

Modélisation et Simulation 3RRR

Judicaël Corpet et Dan Calamia

May 26, 2025

Abstract

Cela fait l'objet de rapport de fin de projet dans le cadre de l' UE MU4RBR05 encadrés par M. Ben Amar et Mme. Pasqui.

1 Introduction

L'objectif de ce projet était de concevoir un modèle de robot manipulateur de type 3RRR sous Solid-Works et de simuler ce modèle sous langage python.

2 Modélisation et Simulation

2.1 Modèle Géométrique Inverse (MGI)

Le modèle géométrique inverse, essentiel pour la commande du robot, consiste à déterminer les angles articulaires des actionneurs (α_i) à partir d'une pose souhaitée de l'effecteur définie par (x_E, y_E, θ_E) . Nous avons le choix entre deux méthodes distinctes pour résoudre notre problème de modélisation, une méthode numérique et une méthode analytique.

2.1.1 Méthode numérique par résolution itérative (Newton-Raphson)

Cette méthode consiste à résoudre numériquement un système non-linéaire de 6 équations à 6 inconnues, représentant les contraintes géométriques. Elle nécessite une estimation initiale des angles articulaires et converge vers la solution par itérations successives.

2.1.2 Méthode analytique basée sur le modèle inverse 2R plan

La seconde méthode exploite une solution fermée analytique inspirée du modèle classique d'un manipulateur plan à deux articulations (2R). Elle permet d'obtenir directement les angles articulaires sans itérations, avec un choix de signe pour gérer les configurations multiples.

2.1.3 Comparatifs des méthodes

Dans notre cas, nous avons principalement utilisé la méthode analytique, car nous travaillons sur un modèle simplifié en deux dimensions où le modèle analytique est simple et rapide à résoudre.

Critère	Méthode analytique	Méthode numérique
Temps de calcul	Très rapide	Plus lent (itératif)
Complexité du modèle	Faible	Élevée
Robustesse	Bonne si atteignable	Variable, dépend de l'initialisation
Nécessité d'une estimation initiale	Non	Oui
Applicabilité	Modèles simples	Modèles complexes et généraux

Table 1: Comparaison des deux méthodes de résolution du modèle géométrique inverse dans le cas d'un robot 3RRR dans un environnement 2D.

2.2 Modèle Cinématique (MCD)

Le modèle cinématique d'un robot parallèle établit la relation entre les vitesses articulaires des actionneurs, notées $\dot{\alpha}_i$, et le torseur des vitesses de l'effecteur (v_x, v_y, ω_z) , exprimé dans le repère local R_E .

On peut écrire cette relation sous la forme suivante :

$$Av_i = B\dot{\alpha}_i \quad (1)$$

$$\begin{pmatrix} \cos \gamma_1 & \sin \gamma_1 & d_1 \\ \cos \gamma_2 & \sin \gamma_2 & d_2 \\ \cos \gamma_3 & \sin \gamma_3 & d_3 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} e_1 & 0 & 0 \\ 0 & e_2 & 0 \\ 0 & 0 & e_3 \end{pmatrix} \begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{pmatrix} \quad (2)$$

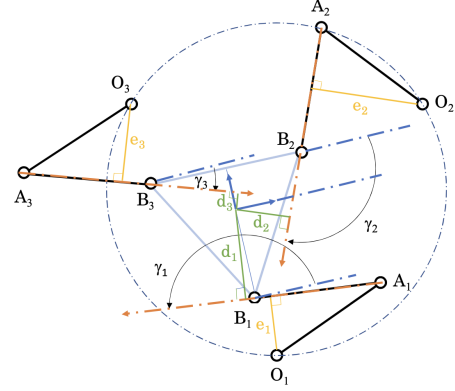


Figure 1: Rappels des paramètres du modèle cinématique

Nous avons résolu le modèle cinématique sous cette forme, tirée des spécificités géométriques et cinématiques du robot, afin d'identifier les singularités de notre système.

2.2.1 Identification des singularités

Il existe deux types de singularités : les singularités parallèles, qui sont liées à la matrice jacobienne A, et les singularités en série, liées à la matrice B.

La première singularité parallèle arrive lorsque que les trois droites A_iB_i sont parallèle (voir figure: 1). Les vitesses orthogonales aux droites A_iB_i deviennent incontrôlable. Mathématiquement, on retrouve cette conditions lorsque le déterminant de la matrice A vaut 0 et que les angles γ_i soient des multiples de π .

La deuxième singularité parallèle a lieu lorsque les trois droites A_iB_i sont concourantes en un point unique. La vitesse angulaire selon l'axe z devient alors incontrôlable. Mathématiquement, on retrouve cette conditions lorsque le déterminant de la matrice A vaut 0.

La troisième singularité est une singularité en série, et elle apparaît lorsqu'un des bras est complètement tendu. Cela reflète une limite de l'espace de travail. Mathématiquement, cela se produit lorsque le déterminant de la matrice B est nul. À noter que cette singularité peut également être détectée lorsque l'un des trois angles β , définis comme les angles entre les droites OA et AB , est égal à zéro. Cela permet d'identifier cette singularité sans passer par le MCD.

