

Sorbonne Université
Master 2 Automatique, Robotique
Parcours Systèmes Avancés et Robotique

Rapport TP1
Robotique mobile: Partie Modélisation/Commande

Suivi de chemin et application au contrôle en convoi

Auteurs : Viviane Bao-3800857
Hao Yuan-21117163

Année Universitaire 2023-2024
15 novembre 2023

Table des matières

I.	Introduction-----	p2
II.	Partie I : Contrôle en position du véhicule de tête-----	p2-p4
III.	Partie II: Contrôle en position et orientation du véhicule de tête-----	p4-p7
IV.	Partie III: Contrôle en convoi-----	p7-p10
V.	Conclusion-----	p10

I. Introduction

L'objectif de ce TP est de mettre en œuvre sous Matlab-Simulink des algorithmes de suivi de chemin pour des robots à roues, et d'illustrer l'application de ces algorithmes au contrôle de véhicules en convoi.

Les modèles Simulink pour les Parties I, II et III sont disponibles dans les versions 2019 et 2023. Pour la Partie I, il faut utiliser les fichiers *capteur1.m*, *anim1.m*. Pour les Parties II et III, il faut utiliser les fichiers *capteur2-3.m* et *anim2-3.m*.

II. Partie I : Contrôle en position du véhicule de tête

L'objectif de cette partie est de synthétiser un contrôleur pour le véhicule de tête afin de faire du suivi de chemin en position. On se donne une trajectoire de référence en position ($x_r(t)$, $y_r(t)$) associée à un point de référence P_r . On se donne un point de contrôle sur le véhicule (choisi sur l'axe central du véhicule, à une distance $D = 2m$ du centre des roues arrières), et on souhaite stabiliser asymptotiquement le point de contrôle vers la trajectoire de référence.

Pour calculer d_f et d_b , on utilise une méthode de dichotomie. En définissant $opt = 10^{-6}m$ comme la distance initiale à la courbe, l'algorithme explore un intervalle de -100 à 100 autour de la position x_c (l'abscisse cartésienne). La valeur *value* représente le carré de la distance à la courbe, mesurée par une norme vectorielle, et est enregistrée conjointement avec la valeur optimale de x_c si elle est plus petite que la précédente. La distance relative est finalement obtenue par la racine carrée en fonction de la position du véhicule par rapport à la courbe (au-dessus ou en dessous). Les coordonnées de la projection sont déduites de $X(f, proj) = (x_c^*, fx_c^*)$.

Pour définir une loi de commande permettant de réguler d_f à une valeur prédéfinie d_f^* , il existe une loi de commande très simple, mais avec une convergence locale et une hypothèse de courbure petite $c(s)$, qui assure seulement que d_f est ultimement petit (pas de garantie de convergence vers 0). Cependant, nous allons appliquer une extension à un contrôleur PI (approche plus complexe et robuste):

$$w_f = -k_1 v_f (d_f - d_f^*) - k_2 v_f \dot{I}_d \quad \text{avec} \quad \dot{I}_d = l_1 v_f (d_f - d_f^*)$$

Le calcul de \dot{I}_d se fait directement via Simulink en réalisant le bloc suivant :

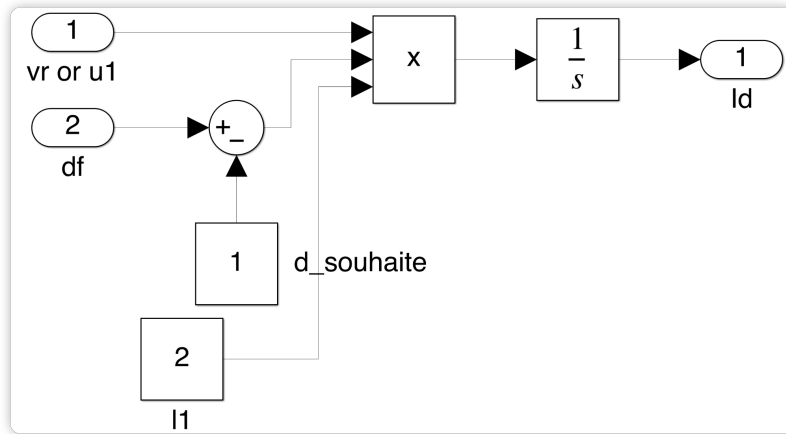


Figure 1: Bloc Simulink pour calculer I_d

I_d est ensuite récupérée dans le contrôleur, permettant ainsi de calculer la loi de commande w_f .

