

TP1 : Contrôle de robots unicycle et bicyclette

Dajing Gu et Alice Phe

November 2020

1 Introduction

1.1 Résumé du cours

Un **modèle cinématique** se définit par les éléments d'assemblage et les liaisons mécaniques, en robotique, il lie la vitesse dans le domaine cartésien et la vitesse articulaire du robot. À l'aide des modèles cinématiques, il y a de nombreuses applications, desquelles trois applications au contrôle sont : la stabilisation de trajectoires, le suivi de chemin et la stabilisation de configurations.

Le contrôle du robot est de minimiser l'erreur entre la position de destination et la position réelle. À l'aide d'algorithmes comme le PID, on peut obtenir une **convergence rapide sans grandes oscillations** en choisissant les composantes idéales (P, I, D).

1.2 Introduction du TP

On cherchera dans ce TD dans différents cas (Unicycle, Bicyclette, Position Goal simple ou multiples) à simuler des déplacements contrôlés de tels robots. Des méthodes de contrôle très simples sont implémenté en utilisant MATLAB.

2 Unicycle

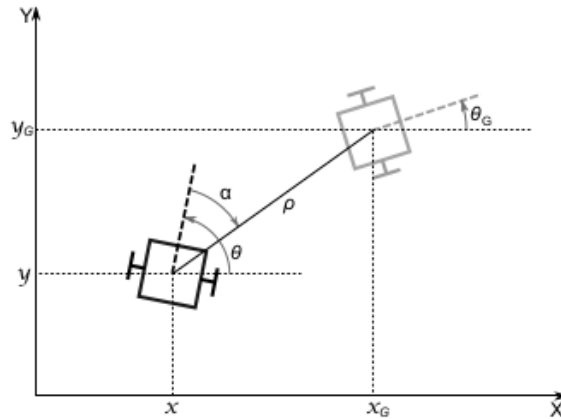


Figure 1: Contrôle d'un unicycle vers une pose

La formule associée est présentée ci-dessous:

$$\rho = \sqrt{(x_G - x)^2 + (y_G - y)^2}$$

$$\alpha = \arctan \frac{(y_G - y)}{(x_G - x)} - \theta$$

$$v = K_\rho \times \rho$$

$$\varphi = K_\alpha \times \alpha$$

$$\text{si } |\alpha| > \alpha_{max}$$

$$\varphi = 0$$

et pour un robot proche du but

$$\beta = \theta_G - \theta$$

$$\omega = K_\beta \times \beta$$

En choisissant $K_\rho = 15$, $K_\alpha = 10$, $K_\beta = 30$ et $\alpha_{max} = \pi/2$, on peut obtenir la trajectoire et l'évolution d'erreur présentés en figure 5. En utilisant la fonction de benchmark `UnicycleToPoseBenchmark`, on obtient une moyenne de temps d'atteinte du but autour de 2000 ms.

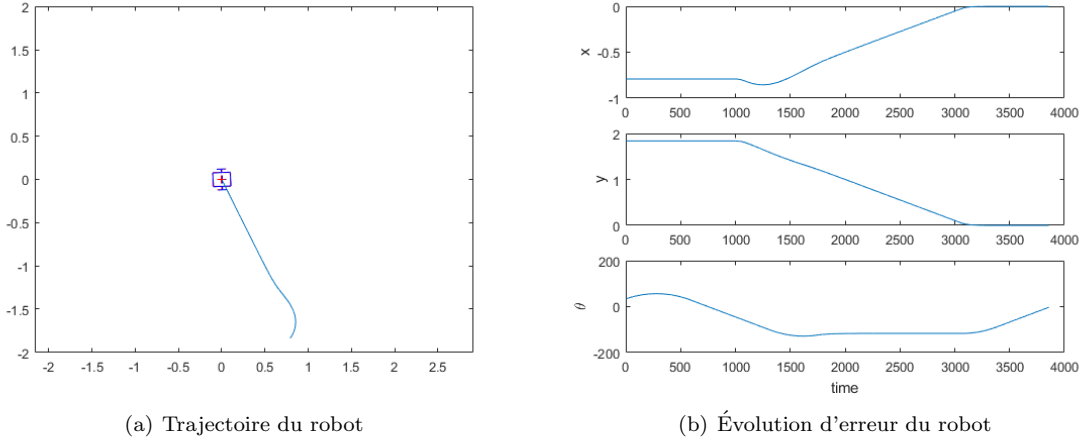


Figure 2: Illustration de résultat

Les équations du modèle sont

$$\begin{cases} \dot{x} = v \cdot \cos(\theta) \\ \dot{y} = v \cdot \sin(\theta) \\ \dot{\theta} = \omega \end{cases}$$

