

Mécatro: Rapport d'automatique

Groupe 7: Royer Jules, Simon Noah-Luc, Colin Matthieu, Bourderieux Armand

Contents

1	Introduction	2
2	Création du contrôleur théorique	3
3	Identification des paramètres physiques	9
4	Implémentation du contrôleur en Arduino	11

Chapter 1

Introduction

Le but de la partie automatique du projet de mécatronique est de créer un contrôleur pour un robot "bolide" suiveur de ligne. Le contrôleur a d'abord été créé théoriquement dans le logiciel Matlab et testé sur un modèle de simulation Simulink, puis il a été implémenté informatiquement en Arduino afin d'être embarqué dans le robot.

Chapter 2

Création du contrôleur théorique

Contrôle d'un mouvement rectiligne uniforme En reprenant les notations et la modélisation du document "Equations de la dynamique du Segway", les équations de la dynamique sous forme d'état sont (on négligera toutes les perturbations):

$$\begin{cases} \dot{p} = u \\ \dot{u} = \frac{1}{\beta} \left(\frac{1}{\rho} k I^+ - m_b d v^2 \right) \\ \dot{\psi} = v \\ \dot{v} = \frac{1}{\gamma} \left(\frac{lk}{2\rho} I^- + m_b d u v \right) \\ \dot{I}^+ = \frac{U^+}{L} - \frac{R}{L} I^+ - \frac{2k}{L\rho} u \\ \dot{I}^- = \frac{U^-}{L} - \frac{R}{L} I^- - \frac{kl}{L\rho} v \\ \dot{y} = u \sin \psi \end{cases}$$

Les entrées sont U^+ , U^- .

Les sorties mesurées sont:

$$\begin{cases} \delta\phi^+ = \delta\phi_{right} + \delta\phi_{left} = \frac{2}{\rho} p \\ \delta\phi^- = \delta\phi_{right} - \delta\phi_{left} = \frac{l}{\rho} \psi \\ c_{LF} = \frac{y}{\cos \psi} \approx y \end{cases}$$

Où on a défini:

- p la distance curviligne parcourue par le robot le long de sa trajectoire. Celle-ci doit figurer dans l'état pour mesurer les angles cumulés.
- u sa vitesse longitudinale.
- ψ l'angle entre l'axe horizontal x et la direction du robot.
- $I^- = I_{right} - I_{left}$ la somme des courants des moteurs.
- $I^- = I_{right} - I_{left}$ la différence des courants.
- $U^+ = U_{right} + U_{left}$ la somme des tensions aux bornes des moteurs.
- $U^- = U_{right} - U_{left}$ la différence des tensions.
- $\beta = M + \frac{2I_y^w}{\rho^2}$
- $\gamma = I_\psi + m_b d^2$
- ϕ^+ la somme des angles des roues. Posons $\alpha = \frac{2}{\rho}$.

- ϕ^- la différence des angles des roues. Posons $\eta = \frac{l}{\rho}$.
- c_{LF} la mesure de l'écart entre le point A et la ligne (approximation en MRU horizontal).

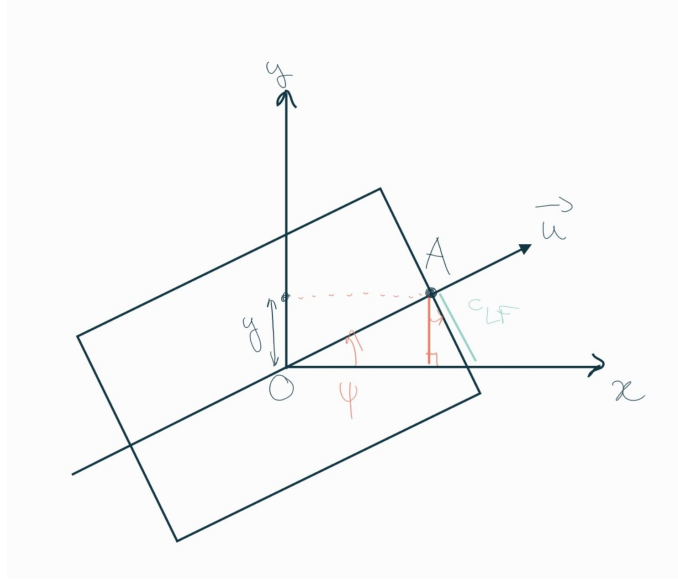


Figure 2.1: Ecart à la ligne dans l'hypothèse d'un faible angle.

