

# TP2: Commande avec anticipation, commande prédictive

Planification et contrôle

ROB316

Yu WANG

December 12, 2021

**Remarque :** Tous les résultats (code, images, etc.) présentés dans ce rapport peuvent être trouvés sur ce gitHub : <https://github.com/Rescon/ROB316>

## 1 Introduction

### 1.1 Résumé du cours

Ce cours introduit le modèle dynamique. Contrairement aux modèles cinématiques, qui tiennent compte des forces, de l'inertie et de la friction du système, les modèles dynamiques offrent une vision plus complexe du système, ce qui permet d'estimer le comportement futur du système par intégration. En tant qu'algorithme qui prend en compte les attentes dans le contrôle, il permet de choisir le meilleur avenir. En pratique, il est difficile à appliquer, mais il est utile pour la validation. Pour atteindre cet optimum en commande prédictive, il existe différentes solutions en fonction de la linéarité du système et de ses contraintes, comme nous allons le voir dans ce TP.

### 1.2 description de TP

Dans ce TP, nous allons voir l'effet d'une commande anticipative et d'une commande prédictive dans une version simplifiée.

A l'aide de Matlab (Octave), nous allons appliquer cet algorithme au suivi de trajectoire. Ensuite, nous trouverons la zone de stabilité pour le contrôle prédictif. Et enfin, la commande prédictive est testée et validée.

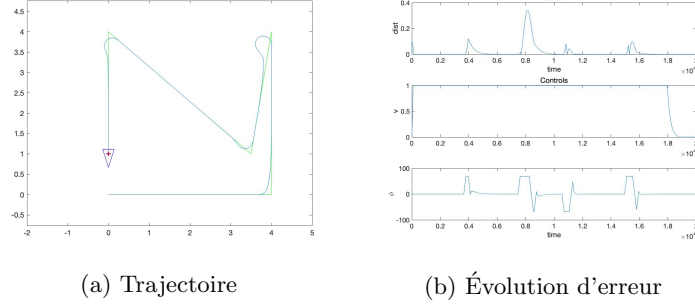


Figure 1: Résultat de PID, total error: 357.7656

## 2 Anticipation

Dans cette section, nous allons comparer les résultats du contrôle du robot par un PID et par MPC.  $K_p$  est fixé à 10 et  $K_\alpha$  est fixé à 5.

### 2.1 Comparaison entre PID et MPC

Sur la base des résultats présentés dans la Figure 1 et la Figure 2, on peut constater que lorsque le robot est contrôlé par PID, la trajectoire réelle du robot est plus proche de la trajectoire prédéterminée. En utilisant la méthode PID, le robot suit une trajectoire constituée d'une série de points sur un chemin prédéterminé. Même avec des oscillations, le robot se déplace toujours le long de cette trajectoire. Avec la méthode MPC, cependant, la trajectoire n'est pas exactement sur la trajectoire. En fait, le MPC est une commande optimale à long terme qui prédit l'avenir et calcule la commande au moyen d'une boucle ouverte (c'est-à-dire en prédisant une plage de valeurs de commande) afin que la commande atteigne la position finale de la meilleure façon possible, de sorte qu'elle ne suit parfois pas le milieu de la trajectoire (ceci est particulièrement évident dans les virages).

### 2.2 Essayez plusieurs horizon: 1, 5, 20, 100, 1000

En essayant différentes valeurs de la taille de l'horizon, dont les résultats sont présentés dans la Figure 3, la Figure 4, la Figure 5, la Figure 6 et la Figure 7, on peut conclure que plus la taille de l'horizon est grande, plus l'erreur de contrôle est importante, plus la trajectoire sélectionnée est éloignée de la trajectoire, et plus la taille du pas vers le futur avant chaque mouvement est grande : cela aide le robot à se déplacer directement vers sa destination finale (par exemple, dans le cas de Window size = 200 et dans le cas de Window size = 1000 ).

