

Défense de solution

APP2 – Intelligence artificielle bioinspirée

Équipe 5

Samuel Laperrière – laps2022

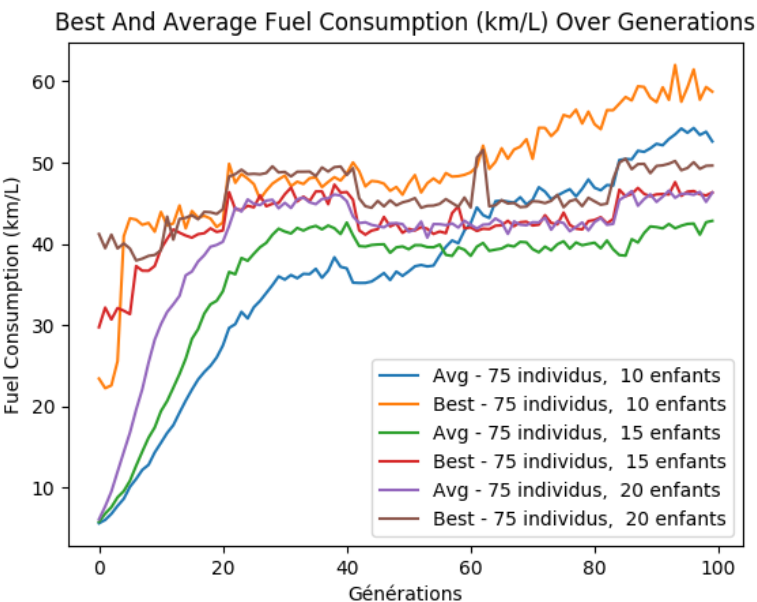
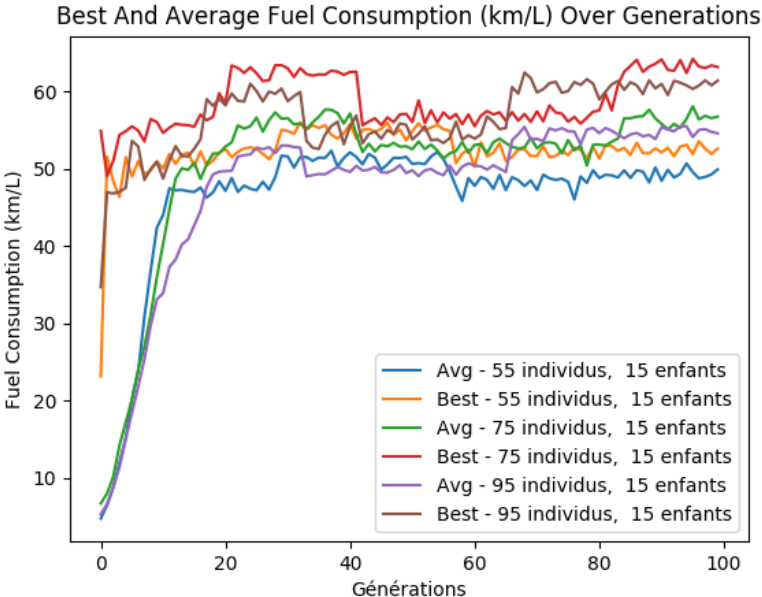
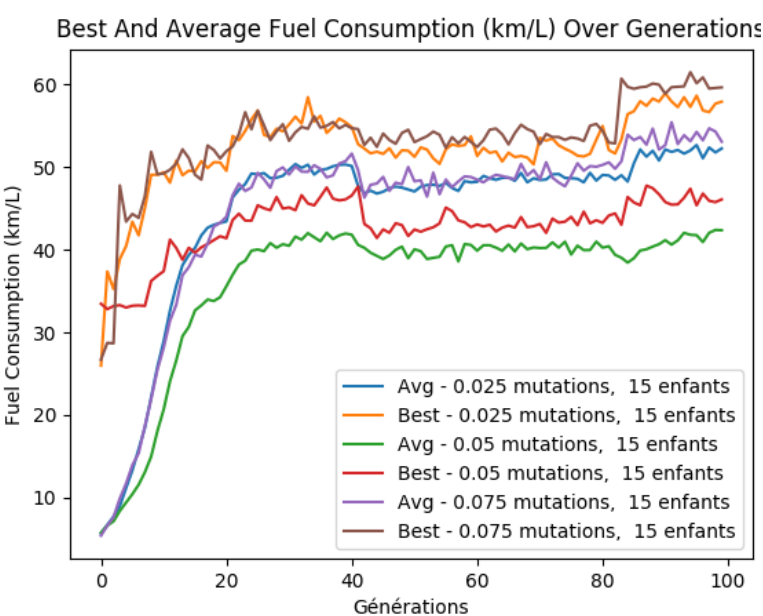
Raphaël Lebrasseur – lebr2112

Charles Murphy – murc3002

Algorithme génétique – encodage et approche

- Population
 - Initialisée de façon aléatoire
 - 8 bits par gène
- *Fitness function*
 - Conduite économique
 - Retourne le rendement en km par litre
 - Conduite sportive
 - Retourne la vitesse maximale
- Reproduction
 - Parents choisis en fonction du *fitness*
 - N enfants remplacent les N individus avec le pire *fitness*
 - 8 parents par enfant (gène 1 du parent 1, gène 2 du parent 2, etc.)
- Mutation
 - Probabilité d'occurrence
 - N bits modifié par la mutation
 - L'index d'un bit muté est aléatoire

Algorithme génétique – Config. économique



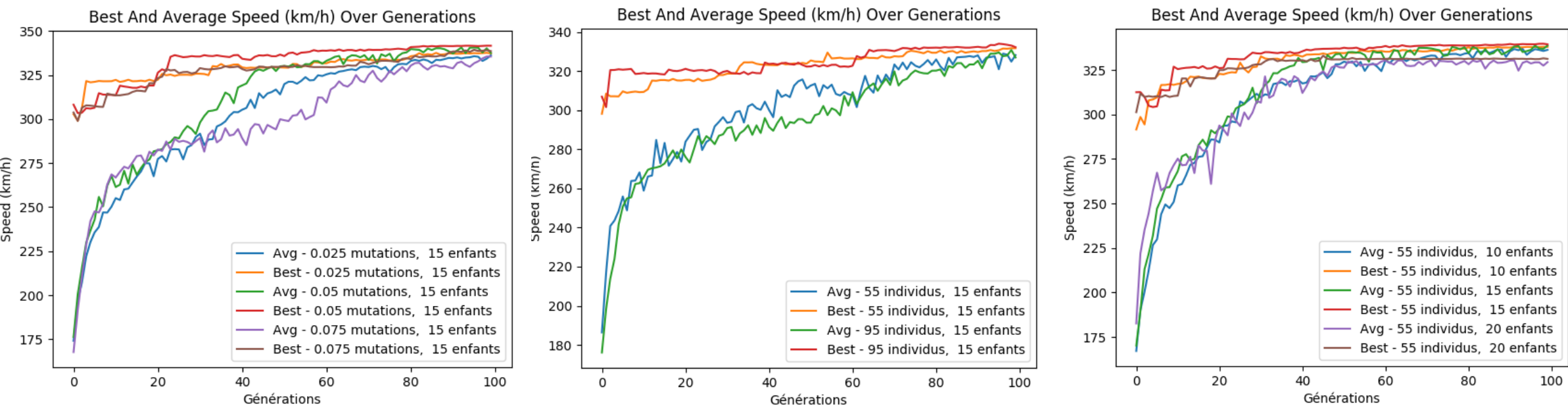
Paramètres

Taille population	Générations	Taux reproduction	Taux mutation	Bits mutés
75 individus	100	15 enfants/Gen	7,5%	2