Recherche des trajectoires d'équilibre Notre trajectoire d'équilibre du mouvement rectiligne uniforme est caractérisée par le système:

$$\begin{cases} u_0 = \bar{u} \\ 0 = \frac{1}{\beta} \left(\frac{1}{\rho} k \bar{I}^+ - m_b d \bar{v}^2 \right) \\ 0 = \bar{v} \\ 0 = \frac{1}{\gamma} \left(\frac{lk}{2\rho} \bar{I}^- + m_b d \bar{u} \bar{v} \right) \\ 0 = \frac{\bar{U}^+}{L} - \frac{R}{L} \bar{I}^+ - \frac{2k}{L\rho} \bar{u} \\ 0 = \frac{\bar{U}^-}{L} - \frac{R}{L} \bar{I}^- - \frac{kl}{L\rho} \bar{v} \\ 0 = \bar{u} \sin \bar{\psi} \end{cases}$$

Avec les sorties linéarisées:

$$\begin{cases} \delta\phi^+ = \alpha\delta p \\ \delta\phi^- = \eta\delta\psi \\ \delta c_{LF} = \delta y \end{cases}$$

Ce qui donne comme trajectoire d'équilibre

$$\begin{cases} \underline{x} = (\bar{p} = u_0 t, \bar{u} = 0, \bar{\psi} = \psi_0 = 0, \bar{v} = 0, \bar{I}^\pm = 0, \bar{y} = 0) \\ \underline{e} = (\bar{U}^+ = \frac{2k}{\rho} u_0, \bar{U}^- = 0) \end{cases}$$

Linéarisé autour de l'équilibre

$$\begin{cases} \delta \dot{p} = \delta u \\ \delta \dot{u} = \frac{1}{\beta} \frac{1}{\rho} k \delta I^+ \\ \delta \dot{\psi} = \delta v \\ \delta \dot{v} = \frac{1}{\gamma} \left(\frac{jk}{2\rho} \delta I^- + m_b du_0 \delta v \right) \\ \delta \dot{I}^+ = \frac{\delta U^+}{L} - \frac{R}{L} \delta I^+ - \frac{2k}{L\rho} \delta u \\ \delta \dot{I}^- = \frac{\delta U^-}{L} - \frac{R}{L} \delta I^- - \frac{kl}{L\rho} \delta v \\ \dot{y} = u_0 \cos \psi_0 \delta \psi + \delta u \sin \psi_0 = u_0 \delta \psi \end{cases}$$

$$\text{eigSys} = \begin{pmatrix} -941.08 \\ -884.05 \\ -72.476 \\ -14.833 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figure 2.2: Valeurs propres du système linéarisé.

Les valeurs propres obtenus sont négatives. Celles qui sont nulles sont dues au fait que certaines variables d'état sont redondantes: p , y , donc le système reste stable.

Simplifications par perturbations singulières Les intensités de courant ont un transitoire bien plus rapide que ceux des grandeurs mécaniques (leurs valeurs propres $\frac{R}{L}$ sont bien supérieures aux autres).

On applique la méthode des perturbations singulières à:

$$\begin{cases} LI^+ = U^+ - RI^+ - \frac{2k}{\rho} u \\ LI^- = U^- - RI^- - \frac{kl}{\rho} v \end{cases}$$

Avec $\varepsilon = L$ "très petit". En tout rigueur il faudrait introduire L_0 pour adimensionner ε .

On obtient:

$$\begin{cases} I^+ = \frac{1}{R} \left(U^+ - \frac{2k}{\rho} u \right) \\ I^- = \frac{1}{R} \left(U^- - \frac{kl}{\rho} v \right) \end{cases}$$

On vérifie la stabilité uniformément exponentielle de la branche d'équilibre avec :

$$\partial_{I^\pm} \begin{pmatrix} \frac{U^+}{L} - \frac{R}{L} I^+ - \frac{2k}{L\rho} u \\ \frac{U^-}{L} - \frac{R}{L} I^- - \frac{kl}{L\rho} v \end{pmatrix} \Big|_{I^\pm = f(x, \varepsilon)} = -\frac{R}{L} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

La branche a donc comme valeurs propres des réels négatifs, donc est ue-stable.

