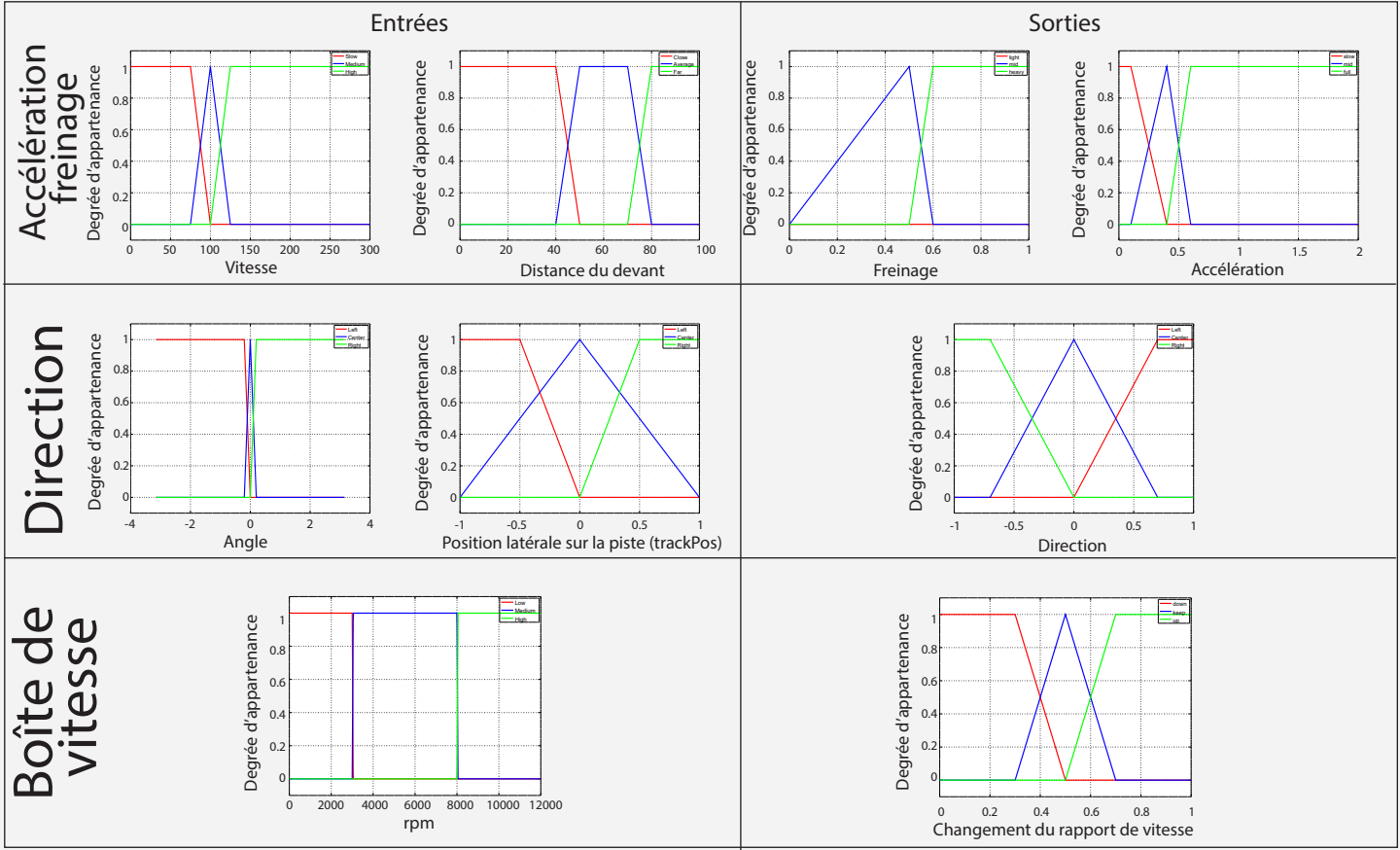
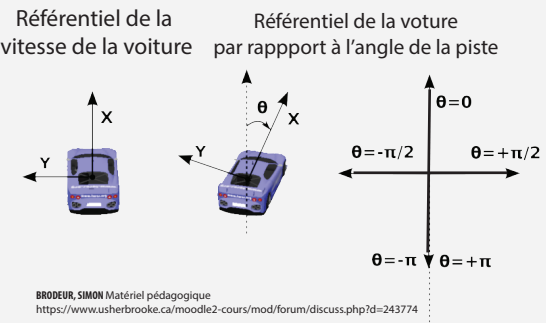


LOGIQUE FLOUE



		Mandani	Sugeno	Réseau de neurone
CLIPPED	AND	$t_{laps} = 2:07:56$ Conduite moins fluide.	$t_{laps} = 2:00:49$ Conduite moins fluide.	$t_{laps} = 2:06:76$ Conduite très fluide.
	OR	La voiture s'arrête après avoir parcourue une petite distance	La voiture s'arrête après avoir parcourue une petite distance.	
SCALED	AND	Configuration de référence. $t_{laps} : 2:07:06$ Conduite fluide.	$t_{laps} = 2:00:47$ Conduite fluide.	
	OR	La voiture s'arrête après avoir parcourue une petite distance	La voiture s'arrête après avoir parcourue une petite distance	
FPS		Entre 6 fps et 12 fps	Entre 12 et 16 fps	Entre 17 et 25 fps



