

# Systemes de conduite autonomes

**BENSALEM Akram**

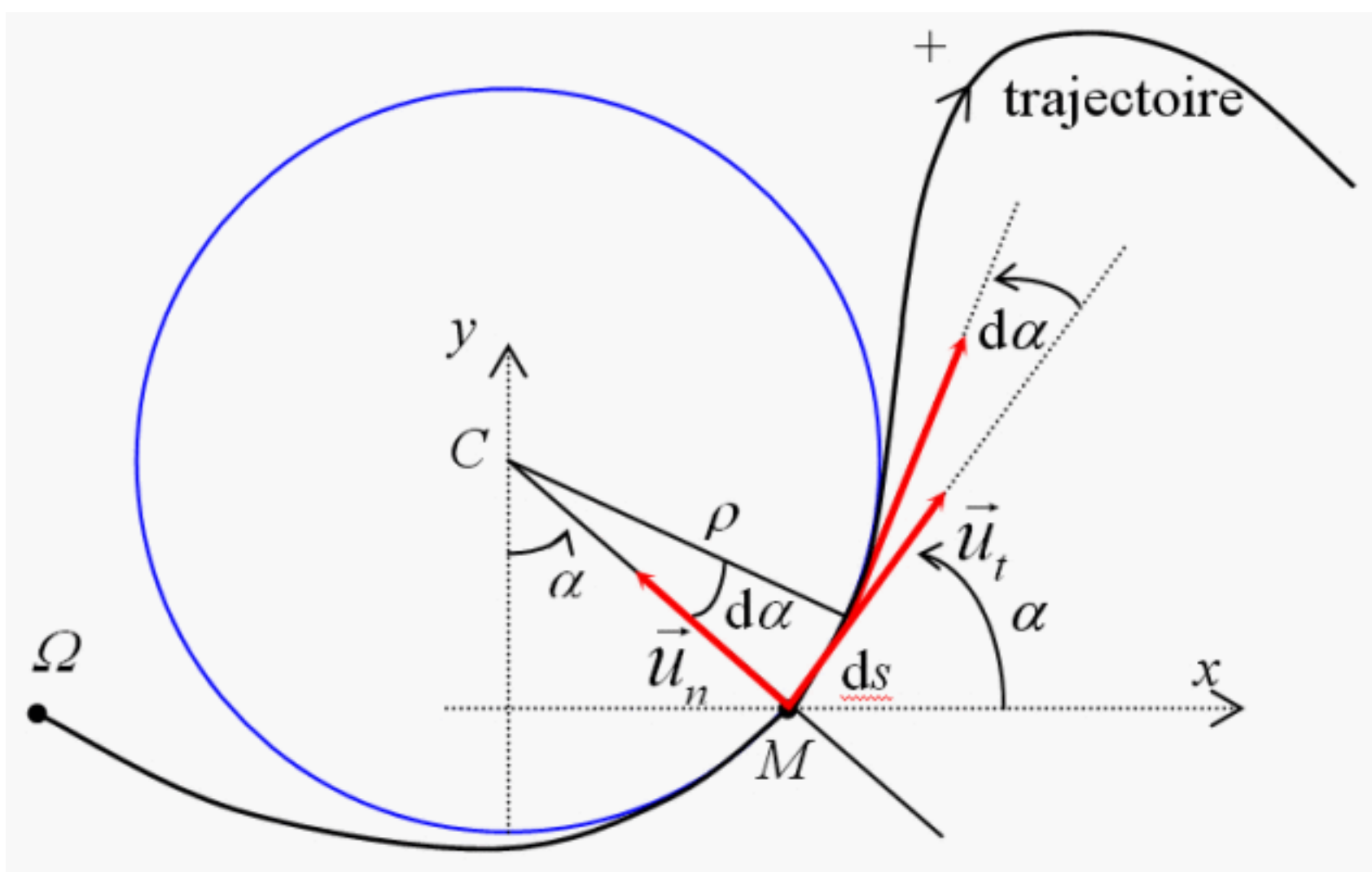
Dans ce document je vais commencer par introduire les concepts de systèmes autonomes de conduite et on va essayer de comprendre les algorithmes et les concepts mathématiques sous-jacents.