Résultats

2	3	4	5	6	Diff	R-Angle	F-Angle	Speed (km/h)	Dist (m)	Fuel (L)	Fuel eco (km/L)
0.139	2.378	2.320	1.383	1.957	9.859	29.53	9.84	86.72	720.0	0.012	59.623

Algorithme génétique – Config. sportive



Paramètres

Taille population	Génération	Taux reproduction	Taux mutation	Bits mutés
75 individus	100	15 enfants/Gen	5%	2

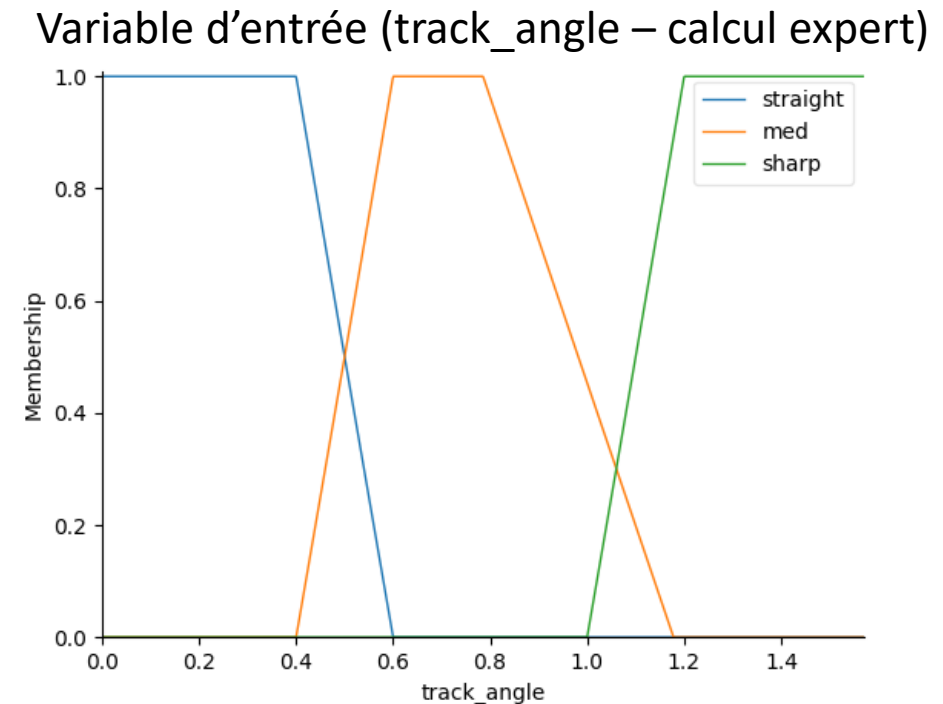
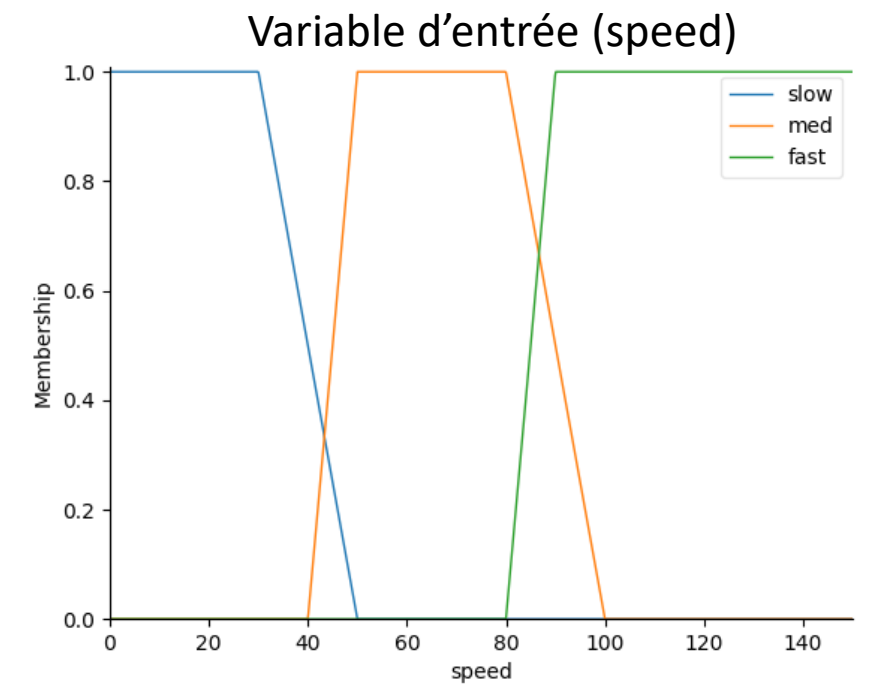
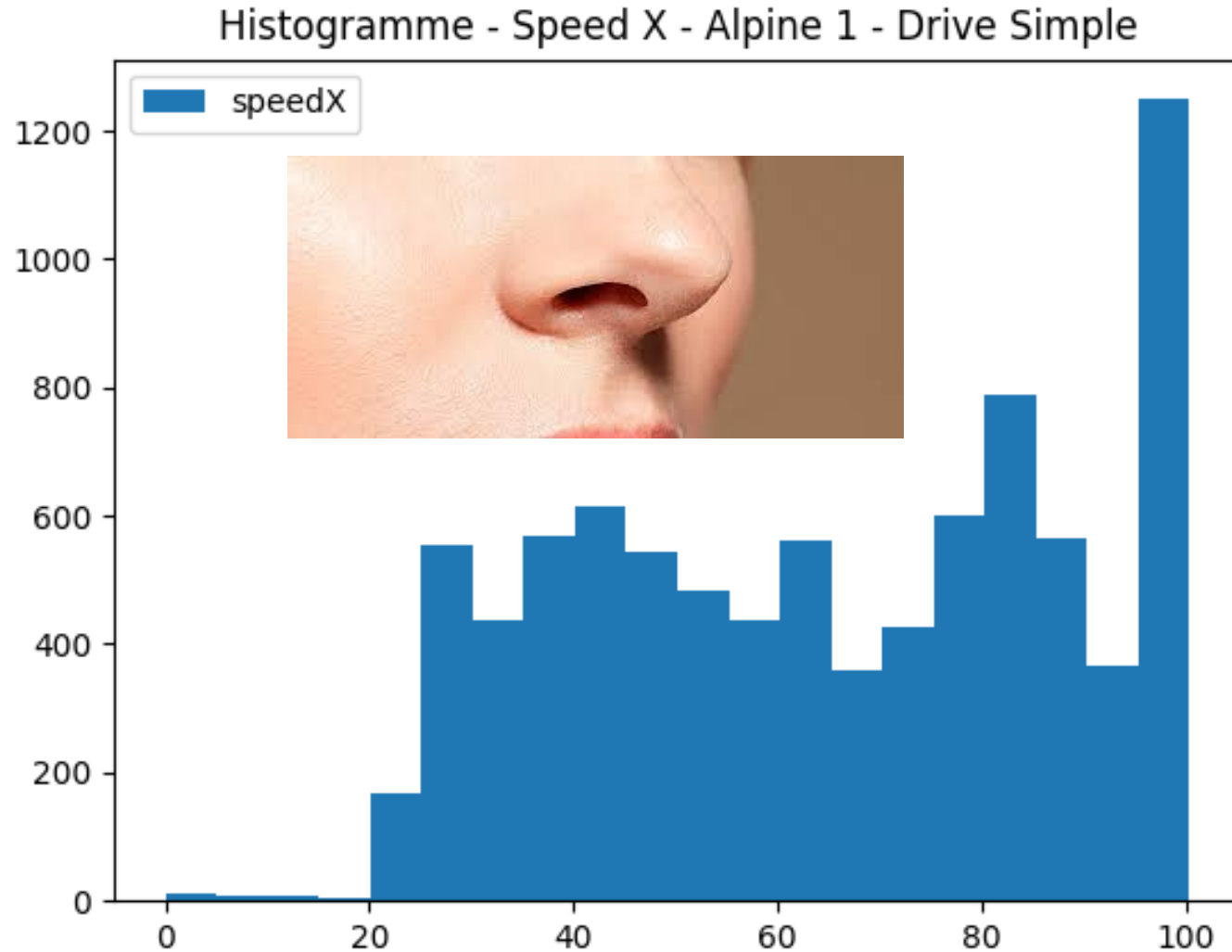
Résultats

