



Rapport du projet Manipmobile

CHAMPANEY Matéo
TRABUT Benjamin
GOMEZ Matthieu
ADOPPO Kylian
AUBRUN Louis
MAES Alexis

Un document présenté pour
le projet du semestre 6 du cursus ingénieur

Centrale supélec

June 12, 2024

Contents

1	Introduction	2
2	Simulation	2
2.1	Motivations	2
2.2	Cinématique inverse	3
2.3	Mise en place sur MATLAB	5
3	Méthode de la moyenne	7
3.1	Principe de la méthode	7
3.2	Test	7
3.3	Commentaire et limite	9
3.4	Détermination du nombre de points pour la moyenne	10
4	Approche cinématique	10
4.1	Le choix de l'asservissement	10
4.2	Géométrie directe	11
4.3	Comment choisir de K ?	14
5	Comparaison des méthodes de résolution	15
5.1	Présentation des trajectoires test	15
5.1.1	Trajectoire sinusoïdale	15
5.1.2	Trajectoire créneau	16
5.2	Résultats pour la trajectoire sinusoïdale	17
5.3	Résultats pour la trajectoire créneau	19
5.4	Bilan	21
6	Conclusion	21

1 Introduction

Lorsque le travail est trop répétitif ou trop long, on préfère aujourd’hui déléguer le travail à des robots plutôt qu’à des humains. Que ce soit pour transporter des objets lourds dans un hangar ou peindre un bateau, on préfère de plus en plus les machines aux humains

Néanmoins, ce choix présente encore des défauts et des contreparties. Ils sont plus lents pour certaines tâches, peuvent être très complexes à concevoir et n’ont pas l’adaptabilité d’un humain. Face à une utilisation toujours plus large des robots dans de plus en plus de domaines et situations, il est nécessaire de progresser sur notre capacité à produire des robots aptes à effectuer des tâches complexes ou des tâches simples efficacement.

C’est pourquoi notre groupe s’est penché sur le cas particulier des bras de robots mobiles. En effet, aujourd’hui de tels robots peuvent être chargés par exemple de tâche de ”pick and place” ou de peinture par exemple. De manière générale, ses robots se doivent de suivre des trajectoires précise pour réaliser différentes tâches. Malheureusement, ceux-ci sont limités dans leur fonctionnement. Généralement, le mouvement de chariot et bras sont dissociés, c'est-à-dire que le robot utilise alternativement le chariot qui lui sert à se déplacer et le bras de robot servant à réaliser les tâches demandées. On va donc chercher à trouver des méthodes pour permettre à un robot à quatre axes, monté sur un chariot, de concilier le mouvement du chariot et le mouvement du bras de robot.

2 Simulation

2.1 Motivations

N’ayant pas accès au robot physique, réquisitionné par les deuxième année, nous avons dû mettre en place une simulation informatique exploitable de notre robot afin d’obtenir nos premiers résultats. La visualisation dans une simulation 3D étant souvent plus parlante que la lecture de courbes. Il a été convenu de s’intéresser dans un premier temps au bras du robot (Fig. 13a), que nous avons schématisé (Fig. 13b).

On note de plus θ_0 l’angle de la liaison pivot d’axe vertical située à la base du bras (elle n’apparaît pas sur le schéma). L’objectif à terme étant de pouvoir diriger le robot, on a besoin de trouver la fonction

$$\begin{aligned} f: \mathbb{R}^4 &\longrightarrow \mathbb{R}^3 \\ (\theta_0, \theta_1, \theta_2, \theta_3) &\longmapsto (x, y, z), \end{aligned}$$

où (x, y, z) sont les coordonnées de la tête du bras du robot. Plus précisément, on s’intéresse à l’inverse de cette fonction (la cinématique inverse), qui nous permettra d’obtenir les valeurs des angles de ses liaisons pivots à partir de la position à laquelle on veut que sa tête soit.

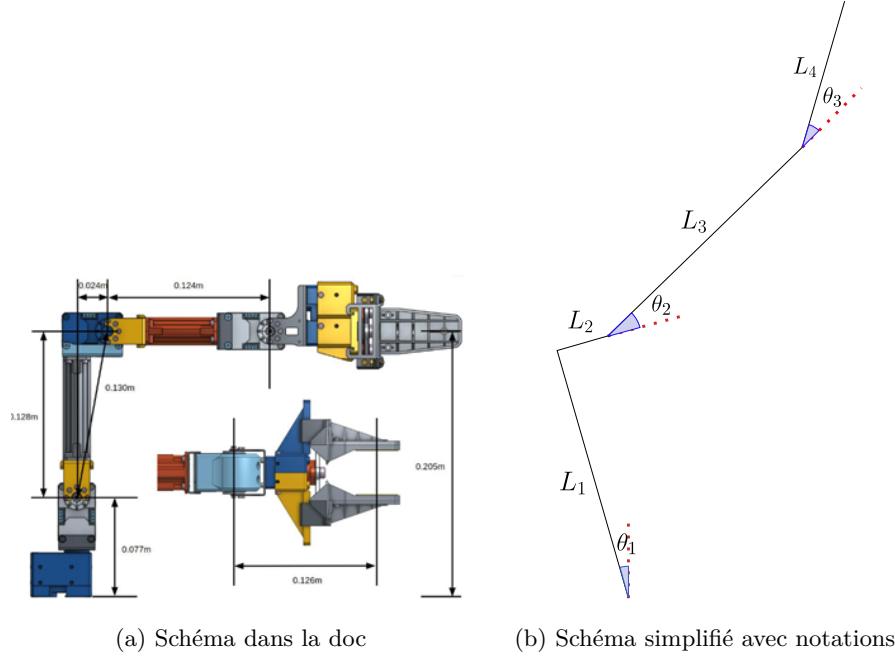


Figure 1

2.2 Cinématique inverse

Le premier problème vient du fait que la fonction f n'est pas nécessairement inversible, plusieurs configurations d'angles du robot mènent à la même position de sa tête.