ALGORITHME GÉNÉTIQUE

Choix des paramètres

Grandeur de la population.
Nombre de bits
Quantité de génération
Probabilité de croisement
Probabilité de mutation

opsize = 40;
nbits=16;
numGenerations = 40;
mutationProb = 0.005;
crossoverProb = 0.6;
NumParams = 8;

Initialisation de la population

Processus aléatoire des valeurs normalisées

Encodage avec contraintes

pour chaque génération

pour chaque individu

Décodage

Les données sont normalisées

Mise à l'échelle

(min, max) pour chaque paramètres à optimiser

Évaluation avec la simulation

Évaluation avec la fonction objective

La solution est l'individu affichant le meilleur résultat de la fonction objective.

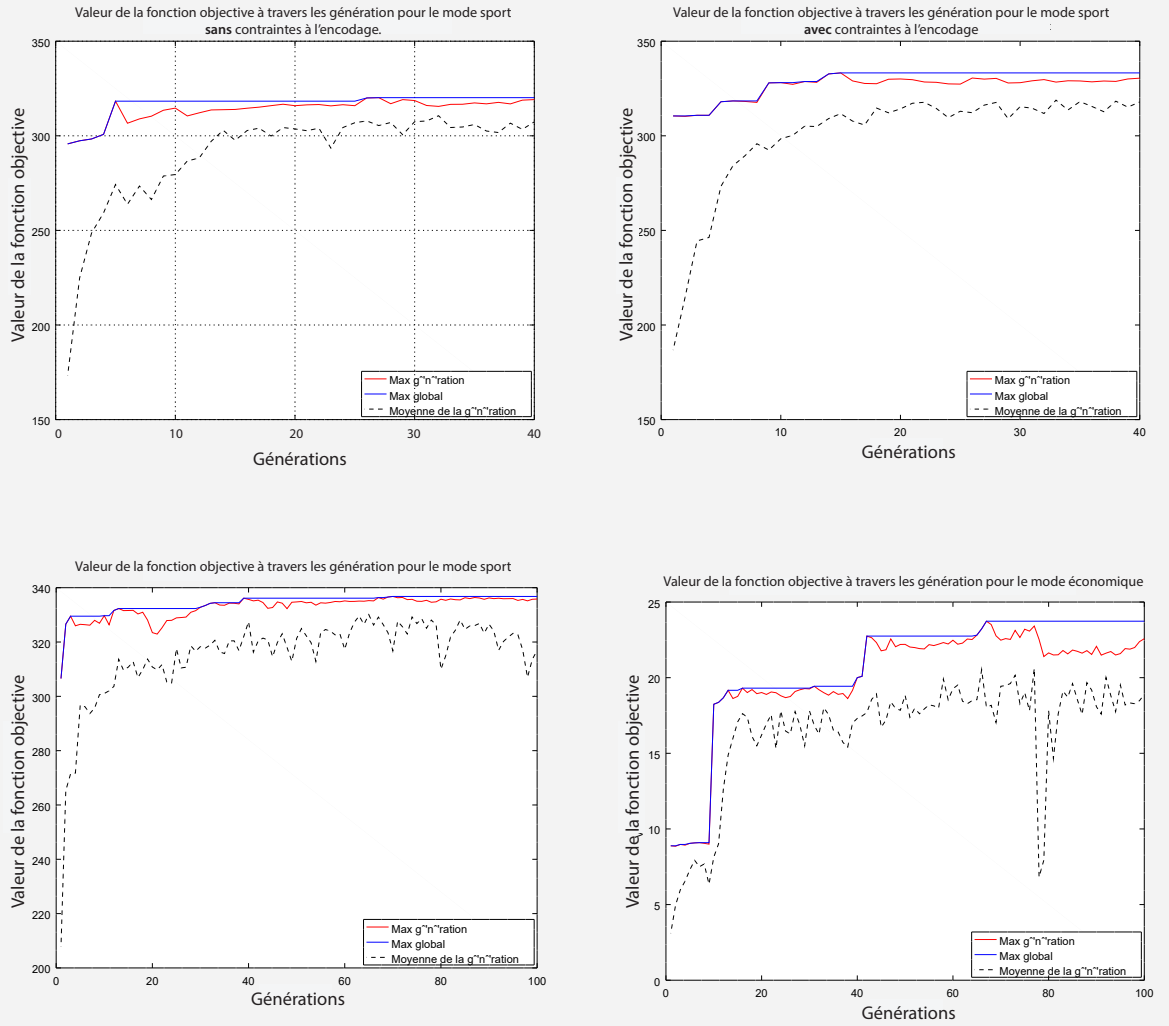
Sélection

Méthode de la roulette par somme cumulative.

