

## **TP1: Logique floue**

### **Exercice 1 :**

On considère un système de régulation de distance entre véhicules basé sur la logique floue.

On souhaite modéliser la variable d'entrée « Distance\_to\_next\_car », représentant la distance séparant un véhicule du suivant.

1. Définir la variable floue « Distance\_to\_next\_car » sur l'univers de discours  $[0, 30]$ .

2. Créer cinq ensembles flous correspondants aux termes linguistiques suivants :

- ✓ v\_small (très petite distance)
- ✓ small (petite distance)
- ✓ perfect (distance idéale)
- ✓ big (grande distance)
- ✓ v\_big (très grande distance)

3. Utiliser des fonctions d'appartenance trapézoïdales et triangulaires pour modéliser ces ensembles flous, avec les paramètres numériques appropriés.

4. Visualiser les fonctions d'appartenance obtenues.

### **Exercice 2 :**

Dans le cadre d'un système de contrôle de la vitesse basé sur la logique floue, on souhaite modéliser la variable d'entrée « Speed\_change », représentant la variation de la vitesse du

véhicule par rapport à un instant précédent.

1. Définir la variable floue Speed\_change sur l'univers de discours  $[-20, 20]$ .

2. Créer trois ensembles flous correspondant aux termes linguistiques suivants :

- ✓ declining (vitesse en diminution),
- ✓ constant (vitesse stable),
- ✓ growing (vitesse en augmentation).

3. Utiliser des fonctions d'appartenance trapézoïdales et triangulaires pour représenter ces ensembles flous.

4. Tracer les fonctions d'appartenance de la variable Speed\_change

### **Exercice 3 :**

Dans un système de régulation de la vitesse automobile fondé sur la logique floue, on souhaite modéliser la variable de sortie « Acceleration\_Adjust », représentant l'ajustement de l'accélération du véhicule à appliquer selon les conditions de conduite.

1. Définir la variable floue de sortie Acceleration\_Adjust sur l'univers de discours  $[-3,3]$ .

2. Créer cinq ensembles flous correspondant aux termes linguistiques suivants :

- ✓ neg\_big (forte décélération),
- ✓ neg\_small (faible décélération),
- ✓ zero (aucune variation),
- ✓ pos\_small (faible accélération),
- ✓ pos\_big (forte accélération).

3. Utiliser des fonctions d'appartenance trapézoïdales et triangulaires pour modéliser ces ensembles flous.

4. Tracer les fonctions d'appartenance de la variable Acceleration\_Adjust

### **Exercice 4 :**

En se basant sur les variables floues d'entrée (Distance\_to\_next\_car et Speed\_change) et la variable de sortie (Acceleration\_Adjust) définies dans les exercices précédents, on souhaite maintenant établir la base de règles floues et simuler le comportement du système de régulation de vitesse.

1. Implémenter les règles suivantes à l'aide de ctrl.Rule et construire le système de contrôle flou (ControlSystem) associé.

- ✓ Si la distance est très petite ('v\_small') ET le changement de vitesse est en déclin ('declining'), alors l'ajustement d'accélération est très négatif ('neg\_small').
- ✓ Si la distance est très petite ('v\_small') ET le changement de vitesse est constant ('constant'), alors l'ajustement d'accélération est très négatif ('neg\_big').
- ✓ Si la distance est très petite ('v\_small') ET le changement de vitesse est croissant ('growing'), alors l'ajustement d'accélération est très négatif ('neg\_big').
- ✓ Si la distance est petite ('small') ET le changement de vitesse est en déclin ('declining'), alors l'ajustement d'accélération est nul ('zero').

- Si la distance est petite ('small') ET le changement de vitesse est constant ('constant'), alors l'ajustement d'accélération est très négatif ('neg\_small').
- Si la distance est petite ('small') ET le changement de vitesse est croissant ('growing'), alors l'ajustement d'accélération est très négatif ('neg\_big')
- ✓ Si la distance est parfaite ('perfect') ET le changement de vitesse est en déclin ('declining'), alors l'ajustement d'accélération est très positif ('pos\_small')
  - ✓ Si la distance est parfaite ('perfect') ET le changement de vitesse est constant ('constant'), alors l'ajustement d'accélération est nul ('zero').
  - ✓ Si la distance est parfaite ('perfect') ET le changement de vitesse est croissant ('growing'), alors l'ajustement d'accélération est très négatif ('neg\_small')
  - ✓ Si la distance est grande ('big') ET le changement de vitesse est en déclin ('declining'), alors l'ajustement d'accélération est très positif ('pos\_big')
  - ✓ Si la distance est grande ('big') ET le changement de vitesse est constant ('constant'), alors l'ajustement d'accélération est très positif ('pos\_small')
  - ✓ Si la distance est grande ('big') ET le changement de vitesse est croissant ('growing'), alors l'ajustement d'accélération est nul ('zero')
  - ✓ Si la distance est très grande ('v\_big') ET le changement de vitesse est en déclin ('declining'), alors l'ajustement d'accélération est très positif ('pos\_big').
  - ✓ Si la distance est très grande ('v\_big') ET le changement de vitesse est constant ('constant'), alors l'ajustement d'accélération est très positif ('pos\_big')
  - ✓ Si la distance est très grande ('v\_big') ET le changement de vitesse est croissant ('growing'), alors l'ajustement d'accélération est très négatif ('pos\_small').

2. Simuler le fonctionnement du système pour les valeurs d'entrée suivantes :

- ✓ Distance\_to\_next\_car = 13
- ✓ Speed\_change = -2.5

3. Calculer la valeur de sortie Acceleration\_Adjust à l'aide de ControlSystemSimulation, afficher le résultat numérique, et visualiser graphiquement le résultat de la défuzzification.