Comme illustré Figure 2, faire pivoter le robot autour de son axe vertical d'un angle π nous permet, en faisant varier θ_2 , d'obtenir la même position pour la tête du robot de deux façons différentes.

On va principalement traiter le problème dans le cas 2D, on se ramène au cas 3D en utilisant la liaison pivot entre bras et chariot pour orienter ce même bras vers le point qu'on cherche à atteindre. Il convient donc de s'imposer quelques conditions pour réduire le nombre de configurations possibles à une seule et obtenir une fonction qui soit injective.

- (i) La tête du robot est la liaison L3-L4, ainsi, l'angle θ_3 n'aura aucun impact sur son emplacement, on réduit le nombre de degrés de liberté et donc de solutions. Il reste alors un degré de liberté supplémentaire qui servira à orienter la pince et dont le fonctionnement sera amené à varier selon la tâche à exécuter. Ainsi la fonction f ne prend plus θ_3 en argument.
- (ii) On prend de sorte que le bras ne puisse pas se « renverser en arrière », s'il lui est demandé d'atteindre un point qui se trouve derrière lui, il s'y rendra

en faisant varier θ_0 . Autrement dit, sur le schéma Figure 3, $\theta_2 + \gamma_2 \in [0, \pi]$ et $\alpha \in [0, \pi]$

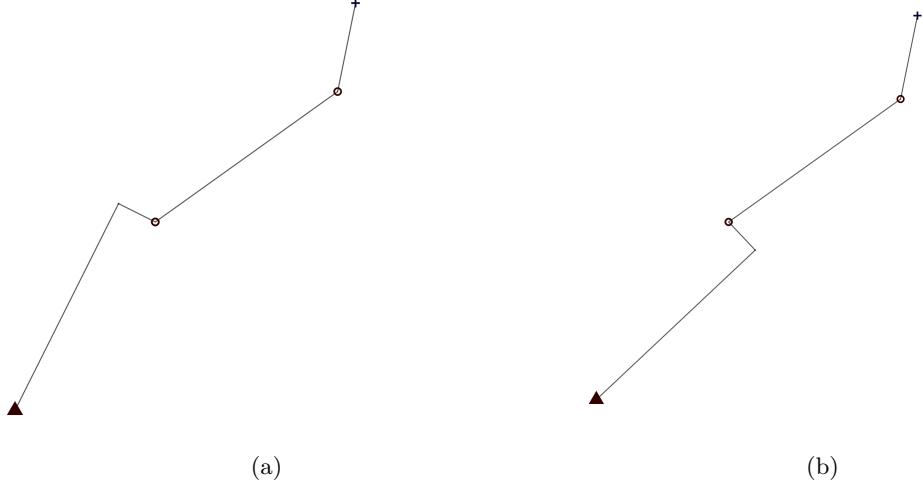


Figure 2: Deux configurations pour une même position

Il s'agit à présent de trouver la formule explicite de f^{-1} . Il est alors judicieux de se placer en coordonnées sphériques (r, θ, ϕ) pour simplifier le problème. On a alors clairement dans un premier temps $\phi = \theta_0$. On cherche ensuite les valeurs de r et de θ , on se place alors dans le plan contenant le bras du robot (Fig. 3)

On a d'abord

$$\theta_1 + \theta = \gamma_1 + \alpha$$

$$\theta_2 + \gamma_2 = \beta$$

On peut ensuite exprimer $\cos(\alpha)$ et $\cos(\beta)$ à l'aide de la formule d'Al Kashi:

$$\cos(\alpha) = \frac{L_3^2 - d^2 - r^2}{2dr}$$

$$\cos(\beta) = \frac{r^2 - L_3^2 - d^2}{2L_3d}$$

Puis γ_1 , γ_2 et d en fonction des paramètres du problème :

$$\gamma_1 = \arctan\left(\frac{L_2}{L_1}\right) \text{ et } \gamma_2 = \gamma_1 + \frac{\pi}{2}$$

$$d = \sqrt{L_1^2 + L_2^2}$$

En combinant ces résultats et (ii) qui nous permet d'appliquer arccos, on obtient la formule explicite de $f^{-1}(r, \theta, \phi)$.

$$\begin{cases} \theta_0 = \phi \\ \theta_1 = \gamma_1 + \arccos\left(\frac{L_3^2 - d^2 - r^2}{2dr}\right) - \theta \\ \theta_2 = \arccos\left(\frac{r^2 - L_3^2 - d^2}{2L_3d}\right) - \gamma_2 \end{cases}$$

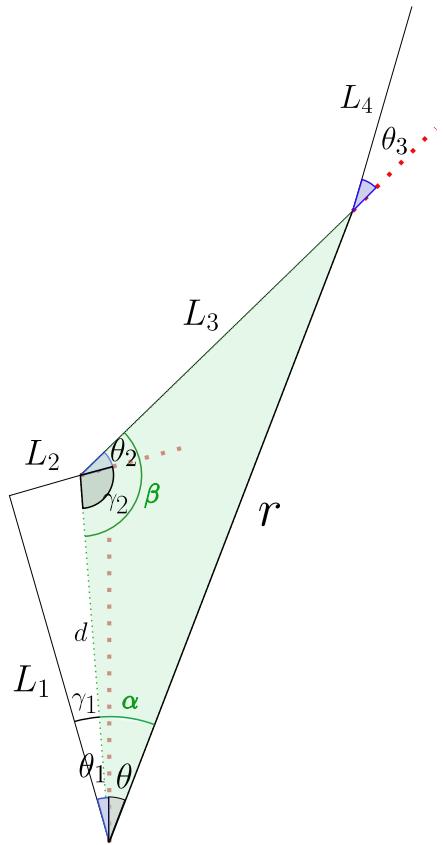


Figure 3

2.3 Mise en place sur MATLAB

Une fois que les formules de cinématique inverse ont été retrouvées, il a fallu coder la simulation sur MATLAB. Nous avons pour cela été orientés vers la Robotic Toolbox de Peter Corke. Hormis la documentation officielle (très austère à appréhender), très peu d'information sur cette toolbox est disponible en ligne. Ce fut donc une des principales difficultés du projet. Après une familiarisation avec ses fonctions et une intégration des formules de cinématique inverse, nous

avons une simulation fonctionnelle qui nous permettra d'effectuer tous les tests dont nous avons besoin (Figure 4).

