

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
ÉCOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Électronique

Conception et réalisation d'un bras manipulateur à 6 degrés de
libertés

CHIKHI Arezki & BABOURI Ayoub

Sous la direction de **Dr. LARBES Cherif** ENP

Présenté et soutenu publiquement le (24/06/2025)

Composition du jury :

Président : Pr. Taghi Mohamed ENP
Promoteur : Dr. LARBES Cherif ENP
Examinatrice : Dr. LANI Fatiha ENP

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
ÉCOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Électronique

Conception et réalisation d'un bras manipulateur à 6 degrés de
libertés

CHIKHI Arezki & BABOURI Ayoub

Sous la direction de **Dr. LARBES Cherif** ENP

Présenté et soutenu publiquement le (24/06/2025)

Composition du jury :

Président : Pr. Taghi Mohamed ENP
Promoteur : Dr. LARBES Cherif ENP
Examinatrice : Dr. LANI Fatiha ENP

الملخص

تُعدّ الروبوتات من الركائز الأساسية في مجال الأتمتة الحديثة، لا سيما من خلال استخدام الأذرع الروبوتية. تتيح هذه الأنظمة تنفيذ مهام معقدة بدقة وتكرارية وأمان. يندرج هذا المشروع في هذا السياق من خلال تصميم وإنجاز ذراع روبوتي بست درجات حرية، يجمع بين النمذجة، التصميم الميكانيكي، والتحكم بمساعدة بيئة ROS. يهدف إلى استكشاف الجوانب التقنية والعملية للروبوتات الصناعية ضمن إطار تعليمي.

الكلمات المفتاحية: الروبوتيك، الذراع الروبوتي، ست درجات حرية، محركات خطوة بخطوة، التصميم بمساعدة الحاسوب، ROS، MoveIt، Gazebo، النمذجة الحركية.

Abstract

Robotics plays a key role in modern automation, especially through the use of robotic arms. These systems are capable of performing complex tasks with precision, repeatability, and safety. This project aims to design and build a 6-degree-of-freedom robotic arm, combining mechanical design, modeling, and control using the ROS framework. It reflects a growing interest in exploring industrial robotics within an educational context.

Keywords : robotics, robotic arm, 6 DOF, stepper motors, CAD, ROS, MoveIt, Gazebo, kinematic modeling, embedded system.

Résumé

La robotique occupe une place centrale dans l'automatisation moderne, notamment à travers l'utilisation des bras manipulateurs. Ces systèmes permettent d'accomplir des tâches complexes avec précision, répétabilité et sécurité. Ce projet s'inscrit dans cette dynamique en proposant la conception et la réalisation d'un bras manipulateur à 6 degrés de liberté, combinant modélisation, conception mécanique et commande assistée par ROS. Il vise à explorer les aspects techniques et pratiques de la robotique industrielle dans un cadre pédagogique.

Mots clés : robotique, bras manipulateur, 6 DDL, moteurs pas à pas, CAO, ROS, MoveIt, Gazebo, modélisation cinématique, système embarqué.

Remerciements

Nous tenons tout d'abord à exprimer notre profonde gratitude à **Monsieur LARBES Cherif**, notre encadrant pédagogique, pour sa disponibilité, ses conseils éclairés et son accompagnement constant tout au long de ce projet. Son expertise et son engagement ont été essentiels à la réussite de ce travail.

Nous remercions également **Monsieur LAHLAH Karim** et **Monsieur BOUSBAI M'hamed** pour leur soutien technique et leur précieuse assistance dans les différentes phases de conception et de réalisation.

Nos remerciements les plus sincères vont à l'entreprise **ACMP**, pour l'accueil chaleureux, les ressources mises à disposition et l'environnement propice au développement du projet. Nous tenons à remercier tout particulièrement **Monsieur BELKHIR Brahim**, dont l'implication et l'aide technique ont été déterminantes pour la concrétisation de notre bras manipulateur.

Nous souhaitons aussi remercier les membres du jury pour l'intérêt qu'ils ont porté à ce travail, et plus particulièrement **Monsieur TAGHI Mohamed**, chef de département, ainsi que **Madame LANI Fatiha**, enseignante et membre du jury, pour leur disponibilité, leurs remarques constructives et leur accompagnement durant tout notre cursus.

Nous n'oublions pas l'ensemble des enseignants du département pour la qualité de leur enseignement, ainsi que nos camarades et amis pour leur entraide, leurs encouragements et les moments de collaboration partagés.

Enfin, nous adressons nos plus vifs remerciements à nos familles pour leur soutien indéfectible, leur patience et leur confiance, qui nous ont permis d'avancer sereinement dans ce parcours.

CHIKHI Arezki et BABOURI Ayoub.

Table des matières

Liste des tableaux

Table des figures

Liste des acronymes

Introduction générale	11
1 Généralités sur la robotique	12
1.1 Introduction	12
1.2 Domaines d'application de la robotique	13
1.3 Composants principaux d'un robot	15
1.4 Paramètres de base d'un bras manipulateur	16
1.5 Outils utilisés dans le projet	17
1.6 Conclusion	19
2 Conception du bras manipulateur à 6 DDL	20
2.1 Introduction	20
2.2 Choix de l'architecture	21
2.2.1 Type de structure adoptée	21
2.3 Étude mécanique	21
2.3.1 Choix des matériaux	21
2.3.2 Stratégies d'allègement de la structure	21
2.3.3 Dimensionnement des segments	23
2.4 Choix des actionneurs	23
2.4.1 Type d'actionneurs sélectionnés	23
2.4.2 Stratégie de sélection des couples	23

2.4.3	Avantages du choix des moteurs pas à pas	24
2.5	Principe de fonctionnement des moteurs pas à pas	24
2.6	Conception assistée par ordinateur (CAO)	25
2.7	Conclusion	28
3	Modélisation du bras manipulateur	29
3.1	Introduction	29
3.2	Définition des repères et conventions	29
3.3	Paramétrage selon Denavit-Hartenberg (DH)	29
3.3.1	Présentation de la méthode DH	29
3.3.2	Pourquoi utiliser la méthode DH ?	30
3.3.3	Paramètres DH pour le bras manipulateur	30
3.3.4	Paramètres DH du bras manipulateur à 6 DDL	30
3.4	Modélisation Cinématique Directe	31
3.4.1	Matrices de transformation homogène	31
3.5	Méthode de Paul pour un robot à 6 DDL	32
3.5.1	Principe général	32
3.5.2	Détermination de θ_1	32
3.5.3	Détermination de θ_2 et θ_3	32
3.5.4	Détermination de θ_4 , θ_5 et θ_6	33
3.6	Calcul des couples et de l'inertie	33
3.6.1	Hypothèse de fonctionnement	33
3.6.2	Calcul du couple gravitationnel	34
3.6.3	Caractéristiques des moteurs et segments	35
3.6.4	Valeurs des couples gravitationnels	35
3.6.5	Calcul du couple d'inertie	35
3.6.6	Moments d'inertie des segments	36
3.6.7	Couple total par segment	36
3.6.8	Transmission du couple via poulie-courroie	36
3.6.9	Couples générés et transmis	37
3.6.10	Capacité maximale de charge	37

3.7	Conclusion	37
4	Commande du bras manipulateur	38
4.1	Introduction	38
4.2	Méthode classique basée sur la modélisation	38
4.2.1	Principe de fonctionnement	38
4.2.2	Implémentation sur ESP32	38
4.3	Commande avec ROS et MoveIt	39
4.3.1	Présentation de ROS	39
4.3.2	Génération du modèle URDF	41
4.3.3	Simulation dans Gazebo	42
4.3.4	Commande réelle avec MoveIt et rosserial	43
4.4	Comparaison entre les deux approches	43
4.5	Étapes de mise en œuvre	44
4.5.1	Conception mécanique	44
4.5.2	Simulation et choix des moteurs et matériaux	44
4.5.3	Usinage des pièces	44
4.5.4	Génération du modèle URDF	45
4.5.5	Configuration MoveIt et ROS	45
4.5.6	Simulation dans Gazebo	46
4.5.7	Configuration de MoveIt avec Rviz	47
4.5.8	Choix et intégration du matériel	47
4.5.9	Intégration ROS sur microcontrôleur	47
4.5.10	Tests et validation	48
4.6	Conclusion	48
	Conclusion générale	49
	Bibliographie	50

Liste des tableaux

2.1	Répartition des moteurs pas à pas selon les articulations	24
3.1	Tableau des paramètres DH	30
3.2	Moteurs utilisés et masses associées	35
3.3	Calcul des couples gravitationnels	35
3.4	Moments d'inertie des segments	36
3.5	Couples totaux par segment	36
3.6	Couples générés et transmis par les moteurs	37
3.7	Poids maximal de la charge supporté par segment	37
4.1	Comparaison des deux méthodes de commande	44

Table des figures

1.1	Robot industriel	13
1.2	Robot mobile	13
1.3	Robot médical	14
1.4	Robot de service	14
1.5	Robot humanoïde	14
1.6	Composants principaux d'un robot	15
1.7	Degrés de liberté d'un bras manipulateur	16
1.8	Espace de travail d'un bras manipulateur	17
1.9	Outils logiciels utilisés	18
2.1	Vue d'ensemble du bras manipulateur à 6 DDL	20
2.2	Exemple d'allègement d'une pièce mécanique	22
2.3	Exemple de structure alvéolaire hexagonale	22
2.4	Moteur pas à pas	23
2.5	Principe de fonctionnement des moteurs pas à pas	25
2.6	Modélisation de la base et de la poulie du bras manipulateur	26
2.7	Modélisation de la 1ère articulation du bras manipulateur	27
2.8	Positionnement des moteurs dans les articulations du bras manipulateur	27
2.9	Modélisation du réducteur planétaire dans la 6 ^e articulation	28
3.1	Bras robotisé en position horizontale	34
3.2	Système poulie-courroie utilisé pour la transmission	36
4.1	Application ROS	39
4.2	Architecture de ROS	39
4.3	Model utilisé pour la génération de l'URDF	42

4.4	Configuration de MoveIt pour le bras manipulateur	45
4.5	Simulation du bras dans Gazebo	46
4.6	Configuration de MoveIt avec Rviz	47

Liste des acronymes

- **DDL** : Degrés De Liberté
- **CAO** : Conception Assistée par Ordinateur
- **ROS** : Robot Operating System
- **URDF** : Unified Robot Description Format
- **Gazebo** : Simulateur robotique 3D compatible avec ROS
- **MoveIt** : Framework ROS pour la planification et le contrôle de mouvement
- **ESP32** : Microcontrôleur Wi-Fi et Bluetooth utilisé dans les systèmes embarqués
- **NEMA** : Norme américaine définissant les dimensions des moteurs pas à pas
- **DH** : Denavit-Hartenberg (méthode de modélisation cinématique)
- **STL** : Stereolithography (format de fichier pour l'impression ou la modélisation 3D)
- **USB** : Universal Serial Bus
- **PCB** : Printed Circuit Board
- **API** : Application Programming Interface
- **ANN** : Artificial Neural Network

Introduction générale

“I don’t know what else to do, but I’m going to put a robot into the real world.”

Joseph Engelberger ("Father of Robotics") [1]

La robotique constitue aujourd’hui un pilier central des avancées technologiques modernes. Que ce soit dans les domaines industriel, médical ou domestique, les robots permettent d’exécuter des tâches complexes avec précision, rapidité et sécurité [2]. Parmi les structures robotiques les plus emblématiques, le bras manipulateur occupe une place prépondérante. Inspiré du bras humain, il reproduit des mouvements articulés et permet de manipuler divers objets dans un espace défini, ce qui le rend indispensable dans des applications telles que les chaînes de production, la chirurgie assistée ou l’exploration spatiale [3].

Ce projet de fin d’études s’inscrit dans cette dynamique, en proposant la conception, la modélisation et la commande d’un bras manipulateur à six degrés de liberté (DDL). Il vise à intégrer les aspects mécaniques, électroniques et logiciels dans un système cohérent, modulaire et évolutif. L’utilisation de moteurs pas à pas, la modélisation cinématique ainsi que la mise en œuvre d’une commande avancée basée sur ROS (Robot Operating System) [4] illustrent concrètement les compétences acquises au cours de notre formation.