2	3	4	5	6	Diff	R-Angle	F-Angle	Speed (km/h)	Dist (m)	Fuel (L)	Fuel eco (km/L)
2.11	1.34	0.96	0.80	0.71	4.41	0.00	1.75	339.53	2820.0	1.696	1.661

Algorithme génétique – synthèse

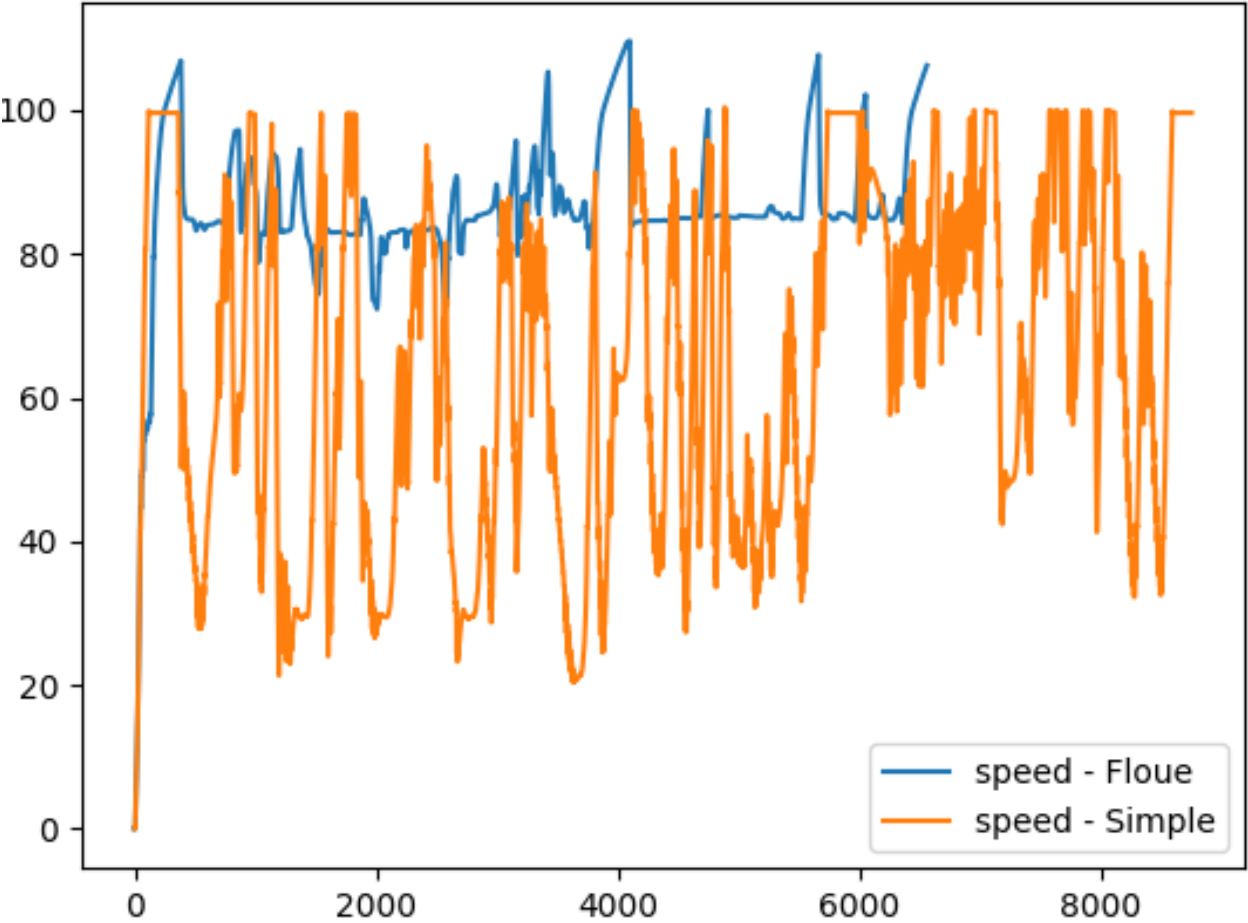
- Pertinence :
 - Simplifie la démarche pour converger vers une solution simple et efficace
 - Bons résultats, sans avoir besoin de décrire toute la dynamique (par opposition à un système expert)
- Améliorations possibles
 - *Fitness function* qui prend en compte l'ordre décroissant des ratios de vitesse
 - Grande probabilité de mutation qui diminue avec le temps
 - Créer une population avec des ratios de vitesse initiaux qui respecte l'ordre décroissant
 - Adapter le nombre de bits par gène

Contrôleur logique floue – Accel

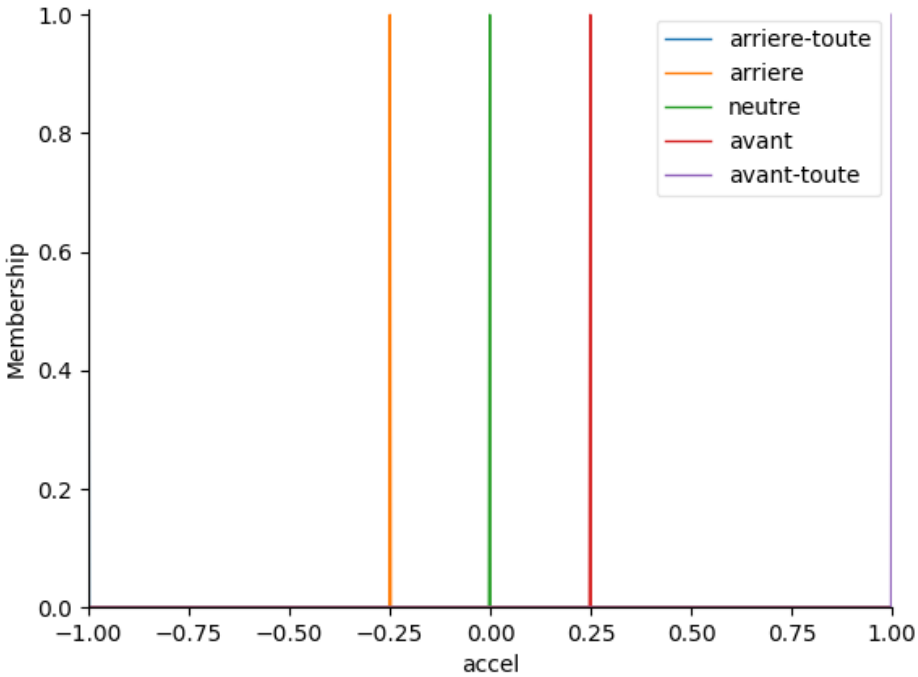


Contrôleur logique floue – Accel (résultats)