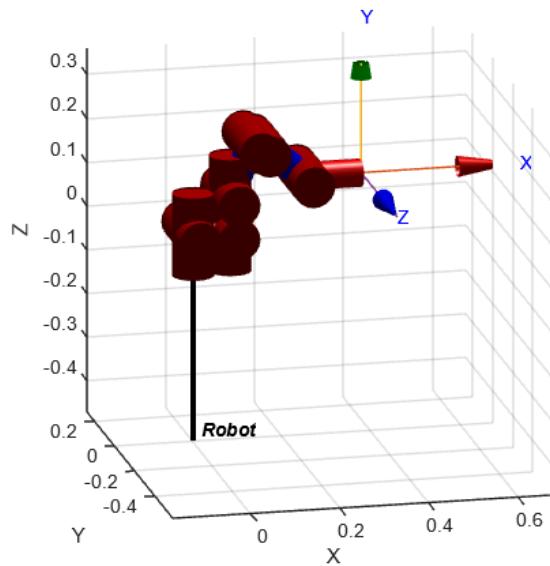


Figure 4: Affichage du bras du robot dans la simulation

3 Méthode de la moyenne

3.1 Principe de la méthode

Face au problème de combinaison entre le mouvement de la base du robot et le mouvement du bras du robot, il est nécessaire d'établir une stratégie pour commander ces deux parties du robot simultanément. Une première approche que nous proposons est de positionner d'abord le chariot, puis le bras du robot.

Pour cela, nous allons essayer de placer le chariot à une position proche des points à atteindre, puis de commander le bras du robot pour atteindre plus précisément ces points. L'algorithme suivant est proposé :

1. Projeter la trajectoire dans l'espace sur le plan (x, y) associé au sol.
2. Pour chaque point de la trajectoire, calculer la moyenne des N prochains points à atteindre.
3. S'il n'y a plus assez de points, utiliser la dernière valeur calculée.

Cette méthode a été codée de deux manières, en Matlab directement à l'aide de fonctions et en Simulink en parallèle de la méthode cinématique que l'on présentera plus tard. On présente dans cette section principalement la démarche et les résultats de l'implémentation passant par des fonctions Matlab.

Cet algorithme permet de transformer une trajectoire spatiale en une trajectoire temporelle. Le chariot suit ensuite cette trajectoire ; les positions, étant la moyenne des projections, seront relativement proches des points à atteindre. Le bras du robot pourra donc atteindre les points de la trajectoire.

Il est important de choisir N de manière adéquate. En effet, une valeur trop grande pourrait éloigner le robot des points de la trajectoire, le rendant incapable de la suivre correctement. À l'inverse, un N trop faible empêcherait le robot de bénéficier de l'avantage de la moyenne des points et le chariot suivrait alors trop près la trajectoire projetée au sol. On perdrait ainsi l'avantage de combiner le bras du robot et le chariot.

3.2 Test

On peut alors tester l'efficacité de la méthode sur des trajectoires simples. On peut entrer par exemple en trajectoire demandé une sinusoïde dans le plan (x, z) .

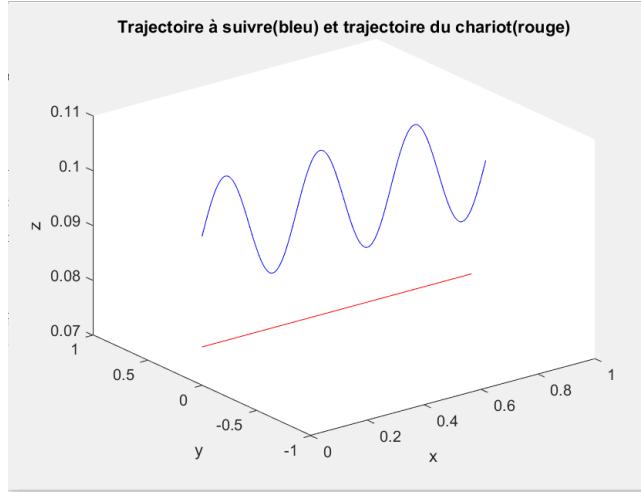


Figure 5: Sinusoïde vue de côté

On voit alors bien que le chariot en rouge suit une droite pour pouvoir être au plus proche de la courbe à suivre. On ajoute d'autres tests pour explorer les limites de notre programme.

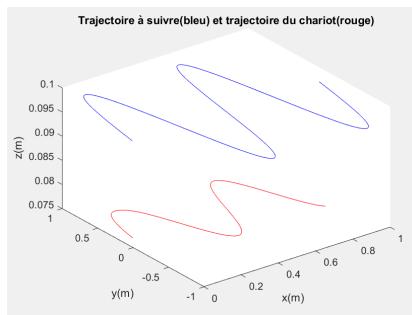


Figure 6: Sinusoïde vue de côté

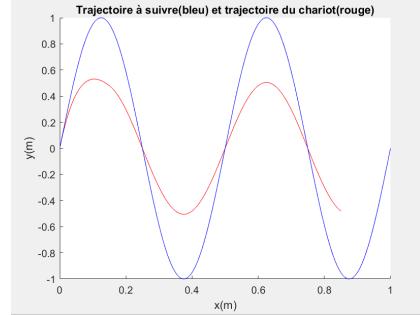


Figure 7: Sinusoïde vue de dessus

On a ici une sinusoïde dans le plan (x, y, z) . On peut voir que le chariot ne suit pas exactement la sinusoïde, et le bras du robot compensera cet écart. On peut remarquer que la courbe en rouge s'arrête avant la fin. Cela vient du fait que lorsque l'on arrive sur les derniers points à traiter, le chariot est bien situé derrière les points à traiter.