Ce mémoire est structuré en quatre chapitres :

Chapitre 1 : Présente les généralités sur la robotique : définitions, domaines d’application, composants principaux d’un robot, ainsi que les paramètres fondamentaux d’un bras manipulateur.

Chapitre 2 : Traite de la conception mécanique du bras manipulateur, en détaillant les choix structurels, les matériaux utilisés, les moteurs pas à pas sélectionnés, et la modélisation assistée par ordinateur (CAO).

Chapitre 3 : Est dédié à la modélisation cinématique du bras. Il expose la définition des paramètres de Denavit-Hartenberg et la représentation mathématique des transformations successives au sein de la chaîne cinématique.

Chapitre 4 : Présente la commande du bras manipulateur. Deux approches sont explorées : une commande standard basée sur les équations cinématiques, et une commande plus avancée via ROS, incluant la simulation dans Gazebo et l’intégration de MoveIt pour la planification de trajectoires.

Enfin, ce mémoire se conclut par une synthèse des résultats obtenus et une ouverture sur des perspectives d’amélioration et d’évolution futures.

Chapitre 1

Généralités sur la robotique

1.1 Introduction

Le terme *robot* est apparu pour la première fois en 1920 dans la pièce de théâtre *R.U.R. (Rossum's Universal Robots)* de l'écrivain tchèque Karel Čapek. Il provient du mot slave « robota », qui signifie « travail forcé » ou « corvée ». Ce terme a ensuite été popularisé dans la littérature de science-fiction, notamment par Isaac Asimov, qui a proposé dans les années 1940 les célèbres trois lois de la robotique [5].

Aujourd'hui, la robotique est une discipline interdisciplinaire regroupant la mécanique, l'électronique, l'informatique et l'automatique, et qui a pour but de concevoir, construire et programmer des machines capables d'exécuter des tâches de manière autonome ou semi-autonome.

La définition exacte d'un robot industriel varie légèrement selon les organisations :

- **Robot Institute of America (RIA)** : « *Un robot est un manipulateur reprogrammable à fonctions multiples. Il est conçu pour déplacer des matériaux, des pièces, des outils ou des instruments spécialisés suivant des trajectoires variables programmées, en vue d'accomplir des tâches très diverses.* » [6].
- **Association Japonaise de Robotique Industrielle** : « *Tout mécanisme permettant d'effectuer, en tout ou en partie, une tâche normalement réalisée par l'homme.* » [7].
- **Association Française de Robotique Industrielle (AFRI)** : « *Un robot industriel est une machine formée de divers mécanismes comportant divers degrés de liberté, ayant souvent l'apparence d'un ou de plusieurs bras se terminant par un poignet capable de maintenir un outil, une pièce ou un instrument de contrôle. En particulier, son unité de contrôle doit contenir un système de mémorisation, et il peut parfois utiliser des accessoires sensitifs et adaptables qui tiennent compte de l'environnement et des circonstances. Ces machines, ayant un rôle pluridisciplinaire, sont généralement conçues pour effectuer des fonctions répétitives, mais sont adaptables à d'autres fonctions.* ».
- **Organisation Internationale de Normalisation (ISO)** : « *Une machine formée par un mécanisme incluant plusieurs degrés de liberté, ayant souvent l'apparence d'un ou plusieurs bras se terminant par un poignet capable de tenir des outils, des pièces ou un dispositif d'inspection.* » [8].

Ainsi, bien qu'il n'existe pas de définition unique, toutes convergent vers l'idée d'un système mécatronique reprogrammable, capable d'interagir avec son environnement et d'exécuter des tâches variées de manière automatique.

Il est important de distinguer un robot d'une machine automatisée : cette dernière exécute un ensemble fixe d'instructions dans un contexte déterminé, tandis qu'un robot possède généralement une certaine capacité de perception et d'adaptation, le rendant plus flexible et intelligent dans l'exécution de tâches variées.

1.2 Domaines d'application de la robotique

La robotique connaît aujourd'hui une diversification croissante de ses domaines d'application, parmi lesquels on distingue :

Robotique industrielle

C'est le secteur le plus ancien et le plus mature. Il inclut principalement les bras manipulateurs utilisés dans les chaînes de production pour des tâches de manipulation, d'assemblage, de soudage, de peinture ou d'emballage.



FIGURE 1.1 – Robot industriel

Robotique mobile

Cette catégorie regroupe les robots capables de se déplacer de manière autonome, tels que les robots d'exploration planétaire, les véhicules autonomes (AGV), ou encore les drones.

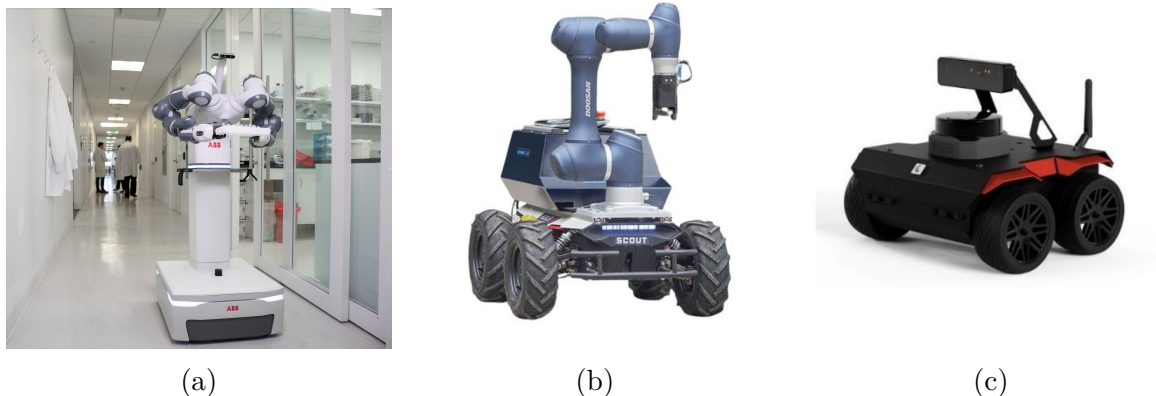


FIGURE 1.2 – Robot mobile

Robotique médicale

Elle comprend les dispositifs robotiques d'assistance chirurgicale (comme le robot Da Vinci) [9], les prothèses intelligentes et les exosquelettes destinés à la rééducation.

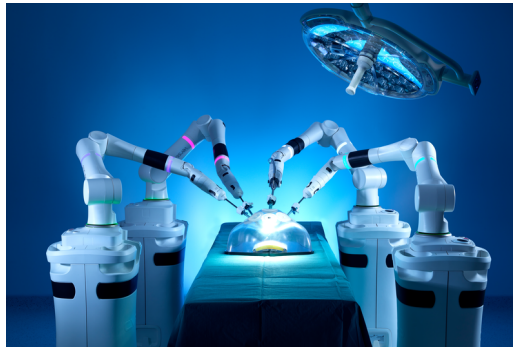


FIGURE 1.3 – Robot médical

Robotique de service

Ce domaine couvre les robots destinés à des applications domestiques ou commerciales (aspirateurs robotisés, robots de livraison, logistique en entrepôts).



FIGURE 1.4 – Robot de service

Robotique humanoïde et recherche

Ce type de robot imite l'apparence ou les mouvements humains. Les robots humanoïdes sont principalement développés à des fins de recherche, d'expérimentation en intelligence artificielle et d'interaction homme-machine.



FIGURE 1.5 – Robot humanoïde

1.3 Composants principaux d'un robot

Un robot, quel que soit son type, est généralement composé des éléments suivants :

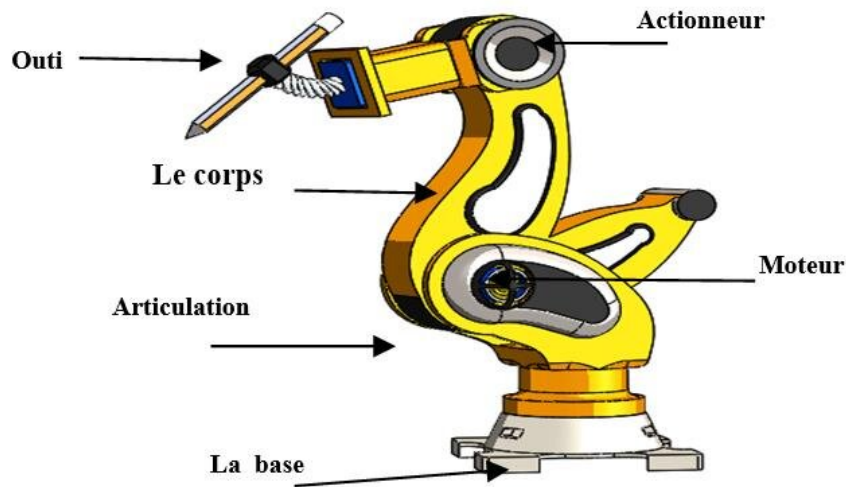


FIGURE 1.6 – Composants principaux d'un robot

Partie mécanique

Elle constitue l'ossature du robot : châssis, bras, articulations. Sa conception dépend de la tâche à accomplir et du degré de liberté requis.

Actionneurs

Ils permettent le mouvement et l'interaction avec l'environnement. Il peut s'agir de moteurs électriques (à courant continu, pas à pas, brushless), de vérins pneumatiques ou hydrauliques.

Capteurs

Ils fournissent au robot des informations sur son état ou son environnement. On trouve par exemple des capteurs de position, de force, de distance, de température, ou des systèmes de vision.

Unité de commande

Elle traite les informations issues des capteurs et envoie des commandes aux actionneurs. Elle peut être constituée d'un microcontrôleur, d'un automate programmable, ou d'un ordinateur embarqué.

Logiciels de contrôle / Intelligence artificielle

Ils permettent la planification, la commande en temps réel, la reconnaissance d'objets, ou l'apprentissage par renforcement dans des contextes plus avancés.

1.4 Paramètres de base d'un bras manipulateur

Degrés de liberté (DDL)

Un degré de liberté correspond à une possibilité de mouvement indépendant (rotation ou translation). Un bras à 6 DDL peut positionner son effecteur final n'importe où dans l'espace tridimensionnel avec n'importe quelle orientation.

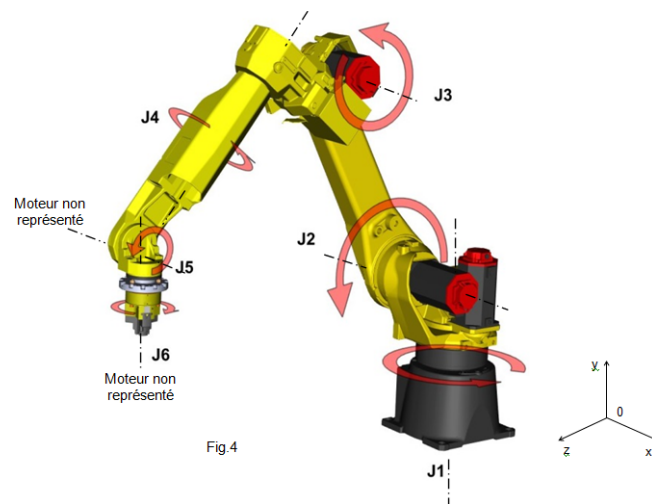


FIGURE 1.7 – Degrés de liberté d'un bras manipulateur

Chaîne cinématique

La structure d'un bras manipulateur peut être en chaîne série (les articulations sont montées les unes à la suite des autres) ou en chaîne parallèle (plusieurs bras interconnectés). La cinématique influence la rigidité, la précision et la complexité du contrôle.

Espace de travail

Il s'agit du volume dans lequel le bras peut déplacer son extrémité. Il dépend de la longueur des segments, des limites des articulations et de l'architecture générale.