Croisement

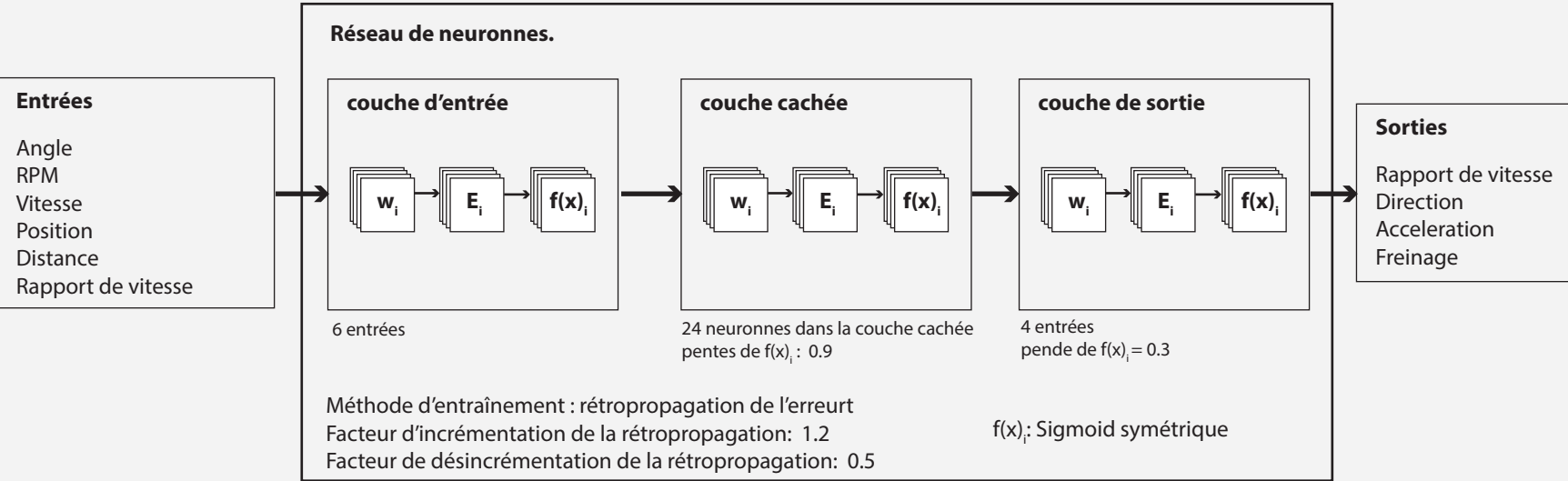
index de croisement déterminé aléatoirement

Mutation



Résultats de l'optimisation:
Mode course: vitesse maximal atteinte: 336.78 Km/h
Économique: consommation sur 100 km : 4.2135 L/100Km

RÉSEAU DE NEURONES



SYSTÈME ANFIS

```
FUNCTION SYSTÈME ANFIS(ENTRÉES)
    DÉFINIR CHAQUE POIDS DE CHAQUE ENTRÉE: K0 à Kn
    X1 = ENTRÉES // LAYER 1
    Y2 = FUZZIFICATION(X1) // LAYER 2
    Y3 = RULES(Y2) // LAYER 3
    Y4 = NORMALISATION(Y3) // LAYER 4
    Y5 = DEFUZZIFICATION(Y4, X1, K) // LAYER 5
    Y = SOMMATION DE NEURONNE(Y5) // LAYER 6
    retour Y

FUNCTION FUZZIFICATION(X1)
    POUR CHAQUE ENTRÉES DU SYSTÈME: X10 à X1n
    Y2i = BELL_ACTIVATION_CURVE(X1i)
    // Y2 est la valeur d'appartenance
    // à la fonction d'activation gaussienne.
    retour Y2

FUNCTION RULES(X2)
    POUR CHAQUE SORTIE DE LA FUZZIFICATION: X20 à X2n
    Y3i = PRODUIT de X2i à X2n
    // Y3 est la valeur d'activation
    retour Y3

FUNCTION NORMALISATION(X3)
    POUR CHAQUE SORTIE DES REGLES : X30 à X3n
    Y4i = X3i / (SOMME DE X3i à X3n)
    // Y4 EST LA VALEUR D'ACTIVATION NORMALISÉ
    RETOUR Y4

FUNCTION DE DEFUZZIFICATION(X4, X1, K1)
    FACTEUR = FACTEUR PONDÉRÉ DE L'ENTREE(X1,K1)
    POUR CHAQUE SORTIE DE LA NORMALISATION : X40 à X4n
    Y5i = X4i * FACTEUR
    // Y5 EST LA VALEUR PONDÉRÉ D'UNE RÈGLE.
    RETOUR Y5

FUNCTION DE SOMMATION DE NEURONNE(X5)
    RETOUR SOMMATION DE TOUS LES X5.

FUNCTION BELL_ACTIVATION_CURVE(X)
    A = CENTER OF BELL CURVE
    B = WIDTH OF BELL CURVE
    C = SLOPE OF BELL CURVE
    RETOUR 1 / (1 + ((X-A)/C)^(2*B))

FACTEUR PONDÉRÉ DE L'ENTREE(X,K)
    POUR CHAQUE ELEMENT DE X CORRESPOND UN ELEMENT K
    SOMME = X1*K1
    RETOUR SOMME
```