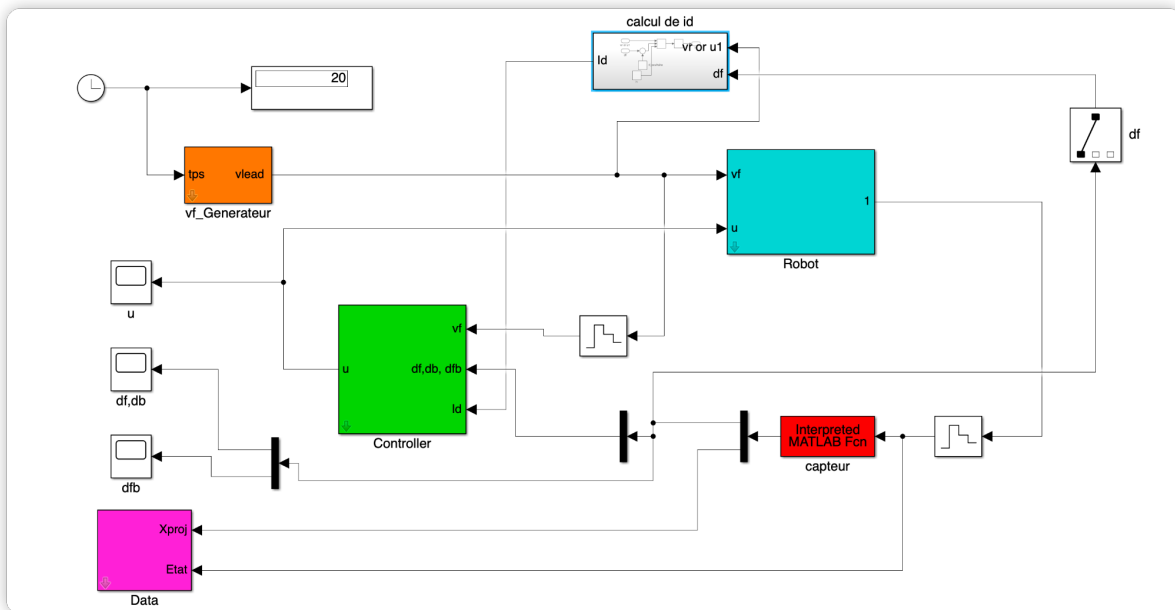
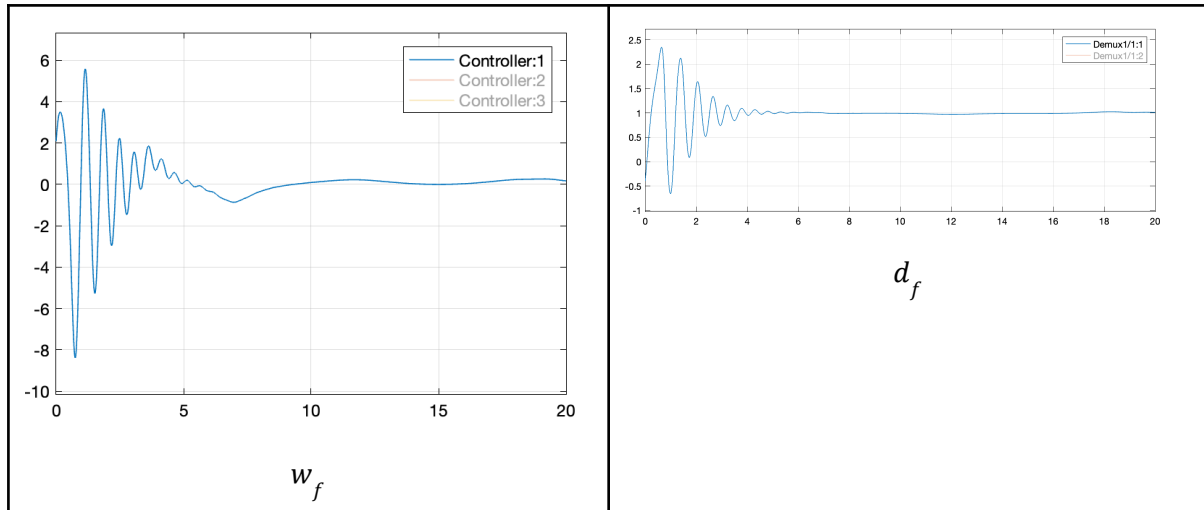


Figure 2: Bloc Simulink complet

❖ Pour $k_1 = 0.5$, $k_2 = 1$, $d_f^* = 1$ et une vitesse positive, voici les courbes obtenues pour w_f et d_f :



Nous observons des grands dépassements initiaux pour w_f avant de converger vers 0. Au début du processus de contrôle, le système peut réagir rapidement à la commande appliquée, ce qui entraîne des dépassements avant que le contrôle se stabilise vers la valeur souhaitée 0.

