

# Atelier de SY27

## Simulation et commande de la voiture pour réguler l'écart latéral

### 1. Objectifs pédagogiques

- Calculer un régulateur qui contrôle le volant d'une voiture de sorte à ce qu'elle suive un chemin défini par une polyligne
- Simuler un système de façon discrète et simuler un régulateur qui contrôle un système dynamique
- Analyser le comportement du régulateur sur une route infiniment droite puis sur une route linéaire par morceaux
- Etudier la robustesse du régulateur vis-à-vis d'erreurs de mesure et de saturations de commande.

### 2. Travail préparatoire

Revoir les cours sur

- la modélisation des véhicules à roues
- la commande de systèmes robotiques non holonomes

### 3. Travail à faire en séance

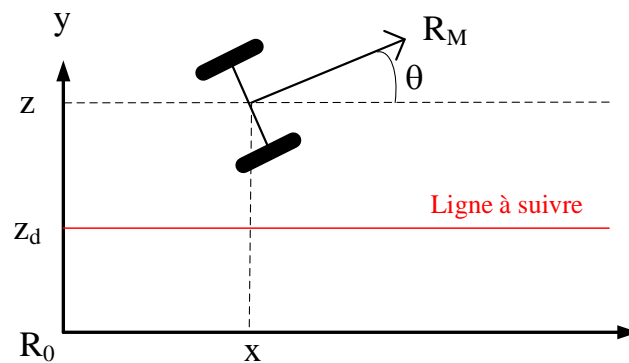
#### 3.1. Prise en main du kit de simulation Scilab

Téléchargez le kit et lancez le programme « tryme ». Il suffit dans l'éditeur « SciNotes » de faire Exécuter. Scilab est très proche de Matlab avec quelques différences mineures comme par exemple le fait que les figures sont toujours en mode « Hold on » et que les commentaires sont « // ».

En utilisant la fonction « draw\_voiture.sci », affichez une voiture aux coordonnées :  $x = -220$  et  $y = 360$ .

### 3.2. Cas d'une route infiniment droite

On reprend le problème de la commande d'une voiture le long d'une ligne horizontale.



Dans ce cas, l'abscisse n'est pas définie et on la choisit arbitrairement égale à 0.

On utilise une fonction spéciale d'affichage ici (draw\_voiture\_ligne.sci).

En vous inspirant du fichier tryme.sce, créez un nouveau script qui simulera la voiture asservie par son régulateur le long de la ligne.

#### 1. Modèle de la voiture

Créez dans le script des fonctions d'évolution et d'observation de la représentation d'état du système.

L'état est  $X=[y, \theta]^T$  et l'observation  $z_d$  (la distance signée à la ligne de consigne).

```
function x_dot=f(x,u,v,L)
endfunction
function z=g(x)
endfunction
```

#### 1. Boucle de simulation

Au début du script, il faut

- Définir les paramètres de la voiture
- initialiser l'état du système
- choisir la valeur de la consigne

La simulation se fait en temps discret.

Une façon de faire est la suivante :

```
duree_simu=5; //en secondes
Te=0.02; //période
t=[0:Te:duree_simu];
n=length(t);
for k=1:n,
    //la simulation se fait ici
end;
```

Pour afficher la voiture pendant la simulation, on peut faire :

```
drawlater();
delete(gca());
isoview(-6,6,-3,8)
draw_voiture_ligne(...)
drawnow()
```

Simulez d'abord votre voiture sans régulateur en discrétisant le système avec la méthode d'Euler.

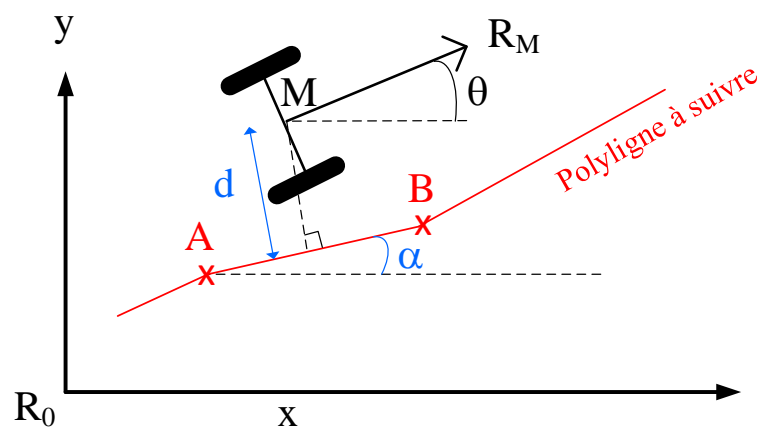
Ensuite, insérez dans la boucle la loi de commande du régulateur avec par exemple les pôles suivants :

```
s1=-2;s2=-2;
```

N'oubliez pas de gérer les saturations.

Enregistrez les signaux de commande et l'écart de sorte à les afficher dans de nouvelles fenêtres à la fin de la simu.

### 3.3. Cas d'une route linéaire par morceaux



On garde le régulateur défini précédemment et on l'utilise pour asservir la voiture à suivre un circuit. Dans la loi de commande, il faut faire attention à soustraire l'angle du segment et, pour éviter les problèmes d'angles apparemment différents alors qu'ils sont proches à  $2\pi$  près, on utilise la fonction « sub\_mod\_pi2 ».

Créez un nouveau script et une nouvelle fonction d'évolution.

Pour la fonction d'observation de la voiture, utilisez la fonction « gc.sci » qui implémente une étape de map-matching minimisant un critère de cap et de distance pour trouver le segment le plus proche du véhicule avec la bonne orientation. Elle retourne l'écart latéral dans le repère de Frenet et l'angle du segment.

Simulez la voiture sur le circuit en l'initialisant comme dans le fichier « tryme ».