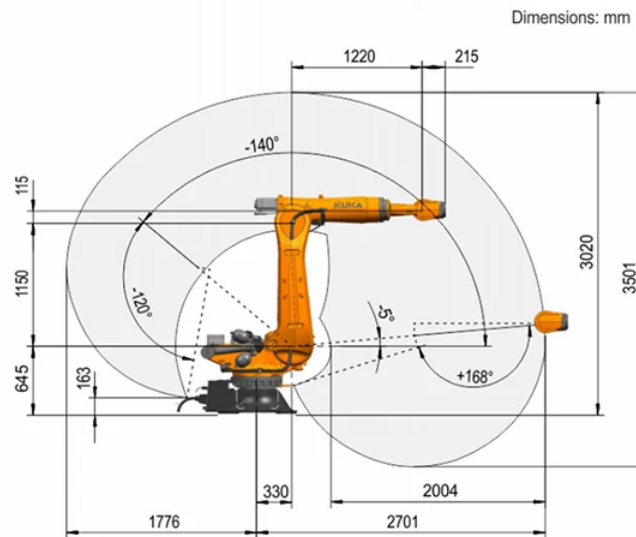


FIGURE 1.8 – Espace de travail d'un bras manipulateur

Charge utile

C'est la masse maximale que peut manipuler le robot sans compromettre sa stabilité ni endommager ses composants. Elle dépend directement des performances des actionneurs et de la structure mécanique.

1.5 Outils utilisés dans le projet

La conception et la réalisation de ce bras manipulateur ont nécessité l'utilisation d'une combinaison d'outils logiciels et matériels, sélectionnés pour leur efficacité et leur compatibilité avec les objectifs du projet.

Outils logiciels

Les outils logiciels suivants ont été utilisés pour la modélisation, la simulation et le contrôle du bras manipulateur :



(a) SolidWorks



(b) ROS

(c) MoveIt



(d) Gazebo

GearTeq



(e) Gearteq

FIGURE 1.9 – Outils logiciels utilisés

- **SolidWorks** : logiciel de conception assistée par ordinateur (CAO) utilisé pour modéliser en 3D les pièces mécaniques du bras et générer automatiquement le fichier URDF du robot via un plugin dédié.
- **Gearteq** : logiciel de CAO utilisé pour la modélisation des poulies et des courroies, permettant de concevoir des systèmes de transmission efficaces et adaptés aux moteurs pas à pas, ce logiciel nous permet de concevoir des poulies avec précision et de les exporter dans des formats compatibles avec SolidWorks, la conception des poulie ainsi que les systèmes de réduction se font par calcul des rapport de réduction, de la vitesse et du couple, ce qui permet d'optimiser les performances du bras manipulateur, et de concevoir les système planétaire avec précision.
- **ROS (Robot Operating System)** : un middleware modulaire largement utilisé en robotique [4], offrant des services de communication inter-processus, de simulation, de planification et de contrôle robotique. ROS a permis de centraliser l'ensemble des composants logiciels du projet.
- **MoveIt** : framework intégré à ROS [10] permettant la planification de mouvement, la détection de collisions et la commande de bras manipulateurs. Il a été utilisé à la fois en simulation (avec Gazebo) et sur le robot réel.
- **Gazebo** : simulateur 3D physique utilisé pour tester virtuellement le comportement du bras manipulateur dans un environnement réaliste avant sa mise en œuvre réelle [11].

Outils matériels

- **ESP32** : microcontrôleur doté de capacités Wi-Fi et Bluetooth, utilisé comme unité de contrôle embarquée. Il a été intégré au système via **rosserial** pour communiquer avec ROS.
- **Drivers DM542** : pilotes de moteurs pas à pas robustes, utilisés pour contrôler avec précision les moteurs du bras tout en assurant un bon rendement en couple.

- **Moteurs pas à pas** : moteurs utilisés pour l'actionnement des articulations, offrant une grande précision de positionnement.
- **Alimentation 24 V** : source d'alimentation centrale utilisée pour fournir l'énergie nécessaire au fonctionnement stable des drivers et moteurs.

Ces outils ont permis de garantir une synergie entre la modélisation, la simulation et la commande du bras manipulateur, tout en assurant la possibilité d'un passage fluide du monde virtuel à l'environnement réel.

1.6 Conclusion

Ce chapitre a permis de poser les bases nécessaires à la compréhension du domaine de la robotique, notamment à travers ses définitions, ses champs d'application, les éléments constitutifs d'un robot, et les paramètres fondamentaux d'un bras manipulateur. Ces généralités permettront d'aborder dans les chapitres suivants les aspects spécifiques à la conception et à la réalisation d'un bras manipulateur à six degrés de liberté.

Chapitre 2

Conception du bras manipulateur à 6 DDL

2.1 Introduction

Ce chapitre présente la phase de conception mécanique du bras manipulateur comportant six degrés de liberté (6 DDL). L'objectif principal est de développer une structure mécanique capable de reproduire des mouvements complexes dans l'espace, tout en étant suffisamment robuste, légère et économiquement réalisable.

Le bras est destiné à des applications de manipulation, nécessitant une mobilité complète dans l'espace, une précision suffisante, et une capacité à porter une charge modérée. Plusieurs contraintes ont orienté les choix techniques, telles que la disponibilité des composants, la simplicité d'usinage, et les performances mécaniques attendues.

La conception du bras a été réalisée en utilisant le logiciel de Conception Assistée par Ordinateur (CAO) **SolidWorks** [12], permettant de modéliser chaque composant, d'assembler les pièces, et de simuler les mouvements.

La forme générale du bras est inspirée des bras manipulateurs industriels, mais adaptée aux spécificités du projet, notamment en termes de taille, de poids et de capacité de charge, un mode de conception a été adopté pour faciliter l'opération d'usinage et d'assemblage.

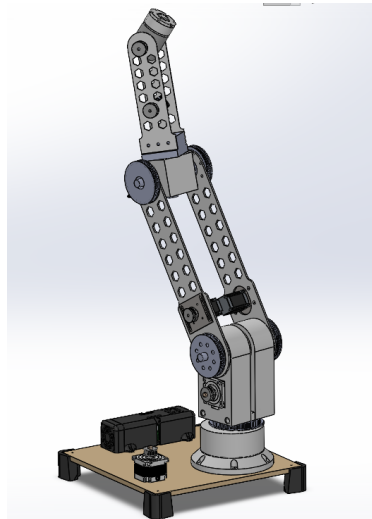


FIGURE 2.1 – Vue d'ensemble du bras manipulateur à 6 DDL

2.2 Choix de l'architecture

2.2.1 Type de structure adoptée

La structure retenue est une architecture en chaîne série composée exclusivement d'articulations rotatives, soit une configuration de type **R-R-R-R-R-R**. Ce choix est motivé par sa simplicité de conception, sa compatibilité avec des actionneurs standards (moteurs rotatifs), et sa capacité à générer un espace de travail tridimensionnel étendu avec six degrés de liberté indépendants.

2.3 Étude mécanique

2.3.1 Choix des matériaux

Les critères principaux ayant guidé le choix des matériaux sont la rigidité, la masse, la disponibilité et le coût.

- **Base** : réalisée en **acier**, pour assurer une excellente stabilité et supporter les efforts transmis par les articulations supérieures. L'acier est choisi pour sa robustesse, sa capacité à amortir les vibrations, et sa bonne résistance aux chocs. La masse volumique de l'acier utilisé est d'environ **7850 kg/m³**.
- **Bras et segments articulés** : réalisés en **aluminium**, afin de réduire le poids total du bras sans compromettre la résistance mécanique. L'aluminium est un matériau léger, usinable et résistant à la corrosion, avec une masse volumique d'environ **2700 kg/m³**.

Ce compromis permet d'obtenir un bras suffisamment rigide pour manipuler une charge utile modérée, tout en limitant le couple nécessaire aux moteurs, et donc la consommation énergétique globale.

2.3.2 Stratégies d'allègement de la structure

Dans le but de réduire la masse globale du bras manipulateur tout en conservant ses performances mécaniques, une attention particulière a été portée à l'**allègement** de la structure. Cette opération consiste à retirer la matière considérée comme non essentielle à la résistance de la pièce, tout en maintenant sa rigidité et sa capacité à supporter les charges.

L'allègement est une démarche fondamentale en ingénierie mécanique, notamment dans les secteurs exigeant un rapport optimal poids/performance, tels que l'aéronautique, l'automobile ou la robotique mobile. Il permet de réduire les efforts nécessaires aux actionneurs, d'améliorer la réactivité du système, et de limiter la consommation énergétique globale.

Optimisation topologique

Une méthode avancée utilisée dans la démarche d'allègement est l'**optimisation topologique** [13, 14]. Il s'agit d'un procédé de conception assistée par ordinateur qui vise à déterminer la répartition idéale de la matière dans un volume donné, en tenant compte des efforts mécaniques appliqués.

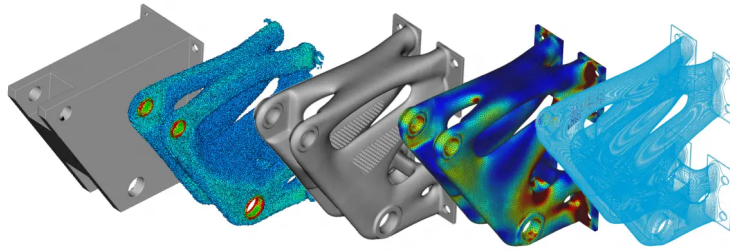


FIGURE 2.2 – Exemple d'allègement d'une pièce mécanique

Le principe est de conserver uniquement la matière utile au cheminement des forces principales, tout en éliminant les zones qui ne participent que peu à la résistance globale. Les résultats de cette méthode conduisent souvent à des formes organiques inspirées de la nature, comme les structures osseuses ou les motifs en toile d'araignée.

Dans certains cas, des motifs spécifiques tels que les **structures alvéolaires hexagonales (nid d'abeille)** [15] peuvent être introduits, offrant un excellent compromis entre légèreté et rigidité.

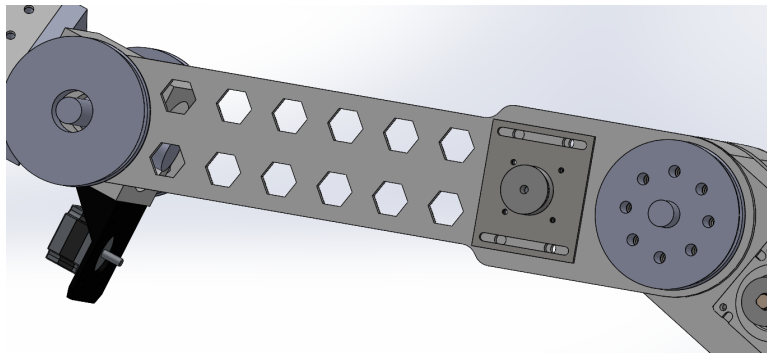


FIGURE 2.3 – Exemple de structure alvéolaire hexagonale

Techniques associées

Différents termes sont associés à cette logique d'allègement [16] :

- **Squelettisation** : consiste à ne conserver que les lignes de force principales dans une structure.
- **Évidement** : suppression de matière en creusant des volumes à l'intérieur des pièces.
- **Ajourage** : introduction d'ouvertures régulières ou irrégulières dans une surface, souvent pour combiner des objectifs esthétiques et fonctionnels.

L'intégration de ces stratégies dès la phase de conception permet de réduire considérablement les masses mises en mouvement, tout en maintenant un niveau de performance mécanique adapté aux contraintes de fonctionnement du bras manipulateur.

Ainsi, la stratégie adoptée est la suivante :

- **Bas du bras (base, 2^e et 3^e articulation)** : moteurs pas à pas à **haut couple**, capables de fournir un couple suffisant pour supporter la masse de l'ensemble des segments supérieurs et la charge utile. Ces moteurs sont volontairement surdimensionnés pour garantir la stabilité du système.
- **Partie supérieure (4^e à 6^e articulation)** : moteurs pas à pas de **couple réduit**, car les charges à mouvoir sont moins importantes. Ce choix permet une réduction du poids total embarqué, ce qui a un impact positif sur les efforts mécaniques et la consommation énergétique.

2.4.3 Avantages du choix des moteurs pas à pas

Le recours aux moteurs pas à pas présente plusieurs avantages dans le contexte de ce projet :

- **Précision angulaire** sans capteurs de retour (boucle ouverte).
- **Contrôle facile** à l'aide de drivers standard (type A4988, DRV8825 ou autres).
- **Maintien en position** sans alimentation constante, grâce au couple résiduel.
- **Réduction des coûts** par rapport aux servomoteurs industriels à couple équivalent.

Un tableau récapitulatif des moteurs sélectionnés selon les articulations pourra être fourni avec les caractéristiques suivantes : couple nominal, intensité, tension d'alimentation, poids, et dimensions.

