Doc Laptime simulation

# Intro

Cette doc explique le fonctionnement global de la simulation, des docs spécifiques explique le fonctionnement des fonctions modélisant le comportement de la voiture en latérale et en longitudinale.

Toute cette simulation s’inspire de la vidéo de Matlab Racing Loung : <https://fr.mathworks.com/videos/matlab-and-simulink-racing-lounge-steady-state-lap-time-simulation-116109.html>

Globalement la simulation prend comme argument une liste de coordonnées X Y dessinant le circuit ainsi que tous les paramètres physiques de la nécessaires à a simulation véhicule.

L’idée est de calculer le diagramme GGV de la voiture, il contient les cercles de traction à différentes vitesses, dans le but ensuite de modéliser la voiture au maximum de ses performances à chaque instant sur le circuit. Cette méthode permet alors de s’affranchir d’un modèle pilote complexe à mettre en place.

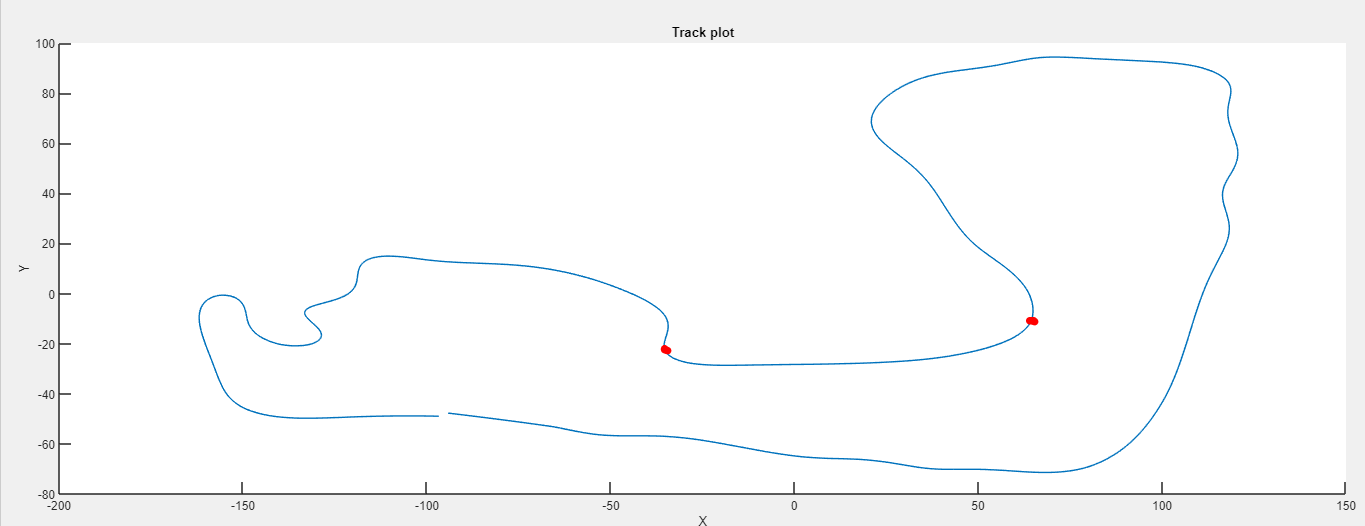
Un point important et le caractère **steady-state** de la simulation, on considère donc que la voiture est tout le temps à l’équilibre il n’y a pas de transitoire !

Dans le document le vert souligne de possibles amélioration à faire sur ce modèle.

# Import du circuit et segmentation

Pour la suite nous aurons besoin de la liste des rayons de courbure du circuit à chaque instant. Pour calculer ces rayons de courbure on utilise la fonction ‘LineCurvature2D’trouvé sur internet qui nous donne la courbure du tracé. La courbure correspond à l’inverse du rayon de courbure.

Pour la suite de la simulation l’emplacement des rayons de courbure minimaux doivent être connu. Cela nous permettra ensuite de découper le circuit en segment entre deux virages.



Sur la figure ci-dessus on peut voir deux point ayant des rayons de courbure correspondant à des minimums locaux.