De façon analogue, une évolution similaire est observée pour d_f , avec une phase initiale de dépassements significatifs avant d'atteindre une stabilisation autour de la valeur 1 (d_f^*).

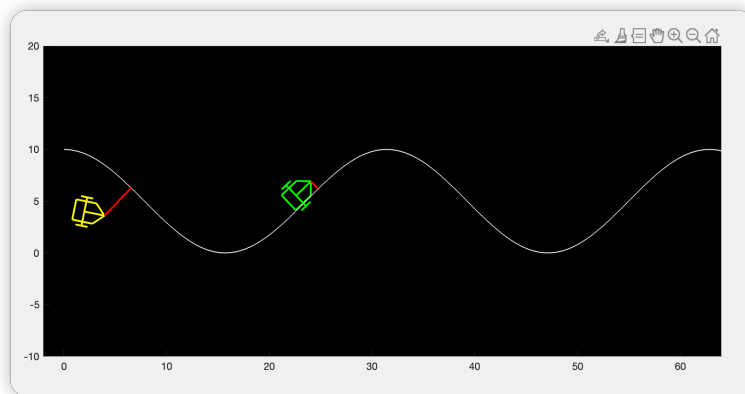


Figure 3: capture d'écran de l'animation (vitesse positive)

L'application de cette loi de commande à une vitesse linéaire du véhicule arbitraire aboutit à un recul du véhicule dans l'animation. Ce comportement suggère que la loi de commande n'assure plus la stabilité du système dans ce cas précis.

III. Partie II: Contrôle en position et orientation du véhicule de tête

Nous souhaitons à présent synthétiser un contrôleur pour le véhicule de tête afin de faire du suivi de chemin en position et orientation.

Pour ce faire, nous décalons le capteur, via le paramètre D , de façon à ce qu'il mesure la distance au chemin à partir du point situé au milieu de l'axe des roues arrière du véhicule. Dans le fichier *capteur.m*, nous devons remplacer les lignes $D = 2$ par $D = 0$. Dans le fichier *anim.m*, dans la partie *définition des rayons de mesure*, nous devons également remplacer la ligne $D = 2$ par $D = 0$.

Pour effectuer un chemin de suivi en position et orientation, nous avons besoin de connaître θ_e , qui représente l'erreur d'orientation entre la tangente à la courbe et l'orientation du véhicule. Une solution pour estimer θ_e est basée sur le linéarisé en faisant l'hypothèse de courbure nulle :

$$\begin{cases} \dot{\hat{d}_f} = v_f \hat{\theta}_e - k_1 |v_f| (\hat{d}_f - d_f) \\ \dot{\hat{\theta}_e} = w_f - k_2 |v_f| (\hat{d}_f - d_f) \end{cases} \quad (*)$$

Il existe une loi de commande très simple, mais avec une convergence locale et une hypothèse de courbure petite $c(s)$, qui assure seulement que \hat{d}_f est ultimement petit (pas de garantie de convergence vers 0). Nous avons décidé de choisir l'extension PID avec :

$$w_f = -k_1 v_f (d_f - d_f^*) - k_2 |v_f| \theta_e - k_3 |v_f| I_d \quad \text{avec } \dot{I}_d = v_f (d_f - d_f^*)$$

Ensuite, nous mettons en place un observateur sur Simulink permettant d'estimer θ_e et de calculer I_d :