TABLE 2.1 – Répartition des moteurs pas à pas selon les articulations

Articulation	Modèle	Couple nominal	Poids (approx.)
Base (1 ^{re})	NEMA 34	5.0 Nm	2.0 kg
2 ^e articulation	NEMA 34	12 Nm	5.0 kg
3 ^e articulation	JK57HS82 (NEMA 23)	2.2 Nm	1.2 kg
4 ^e articulation	JK57HS41 (NEMA 23)	0.6 Nm	0.5 kg
5 ^e articulation	NEMA 17	0.45 Nm	0.3 kg
6 ^e articulation	NEMA 17	0.45 Nm	0.3 kg

2.5 Principe de fonctionnement des moteurs pas à pas

Les moteurs pas à pas fonctionnent en divisant une rotation complète en un certain nombre de pas discrets. Chaque pas correspond à un angle fixe, permettant un contrôle précis de la position angulaire du moteur. Le fonctionnement repose sur l'activation séquentielle des enroulements du moteur, créant un champ magnétique qui attire le rotor vers la position suivante. Le moteur est généralement contrôlé par un driver qui envoie des impulsions électriques aux bobines, faisant tourner le rotor d'un pas à chaque impulsion. La vitesse de rotation est proportionnelle à la fréquence des impulsions, tandis que la position est déterminée par le nombre total d'impulsions envoyées.

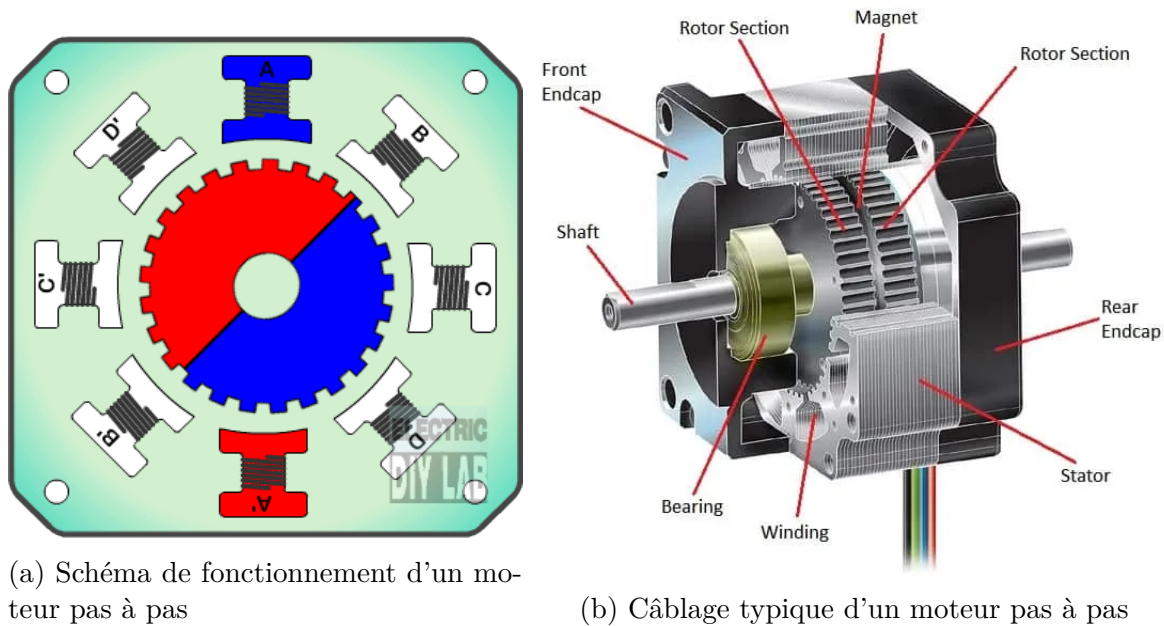


FIGURE 2.5 – Principe de fonctionnement des moteurs pas à pas

2.6 Conception assistée par ordinateur (CAO)

La conception géométrique du bras manipulateur a été réalisée à l'aide d'un logiciel de Conception Assistée par Ordinateur (CAO) **SolidWorks**, permettant la modélisation 3D complète de chaque élément composant le système.

Cette étape permet non seulement de visualiser la géométrie d'ensemble et la cinématique du bras, mais aussi d'évaluer l'encombrement, les collisions éventuelles, la répartition des masses et l'assemblage mécanique des différents composants.

Détail des pièces modélisées

Dans ce projet, chaque pièce a été modélisée individuellement, puis intégrée dans un assemblage global. Les segments du bras, la base, les supports moteurs, les articulations, ainsi que les éléments de fixation ont fait l'objet d'une modélisation précise.

Visualisation par articulation

Afin de mieux comprendre la structure du manipulateur, des vues isométriques ou orthogonales de chaque articulation seront présentées sous forme d'illustrations. Ces vues permettront de montrer les interfaces mécaniques entre les segments, les emplacements des moteurs, ainsi que les degrés de liberté associés à chaque articulation.

Base du bras et Poulie

La base du bras est conçue pour offrir une stabilité maximale. Elle est réalisée en acier, avec des dimensions adaptées pour accueillir les moteurs de la première articulation et supporter les

efforts transmis par les segments supérieurs.

Les poulies ont été conçues pour permettre la transmission du mouvement des moteurs aux articulations, en assurant un bon alignement des axes de rotation. Le nombre de dents ainsi que le type de courroie utilisés sont choisis pour optimiser la transmission du couple tout en minimisant les pertes par frottement. Les poulies ont été réalisées en aluminium pour réduire le poids tout en conservant une bonne résistance mécanique.

La poulie choisie pour la base du bras est de type **HTD5**, permettant une transmission efficace du mouvement des moteurs vers les articulations. La courroie associée est de type **HTD5-5M**, offrant une bonne résistance à l'usure et une faible élasticité.

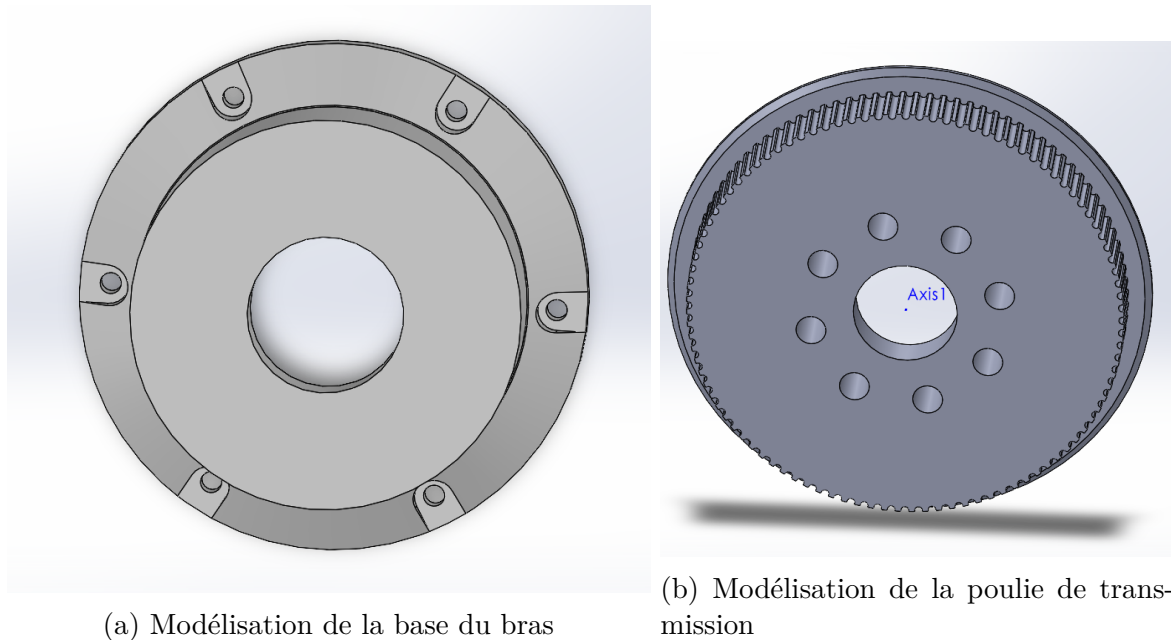


FIGURE 2.6 – Modélisation de la base et de la poulie du bras manipulateur

1ère articulation

La conception de la première articulation est cruciale, car elle constitue le point d'ancrage du bras manipulateur. Elle est équipée d'un moteur pas à pas NEMA 34, capable de fournir un couple élevé pour supporter le poids du bras et des charges éventuelles. Le moteur a été intégré directement dans la 1ère articulation, liée à une poulie de transmission pour augmenter le couple transmis aux segments supérieurs.



FIGURE 2.7 – Modélisation de la 1ère articulation du bras manipulateur

Emplacement des moteurs

L'emplacement des moteurs est un élément clé de la conception, car il influence directement la répartition des masses et les efforts mécaniques dans le bras. Chaque moteur est positionné de manière à minimiser les bras de levier et à optimiser la transmission du couple aux articulations.

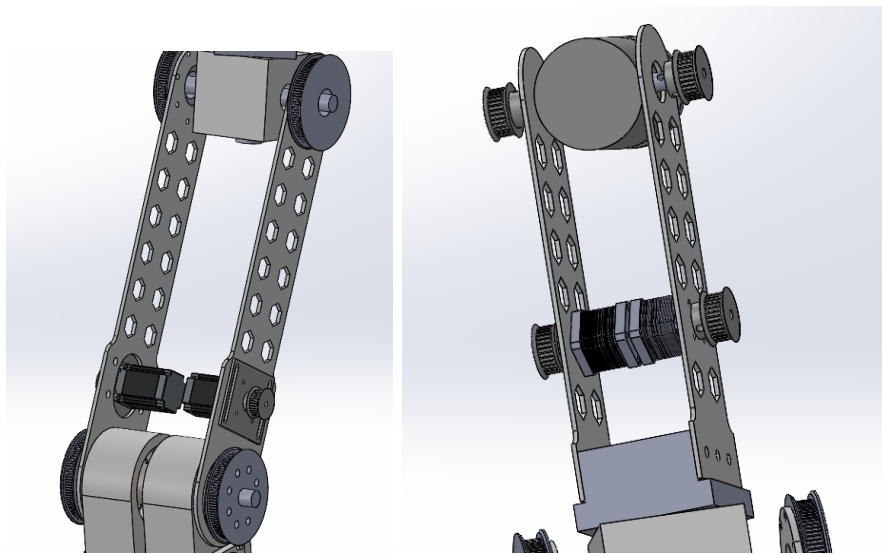


FIGURE 2.8 – Positionnement des moteurs dans les articulations du bras manipulateur

Gearbox et réducteurs

Dans l'articulation finale (6^e articulation), un réducteur de type **planétaire** est intégré pour augmenter le couple transmis à l'effecteur final [18]. Ce réducteur permet de réduire la vitesse de rotation tout en augmentant le couple, ce qui permet de manipuler des charges plus lourdes avec une précision accrue.

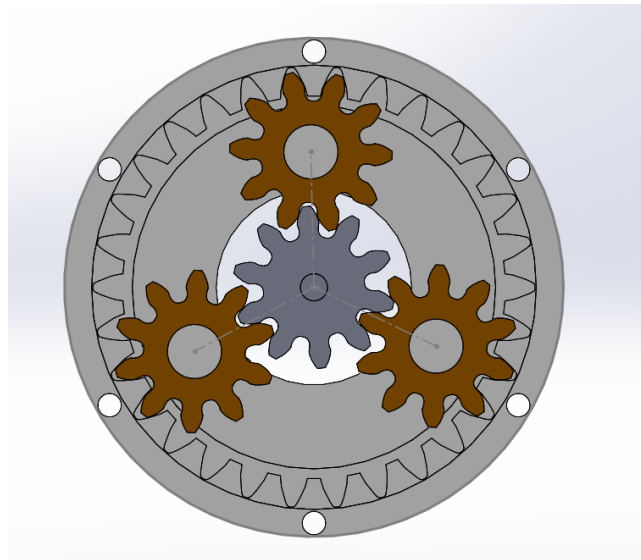


FIGURE 2.9 – Modélisation du réducteur planétaire dans la 6^e articulation

Le réducteur planétaire permet de réduire la vitesse de rotation du moteur avec un rapport de réduction de 1 :4, ce qui permet d'augmenter le couple transmis à l'effecteur final.