Néanmoins comme on le voit sur la figure suivant la courbure du circuit est trop bruité pour obtenir une bonne segmentation :

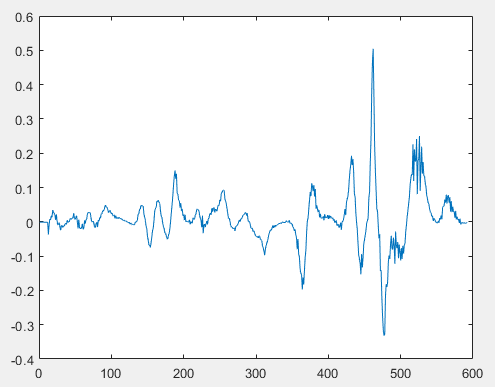
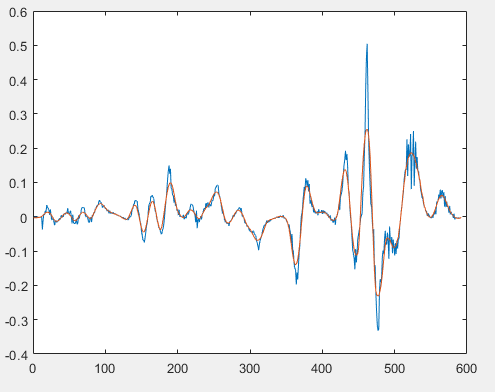


Figure : Tracé de la courbure non filtrée

On lisse donc la courbe avec la fonction ‘sgolayfilt’(ce filtre est cool mais il y a peut-être mieux).



On peut voir en orange la courbe filtrée. On voit que les valeurs pic ne correspondent plus néanmoins le pic à 0.5 serait un rayon de 2m ce qui est impossible selon le règlement et pour la voiture. Il faudrait donc vérifier si ce rayon ainsi lissé correspond encore à la réalité.

Une fois la courbure lissée obtenue on peut obtenir les points de courbures maximaux avec la fonction *extrema.* Cette fonction renvois les indices des extrémaux locaux.

On calcul ensuite le rayon qui est l’inverse de la courbure, on le sature entre -150m et 150m et on remplace les NaN (cas ou la courbure est nulle) par 150m.

On calcul aussi la valeur absolue du rayon pour éviter les erreurs de calcul, on viendra à la fin de la simulation pondérer l’accélération latérale par le signe de R pour s’y retrouver.

On calcul finalement le vecteur D contenant les distances entre chaque point du circuit.

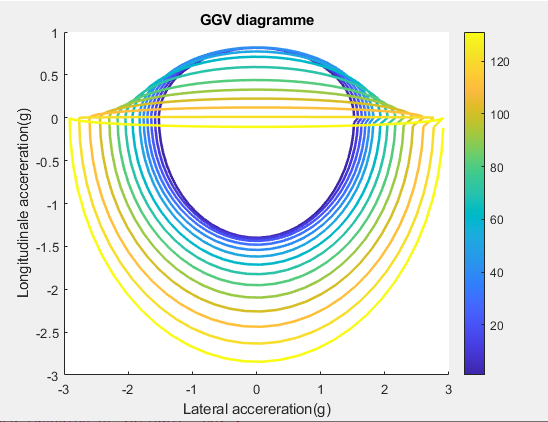
# Calcul du diagramme GGV

Avant de commencer notre boucle sur le circuit on calcule le diagramme GGV pour avoir le potentiel de la voiture. C’est la fonction *makeGGV* qui remplit cette fonction. La variable GGV est constituée de 5 colonne permettant d’avoir accès aux performances maximale de la voiture. L’idée est de pouvoir connaitre les accélérations maximales que la voiture peut fournir à une vitesse donnée

1. La colonne 1 contient l’accélération longitudinale positive
2. La colonne 2 contient l’accélération longitudinale négative
3. La colonne 3 contient l’accélération latérale max
4. La colonne 4 contient les vitesses de calculs des précédentes valeurs
5. La colonne 5 contient le rayon de courbure associé aux accélération latérale

On fait ici des calculs pour une accélération longitudinale pur ou une accélération latérale pur. Dans la suite on fera une approximation elliptique pour avoir un mix entre les deux. Une amélioration pourrait être de simuler des cas mix !

La fonction *plotGGV* permet d’afficher rapidement le diagramme GGV.



Vitesse (km/h)

# Boucle sur les segments

On parcourt ensuite chaque segment du circuit en déterminant la vitesse optimale sur chaque segment. Pour cela avec une initialisation on connait toujours la vitesse de départ du segment (0 pour le premier la vitesse finale de l’autre sinon) on connait calcul ensuite la vitesse maximale de fin de segment en considérant une accélération purement latérale lors de la corde du virage (ie la fin du segment).