Comparatif - Floue vs Simple - Vitesse - Alpine 1



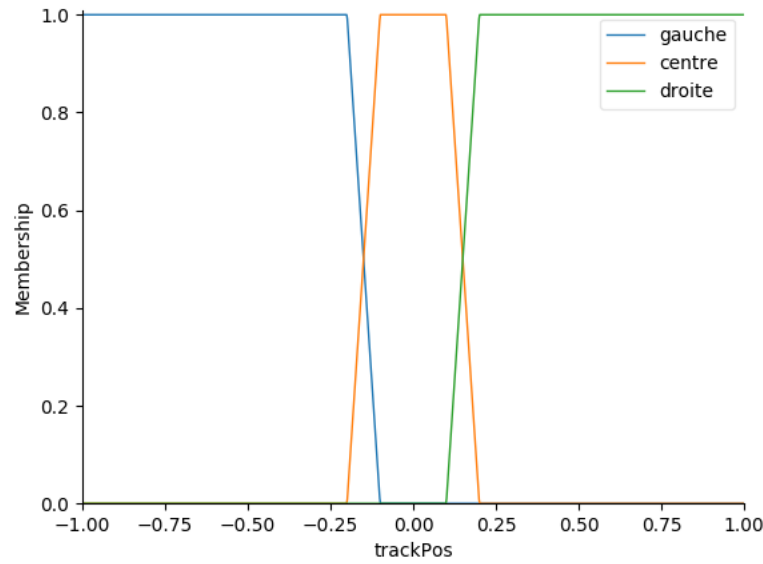
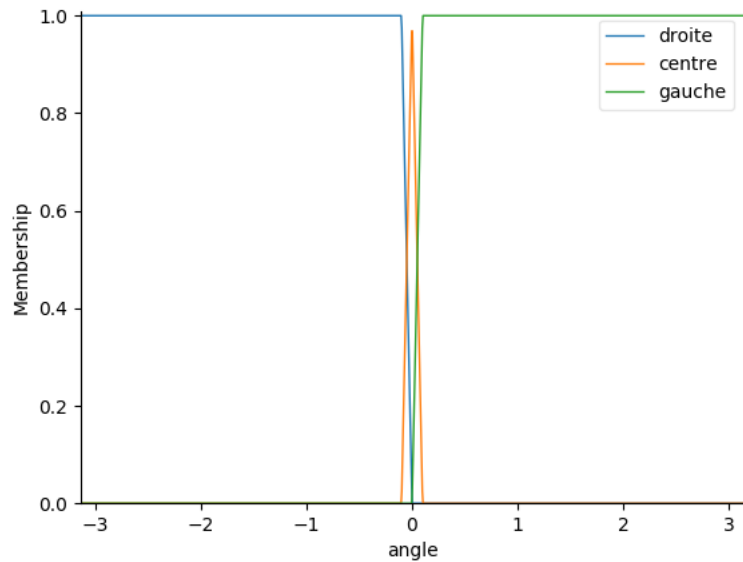
Variable de sortie (Accel)



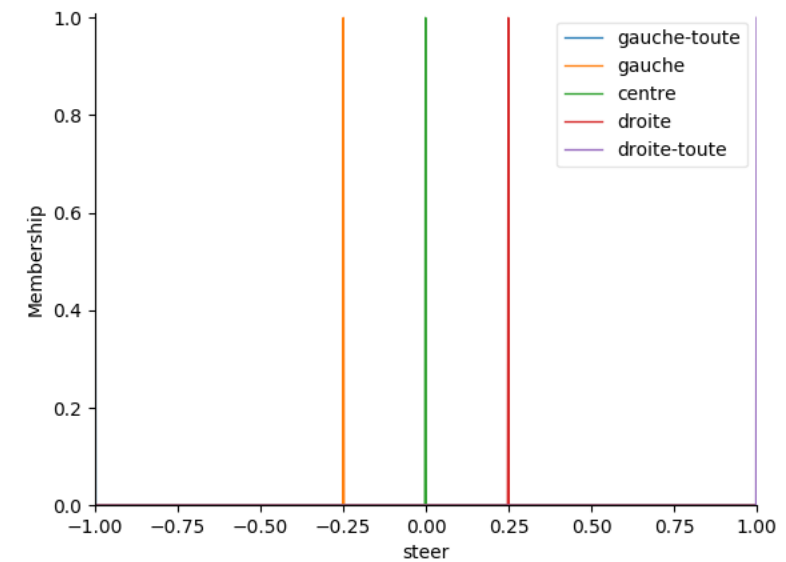
Vitesse Moyenne (km/h)	
Logique floue	64,92
Drive-simple	86,58

Contrôleur logique floue - Steer

Variables d'entrée (angle et trackPos)



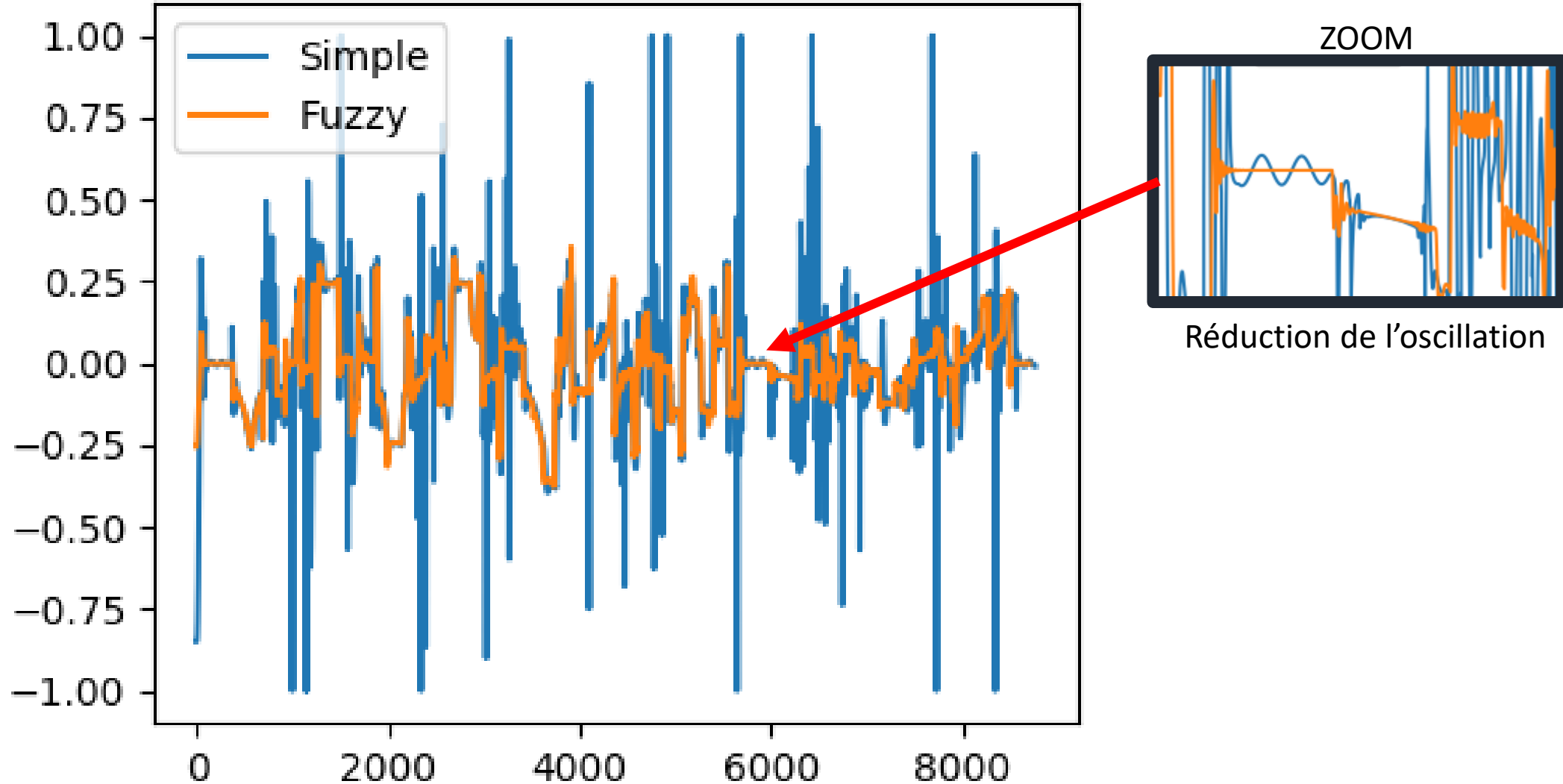
Variable de sortie (Steer)