La conception de ce réducteur a été réalisée en tenant compte des contraintes d'encombrement et de poids, tout en garantissant une transmission efficace du couple.

2.7 Conclusion

Ce chapitre a permis de présenter l'ensemble des choix techniques ayant conduit à la conception du bras manipulateur. L'architecture mécanique adoptée, le choix des matériaux, des moteurs et le travail de modélisation 3D constituent la base sur laquelle s'appuiera la suite du projet, notamment les étapes de modélisation et de réalisation.

Dans le chapitre suivant, nous aborderons les aspects de modélisation cinématique et dynamique du bras, ainsi que les calculs nécessaires pour dimensionner les moteurs en fonction des charges à supporter et des couples à fournir. Ces éléments sont cruciaux pour garantir le bon fonctionnement du bras dans les applications envisagées.

Chapitre 3

Modélisation du bras manipulateur

3.1 Introduction

La modélisation d'un bras manipulateur est une étape essentielle dans tout projet de robotique. Elle permet de représenter mathématiquement la structure mécanique du robot afin de pouvoir en analyser et en contrôler le comportement.

Ce chapitre a pour objectif de modéliser le bras manipulateur à 6 degrés de liberté (DDL) développé dans ce projet. La modélisation sera basée sur la convention de Denavit-Hartenberg (DH) classique, et portera sur la cinématique directe et inverse du bras.

3.2 Définition des repères et conventions

Avant de procéder à l'établissement des équations, il est nécessaire de définir une convention pour l'orientation des axes et des repères.

- L'axe global Z_0 est dirigé vers le haut.
- Les axes X et Y forment le plan horizontal.
- La numérotation des articulations va de la base (1^{ère} articulation) jusqu'à l'effecteur final (6^e articulation).

La convention classique de Denavit-Hartenberg sera utilisée pour associer un repère à chaque articulation.

3.3 Paramétrage selon Denavit-Hartenberg (DH)

3.3.1 Présentation de la méthode DH

La méthode de Denavit-Hartenberg (DH) est une approche standard pour modéliser les robots articulés [19, 20]. Elle permet de décrire la position et l'orientation des segments du robot en utilisant des transformations homogènes. Cette méthode repose sur l'utilisation de repères associés à chaque articulation, facilitant ainsi la représentation des mouvements et la résolution des équations de cinématique.

3.3.2 Pourquoi utiliser la méthode DH ?

La méthode DH présente plusieurs avantages :

- Standardisation : elle fournit une convention claire et universelle pour la modélisation des robots.
- Simplicité : les transformations homogènes sont faciles à manipuler et à combiner.
- Flexibilité : elle s'adapte à différents types de robots, qu'ils soient à 2, 3, 4 ou plus de 6 degrés de liberté.
- Précision : elle permet de décrire avec précision la position et l'orientation de l'effecteur final en fonction des angles des articulations.

3.3.3 Paramètres DH pour le bras manipulateur

La méthode de Denavit-Hartenberg permet de définir les transformations entre chaque paire de repères successifs à l'aide de 4 paramètres :

- a_i : longueur du lien (distance entre les axes Z_i et Z_{i+1} selon X_i)
- α_i : angle entre Z_i et Z_{i+1} autour de X_i
- d_i : distance entre X_i et X_{i+1} selon Z_i
- θ_i : angle entre X_i et X_{i+1} autour de Z_i

Ces paramètres sont utilisés pour construire les matrices de transformation homogène entre les repères associés à chaque articulation.

Les matrices de transformation homogène T_i entre les repères sont données par :

$$T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.3.4 Paramètres DH du bras manipulateur à 6 DDL

Les paramètres DH pour le bras manipulateur à 6 DDL sont définis comme suit :

TABLE 3.1 – Tableau des paramètres DH

i	a_i (mm)	α_i (rad)	d_i (mm)	θ_i (variable)
1	0	$\frac{\pi}{2}$	d_1	θ_1
2	a_2	0	0	θ_2
3	0	$\frac{\pi}{2}$	0	$\theta_3 - \frac{\pi}{2}$
4	0	$-\frac{\pi}{2}$	d_4	θ_4
5	a_5	0	0	$\theta_5 + \frac{\pi}{2}$
6	0	$\frac{\pi}{2}$	d_6	θ_6

3.4 Modélisation Cinématique Directe

3.4.1 Matrices de transformation homogène

- *La Matrice T_1^0 :*

$$T_1^0 = \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *La Matrice T_2^1 :*

$$T_2^1 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *La Matrice T_3^2 :*

$$T_3^2 = \begin{bmatrix} \sin \theta_3 & 0 & -\cos \theta_3 & 0 \\ -\cos \theta_3 & 0 & -\sin \theta_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *La Matrice T_4^3 :*

$$T_4^3 = \begin{bmatrix} \cos \theta_4 & 0 & -\sin \theta_4 & 0 \\ \sin \theta_4 & 0 & \cos \theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *La Matrice T_5^4 :*

$$T_5^4 = \begin{bmatrix} -\sin \theta_5 & -\cos \theta_5 & 0 & -a_5 \sin \theta_5 \\ \cos \theta_5 & -\sin \theta_5 & 0 & a_5 \cos \theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *La Matrice T_6^5 :*

$$T_6^5 = \begin{bmatrix} \cos \theta_6 & 0 & -\sin \theta_6 & 0 \\ \sin \theta_6 & 0 & \cos \theta_6 & 0 \\ 0 & -1 & 0 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *La Matrice de Transformation Globale :*

$$T_5^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 \cdot T_6^5 = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$$

3.5 Méthode de Paul pour un robot à 6 DDL

3.5.1 Principe général

Pour un robot manipulateur à 6 degrés de liberté, la transformation homogène totale entre la base (repère 0) et l'organe terminal (repère 6) s'exprime par le produit des matrices de transformation élémentaires associées à chaque articulation :

$$T_6^0 = T_1 T_2 T_3 T_4 T_5 T_6$$

La méthode de Paul [19] repose sur l'inversion progressive de cette chaîne de transformations afin d'isoler successivement les angles articulaires θ_1 à θ_6 .

3.5.2 Détermination de θ_1

On commence par inverser T_1 :

$$T_1^{-1} T_6^0 = T_2 T_3 T_4 T_5 T_6$$

À partir de la position du point final, on peut estimer :

$$\theta_1 = \text{atan2}(p_y, p_x)$$

où p_x et p_y sont les coordonnées de position de l'organe terminal dans le repère de base.

3.5.3 Détermination de θ_2 et θ_3

On continue l'inversion :

$$T_2^{-1} T_1^{-1} T_6^0 = T_3 T_4 T_5 T_6$$

On extrait ensuite la position du poignet (appelée ****centre du poignet****, ou wrist center), en soustrayant l'effet des articulations 4 à 6 :

$$\vec{P}_{\text{poignet}} = \vec{P}_6 - d_6 \cdot \hat{z}_6$$

Les angles θ_2 et θ_3 peuvent alors être déterminés à l'aide de relations trigonométriques et de la loi des cosinus, comme pour le cas à 5 ddl :

$$r = \sqrt{x'^2 + y'^2}, \quad s = z' - d_1$$

$$\cos \theta_3 = \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2 a_3}$$

$$\theta_3 = \text{atan2} \left(\sqrt{1 - \cos^2 \theta_3}, \cos \theta_3 \right)$$

$$\theta_2 = \text{atan2}(s, r) - \text{atan2}(a_3 \sin \theta_3, a_2 + a_3 \cos \theta_3)$$

3.5.4 Détermination de θ_4 , θ_5 et θ_6

Une fois les trois premières articulations déterminées, on calcule la transformation du bras (base au poignet) :

$$T_3^0 = T_1 T_2 T_3$$

Puis, on isole les rotations du poignet :

$$T_w = (T_3^0)^{-1} T_6^0 = T_4 T_5 T_6$$

En extrayant la matrice de rotation R_w contenue dans T_w , les angles θ_4 , θ_5 et θ_6 sont obtenus via des relations trigonométriques :

$$\theta_5 = \text{atan2} \left(\sqrt{R_{13}^2 + R_{23}^2}, R_{33} \right)$$

$$\theta_4 = \text{atan2}(R_{23}, R_{13})$$

$$\theta_6 = \text{atan2}(R_{32}, -R_{31})$$

3.6 Calcul des couples et de l'inertie

3.6.1 Hypothèse de fonctionnement

Dans le but de déterminer les efforts mécaniques à fournir par les moteurs de l'ensemble articulé, l'étude se base sur le cas limite où le bras robotisé est en position horizontale. Ce scénario représente la condition la plus défavorable en termes de couple gravitationnel.

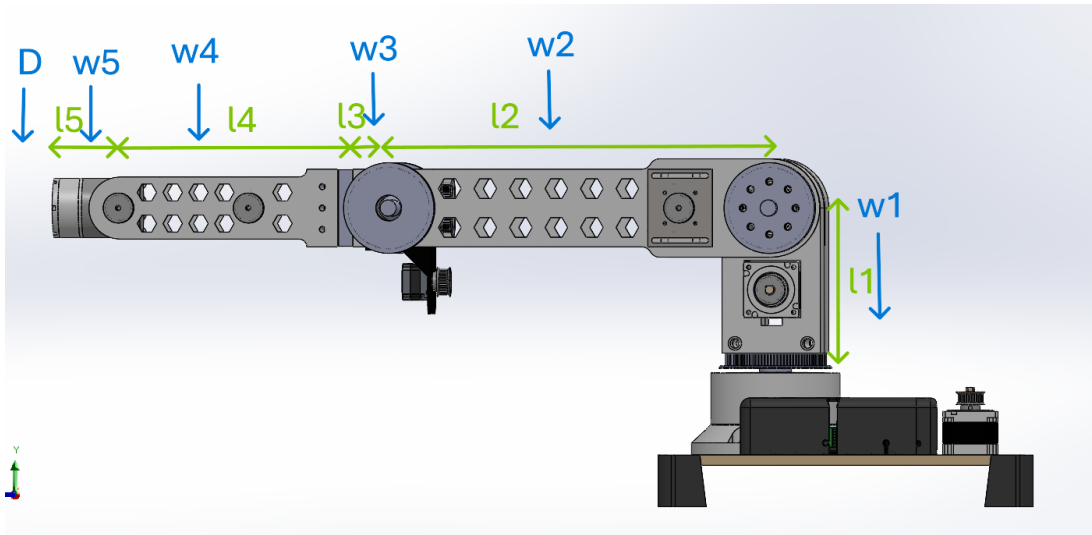


FIGURE 3.1 – Bras robotisé en position horizontale

Les paramètres pris en compte sont :

- W_i : Poids du segment i ;
- A_i : Poids du moteur fixé au segment i ;
- L_i : Longueur du segment i ;
- D : Poids de la charge à l'extrémité du bras.

3.6.2 Calcul du couple gravitationnel

Le couple gravitationnel τ_{gi} généré sur chaque segment est donné par :

$$\tau_{gi} = F_i \times L_i \quad (3.1)$$

où F_i est la somme des forces gravitationnelles appliquées sur le segment, soit :

$$F_i = W_i + A_i + D$$

et L_i est la distance entre l'axe de rotation et le centre de gravité des masses considérées.