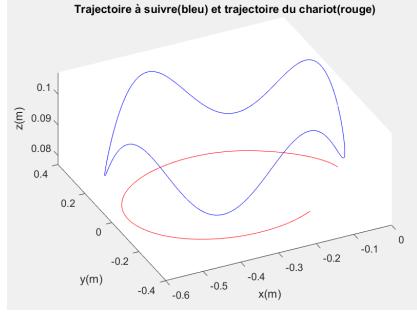


Figure 8: Vue de côté de la trajectoire

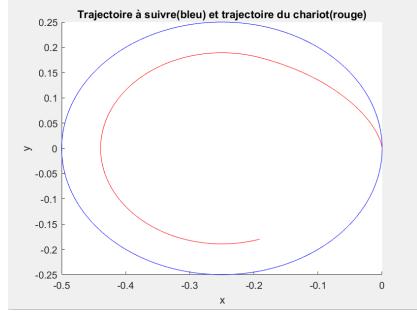


Figure 9: Vue de haut de la trajectoire

On prend une trajectoire plus complexe avec un cercle dans le plan et une sinusoïde dans z . On suit alors une trajectoire proche d'un cercle. On constate que notre programme a des résultats satisfaisants. On remarque également que pour un cercle plus grand, logiquement le chariot suit bien plus précisément la trajectoire du cercle projeté sur le sol. Ce qui semble bien cohérent.

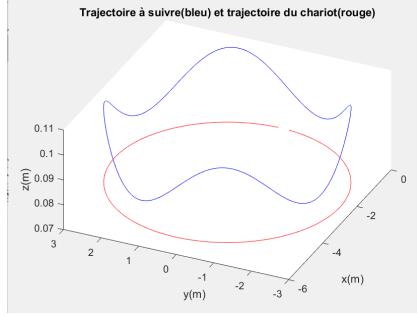


Figure 10: Vue de côté de la trajectoire

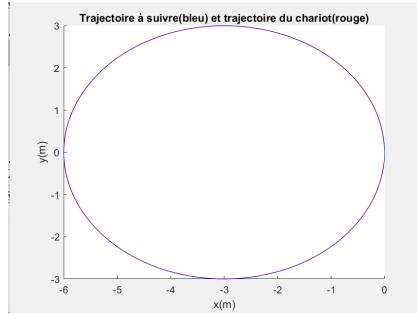


Figure 11: Vue de haut de la trajectoire

3.3 Commentaire et limite

L'association de notre algorithme de moyenne et de la cinématique inverse dont on a pu parler nous permet de déterminer tous les paramètres à imposer aux différents actionneurs de notre robot pour suivre la trajectoire imposé. On obtient en théorie des résultats satisfaisants.

Néanmoins, quelques retenus sont à prendre en compte. Premièrement, on fait l'hypothèse d'holonomie du robot est problématique. On sent bien que le robot pourra avoir du mal à suivre une sinusoïde si la fréquence est trop élevée. On peut prendre en compte que le robot ne bougera en réalité pas si la commande en entrée est trop faible. Avec une fréquence très élevée, la moyenne sera proche d'une droite et le robot réel suivra exactement cette droite, l'amplitude

sinusoïde en entrée sera trop faible.

3.4 Détermination du nombre de points pour la moyenne

La grande difficulté de l'utilisation de ce programme est celle de choisir le nombre de points à utiliser dans la moyenne. En effet, un nombre trop élevé donnera une moyenne trop éloignée de la trajectoire réelle et le bras de robot ne pourra pas atteindre le point à traiter, un nombre trop faible perd de l'intérêt de faire une moyenne. Après de nombreux tests, on remarque que la manière la plus optimale semble de prendre un nombre maximum de points tout en s'assurant que la trajectoire ne sortira pas de la zone atteignable par le bras de robot.

Si on prend une trajectoire en ligne droite qui ne sera possible de suivre que si le robot va à la vitesse maximale. On veut que la distance à laquelle se trouve le point moyen du point à traiter soit inférieur à une longueur L représentant la distance maximal que peut atteindre le bras de robot.

$$\frac{N_{point} * V_{max} * t_{échantillonage}}{2} = L$$
$$N_{point} = \lfloor \frac{2 * L}{V_{max} * t_{échantillonage}} \rfloor$$

On prendra une valeur légèrement inférieure pour se garantir une sécurité

4 Approche cinématique

La seconde approche que nous avons considérée au cours de notre projet est une approche cinématique dans laquelle nous avons réalisé un asservissement de la vitesse du véhicule et du bras articulé.

4.1 Le choix de l'asservissement

Pour réaliser l'asservissement de notre robot, nous avons adopté un schéma bloc agissant directement sur les moteurs qui le composent en contrôlant leur vitesse de sorte à pouvoir suivre la trajectoire qu'on lui donne en entrée. Nous sommes partis sur un schéma-bloc assez simplifié comprenant un correcteur de gain proportionnel K en entrée et avons négligé dans un premier temps le côté dynamique prenant en compte les masses et inerties des différents moteurs.

Voici le schéma-bloc de l'asservissement :

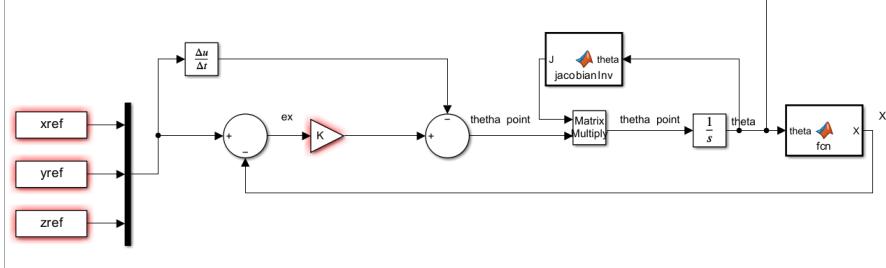


Figure 12: Schéma bloc de l'asservissement cinématique.