- Utilisation de la conjonction (AND)
- Méthode pour la conjonction : **multiplication**
 - Adoucit les valeurs au centre (puisque multiplication de 2 nombres < 1)

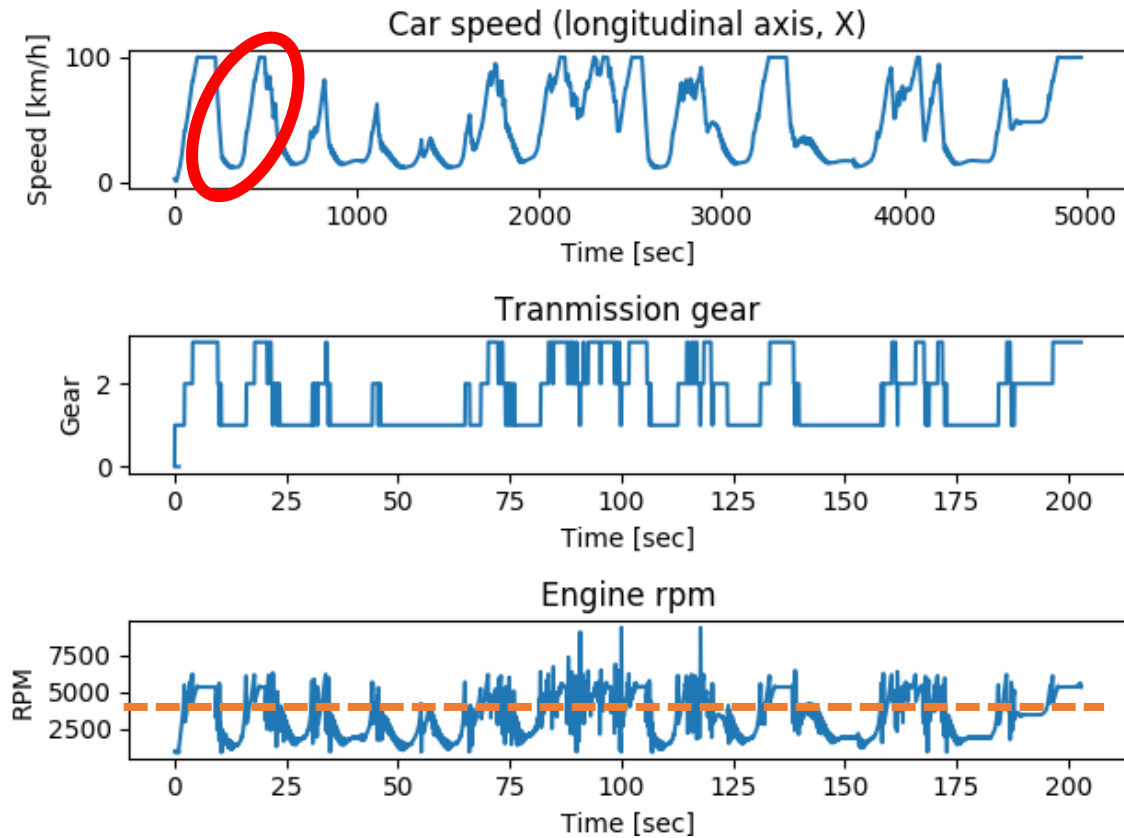
Contrôleur logique floue – Steer (résultats)

Comparatif - Floue vs Simple - Steer - Alpine 1

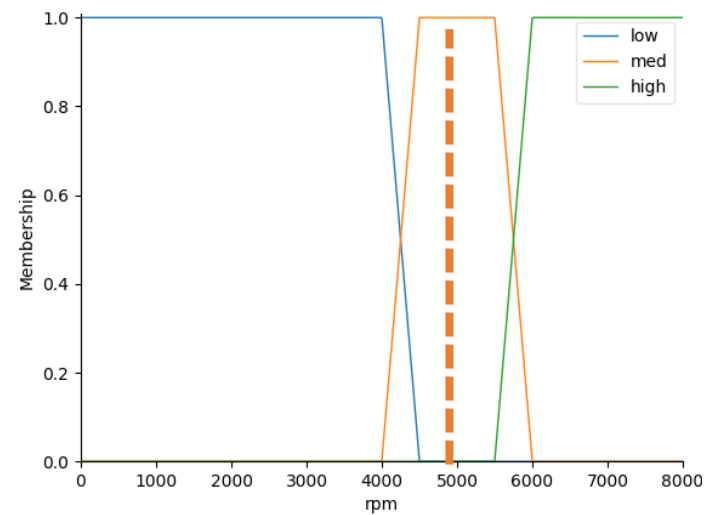


Contrôleur logique floue – Shift

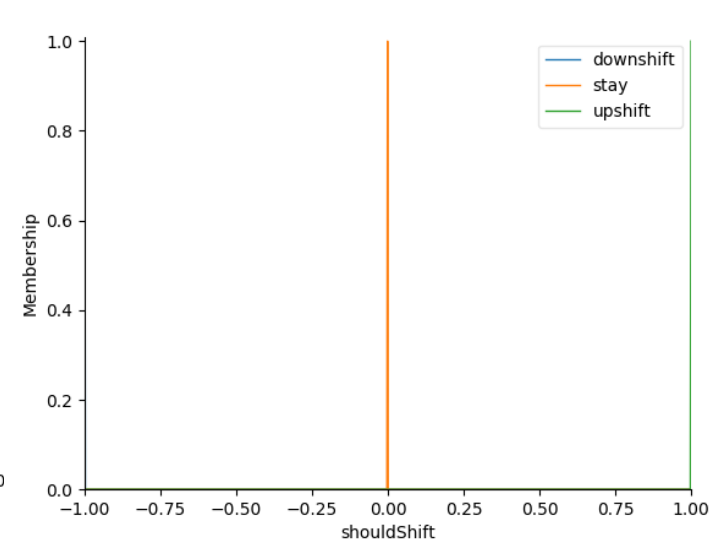
Régime moteur optimal (Drive simple) – Aalborg