Les couples gravitationnels pour les articulations sont les suivants :

$$\begin{aligned} \tau_{1g} &= 0 \\ \tau_{2g} &= \frac{L_2}{2}(W_2 + A_3) + (L_2 + \frac{L_3}{2})(W_3 + A_4) + (L_2 + L_3 + \frac{L_4}{2})(W_4 + A_5) \\ &\quad + (L_2 + L_3 + L_4 + \frac{L_5}{2})W_5 + (L_2 + L_3 + L_4 + L_5)D \\ \tau_{3g} &= \frac{L_3}{2}(W_3 + A_4) + (L_3 + \frac{L_4}{2})(W_4 + A_5) + (L_3 + L_4 + \frac{L_5}{2})W_5 + (L_3 + L_4 + L_5)D \\ \tau_{4g} &= 0 \\ \tau_{5g} &= \frac{L_5}{2}W_5 + L_5D \end{aligned}$$

3.6.3 Caractéristiques des moteurs et segments

Articulation	Moteur	Masse moteur (Kg)	Masse segment (Kg)
A_1	Nema 34	2.0	12.0
A_2	Nema 34	5.0	2.22
A_3	Nema 23 (JK57HS82)	2.2	4.95
A_4	Nema 23 (JK57HS41)	1.2	0.64
A_5	Nema 17	0.6	0.8

TABLE 3.2 – Moteurs utilisés et masses associées

3.6.4 Valeurs des couples gravitationnels

Segment	L_i (mm)	W_i (N)	A_i (N)	τ_{gi} (N.m)
1	224.68	120.0	20	0
2	590.28	22.2	50	$78.5307 + 12.184 \times D$
3	184.84	49.5	12	$15.134 + 6.32 \times D$
4	313.86	6.4	22	0
5	135.86	8.0	6	$0.54 + 1.35 \times D$

TABLE 3.3 – Calcul des couples gravitationnels

3.6.5 Calcul du couple d'inertie

Le couple d'inertie est déterminé à l'aide de la relation suivante :

$$\tau_{Ii} = I_i \times \alpha_i \quad (3.2)$$

où :

- I_i : Moment d'inertie du segment i par rapport à l'axe de rotation ;
- α_i : Accélération angulaire maximale supposée (ici $\alpha_i = 0.542 \text{ rad/s}^2$).

Le moment d'inertie est déterminé à l'aide du théorème de Huygens :

$$I_i = I_i^G + m_i \times d_i^2 \quad (3.3)$$

avec :

- I_i^G : Inertie du segment au centre de gravité (déterminé via SolidWorks) ;
- m_i : Masse du segment ;
- d_i : Distance entre le centre de gravité et l'axe de rotation.

3.6.6 Moments d'inertie des segments

Segment	I_i (N.m)
1	$0.1165 + 1.0652 \times 10^{-2} \times D$
2	$0.0485 + 5.25572 \times 10^{-4} \times D$
3	$2.38189 \times 10^{-3} + 6.27528 \times 10^{-5} \times D$
4	$7.1668 \times 10^{-4} + 6.4736 \times 10^{-4} \times D$
5	$2.4349 \times 10^{-4} + 7.3839 \times 10^{-4} \times D$

TABLE 3.4 – Moments d'inertie des segments

3.6.7 Couple total par segment

Le couple total appliqué à chaque articulation est la somme du couple gravitationnel et du couple d'inertie :

$$\tau_t = \tau_{gi} + \tau_{Ii} \quad (3.4)$$

Segment	τ_t (N.m)
1	$0.1165 + 1.0652 \times 10^{-2} \times D$
2	$78.5307 + 12.184 \times D$
3	$15.134 + 6.32 \times D$
4	$7.1668 \times 10^{-4} + 6.4736 \times 10^{-4} \times D$
5	$0.54 + 1.35 \times D$

TABLE 3.5 – Couples totaux par segment

3.6.8 Transmission du couple via poulie-courroie

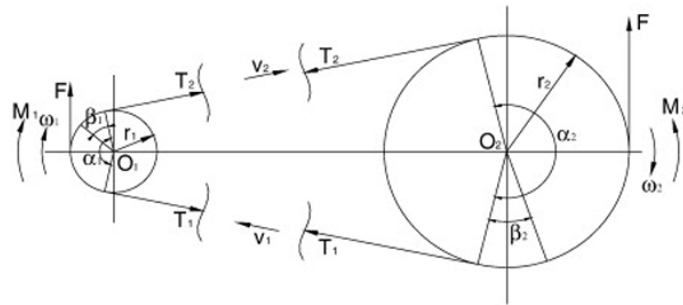


FIGURE 3.2 – Système poulie-courroie utilisé pour la transmission

La transmission du couple entre moteur et segment est assurée par un système poulie-courroie. Ce dernier permet d'augmenter le couple transmis au bras selon la relation :

$$M_2 = M_1 \times \frac{r_2}{r_1} \quad (3.5)$$

avec :

- M_2 : Couple transmis au bras ;
- M_1 : Couple fourni par le moteur ;
- r_2 et r_1 : Diamètres respectifs des poulies du bras et du moteur.

3.6.9 Couples générés et transmis

Segment	Moteur	Couple généré (N.m)	Couple transmis (N.m)
1	Nema 34	5.0	27.5
2	Nema 34	12.5×2	100.0
3	Nema 23 (JK57HS82)	5.0×2	40.0
4	Nema 23 (JK57HS41)	0.6	2.16
5	Nema 17	0.45×2	3.0

TABLE 3.6 – Couples générés et transmis par les moteurs

3.6.10 Capacité maximale de charge

Le poids maximal de la charge que le bras peut supporter est déterminé par la limite de couple admissible par les moteurs et transmissions.

Segment	Poids maximal supporté (kg)
2	1.83
3	4.1
5	1.8

TABLE 3.7 – Poids maximal de la charge supporté par segment

En conclusion, le bras peut supporter une charge maximale d'environ **1.8 Kg** en bout de bras sans compromettre son fonctionnement.

3.7 Conclusion

La modélisation cinématique constitue la base sur laquelle repose le contrôle du bras manipulateur. Elle permet d'établir la relation entre les angles moteurs et la position de l'effecteur, et facilitera l'élaboration de la commande dans les étapes ultérieures du projet.

Ce chapitre a présenté les concepts fondamentaux de la modélisation cinématique, en utilisant la convention de Denavit-Hartenberg pour définir les paramètres du bras manipulateur. La modélisation cinématique directe a été abordée à travers les matrices de transformation homogène, ainsi que les bases de la cinématique inverse.

Dans le chapitre suivant, nous aborderons la commande du bras manipulateur, en utilisant les modèles cinématiques établis pour développer des algorithmes de contrôle adaptés.

Chapitre 4

Commande du bras manipulateur

4.1 Introduction

La commande d'un bras manipulateur consiste à déterminer les signaux à envoyer aux actionneurs pour atteindre une position ou effectuer une trajectoire donnée dans l'espace. Dans ce projet, deux approches complémentaires sont proposées [19, 4] :

- Une méthode classique basée sur la modélisation cinématique directe et inverse, utilisant les paramètres de Denavit-Hartenberg.
- Une méthode avancée basée sur l'utilisation de l'environnement ROS (Robot Operating System) et du framework MoveIt, intégrant la simulation sous Gazebo et le déploiement réel via `rosserial` sur ESP32.

4.2 Méthode classique basée sur la modélisation

4.2.1 Principe de fonctionnement

Cette méthode repose sur la résolution des équations de la cinématique inverse [20, 19] pour obtenir les angles des articulations à partir d'une position cible de l'effecteur. Grâce aux paramètres DH définis au chapitre précédent, il est possible de calculer la transformation globale et d'en déduire les positions articulaires nécessaires.

4.2.2 Implémentation sur ESP32

Le microcontrôleur ESP32 a été utilisé pour piloter les moteurs pas à pas.

Il reçoit les consignes angulaires calculées par la méthode cinématique inverse et les convertit en signaux de commande pour les drivers des moteurs. L'implémentation se fait en C++ avec l'utilisation de la bibliothèque `AccelStepper` [21] pour gérer les moteurs pas à pas.

Le code embarqué implémente :

- La réception des consignes angulaires.
- La commande des drivers des moteurs pas à pas.
- Une gestion simplifiée du timing d'exécution pour garantir la synchronisation des articulations.

4.3 Commande avec ROS et MoveIt

4.3.1 Présentation de ROS

Qu'est-ce que ROS ?

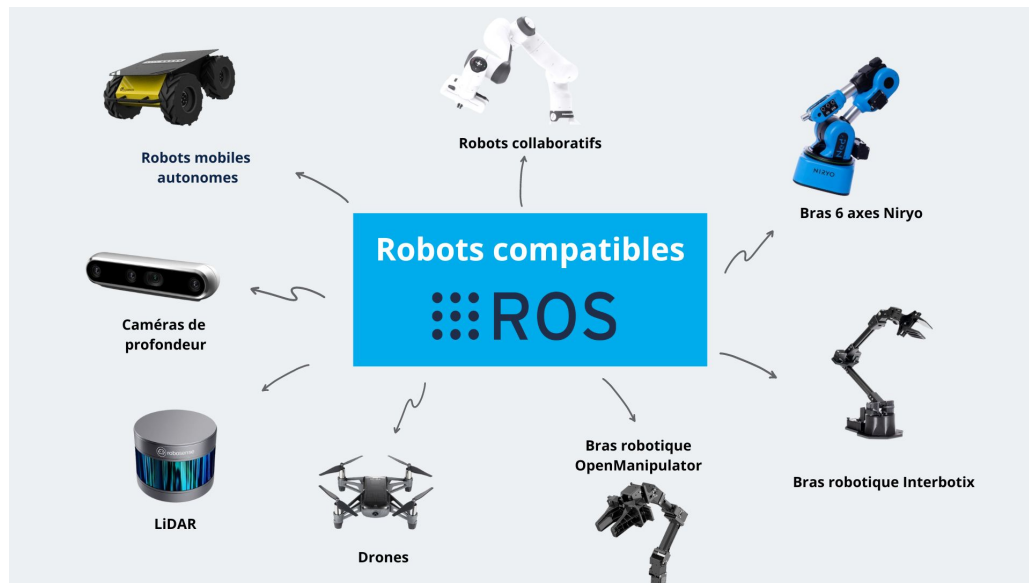


FIGURE 4.1 – Application ROS

Le *Robot Operating System* (ROS) est un middleware open-source [22] qui constitue aujourd'hui une plateforme incontournable dans le domaine de la robotique. ROS fournit une architecture logicielle modulaire permettant de développer, tester et faire interagir des composants logiciels robotiques de manière flexible et efficace.

Architecture générale

ROS est basé sur une architecture distribuée. Chaque fonctionnalité (capteur, actionneur, algorithme, interface utilisateur, etc.) est encapsulée dans un nœud (*node*), qui peut être développé en C++ ou Python. Ces nœuds communiquent entre eux via des canaux appelés *topics* (publications/subscriptions), des services synchrones ou des actions asynchrones.

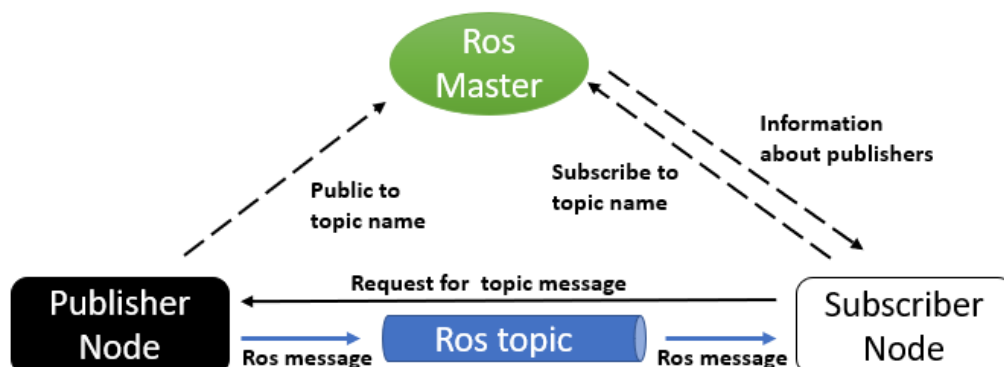


FIGURE 4.2 – Architecture de ROS

Les composants essentiels d'un système ROS sont :

- **Nodes** : programmes individuels qui effectuent des tâches spécifiques.
- **Topics** : canaux de communication pour échanger des messages de manière asynchrone (publisher/subscriber).
- **Services** : appels synchrones de type requête/réponse.
- **Actions** : communication asynchrone orientée vers l'exécution de tâches longues.
- **Messages** : structures de données échangées entre les nœuds.
- **Master** : coordinateur (en ROS 1) qui gère l'annuaire des nœuds.

ROS dans la commande et la navigation

ROS propose plusieurs outils puissants pour la commande de robots :

- **MoveIt** : framework de planification de mouvement pour les bras robotiques. Il prend en charge la cinématique inverse, la planification de trajectoires, la détection de collision, et l'interface avec les moteurs.
- **move_base** : utilisé pour la navigation autonome des robots mobiles. Il combine la localisation, la planification de trajectoire, et l'évitement d'obstacles.
- **tf** et **tf2** : systèmes de gestion des repères de transformation spatiale entre les différentes parties du robot.