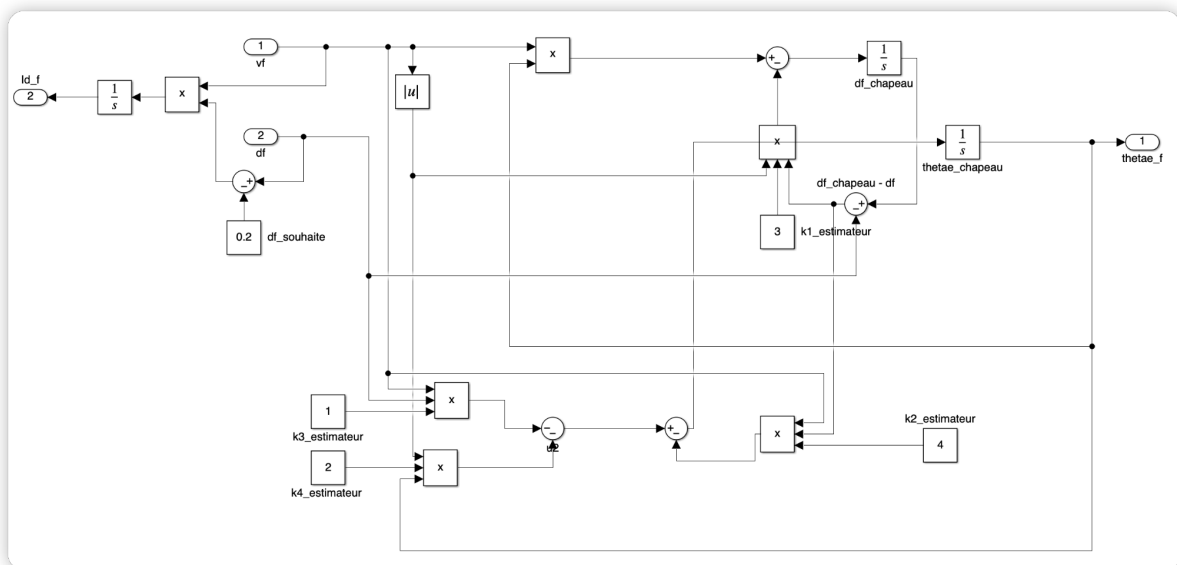


Figure 4: Bloc observateur pour estimer θ_e et calculer I_d

Dans ce bloc, nous avons mis en place le système dynamique (*) . Ici, w_f utilise les paramètres internes de l'observateur ($k_{1\text{ estimateur}}, k_{2\text{ estimateur}}, k_{3\text{ estimateur}}, k_{4\text{ estimateur}},$) . La structure de calcul pour I_d suit la même logique que celle mentionnée dans la Partie I. Ainsi, nous pouvons récupérer la valeur estimée de θ_e et la valeur de I_d dans la contrôleur, permettant donc de calculer la loi de commande définie précédemment.