Système lent simplifié

$$\begin{cases} \delta\dot{p} = \delta u \\ \delta\dot{u} = \underbrace{\frac{1}{\beta} \frac{k}{\rho} \frac{1}{R}}_Q (\delta U^+ - \frac{2k}{\rho} \delta u) \\ \delta\dot{\psi} = \delta v \\ \delta\dot{v} = \underbrace{\frac{1}{\gamma} m_b d u_0}_{b_1} \delta v + \underbrace{\frac{1}{\gamma} \frac{lk}{2\rho} \frac{1}{R}}_{b_2} (\delta U^- - \frac{lk}{\rho} \delta v) \\ \dot{y} = u_0 \delta\psi \end{cases}$$

Découplage en sous-système "Sum" et "Dif" On peut séparer le système global en 2 sous-systèmes indépendants. Le premier, appelé "Sum", régit la dynamique du bolide en ligne droite:

$$\text{Etat} \begin{cases} \delta\dot{p} = \delta u \\ \delta\dot{u} = Q(\delta U^+ - \frac{2k}{\rho} \delta u) \end{cases} \quad \text{Sorties} \begin{cases} \delta y_m^+ = \delta\phi^+ \end{cases}$$

Le second, appelé "Dif", régit la dynamique du bolide dans les virages:

$$\text{Etat} \begin{cases} \delta\dot{\psi} = \delta v \\ \delta\dot{v} = b_1 \delta v + b_2 (\delta U^- - \frac{lk}{\rho} \delta v) \\ \dot{y} = u_0 \delta\psi \end{cases} \quad \text{Sorties} \begin{cases} \delta y_m^- = \delta\phi^- \\ \delta c_{LF} = \delta y \end{cases}$$

On obtient les valeurs propres suivantes:

$\text{eigSysSum01} = \begin{pmatrix} -66.985 & \\ & 0 \end{pmatrix}$	$\text{eigSysDif01} = \begin{pmatrix} -14.593 & \\ & 0 \\ & & 0 \end{pmatrix}$
---	--

Figure 2.3: Valeurs propres des sous-systèmes linéarisés. Environ égales aux plus lentes du linéarisé global.

Modélisation du contrôleur du sous-système "Sum" L'état étant de dimension 2, notre choix s'est porté sur un Proportionnel - Dérivé permettant de contrôler U^+ avec ϕ^+ :

$$\delta U^+ = h_1(\delta\phi_r^+ - \delta\phi^+) - h_2(\delta\dot{\phi}_r^+ - \delta\dot{\phi}^+) = h_1\alpha(\delta p_r - \delta p) - h_2\alpha(\delta u_r - \delta u)$$

$$\text{En Laplace: } \underline{U}^+ = \frac{-h_2 s + h_1}{1} (\underline{\phi}_r^+ - \underline{\phi}^+)$$

Pour ajuster le comportement de notre système en boucle fermée, nous imposons le polynôme caractéristique de $A^+ = \partial_{\underline{x}^+} f$ où f^+ est la fonction de la dynamique de l'état \underline{x}^+ :

$$\chi_{A^+ \text{ désiré}} = s^2 + 2\xi\omega_0 s + \omega_0^2$$

Où l'amortissement ξ et la pulsation propre ω_0 sont à ajuster.

En boucle fermée:

$$\begin{pmatrix} \delta\dot{p} \\ \delta\dot{u} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ \underbrace{-Q\alpha h_1}_{a_1} & \underbrace{Q(\alpha h_2 - \frac{2k}{\rho})}_{a_2} \end{pmatrix}}_{A^+} \begin{pmatrix} \delta p \\ \delta u \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ Q\alpha h_1 & -Q\alpha h_2 \end{pmatrix} \begin{pmatrix} \delta p_r \\ \delta u_r \end{pmatrix} \Rightarrow \chi_{A^+} = s^2 - a_2 - a_1$$

$$\text{Identification} \begin{cases} -a_2 = 2\xi\omega_2 \\ -a_1 = \omega_0^2 \end{cases} \Leftrightarrow \begin{cases} h_2 = \frac{1}{\alpha} \left(\frac{2\xi\omega_0}{-Q} + \frac{2k}{\rho} \right) \\ h_1 = \frac{\omega_0^2}{Q\alpha} \end{cases}$$