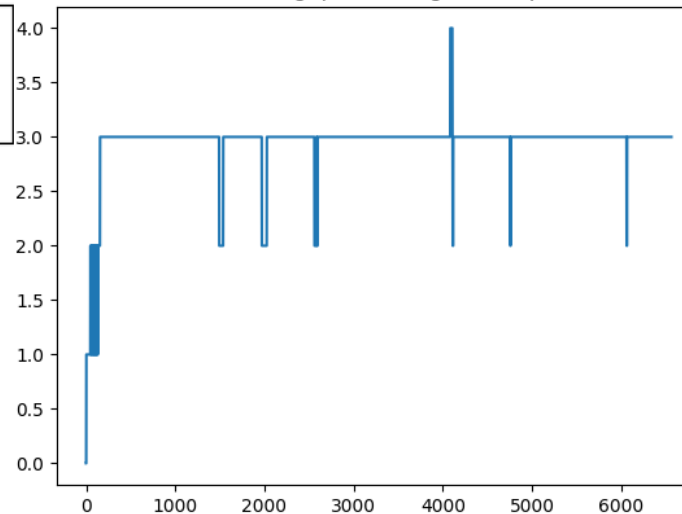
Variable d'entrée (RPM)



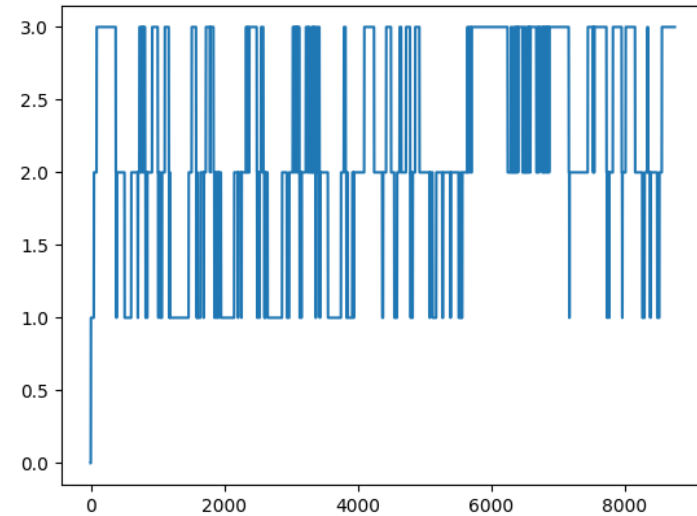
Variable sortie (shouldShift)



Vitesse - Logique floue (gear) - Alpine 1



Vitesse - Drive Simple (gear) - Alpine 1



Contrôleur logique floue – Synthèse

- Pertinence : simplicité de conception
 - “Asservissement” efficace sans solution complexe (Steer, Accélération)
- Shift : solution alambiquée
 - Stabilise le système, mais requerrait beaucoup plus de variables linguistiques pour être vraiment utile (enjeu de dimensionnalité)
- Pas de taille face à drive-bot :
 - Trop de variables linguistiques seraient nécessaires (dimensionnalité)
 - Solution simple pour problème simplifié

Réseau de neurones – approche

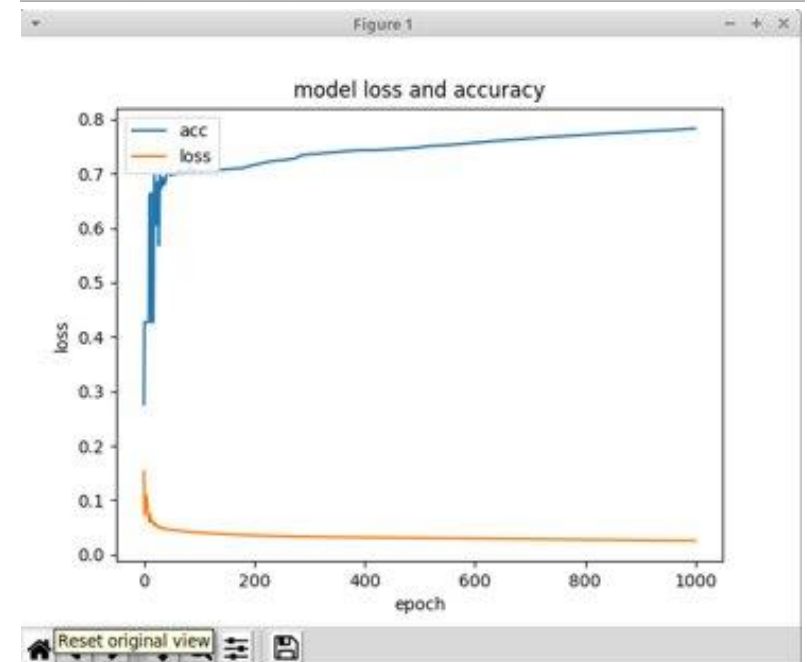
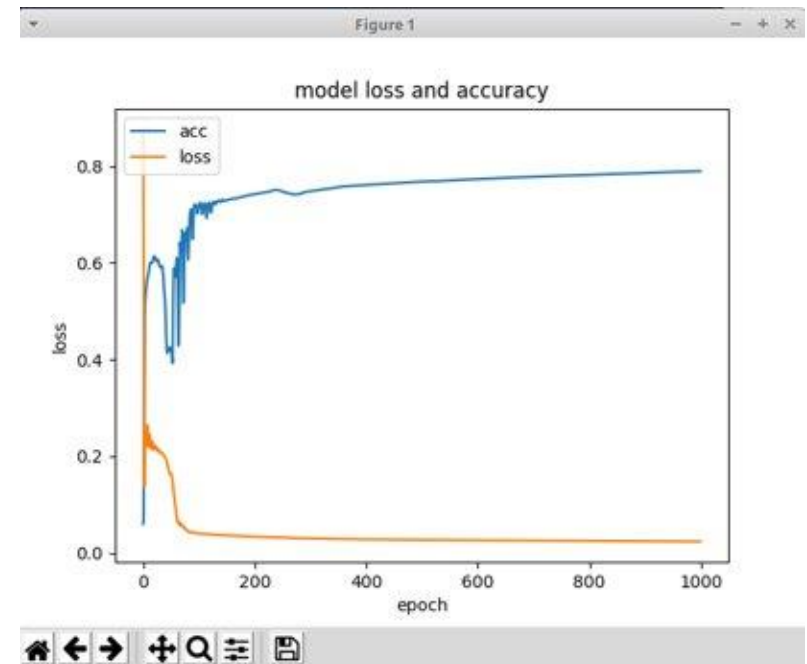
- Paramètres du système
 - 1 couche cachée
 - Activations : *tanh*
 - Le gradient disparaît plus lentement
 - Entrées : 19
 - Angle, vitesse X et Y, *gear*, *rpm*, position par rapport au milieu de la piste
 - Échantillons des capteurs de distances par rapport aux limites de la piste
 - Vitesse de rotation des roues
 - Neurones de la couche cachée: 14
 - Sorties : 4 (*tanh*)
 - *Loss function* : *mse*
 - Réseau de régression et pas de classification (on ne priorise pas un neurone de sortie à la fois)
 - Type d'entraînement : descente de gradient avec momentum
 - *Learning rate* : 0.2
 - *Momentum* : 0.9

Réseau de neurones – traitement des données

- Entrée : *tanh* => [-1, 1]
 - On contraint de -1 à 1 nos données
 - *brake* : on compense la *PWM* avec un passe-bas rudimentaire
 - `0.5 * brake actuel + 0.5 * brake précédent`
- Sortie : *tanh* => [-1, 1]
 - On passe d'une *tanh* à nos données de commande par différents multiplicateurs (selon le contexte)
 - Ex: `gear = round(np.clip(data[0][2], 0, 1) * 5) + 1`
 - (on élimine les *gear* 0 et 1)