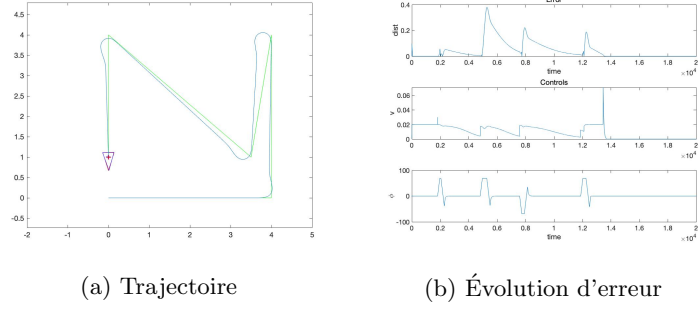


Figure 2: Résultat de MPC avec window size = 5, total error: 866.5171

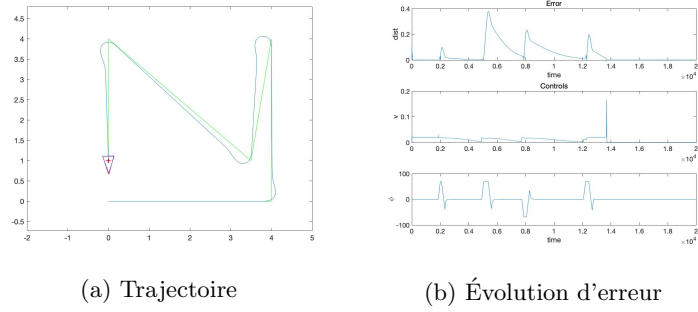


Figure 3: Résultat avec window size = 1, total error: 867.1894

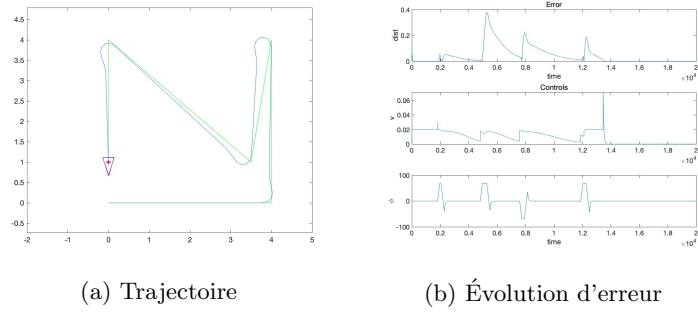


Figure 4: Résultat de MPC avec window size = 5, total error: 866.5171

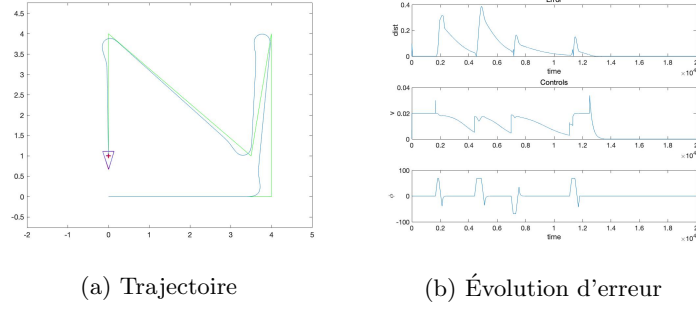


Figure 5: Résultat avec window size = 20, total error: 1015.5479

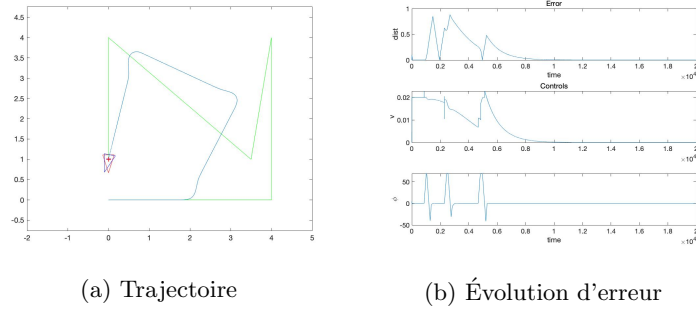


Figure 6: Résultat avec window size = 100, total error: 2405.356

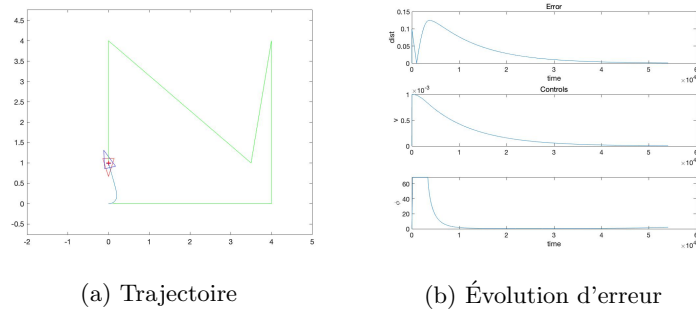


Figure 7: Résultat avec window size = 1000, total error: 1687.9893

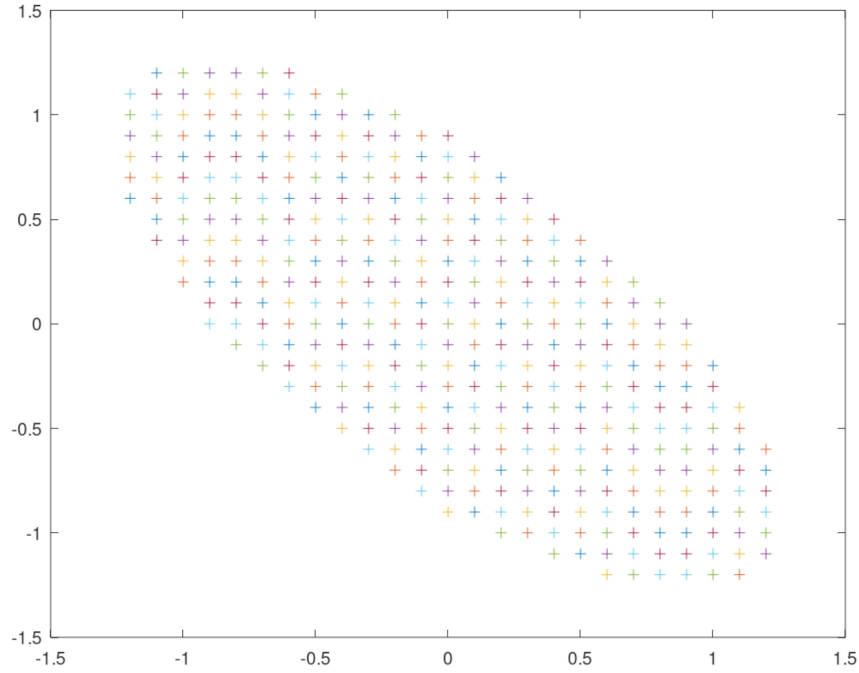


Figure 8: Zone de points stables

### 3 Zone de stabilité d'une commande prédictive

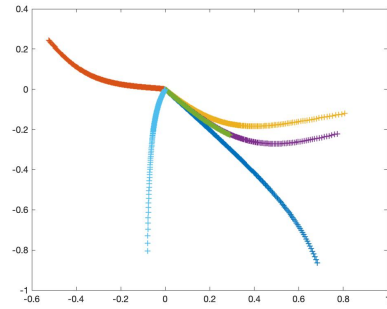
Dans cette section, nous étudions la zone de stabilité de la commande prédictive. Après calcul, on peut obtenir les zones de stabilité de Lyapunov illustrées à la Figure 3.1. Ce sont les endroits