2.2.2 Détection des singularités

Grâce à la géométrie du système et en se référant à la figure 1 on a pu calculer les angles γ_i et les distances e_i et d_i .

Pour chaque bras, on calcule d'abord l'angle relatif entre le segment A_iB_i et l'axe x_e de l'effecteur. Cela revient à mesurer l'orientation de $A_i\vec{B}_i$ par rapport au repère de l'effecteur, en soustrayant l'orientation globale de l'effecteur θ_e .

$$\gamma_i = \text{atan2}(yA_iB_i, xA_iB_i) - \theta_e \quad (3)$$

Ensuite, on calcule les distances orientées d_i , définies comme la norme du segment perpendiculaire à la droite A_iB_i partant du centre de l'effecteur.

$$d_i = \frac{A_i\vec{B}_i \wedge A_i\vec{E}_i}{\|A_i\vec{B}_i\|} \quad (4)$$

Puis, on calcule les distances e_i , qui correspondent à la norme du segment perpendiculaire à la droite A_iB_i , partant de O_i , le centre des liaisons pivot motorisées.

$$e_i = \frac{A_i\vec{B}_i \wedge A_i\vec{O}_i}{\|B_i\vec{A}_i\|} \quad (5)$$

On possède désormais les matrices A et B. On peut facilement calculer les déterminants de ces matrices grâce à la bibliothèque NumPy. Cependant, pour détecter ces singularités en simulation Python, il faut prendre en compte des plages de tolérance lors du calcul des déterminants. Par exemple, dans notre cas, nous avons fixé une tolérance de 10^{-3} pour la détection du déterminant de A afin de mieux corriger certaines approximations survenues lors du calcul des matrices.

2.2.3 Optimisation de l'évitement des singularités

Nous avons choisi de modifier l'orientation de l'effecteur lorsqu'on s'approche trop près d'une singularité, ce qui a pour objectif de modifier la position et les orientations des bras du robot. Lorsque la singularité n'est plus détectée le robot se replace dans son orientation d'origine pour être dans la position la plus optimale. (Comme on peut le voir dans les vidéos envoyées, la simulation du robot sans évitement de singularité, et celle avec évitement.) On peut remarquer que le robot oscille rapidement autour des singularités, on peut régler ce problème en diminuant le pas de temps.

2.3 Simulation Pygame

Notre simulation comprend trois modes distincts, permettant de valider nos compétences dans le cadre de ce projet.

- Le premier mode, activé en appuyant sur la touche « m », correspond au mode libre qui vous permet de déplacer l'effecteur du robot sans contrainte liée aux singularités parallèles, en utilisant les touches fléchées de votre clavier. Cependant, vous pourrez remarquer les singularités en série lorsque le bras est tendu : le robot ne peut plus avancer dans cette direction.
- Le deuxième et le troisième mode tracent respectivement un cercle (en appuyant sur « o ») et un trèfle à quatre feuilles (en appuyant sur « p »). Dans ces modes, le robot évite les singularités parallèles tout en s'assurant de ne pas engendrer de singularités en série, en modifiant l'orientation de l'effecteur.

3 Conception

3.1 Optimisation

L'un des objectifs était de maximiser l'espace de travail. Afin de prendre en compte les contraintes d'impression 3D, nous nous sommes fixés sur une base circulaire de 30 cm de diamètre, avec des segments d'une longueur de 10 cm. L'objectif était donc de déterminer les dimensions optimales de l'effecteur pour avoir la surface de travail maximale. Nous avons ainsi utilisé un algorithme calculant l'ensemble des aires de travail en fonction des paramètres fixés, en tenant compte également des singularités de notre modèle (qui apparaissent en rouge sur la figure 2) mais pas des contraintes physiques. Le résultat théorique nous a donc donné un effecteur de 15 cm de côtés pour obtenir une

surface maximale. Un autre critère nous a permis de déterminer la taille optimale de l'effecteur : la mobilité. Ainsi nous avons pu identifier grâce aux simulations sur Solidworks que la dimension de l'effecteur était proportionnelle à la longueur des segments, c'est-à-dire avec un côté de 10 cm. À partir de ces résultats, nous avons décidé, pour la simulation, de prendre des segments de 120mm de long.