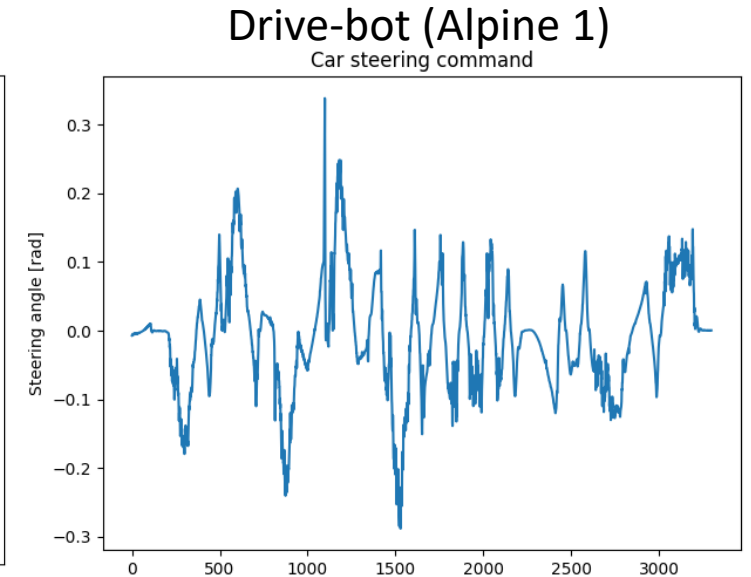
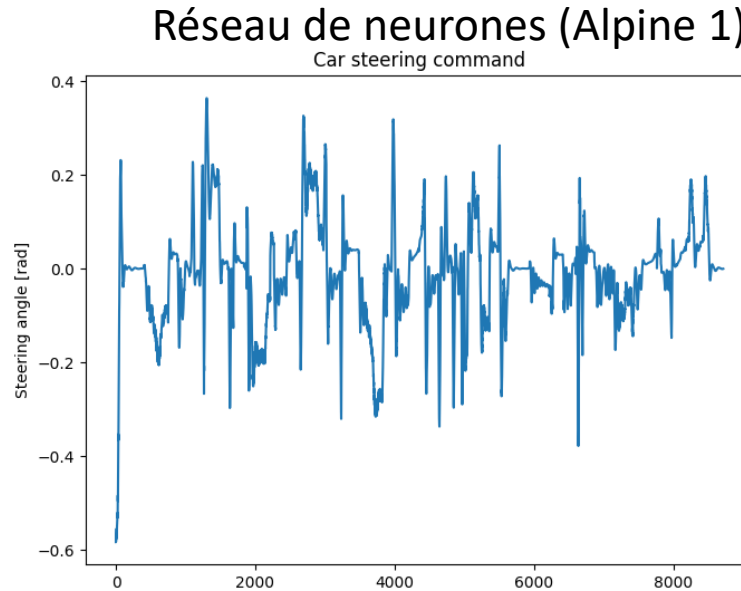
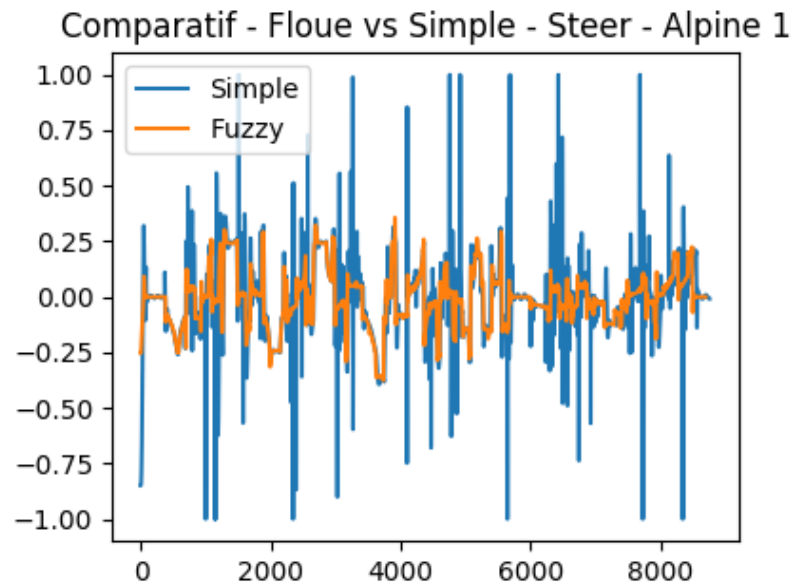
Réseau de neurones – tests & validations

- Partition des données
 - Utilisation d'un tier des données pour l'entraînement des paramètres/calibration des hyperparamètres
 - Données complètes pour les validations du réseau entier
- Influence des conditions de départ sur la convergence:
 - (figure du haut) distribution aléatoire uniforme de -1 à 1 (succès de validation : échec au premier essai)
=> convergence plus lente
 - (figure du bas) distribution aléatoire uniforme de -0.01 à 0.01 (succès de validation: 100 %)
=> convergence plus rapide



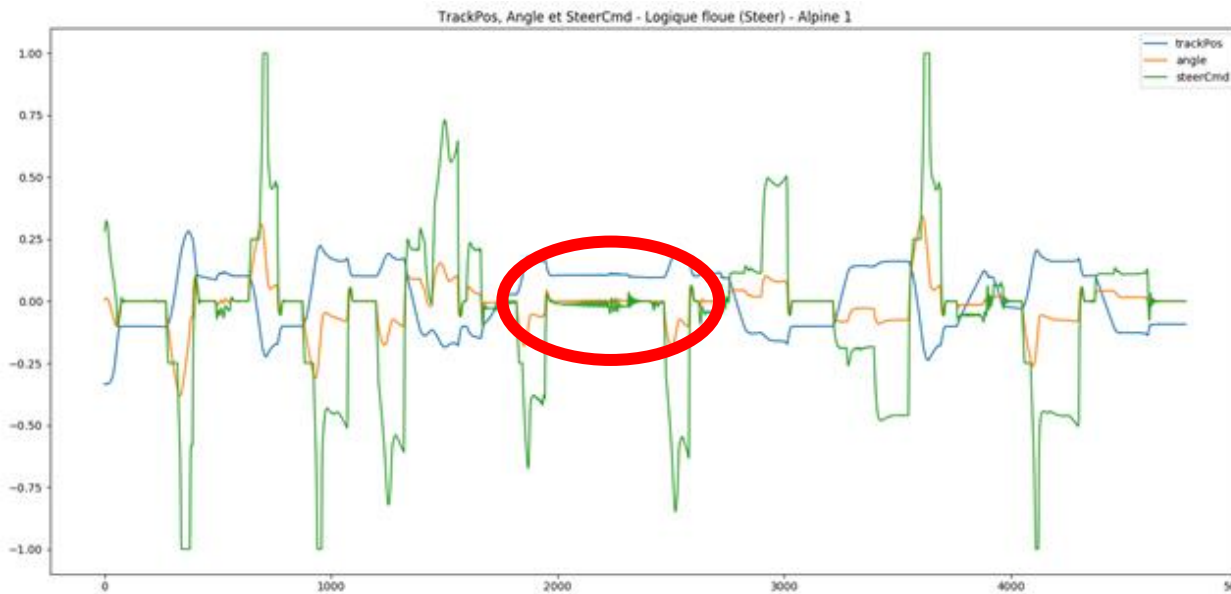
Réseau de neurones vs Logique floue

- La qualité du réseau ne dépasse pas celle des données d'entraînement
- Performance brute (steer) :