## Fonctionnement

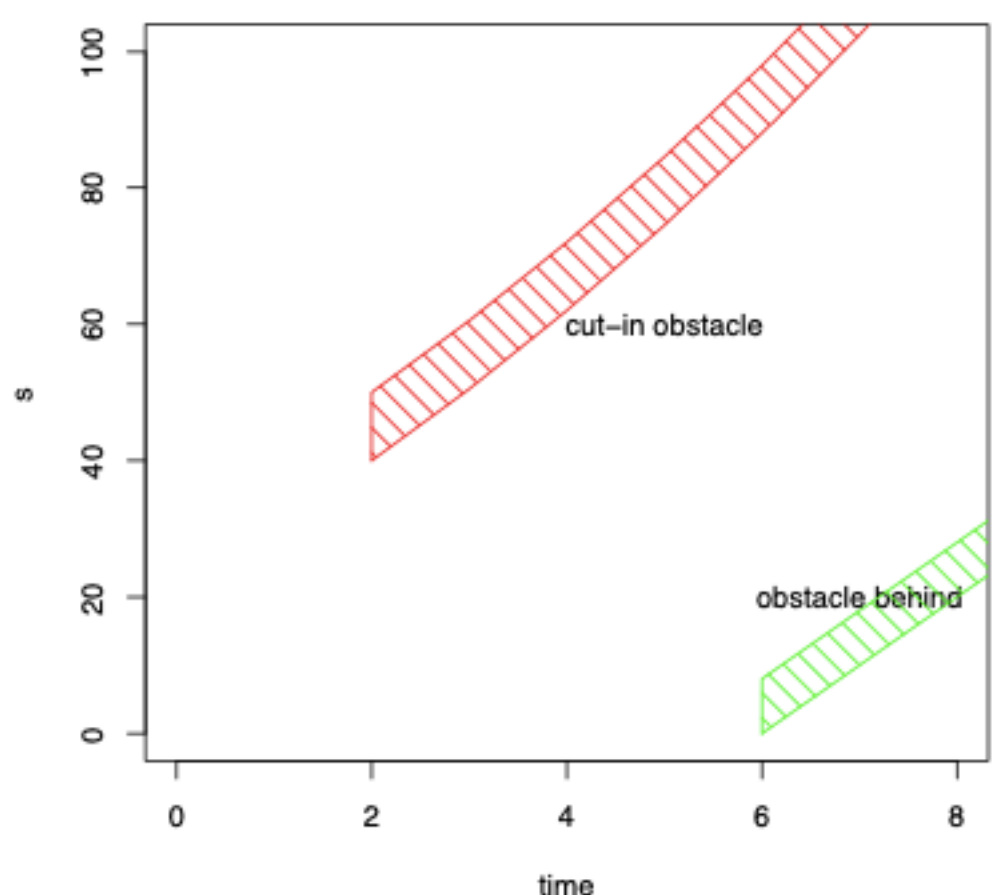
Ces systèmes s'appuient sur une carte routière, une localisation GPS et des capteurs/caméras/radars intégrés. Ces outils récoltent très rapidement les informations sur l'environnement qui sont nécessaires à la prise de décision par la machine. Ces machines automatisées prennent en compte la signalisation routière et respectent le code de la route et notamment les règles de priorité. Le meilleur exemple actuel qu'on pourrait donner ce sont les voitures de la marque Tesla qui peuvent rouler sans pilote et aucune intervention humaine. Ces voitures autonomes sont considérées comme plus sûres car le temps de réaction de ces véhicules n'est que de 100ms avec une planification de la trajectoire sur 200 mètres ou les prochaines 8 secondes. Tous ces éléments déterminent le motion planning.