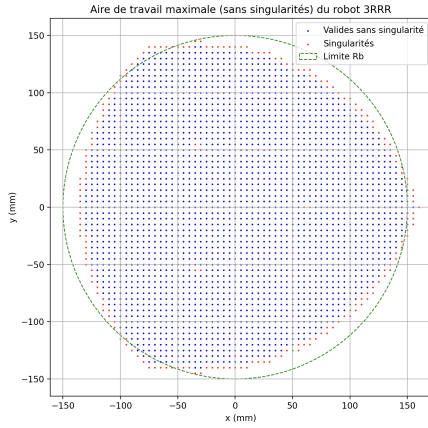


Figure 2: Surface optimale théorique

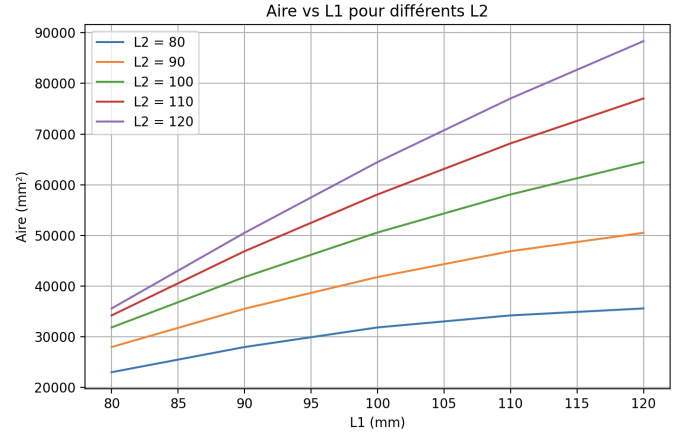


Figure 3: Calcul du L2 optimal en fonction de L1

3.2 Conception avec SolidWorks

La réalisation du robot sur le logiciel SolidWorks a permis d'identifier clairement les contraintes physiques, de les transposer sur le modèle simulé et de tester différentes dimensions de segments, d'effecteur ainsi que l'écartement des moteurs. Les moteurs étaient accessibles sur internet, en open source. Ils ont constitué la base de modélisation du robot pour le construire à l'échelle.

Les contraintes physiques ont ensuite été modélisées par des contraintes de distance entre les pièces.

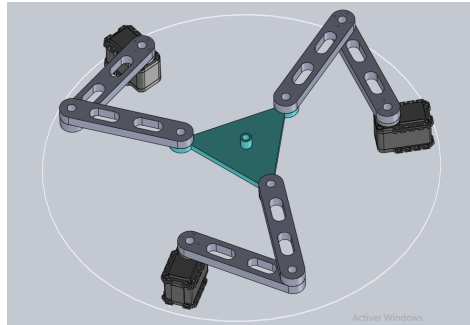


Figure 4: Modèle SolidWorks

4 Conclusion

Ce projet a permis de modéliser et de simuler un robot parallèle 3RRR en utilisant des outils numériques.

La simulation a permis de visualiser les mouvements, détecter les singularités et valider les configurations. Les modèles géométriques directe et inverse ont été mis en oeuvre pour analyser ces singularités et développer un modèle capable de les prendre en compte.

La conception a permis un travail complémentaire permettant de valider les configurations issues de la recherche d'optimisation de l'espace de travail.

Ce projet ouvre la voie à une plus grande optimisation future concernant les dimensions, ainsi qu'au travail d'automatisation de l'effecteur pour suivre et tracer des trajectoires ou des formes données.

5 Annexe

5.1 Rappels des variables utilisées

- x_E, y_E : position (x,y) en mètre de l'effecteur.
- θ_E : orientation en radian de l'effecteur.
- O_i, α_i : position de la première liaison pivot, qui est motorisée ; on note α_i l'angle généré par celle-ci, $\forall i \in \{1, 2, 3\}$.
- A_i, β_i : position de la deuxième liaison pivot ; on note α_i l'angle généré par celle-ci, $\forall i \in \{1, 2, 3\}$.
- B_i : position de la troisième liaison pivot qui relie le bras à l'effecteur $\forall i \in \{1, 2, 3\}$.
- R_b, R_e : rayon de la base (0,13m) et rayon de l'effecteur (0,07m)
- $L1, L2$: longueur du premier et deuxième segment (0,12m)

5.2 Simulation

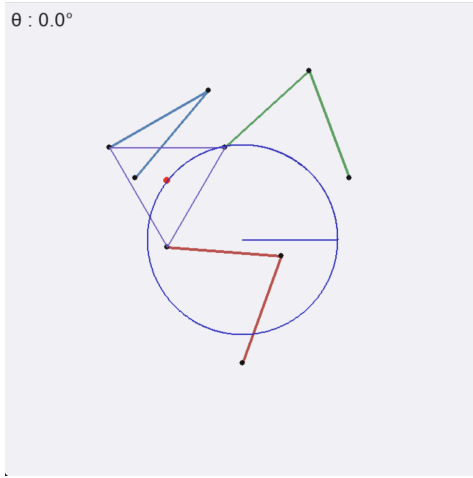


Figure 5: Suivre de trajectoire, traçage d'un cercle

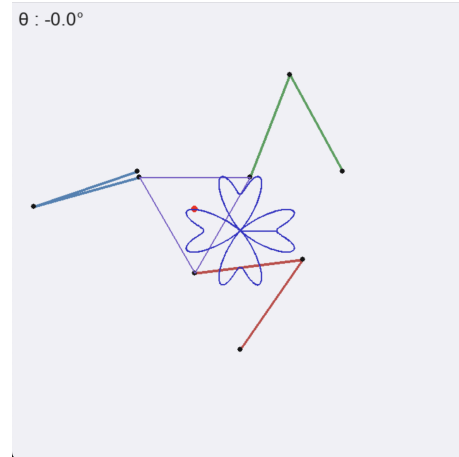


Figure 6: Suivre de trajectoire, traçage d'un trèfle