Et en intégrant on obtient

$$\begin{cases} x_{k+1} = x_k + dt \cdot \dot{x} \\ y_{k+1} = y_k + dt \cdot \dot{y} \\ \theta_{k+1} = \theta_k + dt \cdot \omega \end{cases}$$

On a donc d'après les formules que :

- K_ρ va décider de la vitesse de convergence du robot proportionnellement à la distance au goal. S'il est trop petit, on mettra trop de temps à atteindre le goal, s'il est trop grand, on risque de dépasser le goal et à mettre plus de temps à converger (oscillations pour des ρ petits)
- K_α de la vitesse de rotation loin de la cible. S'il est trop petit on va tourner autour de la cible en faisant des trop grands arcs de cercles. S'il est trop grand, le système va trop réagir vis-à-vis de la correction et "serpenter" vers la cible
- K_β , à l'instar de K_α pour les grandes distances, influe sur la vitesse de rotation proche de la cible. S'il est trop grand, on va avoir un risque de rater l'orientation au niveau de la cible et tourner autour. S'il est trop petit, le robot va tourner trop lentement autour de lui même. Ces modifications ne vont pourtant pas impacter de manière significative le temps moyen d'atteinte du but.

K_ρ	K_α	K_β	time(ms)
1	10	30	9800
40	10	30	9500
15	1	30	NaN
15	20	30	NaN
15	10	1	6825
15	10	60	2009

Table 1: Différents résultats de temps pour K_ρ , K_α , K_β

3 Modèle bicyclette

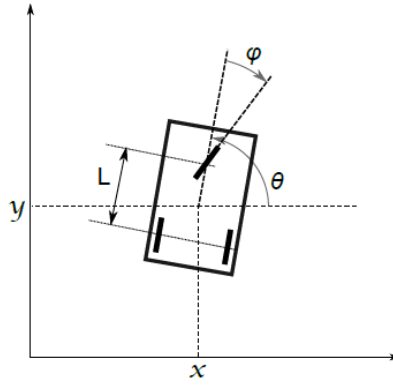


Figure 3: Modèle bicyclette

Les équations du modèle sont

$$\begin{cases} \dot{x} = v.\cos(\theta) \\ \dot{y} = v.\sin(\theta) \\ \dot{\theta} = \frac{v}{L}\sin(\varphi) \\ \dot{\varphi} \end{cases}$$

3.1 Question 2 - contrôle de bicyclette vers un point

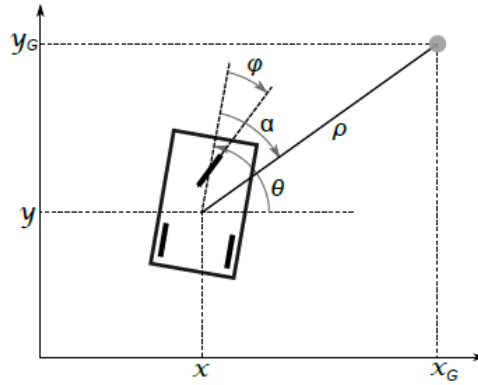


Figure 4: Contrôle d'un modèle bicyclette vers un point

Le formule associée est présentée ci-dessous:

$$\begin{aligned}\rho &= \sqrt{(x_G - x)^2 + (y_G - y)^2} \\ \alpha &= \arctan \frac{(y_G - y)}{(x_G - x)} - \theta \\ v &= K_\rho \times \rho \\ \dot{\varphi} &= K_\alpha \times \alpha\end{aligned}$$

En choisissant $K_\rho = 20$ et $K_\alpha = 10$, on peut obtenir un score **1370** qui est inférieur à 1400. La trajectoire et l'évolution d'erreur sont aussi présentés dans figure 5. Le robot a atteint sa destination mais son orientation finale est différente de celle de but. On implémentera un contrôleur d'orientation dans la question suivante. De même que pour la question précédente un grand K_ρ permet de converger plus vite, et le K_α gère la sensibilité à la correction et ainsi les oscillations ou grands cercles vers la cible. On pourrait augmenter d'autant plus K_ρ mais pour le contrôle vers la position on revient au problème de K_ρ trop grand pour la première question.