# Principes mathématiques

Le EM planner est le nom de l'algorithme permettant de prédire les futurs trajectoires et manoeuvres d'une voiture nomade. L'élaboration de cet algorithme repose sur des principes mathématiques tels que la géométrie différentielle mais aussi la quadrature. L'environnement dans lequel ces voitures autonomes circulent peuvent être apparentées à un espace cartésien  $(x,y,\theta,t)$ . Les coordonnées de l'objet (le véhicule) dans cet espace sont transposées ensuite dans un **repère de Frenet**  $(s, l, dl, ddl, dddl)$  où  $s$  est le trièdre de Frenet qui est un point du repère représentant un objet,  $l$  est un vecteur centré vers l'intérieur de la trajectoire où  $s$  appartient à  $l$  et les axes  $dl, ddl, dddl$  sont respectivement les dérivées première, seconde et troisième du vecteur  $l$ . On imagine que dans la figure ci-dessous que le point  $s$  est représenté par le point  $M$ .



L'algorithme va d'abord identifier une multitude d'obstacles en une fraction de seconde. Les autres véhicules sur la route sont des éléments déterminés comme des obstacles. Après la prise en compte d'obstacles, le EM planner va transposer graphiquement ces informations, avec un horizon de 8 secondes afin de choisir l'allure à adopter. Le graphe représentant l'environnement du véhicule, les zones de faisabilité vont alors donner la vitesse possible et optimale à adopter selon la trajectoire choisie. C'est la **speed optimization**.



Le "dynamic programming path" est un algorithme complémentaire du EM planner. Il permet d'évaluer le coût de déplacement du véhicule selon le chemin suivi et selon les obstacles rencontrés. Il suit une fonction:  $C_{total}(f(s)) = C_{smooth}(f) + C_{obs}(f) + C_{quidance}(f)$

Le coût du "smoothness", c'est-à-dire la régularité du chemin suivi par le véhicule et le confort est donné par :

$$C_{smooth}(f) = w_1 \int (f'(s))^2 ds + w_2 \int (f''(s))^2 ds \\ + w_3 \int (f'''(s))^2 ds.$$

Cela rappelle la quadrature gaussienne qui permet d'approximer une intégrale grâce à une intégration sur plusieurs segments avec les  $w$  représentant des poids. On cherche à déterminer la surface du chemin empruntable par le véhicule. Étant donné que nous nous situons dans un repère de Frenet, nous avons  $f(s)$  qui représente le changement de cap par rapport à la trajectoire initiale,  $f'(s)$  la courbe du chemin et  $f''(s)$  la dérivée de la courbe. Le meilleur compromis doit être trouvé dans le sens longitudinal de manière analogique. En supposant que la voiture ne roule pas trop vite ou trop près du véhicule qui précède, elle doit ralentir sensiblement mais sans précipitation excessive. La facilité et le confort peuvent être mieux décrits par la dérivée de l'accélération latérale ou longitudinale. On rappelle que l'accélération est donné par le vecteur  $\alpha$ .

$$C_{obs}(d) = \begin{cases} 0, & d > d_n \\ C_{nudge}(d - d_c), & d_c \leq d \leq d_n, \\ C_{collision} & d < d_c \end{cases}$$

Ici nous avons la fonction caractéristique de la fonction de coût lié au contournement des obstacles. Elle est nulle si la distance par rapport à dn. La valeur collision est calculée et majorée par la valeur du coût de collision.