Modélisation du contrôleur du sous-système "Dif" Pour contrôler l'état de dimension 3, on a choisit un Proportionnel - Dérivé sur la différence des angles cumulés ϕ^- et un proportionnel sur l'écart à la ligne c_{LF} pour contrôler U^- :

$$\delta U^- = k_1(\delta\phi_r^- - \delta\phi^-) - k_2(\delta\dot{\phi}_r^- - \delta\dot{\phi}^-) + k_5(\delta c_{LFr} - \delta c_{LF}) = k_1\eta(\delta\psi_r - \delta\psi^-) - k_2\eta(\delta\dot{\psi}_r - \delta\dot{\psi}^-) + k_5(\delta y_r - \delta y)$$

$$\text{En Laplace: } \underline{U}^- = \frac{-k_2s + k_1}{1}(\underline{\phi}_r^- - \underline{\phi}^-) + \frac{k_5}{1}(\underline{c}_{LFr} - \underline{c}_{LF})$$

Pour ajuster le comportement de notre système en boucle fermée, nous imposons:

$$\chi_{A^-} \text{ désiré} = s^3 + \sigma \sqrt{6}s^2 + \sigma^2 \sqrt{6}s + \sigma^3$$

En boucle fermée:

$$\begin{cases} \delta\dot{\psi} = \delta v \\ \delta\dot{v} = \underbrace{-k_1 b_2 \eta \delta \psi}_{c_1} + \underbrace{\left(b_1 + b_2 k_2 \eta - \frac{b_2 k l}{\rho}\right) \delta v}_{c_2} - \underbrace{b_2 k_5 \delta y}_{c_3} \end{cases}$$

On a

$$A^- = \begin{pmatrix} 0 & 1 & 0 \\ c_1 & c_2 & c_3 \\ u_0 & 0 & 0 \end{pmatrix}$$

On identifie

$$\begin{aligned} \chi_{A^-} &= s^3 - c_2 s - c_1 s - u_0 c_3 = s^3 + \sigma \sqrt{6} s^2 + \sigma^2 \sqrt{6} s + \sigma^3 \\ &\Leftrightarrow \begin{cases} k_1 = \frac{1}{b_2 \eta} \sigma^2 \sqrt{6} \\ k_2 = \frac{1}{b_2 \eta} \left(-\sigma \sqrt{6} - b_1 + b_2 \frac{k l}{\rho} \right) \\ k_5 = \frac{\sigma^3}{u_0 b_3} \end{cases} \end{aligned}$$

Approximation des contrôleurs avec des dérivées filtrées Pour le sous-système "Sum", le contrôleur en dérivée filtrée s'écrit:

$$\begin{cases} \dot{\eta} = \frac{\Delta\phi^+ - \eta}{\varepsilon} = n_s (\Delta\phi^+ - \eta) \text{ où } \Delta\phi^+ = \delta\phi_r^+ - \delta\phi^+ \text{ et } n_s = \frac{1}{\varepsilon} \\ \delta U^+ = h_1 (1 - T_{ds} n_s) \Delta\phi^+ + h_1 T_{ds} n_s \eta \text{ où } T_{ds} = \frac{h_2}{h_1} \end{cases}$$

De même pour le sous-système "Dif":

$$\begin{cases} \dot{\eta} = \frac{\Delta\phi^- - \eta}{\varepsilon} = n_d (\Delta\phi^- - \eta) \text{ où } \Delta\phi^- = \delta\phi_r^- - \delta\phi^- \text{ et } n_d = \frac{1}{\varepsilon} \\ \delta U^- = k_1 (1 - T_{dd} n_d) \Delta\phi^- + k_1 T_{dd} n_d \eta + k_5 \Delta c_{LF} \text{ où } T_{dd} = \frac{k_2}{k_1} \text{ et } \Delta c_{LF} = \delta c_{LFr} - \delta c_{LF} \end{cases}$$

Réponses fréquentielles des systèmes fermés modélisés en Matlab Pour le système "Sum", nous ne voulions pas trop accélérer la dynamique, ainsi nous avons pris $\omega_0 = |\lambda^+|$ la valeur propre du système, et l'amortissement $\xi = \frac{1}{\sqrt{2}}$, afin d'avoir en boucle fermée (avec $n_s = 150$):

Pour le sous-système "Dif", nous avons accéléré la dynamique avec $\sigma = 15$

<code>eigSysSumBF =</code>		<code>eigSysSumBFApprox =</code>	
		$-35.416 +$	$0i$
$-47.365 +$	$47.365i$	$-40.784 +$	$68.347i$
$-47.365 -$	$47.365i$	$-40.784 -$	$68.347i$

Figure 2.4: Valeurs propres de "Sum" avec contrôleur PD, (gauche: contrôleur idéal, droite: dérivée filtrée). Elles sont inférieures à celles en boucle ouverte.

