

Session S8 Électrique & Informatique

APP

Unité 2

Semaines 4, 5 et 6

**INTELLIGENCE ARTIFICIELLE
BIO-INSPIRÉE : LOGIQUE FLOUE,
RÉSEAUX DE NEURONES
ARTIFICIELS ET ALGORITHME
GÉNÉTIQUE - APPRENTISSAGE
MACHINE II**

Documentation des outils

**Département de génie électrique et de génie informatique
Faculté de Génie
Université de Sherbrooke**

Automne 2019

Note : En vue d'alléger le texte, le masculin est utilisé pour désigner les femmes et les hommes.

Document GIA792A19DocumentInformationComplementaireProbleme.pdf

Version 7, 9 septembre 2019

Rédigé par Simon Brodeur, Msc.A. et Jean Rouat, Ph.D.en collaboration avec

Copyright © 2005, ... 2019 Département de génie électrique et génie informatique. Université de Sherbrooke

Tuteur: Jean Rouat, Ph.D.

Réalisé avec TeXshop et L^AT_EX.

Table des matières

1	Introduction	4
1.1	Simulateur de voiture et interface Python	4
1.2	Description des structures de données	4
1.2.1	Mode optimisation	4
1.2.2	Mode contrôle	5

1 Introduction

Une machine virtuelle Ubuntu LINUX comprenant l'ensemble des logiciels (simulateur de voiture et outils d'intelligence artificielle) nécessaires pour la résolution du problème est fournie. Cette machine virtuelle inclut les environnements Python et **TORCS – "Open Racing Car Simulator"**.

1.1 Simulateur de voiture et interface Python

Commencez par vous familiariser avec le simulateur de voiture et l'interface en examinant le fichier *main.py* situé dans le répertoire *drive-simple* du code fournit. Ce fichier fait parcourir un circuit complet à la voiture tout en enregistrant à chaque instant l'état de la voiture. Les données enregistrées sont stockées dans le répertoire *drive-simple/recordings*. Vous pouvez vous inspirer de l'exemple de code situé dans le répertoire *visualization* pour examiner les variables d'entrées et de sortie du simulateur.

ATTENTION NE PAS MODIFIER les modules **torcs.control.core** ainsi que **torcs.optim.core**. Ces fichiers réalisent l'interface entre le simulateur et Python.

1.2 Description des structures de données

1.2.1 Mode optimisation

En mode optimisation, votre algorithme implémenté sous Python transmet un vecteur de 8 paramètres au simulateur (voir Tableau 1 pour description). Physiquement, les paramètres correspondent principalement aux ratios (ou rapports) de vitesse de la transmission pour les 5 dernières vitesses, comme illustré à la figure 2. Le simulateur roule le véhicule avec ces paramètres pour un temps défini. Il retourne ensuite des mesures de performance sur la voiture. Ce mécanisme est illustré à la figure 1. Vous devrez exploiter ce mécanisme pour implémenter la méthode d'optimisation par algorithme génétique.

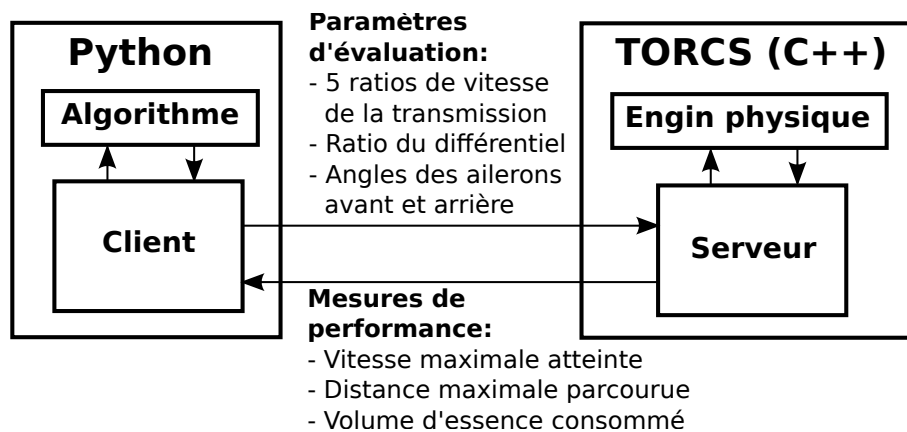


FIGURE 1 – Mécanisme de communication entre le code Python et TORCS pour la simulation en mode optimisation.

Description	Valeur typique
Ratio de la 2e vitesse	2.5
Ratio de la 3e vitesse	1.5
Ratio de la 4e vitesse	1.5
Ratio de la 5e vitesse	1.5
Ratio de la 6e vitesse	1.0
Ratio du différentiel arrière	4.5
Angle de l'aileron arrière	14.0
Angle de l'aileron avant	6.0

TABLE 1 – Description des paramètres à optimiser sur la voiture. Tous les rapports de la boîte de vitesse sont en unités SI et définis dans l'intervalle $[0.1, 5]$. Le rapport du différentiel arrière est en unité SI et défini dans l'intervalle $[1, 10]$. Les angles des ailerons avant et arrière sont en degrés et définis dans l'intervalle $[0, 90]$.

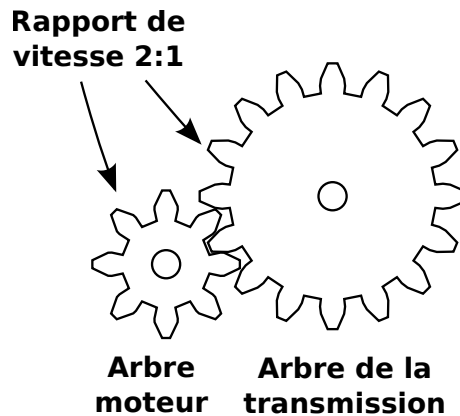


FIGURE 2 – Exemple de rapport de vitesse 2 : 1 pour la transmission de la voiture.