En considérant $\mathbf{X} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ le vecteur position du robot, la lecture du schéma donne :

$$e_X = X^{ref} - X$$

où e_X représente l'erreur sur la position du robot. Donc

$$\dot{e}_X = \dot{X}^{ref} - \dot{X}$$

où \dot{e}_X représente l'erreur sur la vitesse du robot.

$$\dot{e}_X = \dot{X}^{ref} - J(\theta)\dot{\theta}$$

En effet, nous avons $X = f(\theta)$ (Méthode géométrie Directe). Nous avons donc $\dot{X} = J(\theta)\dot{\theta}$ où $J = \frac{\partial f}{\partial \theta}$

Il en découle :

$$\dot{e}_x = \dot{X}^{ref} - J(\theta)J^{-1}(X^{ref} + K e_X)$$

Finalement,

$$\dot{e}_X = -K e_X$$

Remarque Dans cette partie, nous avons fait l'hypothèse que le dernier angle θ_3 est déterminé de façon à ce que la pince soit horizontale. Cependant, dans un souci de généralisation des tâches effectuées par le robot, nous pouvons étendre la dimension de X et X_{ref} de façon à inclure un angle entre la pince et la surface ou objet qu'il manipule. On peut alors, dans le cas d'un robot qui peint une coque de bateau, augmenter X_{ref} d'un paramètre ϕ pour imposer la pince d'être orthogonale à la surface en train d'être peinte.

4.2 Géométrie directe

Notons f la fonction donnant la géométrie directe du robot. Autrement dit. on aura $X = f(\theta)$ avec X le vecteur position de la pince et θ le vecteur donnant les paramètres du robot (position du chariot et angles du bras articulé).

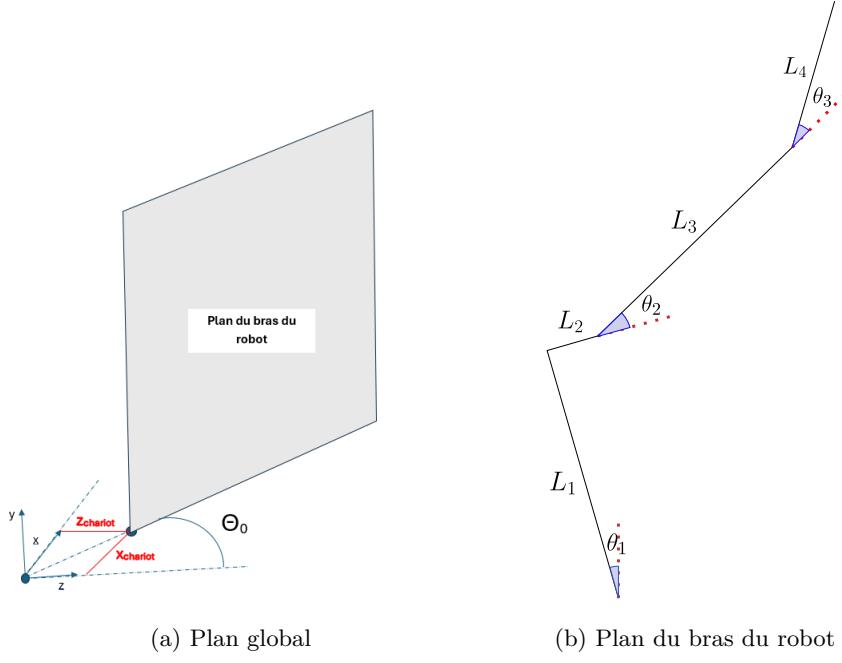


Figure 13

Pour déterminer la fonction f , on commence par utiliser la méthode de la séquence de transformation élémentaire (Elementary transform sequence) pour déterminer les différentes matrices de transformations :

$$E_1 = T_{t_x}(x_{chariot}) = \begin{pmatrix} 1 & 0 & 0 & x_{chariot} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_2 = T_{t_z}(z_{chariot}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z_{chariot} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_3 = T_{R_y}(\theta_0) = \begin{pmatrix} \cos \theta_0 & 0 & \sin \theta_0 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_0 & 0 & \cos \theta_0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_4 = T_{R_z}(\theta_1) = \begin{pmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_5 = T_{t_y}(L_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_6 = T_{t_x}(L_2) = \begin{pmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_7 = T_{R_z}(\theta_2) = \begin{pmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_8 = T_{t_x}(L_3) = \begin{pmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_9 = T_{R_z}(\theta_3) = \begin{pmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_{10} = T_{t_x}(L_4) = \begin{pmatrix} 1 & 0 & 0 & L_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ainsi, en faisant le produit de toutes ces matrices, on obtient la matrice de transformation globale.

$$T = E_1 \times E_2 \times E_3 \times E_4 \times E_5 \times E_6 \times E_7 \times E_8 \times E_9 \times E_{10}$$

Enfin, on peut en déduire les coordonnées du vecteur position X :

$$T = \begin{pmatrix} R_n^0 & X \\ 0_{1 \times 3} & 1 \end{pmatrix}$$

$$X = \begin{pmatrix} x_{chariot} + \cos \theta_0 (\cos \theta_1 (L_4 \cos(\theta_2 + \theta_3) + L_3 \cos \theta_2 + L_2) - \sin \theta_1 (L_4 \sin(\theta_2 + \theta_3) + L_3 \sin \theta_2 + L_1)) \\ \sin \theta_1 (L_4 \cos(\theta_2 + \theta_3) + L_3 \cos \theta_2 + L_2) + \cos \theta_1 (L_4 \sin(\theta_2 + \theta_3) + L_3 \sin \theta_2 + L_1) \\ z_{chariot} + -\sin \theta_0 (\cos \theta_1 (L_4 \cos(\theta_2 + \theta_3) + L_3 \cos \theta_2 + L_2) - \sin \theta_1 (L_4 \sin(\theta_2 + \theta_3) + L_3 \sin \theta_2 + L_1)) \end{pmatrix}$$

Ayant déterminé la fonction f , on peut calculer la jacobienne du robot :

$$J(\theta) = \left(\frac{\partial X}{\partial \theta_i} \right)$$

Grâce aux outils pré-définis par MATLAB, nous pouvons directement obtenir la matrice inverse de la jacobienne et compléter la case assignée du schéma bloc.