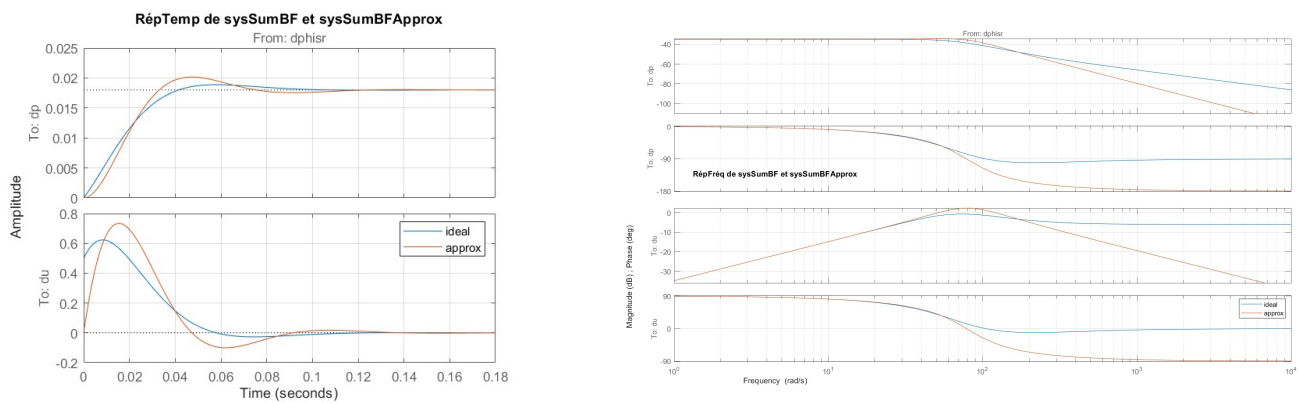


Figure 2.5: Réponses temporelles et fréquentielles du système "Sum" contrôlé. Avec le contrôleur avec dérivée filtrée, la réponse oscille un peu et est en retard. La réponse sur u à une marge de phase de 180° et une marge de gain infinie ce qui est satisfaisant.

Chapter 3

Identification des paramètres physiques

Nous avons rassemblé toutes les valeurs numériques dans ce Google Sheet.

Estimation de la constante de couple k Les équations électriques à l'équilibre s'écrivent

$$\begin{cases} 0 = U - RI - k\Omega \\ \tau = kI \end{cases}$$

En se plaçant sans couple, donc sans courant, et en utilisant les courbes constructeurs, on calcule $k = \frac{U}{\Omega} \text{ V}/(\text{rad.s}^{-1})$.



Figure 3.1: Courbes de fonctionnement fournies par Pololu.

Estimation des masses des pièces Toutes les pièces ont été pesées.

On a approximé la masse de l'arbre moteur et du rotor à la différence entre la masse pesée du moteur, et celle donnée par la CAO.

Estimation des moments d'inertie Pour le moment d'inertie de l'unité roue + engrenage + rotor, notre première approche a été d'approximer la géométrie du système à celle d'un disque, et d'utiliser la matrice d'inertie (cellule D34 du GSheet):

$$I_w(C_w) = \begin{pmatrix} m(\rho^2/4 + l_w^2/2) & 0 & 0 \\ 0 & m(\rho^2/4 + l_w^2/2) & 0 \\ 0 & 0 & m(R^2/4 + l^2/12) \end{pmatrix}$$

Comme autre approche, nous avons mesuré le temps de réponse de la vitesse angulaire de la roue à un échelon de 12V aux bornes du moteur. Nous avons mesuré la somme des positions angulaires, que nous avons filtrée avec un filtre passe bas, en supposant que cela ne fausse pas sensiblement le temps de réponse.