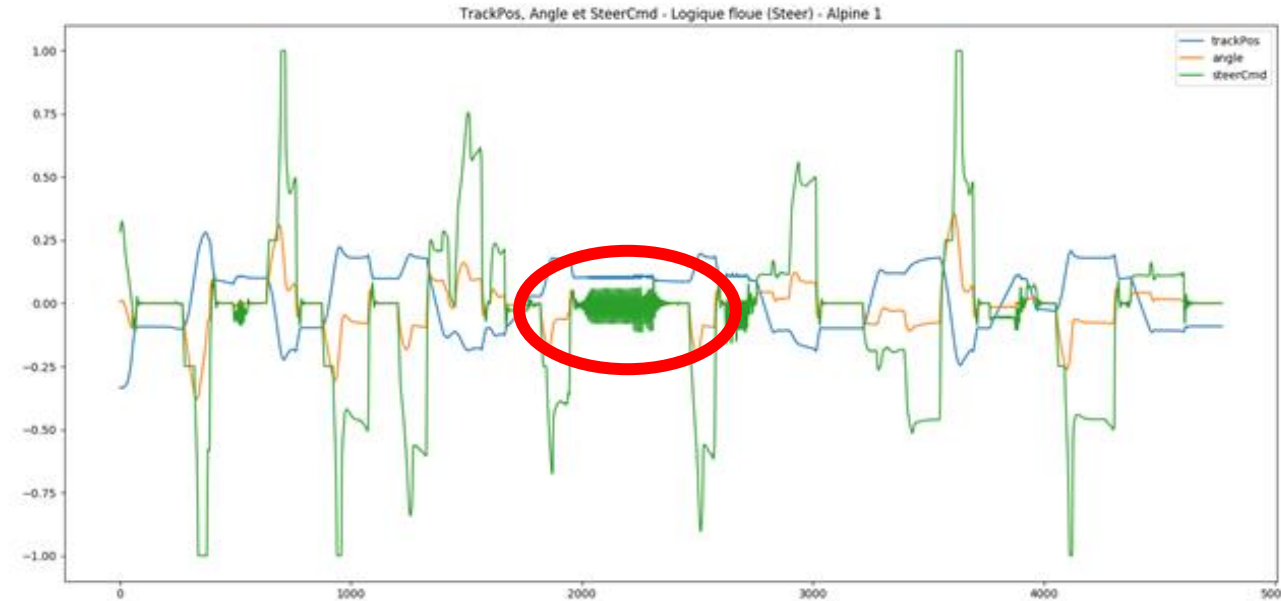


Annexes

Annexe A - Contrôleur logique floue – Steer – Multiplication vs Max



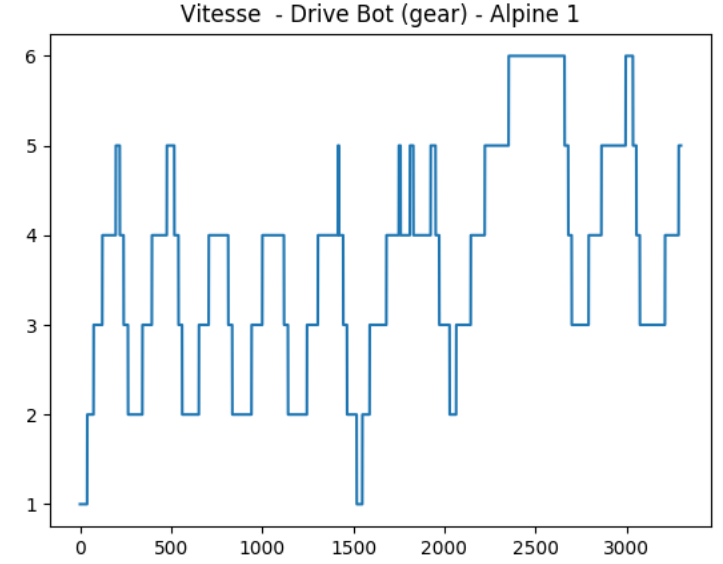
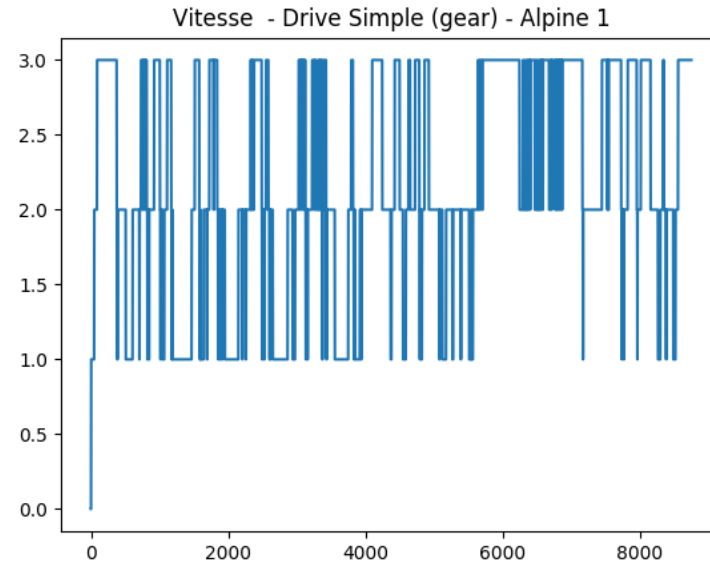
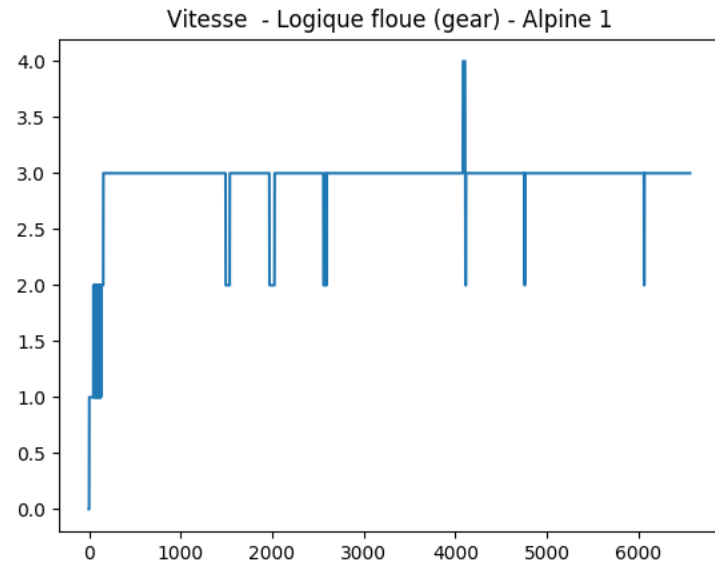
Meilleure solution : Sugeno avec méthode **multiplication**.



Comparatif : même chose, avec méthode **max**.

Oscillations plus importantes dans les lignes droites.

Annexe B - Contrôleur logique floue – Shift – Comparatif des contrôleurs



Annexe C – Règles de logique floue – Steer

Droite	Centre	Gauche	Gauche-toute
Centre	Droite	Centre	Gauche
Gauche	Droite-toute	Droite	Centre
trackPos/Angle	Gauche	Centre	Droite

Annexe D – Règles de logique floue – Accélération

Sharp	Avant-toute	Avant-toute	Arrière-toute
Med	Avant	Avant	Avant
Straight	Avant-toute	Avant-toute	Avant
Track_angle*/Speed	Slow	Med	Fast

* track_angle : le calcul d'estimation de la courbe (valeur absolue) de drive-simple

Annexe E – Règles de logique floue – Shift

High	Upshift
Med	Stay
Low	Downshift
<u>RPM</u>	