

Optimisation de la trajectoire idéale sur un circuit de sport automobile

Passionné par la course automobile depuis longtemps, ce sujet permet de relier cette passion à la physique et l'informatique. Par ailleurs, j'ai toujours été intéressé par la discrétisation et l'implémentation de modèles physiques, ainsi ce sujet permet de combiner plusieurs de mes centres d'intérêts.

La course automobile, et notamment la Formule 1, est un sport plutôt récent comparé à d'autres. Cependant, encadrés par des règles strictes et un travail acharné des pilotes cherchant à s'améliorer, il figure parmi les sports intenses et techniques, que ce soit pour les pilotes ou leurs équipes.

Positionnement thématique (ÉTAPE 1) :

- *INFORMATIQUE (Informatique pratique)*
- *PHYSIQUE (Mécanique)*

Mots-clés (ÉTAPE 1) :

Mots-clés (en français) Mots-clés (en anglais)

Simulation informatique *Computer simulation*

Mécanique Newtonienne *Classical mechanics*

Apprentissage génétique *Genetic learning*

Sport automobile *Motorsport*

Modélisation informatique *Computer models*

Bibliographie commentée

La recherche de la trajectoire optimale en course automobile est un problème abordé de plusieurs manières différentes, notamment à cause des grandes variations des contraintes des différentes courses. Notamment, on retrouve des méthodes par réseaux de neurones, plus particulièrement par algorithmes génétiques[1], ou par génération procédurale[2]. La génération procédurale donne des très bons résultats pour des circuits simples d'après l'article 2, mais s'étend mal à un problème plus vaste comme la F1 (La variété des circuits sur une année étant

incomparable à d'autres compétitions), et s'applique plus aux courses Américaines type IndyCar 500. Pour les réseaux de neurones, un réseau classique ou un réseau récurrent sont les plus appropriés pour ce type de problème[3], et permettent une très bonne approximation des informations disponibles à un vrai pilote, contrairement à un réseau convolutionnel.

D'après, ces articles le point limitant de cette modélisation semble être la résolution du circuit. En effet, une matrice 5000x5000 limite le circuit à une précision de 0.25m par pixel dans le cas le plus favorable (Circuit de Monaco), voire 0.5m (Circuit de Spa-Francorchamps) [5]. Peu d'articles donnent la résolution réelle utilisée, mais en vue des marges de confiance de leurs résultats, on peut s'attendre à une résolution aux alentours de 10000x10000, donnant au pire 0.25m de fautes, et au mieux 0.125m. Pour un ordre de grandeur, un circuit de F1 est large de 10-17m, et une voiture de 2m [5].

Cependant, il reste encore un grand nombre de questions. Combien de neurones/couches sont optimales pour ce type de problème?[6][7] Quels sont les paramètres d'entrées du réseau de neurone qui saisissent le meilleur équilibre entre réalisme (ce que le pilote peut voir), prédiction (ce que le pilote sait déjà sur le circuit) et complexité (nombre de neurones, temps de calcul), qui permettrait au réseau de neurone de converger en temps raisonnable[7]. L'obtention de ces paramètres semble [1][2] semble être les points les plus complexes, nécessitant probablement des algorithmes de recherches matricielles ou des techniques calculatoires. Tous ces calculs vont tourner autour de matrices et systèmes linéaires, un domaine où Python n'a que peu de désavantages face au C/C++ grâce à la bibliothèque Numpy. Ainsi, comme choisi dans les articles [1][4], ce langage semble offrir le meilleur compromis performance et facilité d'usage, en se restreignant aux bibliothèques Numpy, Scipy et Multiprocessing.

Problématique retenue

Comment peut-on déterminer la trajectoire optimale sur un circuit de Formule 1?

Objectifs du TIPE du candidat

- Implémenter une modélisation physique d'une voiture, en Python
- Réaliser un réseau de neurones récurrent, en Python, C et Swift
- Obtenir des paramètres d'entrées réalistes et manipulables efficacement, en C, Swift, Python et Metal
- Entraîner le réseau de neurones sur l'ensemble des circuits de Formule 1
- Confronter les résultats théoriques aux résultats de vrais pilotes sous les mêmes hypothèses

Références bibliographiques (ÉTAPE 1)

- [1] PIER LUCA LANZI : Searching for the optimal racing line using genetic algorithms : *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, IEEE*
- [2] YING XIONG : Racing Line Optimization : *Masters Thesis, Shanghai Jiao Tong University with Massachusetts Institute of Technology, 2010*
- [3] ARAVINDPAI PAI : Analyzing Types of Neural Networks in Deep Learning : <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/>
- [4] S. RISUCHAT : Development of a car physics engine for games : *Masters Thesis, Bournemouth University, 2012*
- [5] FÉDÉRATION INTERNATIONALE DE L'AUTOMOBILE : FIA 2023 Sporting Regulations - Formula 1 : https://www.fia.com/sites/default/files/fia_2023_formula_1_sporting_regulations_-_issue_2_-_2022-09-30.pdf
- [6] MARCO GADOLA, DAVID VETURRI, DANILO CAMBIAGHI : A Tool for Lap Time Simulation : *Research Paper, University of Brescia, 1996*
- [7] VINCENT BARRA, LAURENT MICLET, ANTOINE CORNUÉJOLS : Apprentissage artificiel - 4e édition, Concepts et algorithmes : 2021, ISBN-13 978-2-416-00104-8

DOT

- [1] : Septembre - Octobre: Mise en place du modèle théorique
- [2] : Fin octobre: Recherches sur les méthodes d'apprentissages et techniques d'optimisations
- [3] : Novembre-Décembre: Premier code fonctionnel
- [4] : Janvier-Mars: Améliorations, recherches approfondies sur l'intelligence artificielle et l'apprentissage génétique
- [5] : Mai: Finalisation de l'implémentation en utilisant la carte graphique, correction de problèmes et génération des résultats finaux