisses aboutit à un accroissement important du nombre de règles à définir par l'expert.

2.3.2. Interface de fuzzification

Les opérateurs utilisés dans la commande floue agissent sur des sous-ensembles flous. Par conséquent, il est nécessaire de transformer les variables non floues provenant du mode extérieur en des sous-ensembles flous. Pour se faire, on utilise un opérateur dit de fuzzification qui associe à une mesure de la variable x_0 une fonction d'appartenance particulière $\mu_{x0}(x)$.

Le choix de l'opérateur de fuzzification dépend de la confiance que l'on accorde aux mesures effectuées. Ainsi si la mesure x_0 est exacte, les sous ensemble floue X_0 doit être représentée par un fait précis. Par conséquent, on utilise comme opérateur de fuzzification la transformation dite de singleton. La fonction d'appartenance du sous-ensemble flous X_0 est alors définie par:

$$\mu_{x_0} : U \to U, \mu_{x_0}(x) = 1 si \quad x = x_0; \quad \mu_{x_0}(x) = 0 \quad si \quad x \neq x_0 \tag{2.15}$$

La Figure 2.17 montre l'aspect de cette fonction d'appartenance

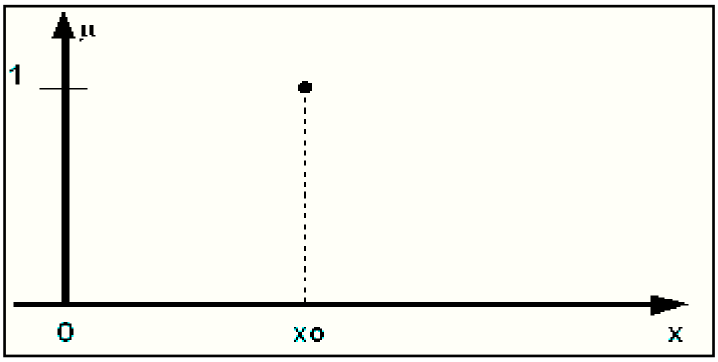


Figure 2.17– Méthode de Fuzzification pour une mesure exacte

Ainsi, le sous ensemble floues X_0 réalisé par cette méthode de fuzzification ne comprend que l'élément x_0. Par contre, si la mesure de la variable est incertaine, par exemple à cause de bruit, le sous ensemble flou X_0 doit être représentée par un fait imprécis. On utilise alors une méthode de fuzzification qui associe à la variable mesurée x_0 une fonction d'appartenance telle que, par exemple:

$$\mu_{x_0}(x) = \max\left\{0;1 - \frac{|x - x_0|}{\varepsilon}\right\} \tag{2.16}$$

La représentation graphique de cette fonction est représentée par la Figure 2.18. Ce sous-ensemble flou comprend donc la mesure x_0 avec une appartenance unité et les valeurs voisines de x_0 avec une appartenance inversement proportionnelle à l'écart avec x_0.

La base du triangle (ε) est fonction de l'importance relative des erreurs de mesures. En effet, plus elles sont importantes, plus la mesure de la variable x_0 devient imprécise, et donc, plus le triangle doit s'élargir.

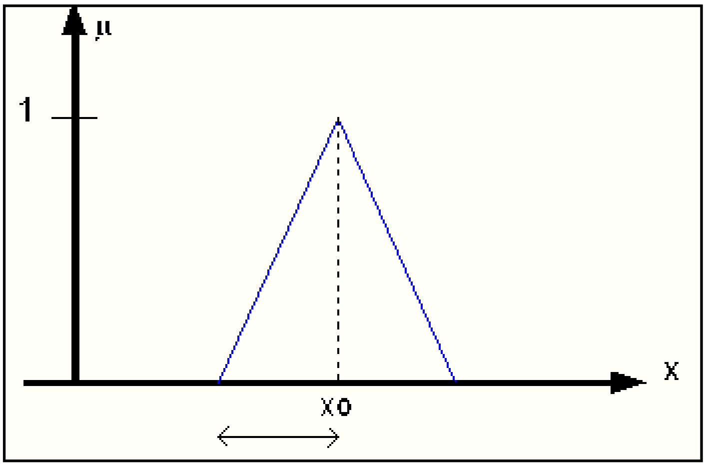


Figure 2.18– Méthode de fuzzification pour une mesure incertaine

2.3.3. Mécanismes d'inférence

A partir de la base de règles (fournie par l'expert) et du sous ensemble flou X0 correspondant à la fuzzification du vecteur de mesure $x_0=[x_{0,1},............ x_{0,n}]^T U$, le mécanisme d'inférence calcule le sous-ensemble flou $\mu(x_0)$ relatif à la commande du système

En général, plusieurs valeurs de variables floues, convenablement défini par des fonctions d'appartenance, sont liées entre elles par des règles, afin de tirer des conclusions. On parle alors de déductions floues. Dans ce contexte, on peut distinguer deux genres de règles d'inférences:

- Inférence avec une seule règle
- Inférence avec plusieurs règles

Dans les inférences de régulateur par logique floue interviennent les opérateurs ET et OU. L'opérateur ET s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur OU lie les différentes règles.

Comme on l'a montré dans le paragraphe opérateurs flous (2.2.2), il existe plusieurs possibilités pour réaliser ces opérateurs qui s'appliquent aux fonctions d'appartenance. On introduit alors la notion de méthode d'inférence. Elle détermine la réalisation des différents opérateurs dans une inférence, permettant ainsi un traitement numérique de cette dernière.

Pour le réglage par logique floue, on utilise en général, une des méthodes suivantes:

- Méthode d'inférence max-min.
- Méthode d'inférence max-prod.
- Méthode d'inférence somme-prod.

2.3.4. Interface de défuzzification

Comme on l'a vu à la section précédente, les méthodes d'inférences fournissent une fonction d'appartenance résultante μRES(z) pour la variable de sortie z. Il s'agit donc d'une information floue. Les actionneurs actuels, utilisées dans les boucles de commande ne s'accompagnent pas de ce genre de décision, il convient de la transformer en une grandeur de commande précise: c'est le but de l'étape de défuzzification. Les méthodes les plus couramment utilisées sont:

- La méthode de centre de gravité
- La méthode de maximum
- La méthode des surfaces
- La méthode des hauteurs

2.5. Conclusion

La logique floue ouvre des possibilités remarquables de codification des connaissances des experts. Cependant, les applications utilisant la logique floue ne sont pas fondamentalement plus performantes. Elles sont tout simplement plus faciles à réaliser et à utiliser : l'utilisation faite par la logique floue d'expressions du langage courant permet au système flou de rester compréhensible pour les personnes non expertes. C'est ainsi que des machines complexes peuvent devenir plus conviviales cause à l'utilisation de la logique floue.

Malheureusement la manipulation de règles non précises peut générer un nombre d'erreurs non négligeable. La mise en place d'un système flou nécessite donc une attention particulière lors de la phase de teste de manière à détecter Les éventuelles aberrations du système.