Dans notre projet on a opter pour l'utilisation de MoveIt sous ROS qui s'aligne parfaitement avec nos attentes et constitue un outil puissant et d'actualité utilisé en robotique pour commander les bras manipulateurs.

Simulation avec Gazebo

ROS s'intègre de manière native avec **Gazebo**, un simulateur physique 3D permettant de tester virtuellement les comportements du robot dans un environnement réaliste. Les principales caractéristiques de Gazebo sont :

- Simulation de la physique (gravité, collisions, frictions, masse).
- Capteurs virtuels (caméras, LIDAR, IMU, fin de course).
- Compatibilité avec les modèles URDF/Xacro.
- Intégration avec ROS via des plugins (ex : **gazebo_ros**).

Cette plateforme permet de valider les algorithmes de commande, la géométrie du robot et les interactions avec l'environnement avant tout déploiement réel. Dans notre projet on a commencer par intégrer notre robot dans gazebo, de tester la capacité des moteurs à générer les couples nécessaire aux mouvements du robot ainsi que de valider nos choix des matériaux et structure générale du robot avant de passer à sa phase de réalisation.

Avantages de ROS en robotique

ROS offre de nombreux avantages :

- Un écosystème riche de bibliothèques et d'outils prêts à l'emploi.
- Une forte communauté active.
- La ré-utilisabilité des modules (détection, navigation, manipulation...).
- La possibilité de simuler, tester, visualiser et contrôler des robots de manière cohérente.

Ainsi, ROS s'impose comme un outil complet et puissant, facilitant le développement de projets robotiques complexes et évolutifs.

MoveIt : Planification de mouvement pour bras manipulateurs

MoveIt est un framework ROS dédié à la planification [23] et au contrôle des bras robotiques. Il constitue une solution complète pour la manipulation, combinant plusieurs modules essentiels permettant d'assurer le fonctionnement d'un bras manipulateur de manière fluide, sûre et efficace.

Les fonctionnalités clés de MoveIt incluent :

- **Cinématique inverse et directe** : MoveIt utilise des solveurs (comme KDL ou IK-Fast) pour calculer les configurations articulaires permettant d'atteindre une position et orientation cible.
- **Planification de trajectoire** : grâce à des algorithmes comme RRT, OMPL, CHOMP, il génère des chemins optimisés dans l'espace des configurations tout en évitant les obstacles.
- **Détection de collisions** : MoveIt intègre le moteur FCL (Flexible Collision Library) pour vérifier en temps réel les collisions possibles entre les parties du robot ou avec l'environnement.
- **Contrôle en temps réel** : MoveIt peut envoyer des commandes vers le matériel physique ou vers un simulateur (comme Gazebo) via des interfaces de contrôle (MoveGroup, contrôleurs ROS).
- **Intégration avec RViz** : l'utilisateur peut visualiser et interagir avec le robot en temps réel (déplacement d'un end-effector, suivi de trajectoires, tests de collision).

MoveIt repose sur la description URDF ou Xacro du robot. Pour faciliter la configuration initiale, l'outil **MoveIt Setup Assistant** permet de générer les fichiers de configuration nécessaires (groupe d'articulations, contraintes, zones de travail, etc.).

Enfin, MoveIt est hautement extensible et supporte aussi :

- La manipulation à deux bras.
- L'interfaçage avec des capteurs (caméras, force/torque).
- La commande assistée par apprentissage (Machine Learning).

Grâce à sa flexibilité et sa puissance, MoveIt est devenu l'outil de référence pour les projets de manipulation robotique, tant en simulation que sur des robots physiques.

4.3.2 Génération du modèle URDF

La modélisation mécanique du bras a été effectuée sous SolidWorks. Afin d'intégrer ce modèle dans l'environnement ROS, le plugin **SW2URDF** (SolidWorks to URDF Exporter) [24] a été utilisé. Ce plugin permet de générer automatiquement les fichiers URDF, incluant la géométrie, les jointures, et les liens, à partir de l'assemblage 3D conçu.

Mais pour ce faire on commence par modifier notre model SolidWorks, d'abord pour avoir un meilleur résultat avec un temps de calcul réduit, nous devant avoir un modèle minimaliste, c'est à dire que le modèle qu'on vas utiliser pour générer le fichier URDF n'est pas le modèle complet du robot mais plutôt une version qui ne contient que l'essentiel à savoir on supprime tout roulement, poulie et courroie, vis et autre fixations, les moteurs, ...etc. Le but est de ne garder que la structure générale qui définit les liaisons et contraintes du robot. En suite, on

devrait fixer les axes et pour chaque degrés de liberté, effectivement le plugin SW2URDF le fait automatiquement mais des fois des lèges ajustements s'impose.

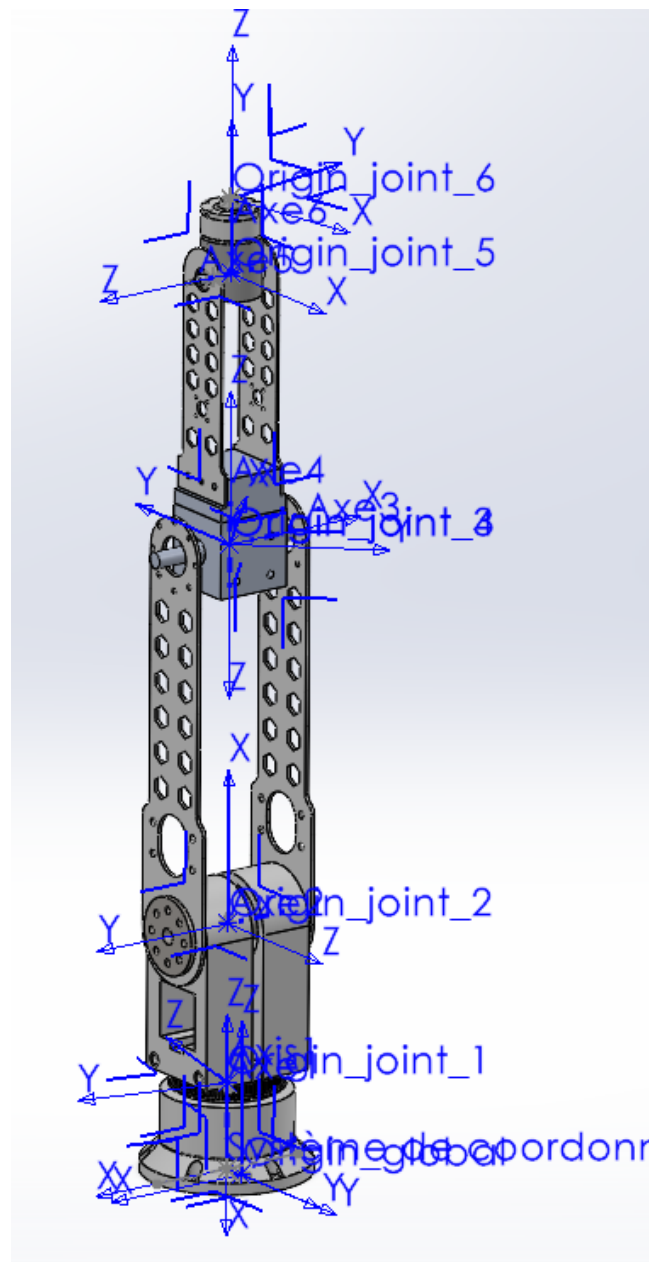


FIGURE 4.3 – Model utilisé pour la génération de l'URDF

4.3.3 Simulation dans Gazebo

Gazebo est un simulateur 3D [25] qui permet de tester virtuellement les mouvements du robot. Le modèle URDF du bras a été chargé dans Gazebo via ROS, et les commandes de mouvement ont été envoyées à l'aide de MoveIt. Les tests ont permis de vérifier :

- La validité de la configuration cinématique.
- La capacité du planificateur MoveIt à atteindre différentes cibles dans l'espace de travail.
- Le respect des contraintes physiques simulées (poids, collisions).

Avec Gazebo on a pu simuler notre robot dans un environnement virtuel qu'on a construit avec Gazebo, et donc de tester des applications de pick and place et de tester l'efficacité de la génération de trajectoire ainsi que de la cinématique inverse.

4.3.4 Commande réelle avec MoveIt et roserial

Après validation en simulation, l'implémentation réelle a été effectuée en remplaçant le simulateur par l'interface matérielle. MoveIt continue d'envoyer les trajectoires, mais elles sont désormais transmises via `roserial` [26] à l'ESP32, qui assure le pilotage temps réel des moteurs.

Cette architecture présente plusieurs avantages :

- Réutilisation de la même configuration que la simulation.
- Accès à toute la puissance de MoveIt pour la planification de trajectoires complexes.
- Communication bidirectionnelle pour envisager un retour capteur malgré que pour des moteur stepper la commande se fait en boucle ouverte.

4.4 Comparaison entre les deux approches

La méthode classique se distingue par une complexité d'implémentation relativement faible, ce qui la rend accessible pour des applications simples. Cependant, cette simplicité se fait souvent au détriment de la précision, qui dépend fortement de la qualité du modèle et des calculs effectués manuellement. Elle offre également une souplesse de modification limitée, rendant l'adaptation à de nouveaux besoins plus contraignante. De plus, elle ne permet généralement pas de simulation avant déploiement, ce qui complique le processus de validation. Enfin, son évolutivité reste restreinte, notamment lorsqu'il s'agit d'intégrer des capteurs ou des modules d'intelligence artificielle.

En revanche, l'approche basée sur ROS et MoveIt présente une complexité d'implémentation moyenne à élevée, en raison de l'apprentissage nécessaire des outils et de la configuration initiale. Cette complexité est compensée par une précision optimisée grâce aux fonctionnalités avancées de MoveIt. La souplesse de modification est excellente, en grande partie grâce à l'utilisation de fichiers URDF et aux nombreux paramètres configurables. Cette approche permet aussi une simulation réaliste dans Gazebo avant tout déploiement réel, offrant un environnement de test sécurisé. Enfin, elle est hautement évolutive, facilitant l'ajout de capteurs, de modules de perception, ou même d'algorithmes d'intelligence artificielle.

Tableau comparatif des deux approches

En résumé, voici un tableau comparatif des deux approches :

Critère	Méthode classique	ROS + MoveIt
Complexité d'implémentation	Faible	Moyenne à élevée
Précision	Dépend du modèle et des calculs	Très bonne (optimisée)
Souplesse de modification	Faible	Très bonne (URDF / paramétrage MoveIt)
Simulation avant déploiement	Non	Oui via Gazebo
Évolutivité (ajout capteurs, IA, etc.)	Limitée	Très bonne

TABLE 4.1 – Comparaison des deux méthodes de commande

4.5 Étapes de mise en œuvre

La réalisation du bras manipulateur à 6 degrés de liberté a suivi plusieurs étapes clés, intégrant à la fois la modélisation mécanique, la génération du modèle URDF, la configuration de MoveIt, et l'intégration matérielle. Voici un aperçu des principales étapes :

4.5.1 Conception mécanique

La première phase du projet a consisté en la modélisation mécanique du bras manipulateur. Pour cela, nous avons utilisé **SolidWorks**, un logiciel de conception assistée par ordinateur (CAO), qui nous a permis de créer un modèle 3D fidèle et articulé. Chaque segment du bras, chaque joint ainsi que les pièces de support ont été conçus en respectant les contraintes dimensionnelles, les plages de mouvements mécaniques, et les spécificités d'assemblage. Une attention particulière a été portée à l'ergonomie et à la possibilité d'imprimer ou usiner les pièces.

4.5.2 Simulation et choix des moteurs et matériaux

La simulation dans **SolidWorks** a permis de valider les choix de conception et de matériaux. Les moteurs pas à pas ont été sélectionnés selon leurs couples et leur précision, adaptés aux mouvements du bras.

Les matériaux utilisés pour les pièces imprimées ou usinées ont été sélectionnés en fonction de leur résistance mécanique, leur poids et leur coût. Des plastiques techniques comme l'ABS ou le PLA ont été privilégiés pour les pièces imprimées, tandis que de l'aluminium et acier ont été envisagés pour les pièces usinées.

4.5.3 Usinage des pièces

Les pièces du bras ont été ensuite usinées en utilisant différentes techniques d'usinage, chaque pièce a été découpée et réalisée selon les spécifications du modèle 3D, notamment les matériaux choisis et les dimensions exactes.