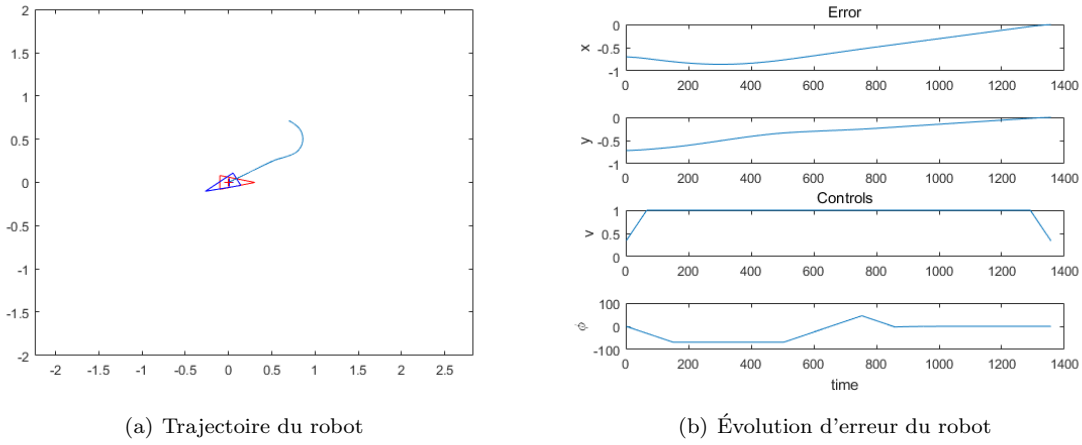


Figure 5: Illustration de résultat $K_\rho = 20$, $K_\alpha = 10$

3.2 Question 3 - contrôle de bicyclette vers une position

On cherche cette fois à étudier le contrôle d'une bicyclette

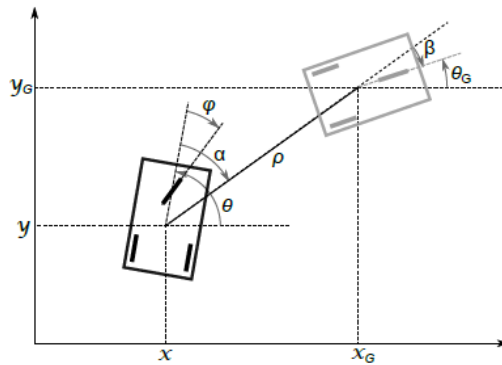
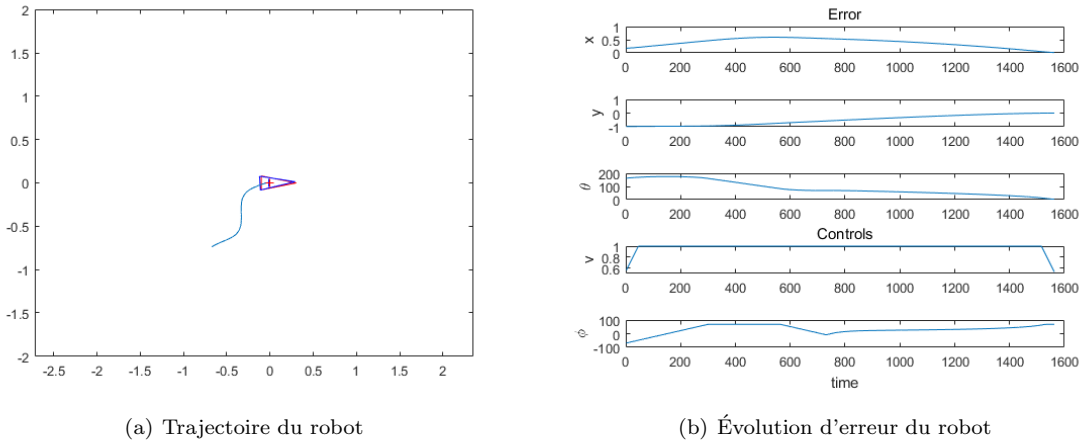


Figure 6: Contrôle d'un modèle bicyclette vers une position

La formule associée est présentée ci-dessous:

$$\begin{aligned}\rho &= \sqrt{(x_G - x)^2 + (y_G - y)^2} \\ \alpha &= \arctan \frac{(y_G - y)}{(x_G - x)} - \theta \\ \beta &= \theta_G - \arctan \frac{(y_G - y)}{(x_G - x)} \\ v &= K_\rho \times \rho \\ \dot{\phi} &= K_\alpha \times \alpha + K_\beta \times \beta \\ &\text{avec } K_\beta < 0\end{aligned}$$

En choisissant $K_\rho = 20$, $K_\alpha = 10$ et $K_\beta = -5$ on peut obtenir un score **1620**, qui est inférieur à 1800. La trajectoire et l'évolution d'erreur sont aussi présenté dans figure 7. Le robot a atteint la destination et son orientation finale est la même que celle de but.