4.3 Comment choisir de K ?

En reprenant le résultat de la 4.1, si nous voulons obtenir une convergence exponentielle de l'erreur, e_X il faut choisir K de façon à ce que $e^{-tK} \xrightarrow[t \rightarrow +\infty]{} 0_3$

Nous avons donc opté pour le choix d'une matrice diagonale à coefficient strictement positif sur la diagonale. Cependant, il nous faut à nouveau choisir les coefficients. La théorie derrière le modèle nous invite à choisir K le plus grand possible pour que la convergence soit la plus rapide possible.

Cependant, dans la pratique, il ne nous est pas possible de considérer des valeurs des coefficients de K les plus grands possibles. En effet, ces derniers entraîneront une convergence trop rapide que le système ne pourra supporter, car ils nécessiteront des vitesses de rotation des angles du bras trop importantes.

Afin de choisir au mieux K nous avons décidé de tester expérimentalement plusieurs valeurs de K et de retenir celle qui présente les meilleurs résultats. N'ayant pas pu avoir accès au robot lors de nos expérimentations, nous nous

sommes contentés pour notre simulation informatique du robot de choisir des valeurs modérément élevées pour assurer une convergence rapide du système, mais qui semblent cohérents avec le suivi des moteurs.

5 Comparaison des méthodes de résolution

Dans cette partie, nous allons maintenant comparer les deux méthodes de résolution. Dans un premier temps, on remarque que les mouvements du bras et du chariot sont couplés dans tous les cas où ils peuvent l'être. En effet, il est possible qu'un mouvement du bras, puis du robot, puisse être observé découplé, mais cela représente seulement des cas très particuliers où ce découplage est la solution optimale.

Afin de comparer les deux méthodes de résolution, nous nous sommes appuyés sur 4 critères facilement identifiables pour chaque trajectoire donnée :

- (C.1) Quelle méthode donne des commandes périodiques dans les liaisons pour une trajectoire périodique
- (C.2) Quelle méthode demande des vitesses réalistes dans les liaisons ?
- (C.3) Quelle méthode est la plus précise ?
- (C.4) Quelle méthode optimise le mieux la trajectoire du chariot ?

Les critères (C.1) et (C.2) sont des critères plus forts que les critères (C.3) et (C.4). En effet, si une méthode ne donne pas une commande périodique en sortie pour une trajectoire périodique en entrée, on peut s'attendre à une trajectoire imprédictible sur le robot réel, et donc potentiellement dangereuse. De la même manière, si les vitesses demandées dans les liaisons sont trop élevées, cela peut être dangereux sur un robot réel.

Le critère (C.4) est beaucoup plus difficile à mesurer. En effet, il est adapté seulement aux trajectoires qui présentent une réelle distinction entre les deux méthodes pour le comportement du robot. Dans la majorité des cas, il n'est pas possible de voir facilement quelle trajectoire est meilleure.

Afin d'évaluer plus précisément les performances des deux méthodes, nous allons définir deux trajectoires test ainsi que divers critères de performances à comparer.

Une animation est disponible pour chaque trajectoire étudiée dans le dossier "Animations trajectoires test"

5.1 Présentation des trajectoires test

5.1.1 Trajectoire sinusoïdale

On souhaite définir dans un premier temps une trajectoire facile à réaliser pour le robot, et simple à analyser. Nous avons donc choisi de définir une trajectoire sinusoïdale dans l'espace que le robot devra suivre. Elle est définie de cette manière :

- la coordonnée en x est fixée constante
- la coordonnée en y est une fonction linéaire du temps
- la coordonnée en z est une fonction sinusoïdale du temps

Voici un exemple dans l'espace de trajectoire que peut suivre la pince (Fig 13).

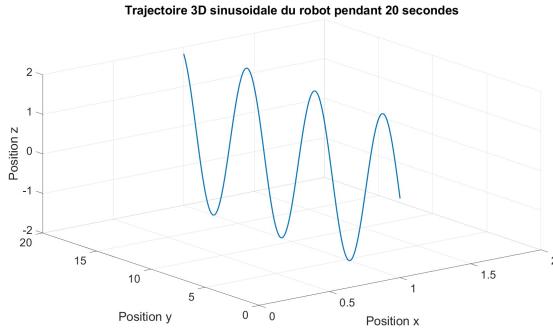


Figure 14: Exemple de trajectoire sinusoïdale pour la pince du robot

5.1.2 Trajectoire créneau

Dans un second temps, on souhaite définir une trajectoire qui soit toujours simple à réaliser, mais qui s'approchait plus de ce que pourrait réaliser un robot réel. Nous avons donc choisi l'exemple d'une peinture de mur : on imagine que le robot tient dans sa pince un pinceau ou un rouleau et qu'il recouvre l'entièreté d'un mur de longueur finie, et de hauteur suffisamment faible pour que le bras puisse atteindre le haut. On peut donc définir la trajectoire créneau que va suivre le robot de cette manière

- la coordonnée en x est fixée constante tout au long de la simulation
- le schéma à répéter est :
 - on fixe la coordonnée en y et la coordonnée en z augmente (on monte)
 - on fixe la coordonnée en z et la coordonnée en y augmente (on avance)
 - on fixe la coordonnée en y et la coordonnée en z diminue (on descend)
 - on fixe la coordonnée en y et la coordonnée en z augmente (on avance)
- on répète ce schéma tant que le mur n'est pas rempli
- une fois que le mur est rempli, le robot s'arrête

Voici un exemple dans l'espace de trajectoire que peut suivre la pince (Fig 14).