Les données constructeurs fournissent une relation linéaire entre la vitesse angulaire et le couple d'entrée.

$$\omega = a - b\tau$$

On a essayé d'exploiter le théorème du moment cinétique à l'unité roue+rotor:

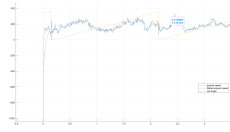


Figure 3.2: Vitesse angulaire (degrés/s) de la roue en fonction du temps (s).

$$I_w^y \dot{\omega} = \tau = \frac{a}{b} - \frac{\omega}{b}$$

$$\dot{\omega} + \frac{\omega}{T} = \frac{a}{T} \text{ où } T = I_w^y b$$

Estimation des constantes électriques La résistance et l'inductance interne du moteur ont été trouvées sur Internet ici.

Chapter 4

Implémentation du contrôleur en Arduino

Structure Après discussion avec M. Di Meglio, le contrôleur que nous voulions implémenter en Arduino se composait de :

- Un contrôleur PD sur le sous-système "Sum" prenant en entrée la mesure de la somme des angles, renvoyant la somme des tensions des deux moteurs, et qui régit la vitesse en translation
- Un contrôleur PD sur le sous-système "Diff", prenant en entrée la mesure de la distance par rapport à la ligne (donnée par le Line Follower Array), renvoyant la différence des tensions des deux moteurs, et qui régit la vitesse en rotation.

Contrôle de l'écart à la ligne Nous avons d'abord implémenté un contrôleur P sur le sous-système "Diff" en imposant la somme des tensions (ce qui revient à imposer une vitesse longitudinale en ligne droite). Le code est disponible dans le fichier TODO: INSERER FICHIER

Après avoir effectué cette implémentation et avoir trouvé des gains satisfaisants, nous avons implémenté le contrôleur PD sur le sous-système "Diff", puis réglé les gains de ce contrôleur à partir d'observations de la performance du robot sur le circuit. Voici quelques questions qui guidaient notre réflexion :

- Le robot arrive-t-il à tourner assez dans les virages ? Si non, augmenter le gain proportionnel.
- Le robot est-il assez nerveux ? Si non, augmenter le gain dérivé. A l'inverse si oui (visible par des oscillations très rapides), baisser le gain dérivé.

Le code correspondant est disponible dans le fichier TODO: INSERER FICHIER.

Nous nous sommes également questionnés sur l'utilité de l'implémentation d'un PID plutôt qu'un PD sur l'écart par rapport à la ligne, mais nous avons abandonné cette idée en raison de la quantification du Line Follower Array :

- Avant l'implémentation, nous avions l'intuition que l'ajout d'un terme intégral n'allait pas être satisfaisant à cause de la quantification du capteur, qui est importante : il est très dur d'arriver parfaitement au "0" du Line Follower Array, même en étant sur la ligne. Il n'aurait donc pas été possible de se débarrasser de l'erreur statique, même avec un terme intégral. Au contraire, ce terme rajouterait des erreurs "numériques".
- De brefs tests sur le circuit ont confirmé cette intuition.

Contrôle de la vitesse en translation Après avoir implémenté le contrôleur PD sur l'écart à la ligne, nous avons essayé d'implémenter le contrôle de la vitesse en translation, matérialisé par la somme des tensions des deux moteurs. Le code correspondant est disponible dans TODO:INSERER FICHIER

Néanmoins, les gains proposés par Matlab sur ce sous-système n'étaient pas satisfaisants et la performance du robot sur le circuit était moins bonne qu'en imposant une somme des tensions constante. Un réglage plus manuel des gains n'a pas donné de résultat satisfaisant non plus. Nous avons donc décidé de ne pas implémenter le contrôleur PD sur la somme des angles mesurés par les encodeurs.

Cependant, nous avons vite identifié que notre robot n'allait pas à une vitesse optimale en ligne droite, et nous nous sommes demandés comment nous pouvions améliorer ce comportement sans utiliser ce contrôleur. Nous avons opté pour une solution non-linéaire de "seuil" :

- Si l'écart à la ligne est supérieure à une certaine valeur seuil pré-définie, nous considérons que le robot est dans un virage, et nous adaptons sa vitesse en fonction en baissant la valeur de la commande de la somme des tensions des deux moteurs vers une valeur pré-définie.
- Sinon, nous considérons que le robot est en ligne droite et nous augmentons la valeur de la commande de la somme des tensions à une valeur pré-définie.

Le code correspondant, qui est celui utilisé lors du challenge, est disponible dans `TODO:INSERER FICHIER`.