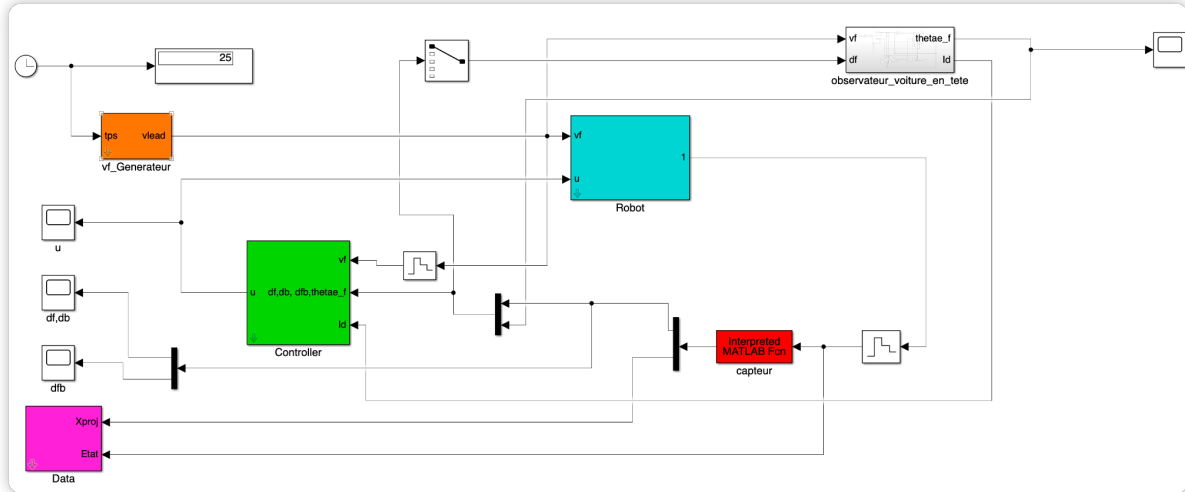
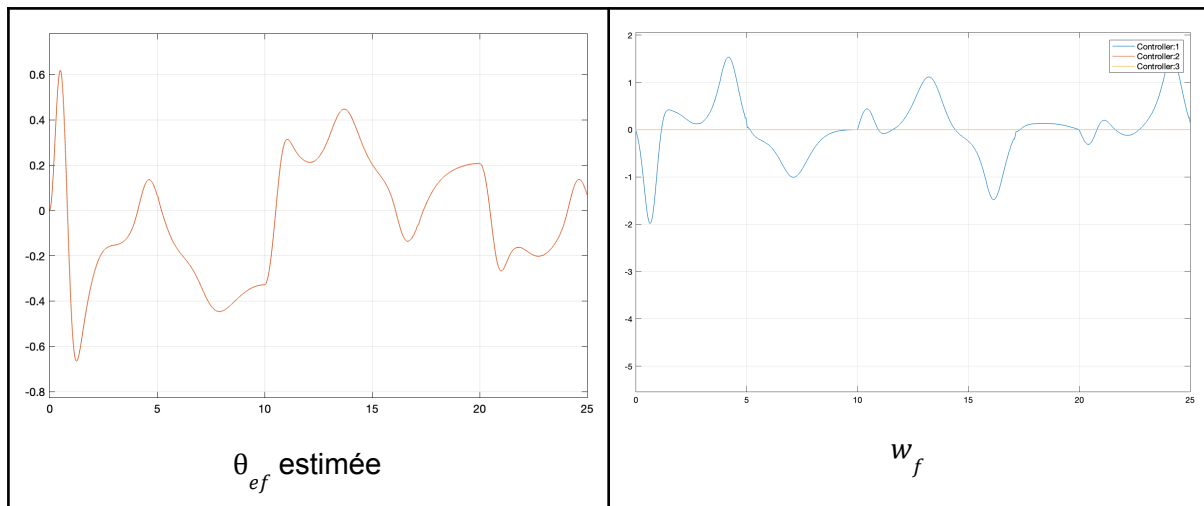
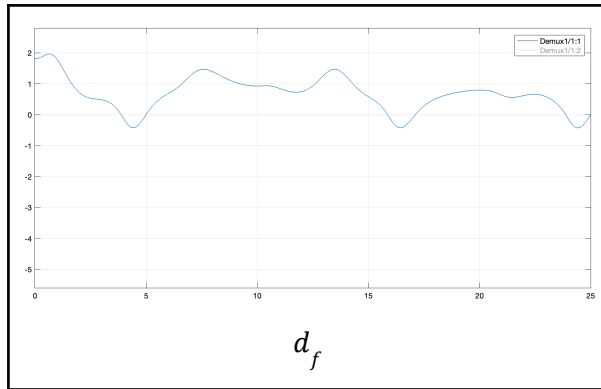


Figure 5: Bloc Simulink complet

❖ Pour $k_1 = 0.5, k_2 = 1, k_3 = 0.1, d_f^* = 0.2$, et une vitesse arbitraire, voici les courbes obtenues pour θ_{ef} estimée, w_f et d_f :





Les oscillations observées sur la courbe de d_f peuvent résulter d'un choix inadéquat des gains, ce qui entraîne un temps de réponse prolongé. Cela signifie que le contrôleur met du temps à ramener d_f vers la valeur souhaitée, ce qui peut conduire à des oscillations prolongées avant la stabilisation.

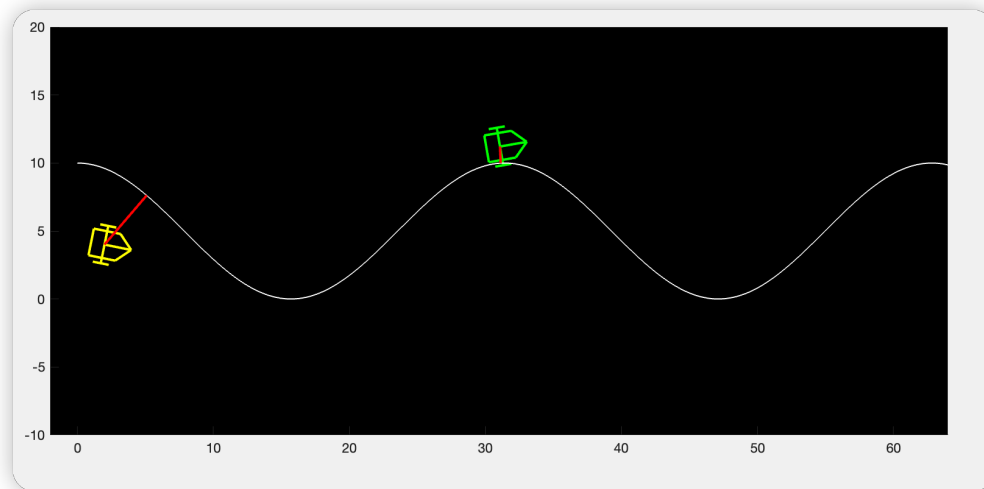


Figure 6: capture d'écran de l'animation (vitesse arbitraire)