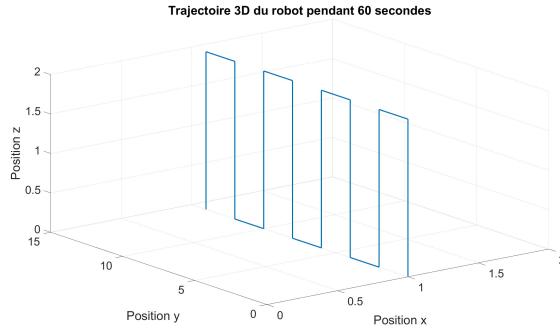


Figure 15: Exemple de trajectoire créneau pour la pince du robot

5.2 Résultats pour la trajectoire sinusoïdale

Animation à regarder : animation ass creneau.mp4 et animation moy creneau.mp4

L'exemple présenté ici a été réalisé pour les valeurs d'entrée suivantes :

- pulsation : 1
- Amplitude de z : 0.1
- déphasage : 0
- décalage constant de z : 0.1
- valeur constante de x : 0.1
- valeur initiale de y : 0.01
- vitesse selon y : 0.05

La première chose qu'on observe est que dans toutes les liaisons, la commande va bien être périodique et demander des vitesses réalisables. Les critères (C.1) et (C.2) ne permettent donc pas de choisir une méthode par rapport à l'autre. On en déduit aussi que les deux méthodes vont être utilisables pour commander le robot.

Pour montrer qu'une méthode est préférable à l'autre, on va devoir s'appuyer sur les critères (C.3) et (C.4)

Pour le critère (C.3), on observe déjà sur l'animation pour la méthode d'asservissement un léger décalage entre la trajectoire d'entrée, et la trajectoire de la pince du robot. Ce décalage s'explique par l'utilisation d'un correcteur proportionnel. Ce décalage peut être réduit avec l'augmentation du gain, mais

ne peut jamais être nul. Pour la simulation en exemple, on a utilisé un gain de $K = 10$ car l'écart est très proche de l'écart asymptotique, mais assez faible pour que les vitesses (notamment dans la partie transitoire) restent réalistes. Ce décalage est notamment prouvé directement par les courbes de sortie de la simulation Simulink ci-dessous (Fig 15).

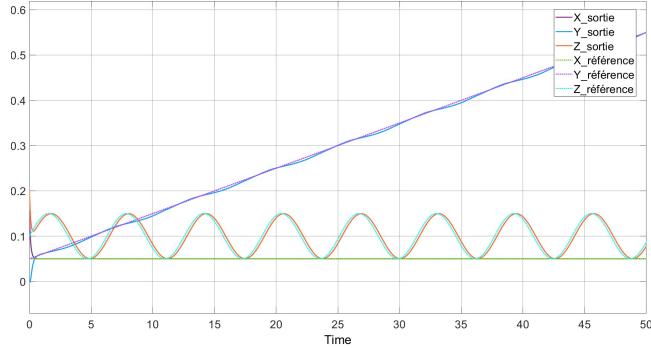


Figure 16: Évolution des coordonnées d'entrée et sortie de la pince en fonction du temps

Par ailleurs, la méthode des moyennes ne présente pas de décalage en sortie. Cela vient du fait que la méthode n'utilise pas de boucle d'asservissement, et qu'aucun décalage n'est créé dans la fonction directe.

Donc pour le critère (C.3) la méthode des moyennes est préférable.

Pour le critère (C.4), comme précisé plus tôt, on peut uniquement faire la différence que si elle est vraiment visible. En effet, dans notre exemple, sur l'animation, on observe que le chariot avance et recule sur l'axe y et aussi sur l'axe x. C'est aussi ce que l'on peut observer sur la courbe de trajectoire qu'à suivi le chariot dans le plan du sol ci-dessous (Fig 16).

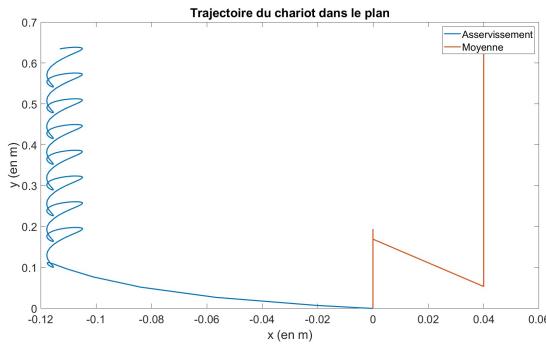


Figure 17: Trajectoire du chariot pour les deux méthodes dans le plan du sol (Oxy)

En excluant les points de la partie transitoire (qui sont ceux pour y faible), on observe bien une trajectoire rectiligne pour le chariot avec la méthode des moyennes en orange (et sans retour en arrière, ce qui peut être vu sur l'animation), tandis que la trajectoire avec la méthode d'asservissement en bleu présente des mouvements supplémentaires sur l'axe x et des retours en arrière si on regarde seulement l'axe y. Ces mouvements supplémentaires ne sont pas dû à la valeur du correcteur K (car on les retrouve identiques si on change sa valeur), mais bien à cause du calcul direct (avec l'utilisation de la Jacobienne et de la géométrie directe).

Donc pour le critère (C.4) aussi la méthode des moyennes est préférable.

Par conséquent, pour une trajectoire sinusoïdale, la méthode des moyennes est préférable à la méthode d'asservissement, mais cette dernière donne quand même des commandes utilisables.

