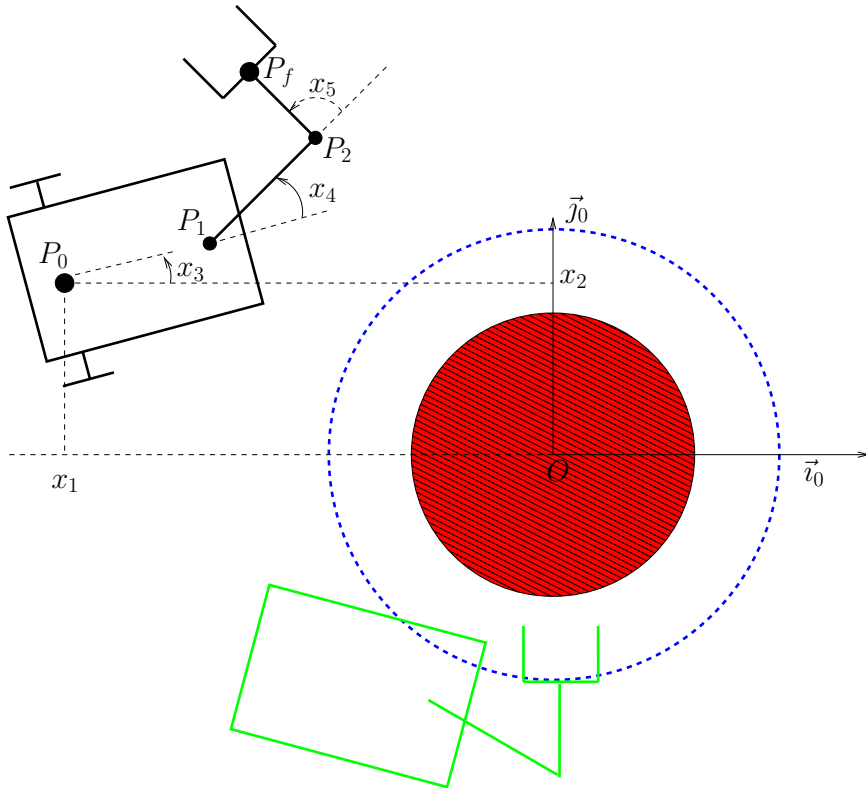


Projet sur la commande MPC – Master 2 SAR

Intitulé: *Commande d'un manipulateur mobile*

L'objectif de ce projet est de synthétiser une loi de commande de type MPC, à partir de la toolbox `do-mpc` vue en séance de TP, afin de contrôler un robot manipulateur. Le problème que l'on cherche à résoudre est le suivant. Partant d'une configuration initiale arbitraire (par exemple la configuration en noir sur la figure), le robot doit se déplacer de façon à amener la pince du manipulateur en face de l'obstacle rouge, à une distance prédéfinie du centre de l'obstacle matérialisée par le cercle bleu, et à un endroit particulier sur ce cercle (par exemple la configuration verte de la figure). Autrement dit, on doit contrôler précisément la position et l'orientation de la pince pour la faire converger vers des position et orientation de référence, tout en évitant les collisions. La position sur le cercle bleu, qui définit la position et orientation de référence de la pince, est évidemment définie à l'avance. Par ailleurs, outre l'évitement des collisions, le robot est soumis à plusieurs contraintes: contraintes sur la vitesse linéaire et angulaire du robot mobile, sur le débattement articulaire des axes du bras manipulateur, sur les accélérations.



On rappelle, avec les notations de la figure, les équations de la cinématique du système:

$$\begin{cases} \dot{x}_1 &= v_1 \cos x_3 \\ \dot{x}_2 &= v_1 \sin x_3 \\ \dot{x}_3 &= v_2 \\ \dot{x}_4 &= v_3 \\ \dot{x}_5 &= v_4 \end{cases} \quad (1)$$

où v_1 correspond à la vitesse de roulement du robot, v_2 à sa vitesse de rotation instantanée, et v_3, v_4 correspondent aux vitesses articulaires des deux axes du manipulateur. Afin de prendre en compte les limites d'accélération, on ajoute à cette cinématique un modèle simplifié de la dynamique (on notera, évidemment, que ce n'est pas le vrai modèle dynamique au sens de la mécanique):

$$\begin{cases} \dot{v}_1 &= u_1 \\ \dot{v}_2 &= u_2 \\ \dot{v}_3 &= u_3 \\ \dot{v}_4 &= u_4 \end{cases} \quad (2)$$

où u_1, \dots, u_4 , homogènes à des accélérations, correspondent aux variables de commande. La géométrie est définie par les relations suivantes:

- distance entre P_0 et P_1 : $\ell_1 = 0.4m$;
- distance entre P_1 et P_2 : $\ell_2 = 0.4m$;
- distance entre P_2 et P_f : $\ell_3 = 0.3m$;
- rayon de l'obstacle: $0.4m$;
- rayon du cercle bleu: $0.6m$.

Afin d'éviter des collisions entre le robot et l'obstacle, on impose les contraintes "anti-collision" suivantes:

- distance entre P_0 et l'obstacle supérieure à $0.4m$;
- distance entre P_1 et l'obstacle supérieure à $0.3m$;
- distance entre P_2 et l'obstacle supérieure à: $0.1m$;
- distance entre P_f et l'obstacle supérieure à: $0.15m$.

On admettra que lorsque ces contraintes sont satisfaites, le robot ne peut pas entrer en collision avec l'obstacle. On ajoute à ces contraintes anti-collision des contraintes articulaires sur les axes du manipulateur:

$$|x_4| \leq \gamma, \quad |x_5| \leq \gamma \quad (3)$$

où $\gamma > 0$ est un paramètre dont on étudiera par la suite l'impact sur les trajectoires du robot. On introduit également des limites sur les vitesses:

$$|v_i| \leq 1.5, \quad \forall i = 1, \dots, 4$$

Enfin, on prend en compte les limites de puissance du robot via les contraintes sur les commandes suivantes:

$$|u_i| \leq 2, \quad \forall i = 1, \dots, 4 \quad (4)$$

1. Exprimer les contraintes anti-collision en fonction des variables d'état du système.
2. En vous inspirant de l'exemple traité en TP, développer avec do-mpc un contrôleur MPC pour résoudre le problème posé. Afin de vous guider dans cet exercice, voici plusieurs consignes/précisions:

- Critère à optimiser: vous testerez les deux critères ci-dessous, **en discutant les résultats obtenus** (attention, comme indiqué en cours, rien ne garantit que l'optimisation va fournir une "bonne solution"):

$$J_1() = \|p - p_{ref}\|^2 + (\theta - \theta_{ref})^2 \quad (5)$$

$$J_2() = \|p - p_{ref}\|^2 + (\theta - \theta_{ref})^2 + v_3^2 + v_4^2 \quad (6)$$

où p et p_{ref} correspondent respectivement à la position de la pince (effecteur) du manipulateur et la position de référence (objectif), et θ et θ_{ref} correspondent respectivement à l'orientation de la pince du manipulateur et l'orientation de référence (objectif).

- Pas de temps d'échantillonnage: vous pourrez prendre un pas de temps d'échantillonnage de $\delta = 0.05$ seconde.
 - Horizon d'optimisation: vous pourrez tester plusieurs longueurs d'horizon d'optimisation telles que $T = 10, 20, 50$.
 - Prise en compte des contraintes anti-collision: afin de prendre en compte les contraintes anti-collision, vous pourrez vous référer sur le site de "do-mpc" à l'exemple du double pendule inversé, qui contient une simulation avec évitement d'obstacle. En particulier, le code de cet exemple sur GIT vous permettra de voir quelles fonctions peuvent être utilisées pour intégrer ces contraintes dans le schéma MPC.
3. Tester les performances du contrôleur en étudiant l'impact sur les performances des éléments suivants:
 - Pas de temps d'échantillonnage;
 - Horizon d'optimisation;
 - Valeur du paramètre γ définissant les limites articulaires (on pourra considérer plusieurs valeurs comprises en $\pi/4$ et π);
 - Condition initiale (on considèrera différentes conditions initiales nécessitant, ou non, de contourner l'obstacle);