1.2.2 Mode contrôle

En mode contrôle, votre algorithme implémenté sous Python récupère l'état de la voiture dans la simulation et transmet en retour une commande à exécuter. Ce mécanisme est illustré à la figure 3. Vous devrez exploiter ce mécanisme pour implémenter le contrôle de la direction, de l'accélérateur, des freins et de la transmission par logique floue et/ou réseau de neurones artificiels. Le dictionnaire Python accessible et décrivant l'état de la voiture comporte les champs décrits au tableau 2. Le dictionnaire Python permettant le contrôle de la voiture requiert quant à elle les champs décrits au tableau 3.

En mode de contrôle, les données provenant du simulateur sont enregistrées sur le disque. Il est possible de conduire soi-même ou de faire conduire la voiture par un algorithme optimal sur plusieurs pistes (voir *drive-human/main.py*). À partir de l'enregistrement des données de la voiture pour chacune des pistes, une base de données peut être construite. Pour visualiser l'ensemble des données de la voiture pour un exemple de piste, voir le script *visualization/main.py*.

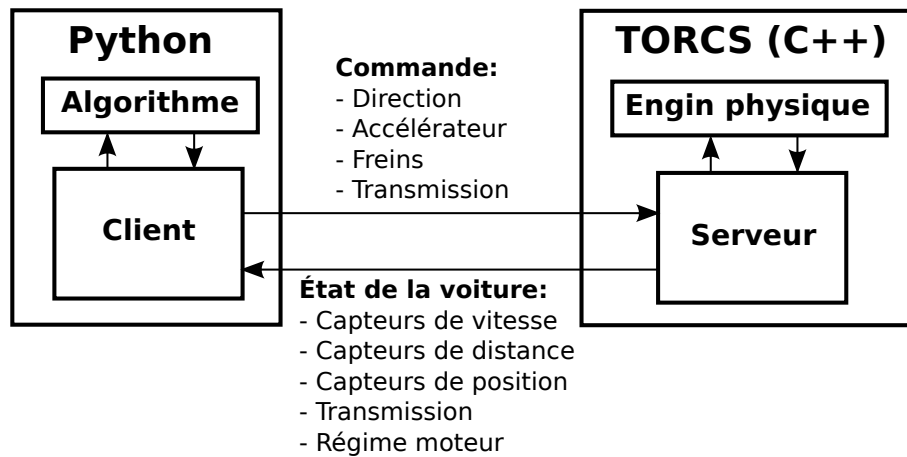


FIGURE 3 – Mécanisme de communication entre le code Python et TORCS pour la simulation en mode contrôle.

Nom	Description
angle	Angle entre la direction de la voiture et la direction de l'axe de la piste. Intervalle de $[-\pi, \pi]$. [rad]
damage	Dommages actuels sur la voiture (plus la valeur est élevée, plus le dommage est élevé). [points]
curLapTime	Temps écoulé depuis le dernier tour. [s]
distFromStart	Distance parcourue de la voiture à partir de la ligne de départ selon la piste. [m]
fuel	Niveau actuel d'essence dans le réservoir. [litres]
distRaced	Distance parcourue de la voiture depuis le début de la course. [m]
gear	Vitesse actuelle de la transmission. -1 si en reculs, 0 si neutre et de 1 à 6 pour les vitesses avant.
lastLapTime	Temps enregistré pour compléter le dernier tour de piste. [s]
rpm	Nombre de rotations par minute du moteur de la voiture, dans l'intervalle $[0, 10000]$. [rpm]
speedX	Vitesse de la voiture selon l'axe longitudinal (i.e. vitesse avant). [km/h]
speedY	Vitesse de la voiture selon l'axe transversal (i.e. vitesse de côté). [km/h]
track	Vecteur de 19 capteurs de distance : chaque capteur représente la distance entre la bordure de la piste (i.e. l'asphalte) et la voiture. Les capteurs sont orientés à chaque 10 degrés de $-\pi/2$ et $+\pi/2$ à l'avant de la voiture. Les capteurs ont une portée de 100 m. Quand la voiture est en dehors de la piste, les valeurs des capteurs ne sont pas fiables ! [m]
trackPos	Distance entre la voiture et l'axe de la piste. La valeur est normalisée par rapport à la largeur de la piste : 0 quand la voiture est sur l'axe (i.e. au centre de la piste), -1 quand la voiture est à la bordure droite de piste et +1 lorsque sur la bordure gauche. Les valeurs supérieures à 1 ou inférieures à -1 indiquent que la voiture se situe à l'extérieur de la piste !
wheelSpinVel	Vecteur de 4 capteurs représentent la vitesse de rotation de chacune des roues. [rad/s]

TABLE 2 – Description des variables décrivant l'état de la voiture durant la simulation.

Tableau adapté de [Software Manual of the Car Racing Competition @ WCCI2008](#)

Nom	Description
accel	Pédale de gaz virtuelle (0 ne signifiant aucun gaz, 1 plein gaz), dans l'intervalle [0,1].
brake	Pédale de frein virtuelle (0 ne signifiant aucun freinage, 1 plein freinage), dans l'intervalle [0,1].
gear	Vitesse actuelle de la transmission. -1 si en reculons, 0 si neutre et de 1 à 6 pour les vitesses avants.
steer	Angle de la direction. -1 et +1 correspondent respectivement à une direction pleinement à gauche et à droite, correspondant à un angle de $\pi/4$ rad.

TABLE 3 – Description des variables décrivant une commande de la voiture durant la simulation.
Tableau adapté de [Software Manual of the Car Racing Competition @ WCCI2008](#)