5.3 Résultats pour la trajectoire créneau

Animation à regarder : animation ass creneau.mp4 et animation moy creneau.mp4

L'exemple présenté ici a été réalisé pour les valeurs d'entrée suivantes :

- Longueur des segments selon z : 0.1
- Longueur des segments selon y : 0.1
- vitesse de déplacement le long des segments : 0.5
- Longueur totale du tracé : 1
- valeur constante de x : 0.05

La première chose qui saute aux yeux lorsqu'on regarde les animations est le comportement du robot qui semble chaotique pour la méthode des moyennes. Si la pince suit bien la trajectoire d'entrée, le robot ne semble pas du tout suivre de trajectoire périodique, comme on pourrait s'y attendre. Et c'est bien ce que l'on observe dans les consignes pour les différentes liaisons, comme pour l'angle θ_2 dans le bras (Fig 17)

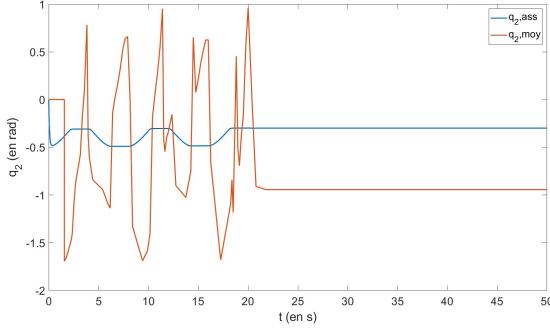


Figure 18: Evolution de θ_2 pour les deux méthodes en fonction du temps

Sur ce graphique, on observe bien une commande périodique de l'angle θ_2 pour la méthode d'asservissement (avant d'arriver à la position finale vers 18s). Ce n'est pas le cas pour la méthode des moyennes. On observe bien une non-périodicité avant la partie finale.

Ce comportement chaotique du robot vient de la méthode de résolution des moyennes. En effet, si elle est adaptée à des trajectoires plus simples, comme les sinusoïdes, elle ne l'est pas pour des trajectoires montrant des variations abruptes, comme ici lors du passage du déplacement selon y vers le déplacement selon z. Par conséquent, le calcul de moyenne pour la position du chariot donne des commandes chaotiques, ce qui fait que le bras doit aussi compenser et suit lui aussi une trajectoire chaotique.

Une autre conséquence de l'erreur du calcul de moyenne est la présence de pics de vitesse presque infinie dans les liaisons. En effet, si on reprend l'exemple de la consigne en θ_2 , on observe bien des déplacements abrupts aux changements d'axe de déplacement de la pince. La vitesse dans cette liaison peut atteindre presque 5rad/s, et le sens de rotation change de sens de façon quasi instantanée, ce qui n'est physiquement pas possible dans la réalité (compte tenu de l'inertie des solides). Donc cette commande ne serait pas réalisable pour un robot dont les masses ont été prises en compte.

Par conséquent, on préférera la méthode d'asservissement pour une trajectoire crénée d'après les critères (C.1) et (C.2).

Même si ces critères sont plus faibles, on peut tout de même citer que le critère (C.3) montre que la méthode des moyennes est préférable à celle de l'asservissement au niveau de la précision (pour la même raison que pour la trajectoire sinusoïdale) et qu'on ne peut pas choisir facilement de méthode préférable pour le critère (C.4) du fait des défauts du déplacement du chariot pour les deux méthodes.

5.4 Bilan

Voici un tableau qui récapitule la méthode préférable par critère pour la trajectoire sinusoïdale et la trajectoire créneau

Critère	Trajectoire sinusoïdale	Trajectoire créneau
(C.1)	Non différentiables	Asservissement
(C.2)	Non différentiables	Asservissement
(C.3)	Moyennes	Moyennes
(C.1)	Moyennes	Non différentiables

Finalement, la méthode d'asservissement va toujours être utilisable, Quelle que soit la trajectoire, tandis que la méthode des moyennes ne sera pas toujours utilisable, mais sera préférable à celle d'asservissement dans certains cas.

6 Conclusion

Notre travail cette année nous a donc permis de mettre en place plusieurs algorithmes répondant à notre problème. Un premier basé sur la méthode des moyennes implémenté en MATLAB et utilisant la cinématique inverse, un second utilisant de nouveau la méthode des moyennes implémenté cette fois sous Simulink, et une autre méthode basée sur un asservissement en position du robot. Ces deux programmes présentent des résultats variables selon les situations, mais ils permettent d'obtenir des résultats intéressants pour le problème que l'on s'était posé.

Nos programmes présentent des aspects intéressants comme la capacité à dissocier le mouvement du chariot et du bras de robot pour la méthode des moyennes, ou justement d'aligner les deux sans distinction dans l'asservissement. En accord avec notre objectif, l'étude et les programmes peuvent être facilement adaptés à d'autres robots que le TurtleBot que l'on a utilisé comme support lors de ce projet. Le projet est donc adaptable et aborde plusieurs méthodes pour permettre un fonctionnement dans de nombreuses conditions. Le programme a de bons résultats pour des trajectoires avec un minimum de mouvement brusque, comme pourrait le faire un robot lors d'une tâche de pick and place. L'objectif semble être réalisé dans la mesure où le robot peut en effet effectuer des tâches tout en effectuant en simultané un mouvement du chariot et du bras.

Néanmoins, il est nécessaire de prendre en compte plusieurs problèmes. Le programme ne permet pas encore de prendre en compte les conditions imposées sur la position de la pince. Si les conditions imposées par la tâche sont trop complexes, il sera nécessaire de changer en profondeur les programmes que nous avons rédigés lors de notre projet. De plus, avant l'application à un vrai robot, il est nécessaire de prendre en compte toute la dynamique du robot en question. Enfin, on peut également citer l'holonomie du robot et la nécessité de modifier le correcteur utilisé pour obtenir des résultats satisfaisants.

En conclusion, notre travail permet bien de donner les commandes du robot à suivre pour réaliser certaines tâches. Dans la continuité de ce projet, il serait

nécessaire de les tester sur un robot réel pour étudier le comportement dynamique et parfaire notre approche ainsi que d'optimiser les programmes avec des approches plus fines. La suite de ce projet permettrait alors des solutions plus efficaces et plus facilement applicables à la géométrie et à la dynamique de chaque robot, qui pourront donc être plus facilement mises en place pour répondre aux enjeux de pénétrabilité, de santé et de temps d'aujourd'hui et de demain.

Nous notons dans ce projet l'importance du travail de Peter Cork et de sa toolbox qui nous a permis de créer le robot et de générer des animations.

Nous tenions à l'issu de ce projet remercier notre encadrant ainsi que les professeurs et encadrant d'APP-API pour leur retour et pour le temps qu'ils ont passé à s'assurer du bon déroulement de notre projet.