La fonction guidance quant à elle détermine la différence de coût entre un chemin dit idéal, c'est-à-dire un chemin sur lequel on ne rencontre aucun obstacle et avec un temps de parcours minimal, et le chemin possible pour le véhicule.

$$C_{guidance}(f) = \int (f(s) - g(s))^2 ds$$

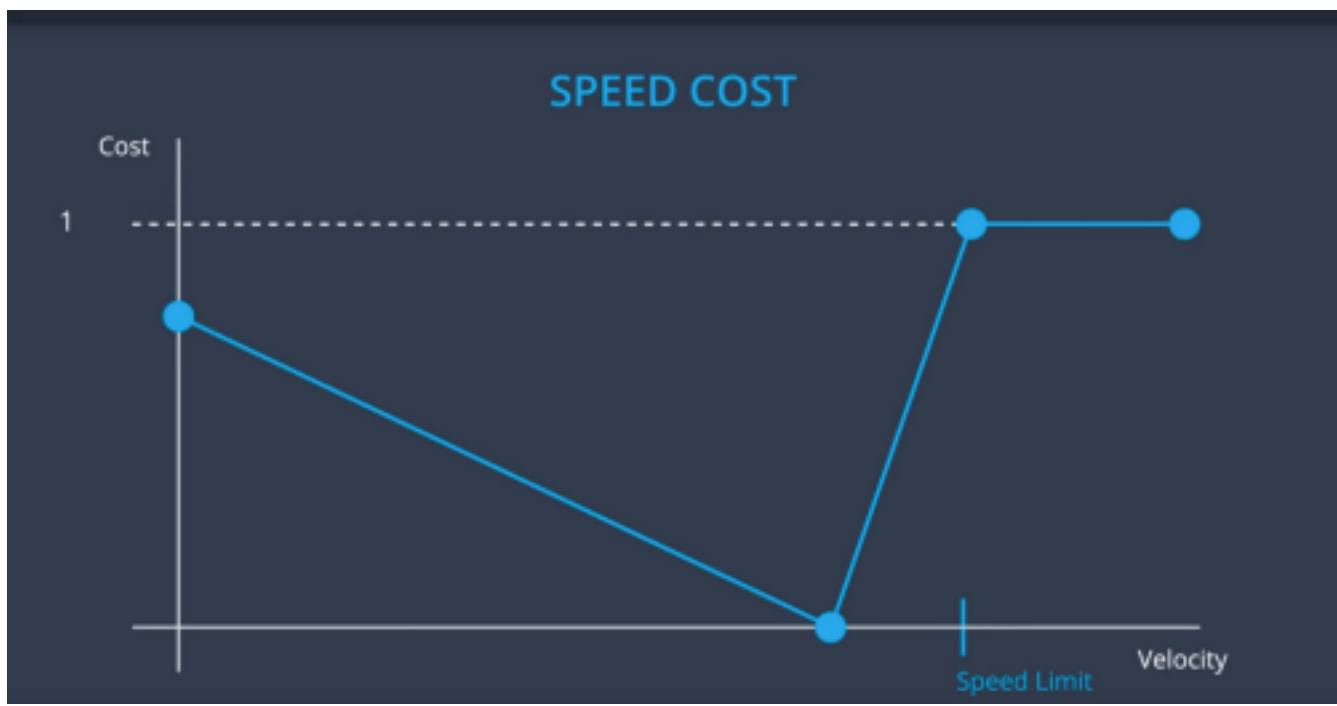
La fonction  $g$  représentant le chemin idéal.

Si on revient à la fonction de coût total on obtient:

$$C_s(f) = w_1 \int (f'(s))^2 ds + w_2 \int (f''(s))^2 ds \\ + w_3 \int (f'''(s))^2 + w_4 \int (f(s) - g(s))^2 ds.$$

L'objectif pour les algorithmes en général c'est de minimiser le temps d'exécution, qui, représente un coût pour la machine. L'objectif pour les systèmes de conduite autonome est le même, réduire la fonction de coût au minimum.

Puis le chemin projeté par la machine est long et moins il sera précis avec le temps, c'est pour ça que les itérations de l'algorithme se font avec une fréquence en ms.