Dans les deux scénarios, que ce soit avec une vitesse positive ou une vitesse arbitraire, la loi de commande garantit la stabilité du système, ce qui n'était pas le cas avec la solution de la Partie I. Effectivement, dans la première partie où seul le contrôle de la position était implémenté, la robustesse du système pour des vitesses arbitraires n'était pas garantie. En revanche, dans la deuxième partie où la commande était étendue pour contrôler à la fois la position et l'orientation, la stabilité du système était assurée même pour des vitesses arbitraires. Cette extension de la loi de commande a renforcé la capacité du système à maintenir sa stabilité dans des conditions plus variées et plus complexes.

IV. Partie III: Contrôle en convoi

Nous voulons maintenant étendre la solution développée de sorte que le véhicule suiveur suive le chemin et régule sa distance au véhicule de tête à une valeur prédéfinie d_{fb}^* .

Afin d'y parvenir, nous devons utiliser la même approche adoptée dans la Partie II ; la loi de commande pour contrôler le véhicule suiveur ainsi que la structure de l'observateur associée demeurent identiques à celles pour le véhicule de tête.

$$w_b = -k_4 v_b (d_b - d_b^*) - k_5 |v_b| \theta_{eb} - k_6 |v_b| I_{db}$$

$$\text{avec } \dot{I}_{db} = v_b (d_b - d_b^*)$$

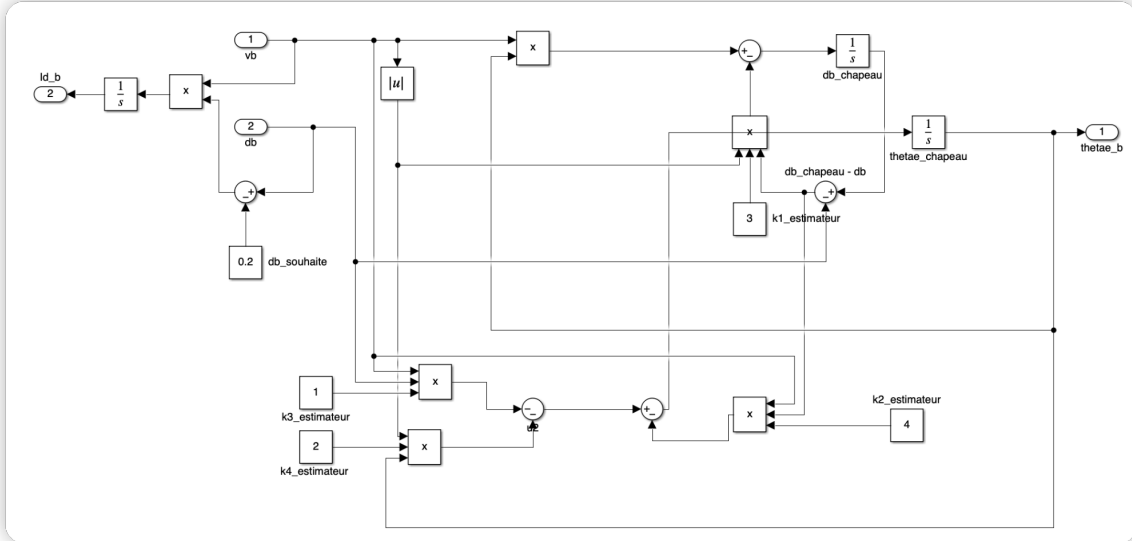


Figure 7: Bloc observateur du véhicule suiveur

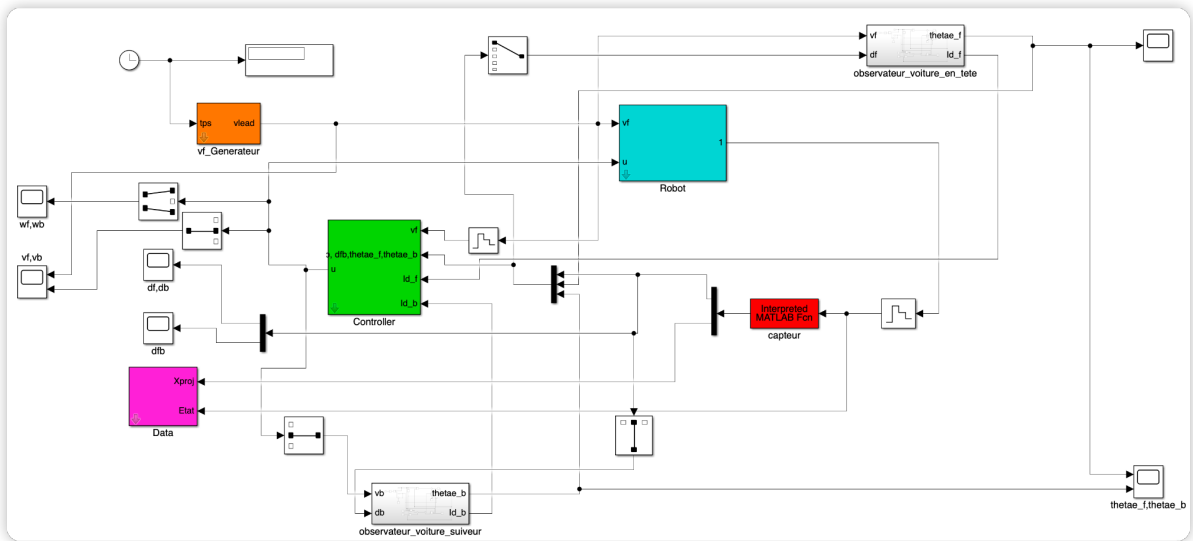


Figure 8: Bloc Simulink complet

Ici, nous devons définir v_b . Nous avons $d_{fb} = v_f - v_b$
 $= -k_7 (d_{fb} - d_{fb}^*)$

$$\text{Ainsi, } v_b = v_f + k_7 (d_{fb} - d_{fb}^*)$$

Pour garantir le contrôle en convoi des deux véhicules, nous avons ajusté le troisième gain de la loi de commande pour le véhicule de tête, qui est désormais de $k_3 = 0.05$.

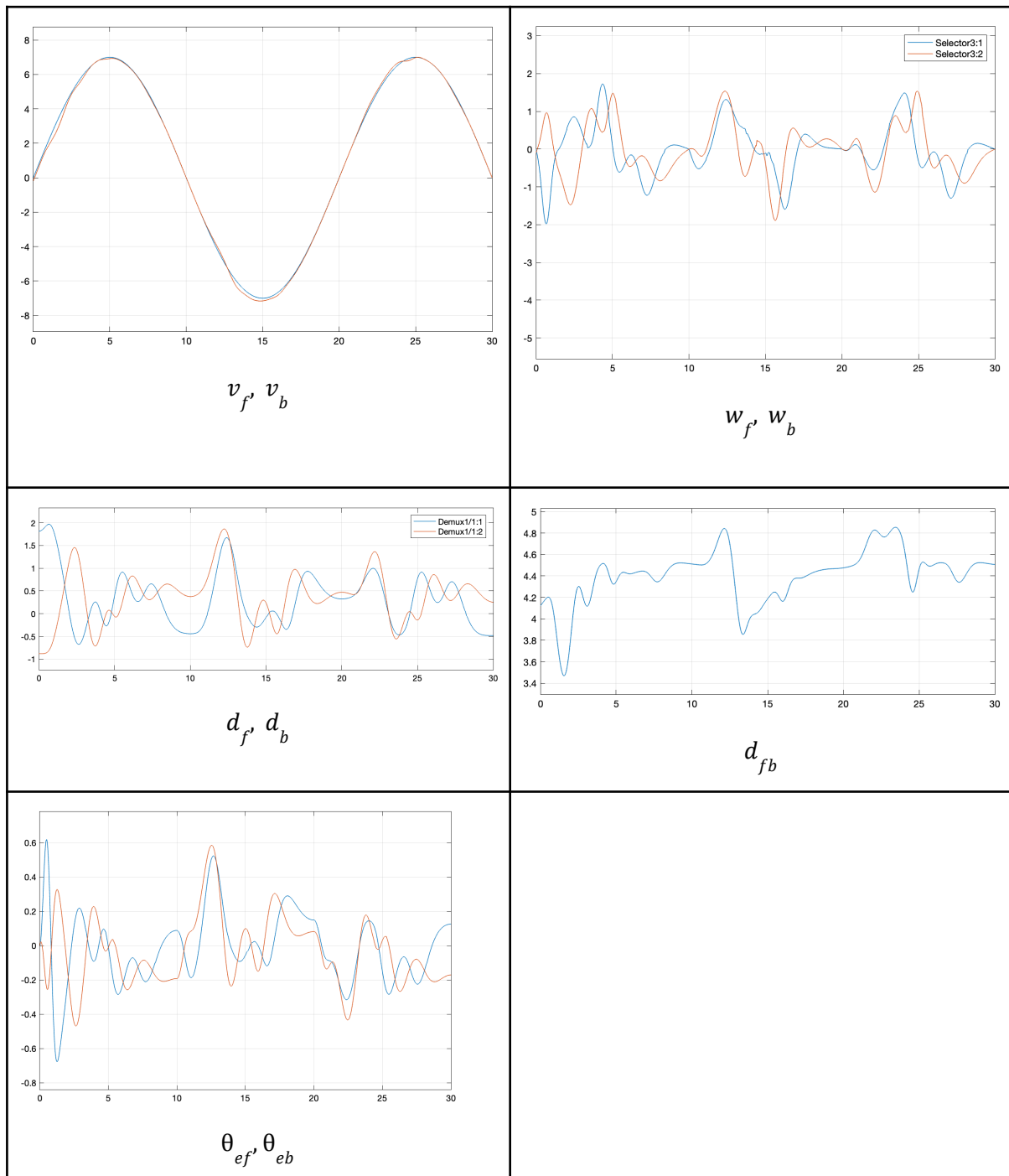
Les gains de la loi de commande pour le véhicule suiveur sont définis comme suit :