Le modèle bicyclette ne permet pas de tourner sur lui-même ni de reculer : pour arriver de façon optimale à la cible avec la bonne orientation, il faut que le robot ait déjà la bonne orientation assez loin de la cible. De même que pour la première question, on retrouve les mêmes problèmes avec K_ρ , K_α non ajustés, avec la conséquence supplémentaire que pour les valeurs trop grandes de K_α on va arriver avec la mauvaise orientation sur la cible et tourner à l'infini pour trouver la bonne combinaison position/ orientation. Le système ne pouvant se translater, K_β est la correction d'orientation une fois la bonne orientation obtenue pour arriver jusqu'à la position. Si K_β est trop petit, il n'y pas de correction, s'il trop grand, le système va être trop sensible et arriver avec la mauvaise orientation.

3.3 Question 4 - contrôle de bicyclette selon un chemin

En choisissant $K_\rho = 20$, $K_\alpha = 10$, même s'il y a quelques oscillations à chaque nouveau tournant, le robot a réussi à atteindre son but en suivant une série de points avec une erreur de **372**.

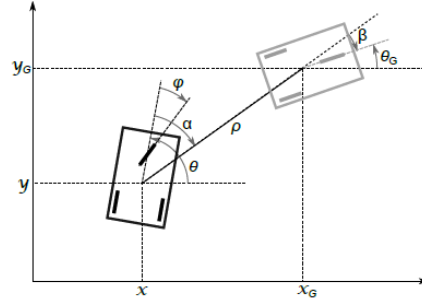
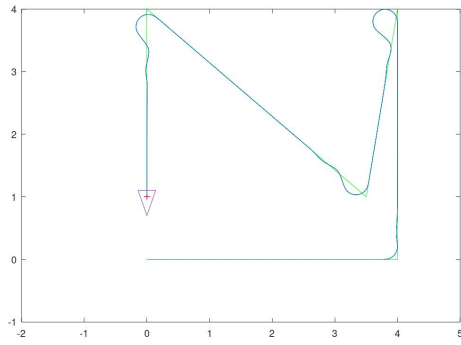
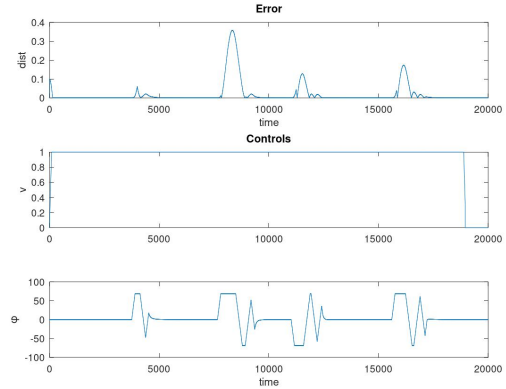


Figure 8: Contrôle d'un modèle bicyclette selon un chemin



(a) Trajectoire du robot

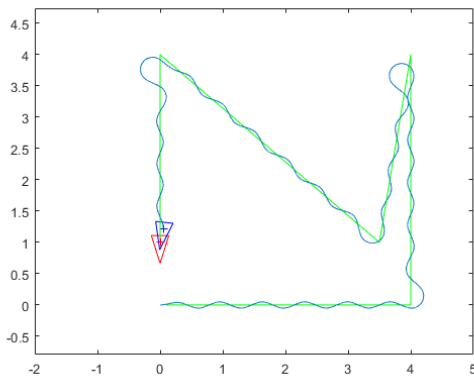


(b) Évolution d'erreur du robot

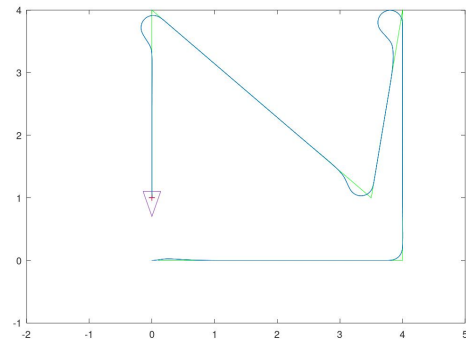
Figure 9: Illustration de résultat avec $K_\rho = 20$, $K_\alpha = 10$

En choisissant un grand K_α , la sensibilité du système fait que le système va avoir beaucoup d'oscillations avant d'atteindre la stabilité pour chaque portion de la trajectoire. Mais pour un K_α petit, le robot va réaliser de très grands arcs de cercle lors d'un passage à une nouvelle portion.

L'influence de K_ρ joue pour l'erreur du système lors des tournants et la vitesse de convergence, mais change peu l'erreur totale.



(a) Trajectoire du robot avec un grand K_α



(b) Trajectoire du robot avec un petit K_α

Figure 10: Illustration de résultat