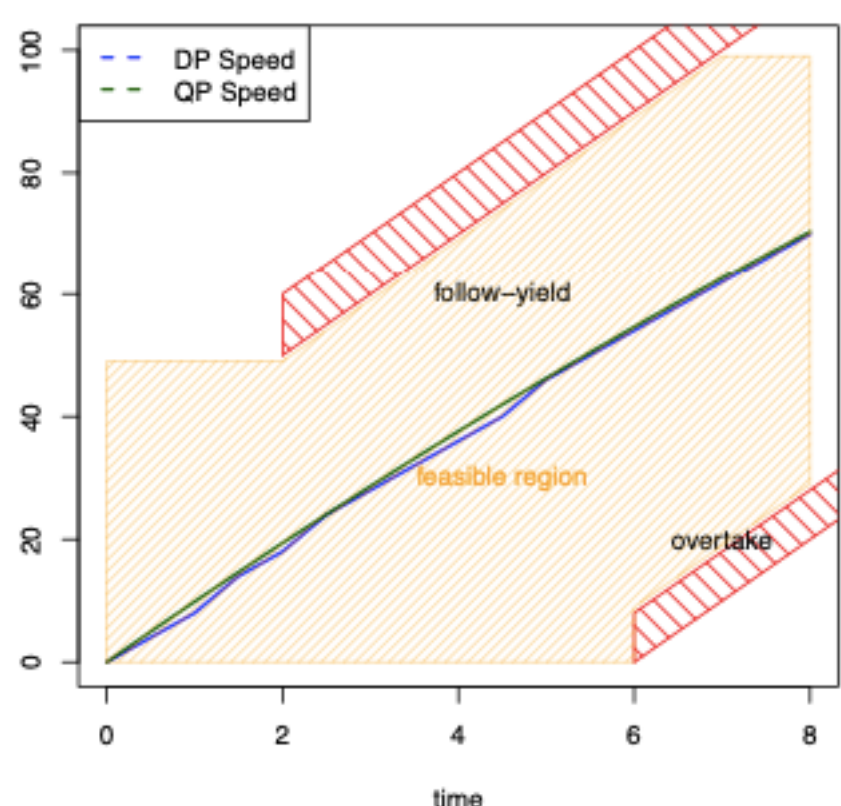
En prenant en compte la vitesse de déplacement du véhicule, on obtient un graphe de la fonction coût où la valeur 1 correspond à l'intégralité des coûts (100%).

$$\begin{aligned} f(s) + \sin(\theta)l_f + w/2 &\leq f(s) + f'(s)l_r + w/2 \\ &\leq l_{\text{left corner bound}} \end{aligned}$$

thêta représente un angle de déviation et plus  $\theta$  est petit et plus la marge d'erreur par rapport à la position d'un obstacle est faible. Là on juge le repérage par rapport à la diagonale gauche du véhicule, la même opération complémentaire s'opère sur la diagonale droite. Tous ces éléments déterminent le chemin optimal et sont essentiels à la sécurité du véhicule car elles sont le fruit de distances latérales.

$$\begin{aligned} f(s) + \sin(\theta)l_f + w/2 &\leq f(s) + f'(s)l_r + w/2 \\ &\leq l_{\text{left corner bound}} \end{aligned}$$

Finalement à partir de tous les éléments récupérés à un instant  $t$ , le système de deep learning construit une sorte de projection de l'environnement avec les surfaces possibles empruntées par le véhicule. Le chemin optimal (superposition de la ligne verte et bleue) est couplée à la vitesse, tous deux découlent du chemin prédit par le système de deep learning.



## BIBLIOGRAPHIE ET SOURCES

<https://arxiv.org/pdf/1807.08048v1.pdf>

<https://arxiv.org/pdf/2006.14480v2.pdf>

<http://www.diva-portal.org/smash/get/diva2:1366887/FULLTEXT01.pdf>

<https://www.youtube.com/watch?v=1L0TKZQcUtA>

<https://www.youtube.com/watch?v=Aut32pR5PQA>

[http://res-nlp.univ-lemans.fr/NLP\\_C\\_M01\\_G01/co/Contenu%2045.html](http://res-nlp.univ-lemans.fr/NLP_C_M01_G01/co/Contenu%2045.html)