$$k_4 = 0.5, k_5 = 1, k_6 = 0.05$$

$$k_7 = 0.5$$

Par ailleurs, nous avons imposé la valeur de d_b^* à 0.2 et celle de d_{fb}^* à 4.5.

Nous obtenons les courbes suivantes pour une vitesse arbitraire :



D'après les résultats obtenus, nous pouvons voir que les deux véhicules ont un comportement similaire. De plus, d_{fb} tend vers d_{fb}^* avec des oscillations. Cela pourrait résulter d'un choix de gains de commande sous-optimal pour les deux lois de contrôle, entraînant ainsi des oscillations lors de la tentative de correction du système.

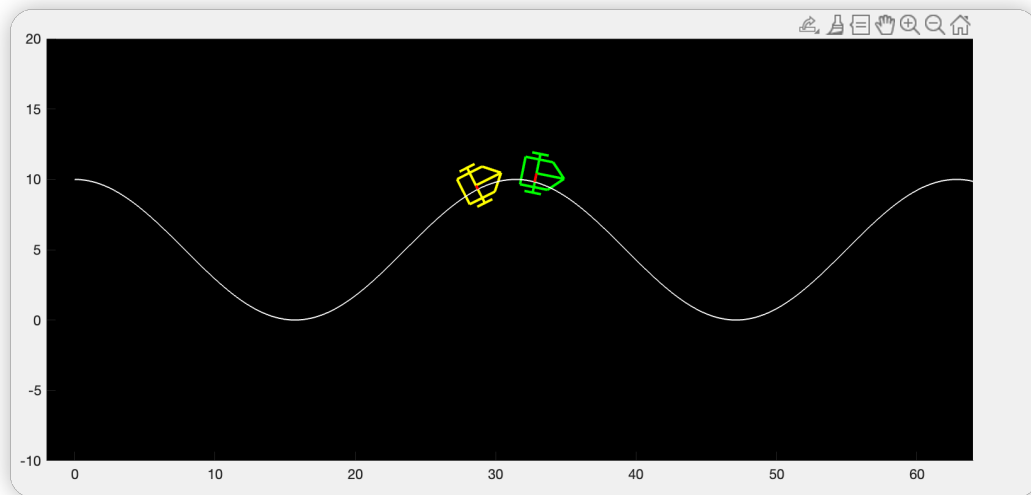


Figure 9 : capture d'écran de l'animation

V. Conclusion

Ce TP a mis en lumière l'importance capitale du choix et de l'ajustement précis des paramètres de contrôle dans la conception de lois de commande pour les systèmes de véhicules. Nous avons exploré comment des réglages inadaptés peuvent entraîner des oscillations et affecter la stabilité du système.

L'analyse des diverses stratégies de contrôle a démontré que la prise en compte simultanée de la position, de l'orientation est cruciale pour assurer une stabilité robuste du système.