où les conditions de stabilité sont vérifiées. Dans la Figure 8, nous pouvons voir que toutes les points qui satisfont à la condition de stabilité forment une ellipse.

### 4 Commande prédictive

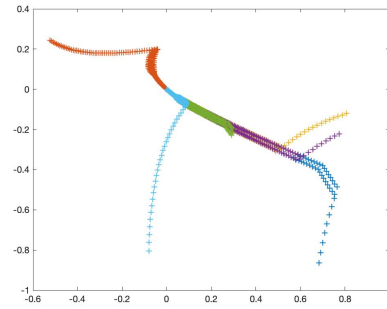
Dans la question précédente, nous avons obtenu la commande  $K$  est  $[2.118, 2.118]$ . Pour ce problème, nous allons utiliser `test_mpc` pour le tester  $u = Kx$  et le résultat est montré dans la Figure 9a. Dans cette figure, on peut voir que toutes les lignes sont concentrées au point  $(0, 0)$ , ce qui est probablement l'état stable.

La Figure 9a montre six trajectoires simulées en utilisant la commande prédictive. Nous pouvons constater que la courbe tend à converger vers la position de référence, mais c'est plus difficile et la trajectoire la plus courte

n'est pas choisie, et la courbe a tendance à vouloir revenir sur l'axe menant à la position de référence.



(a) Trajectoire stabilisée



(b) Trajectoires avec commande prédictive

Figure 9: Résultats de **simulateMPC**