Ensuite, les pièces ont été assemblées pour former le bras manipulateur. Les roulements, poulies

et autres éléments mécaniques ont été intégrés pour assurer la fluidité des mouvements. Chaque articulation a été testée pour vérifier la liberté de mouvement et l'absence de frottements.

4.5.4 Génération du modèle URDF

Une fois la modélisation terminée, nous avons exporté le modèle 3D sous forme de fichiers URDF (*Unified Robot Description Format*), indispensables pour l'intégration dans l'écosystème ROS. Cette opération a été facilitée par l'utilisation du plugin *sw_urdf_exporter* pour SolidWorks, permettant de générer automatiquement les fichiers URDF, les maillages STL, et les textures associées. Ces fichiers décrivent la géométrie, la cinématique et les propriétés physiques du robot.

4.5.5 Configuration MoveIt et ROS

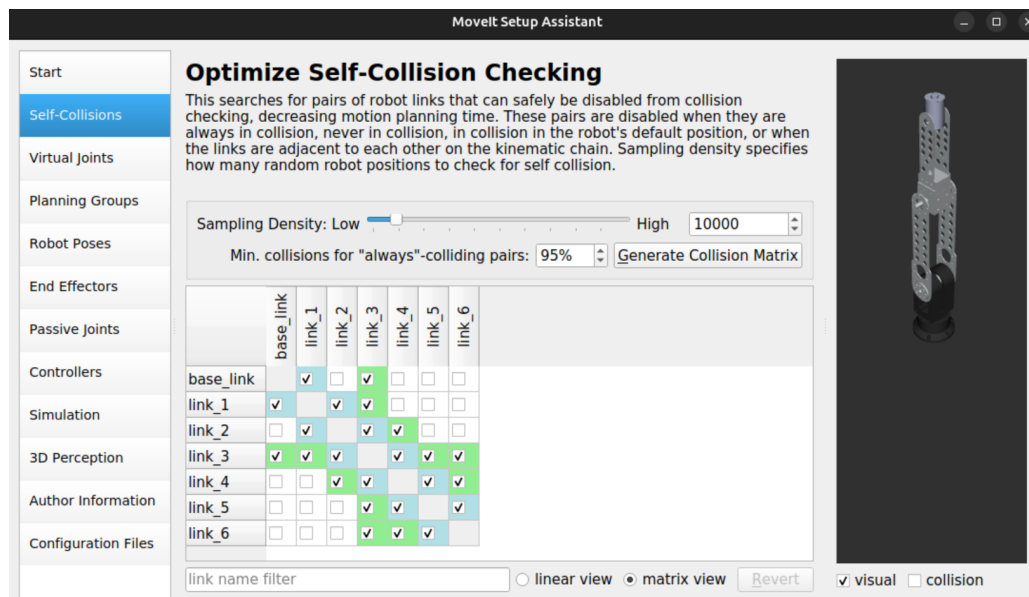


FIGURE 4.4 – Configuration de MoveIt pour le bras manipulateur

L'environnement **MoveIt** a ensuite été configuré pour permettre la planification et la commande du bras. Nous avons utilisé l'outil **MoveIt Setup Assistant** afin de :

- définir les groupes de joints et d'actions,
- configurer les limites de mouvement (angles, vitesses),
- intégrer le modèle URDF,
- générer les fichiers de configuration nécessaires à la commande du bras sous ROS.

4.5.6 Simulation dans Gazebo

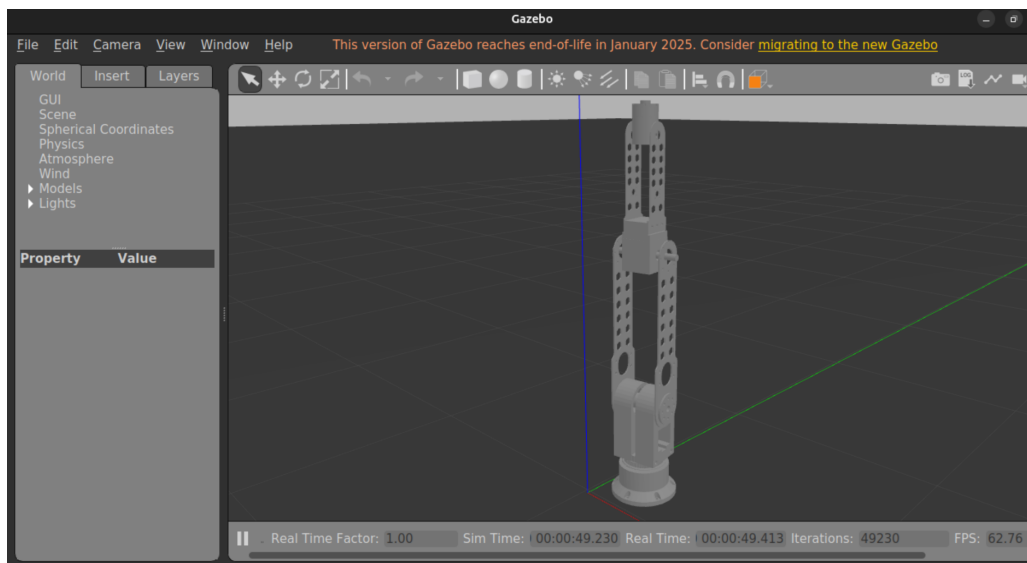


FIGURE 4.5 – Simulation du bras dans Gazebo

Pour valider les comportements attendus, le modèle du robot a été intégré dans le simulateur **Gazebo**. Cet environnement nous a permis de tester :

- la stabilité du bras dans des scénarios virtuels,
- la réponse aux commandes de mouvement,
- les interactions potentielles avec l'environnement.

La simulation a joué un rôle crucial pour corriger les erreurs cinématiques, ajuster les paramètres de jointure et vérifier le bon fonctionnement de la planification dans MoveIt avant de passer à la partie matérielle.

4.5.7 Configuration de MoveIt avec Rviz

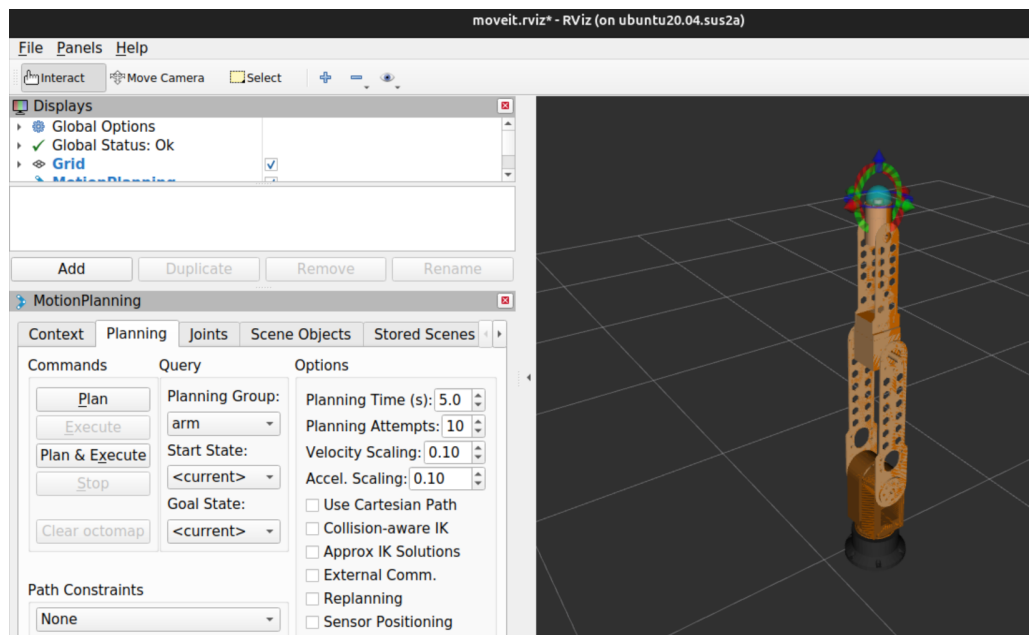


FIGURE 4.6 – Configuration de MoveIt avec Rviz

L'outil **Rviz** a été utilisé pour visualiser le robot et les trajectoires planifiées par MoveIt. C'est un outil de visualisation 3D intégré à ROS qui permet de représenter graphiquement les robots, les capteurs, et les données en temps réel. Il a été configuré pour afficher le modèle du bras, les trajectoires planifiées, et les points d'intérêt (cibles, obstacles). Rviz a permis de valider visuellement les trajectoires, d'ajuster les paramètres de planification, et de collision. Il a également servi à manipuler le bras en temps réel, en déplaçant l'effecteur final et en observant les réponses du système, ce qui a facilité le réglage des paramètres de MoveIt et la vérification de la cinématique inverse.

4.5.8 Choix et intégration du matériel

Une fois la partie logicielle maîtrisée, nous avons procédé à la sélection et à l'intégration des composants électroniques :

- **Carte ESP32** : microcontrôleur principal chargé d'exécuter les instructions de commande, compatible avec ROS via *rosserial*.
- **Drivers DM542** : utilisés pour piloter les moteurs pas à pas avec précision et puissance suffisante.
- **Moteurs pas à pas** : Les moteurs ont été choisis en fonction de leur couple, et de leur poids, adaptés aux segments du bras.
- **Alimentation 24V** : adaptée aux besoins énergétiques des moteurs et des drivers.

4.5.9 Intégration ROS sur microcontrôleur

La communication entre ROS et l'ESP32 a été assurée grâce à **rosserial**, un package permettant à un microcontrôleur de recevoir et d'émettre des messages ROS via un port série. Nous avons développé un firmware embarqué sur l'ESP32 pour :

- interpréter les messages de trajectoire provenant de MoveIt,
- convertir ces trajectoires en signaux de commande pour les drivers DM542,
- gérer les entrées/sorties (capteurs, boutons, etc.) et la sécurité du système.

4.5.10 Tests et validation

Une fois l'intégration matérielle terminée, des tests ont été effectués pour valider le fonctionnement du bras dans des conditions réelles. Les tests ont inclus :

- la vérification de la précision des mouvements,
- la réponse aux commandes de MoveIt,
- la gestion des erreurs et des situations de collision,
- l'évaluation de la robustesse du système dans des scénarios variés.

Les ajustements nécessaires ont été réalisés en fonction des résultats obtenus, notamment en termes de calibration des moteurs, de réglages des limites de mouvement, et d'optimisation des trajectoires.

En résumé, la mise en œuvre du projet a nécessité une parfaite synergie entre les outils de conception logicielle (SolidWorks, ROS, MoveIt, Gazebo) et les éléments matériels (ESP32, moteurs pas à pas, alimentation, drivers), avec un souci constant d'intégration, de modularité et de précision.

4.6 Conclusion

Ce chapitre a présenté les différentes méthodes de commande du bras manipulateur, en mettant l'accent sur la modélisation cinématique et l'utilisation de ROS avec MoveIt. La méthode classique offre une approche simple mais limitée, tandis que l'intégration de ROS et MoveIt permet une flexibilité, une précision et une évolutivité supérieures. La réalisation du bras manipulateur a suivi un processus rigoureux, de la conception mécanique à l'intégration matérielle, en passant par la simulation et la configuration logicielle.

Conclusion générale

La robotique représente aujourd'hui un domaine d'innovation majeur, au cœur de la révolution industrielle et technologique. Elle permet de repousser les limites de l'automatisation, en apportant des solutions concrètes à des problématiques complexes dans des secteurs aussi variés que l'automobile, l'aéronautique, la logistique, ou encore la médecine. Parmi les éléments clés de cette discipline, les bras manipulateurs occupent une place centrale en raison de leur capacité à imiter les gestes humains et à effectuer des tâches de précision, parfois dans des environnements hostiles ou inaccessibles à l'homme.

Dans le cadre de ce projet, nous avons conçu et réalisé un bras manipulateur à six degrés de liberté, intégrant des moteurs pas à pas à haut couple, une structure mécaniquement optimisée, et une architecture de commande avancée basée sur le système ROS. L'objectif principal était de créer un système robotique fonctionnel, polyvalent et modulaire, tout en abordant de manière méthodique toutes les étapes fondamentales d'un projet de robotique moderne.

Le travail a débuté par une étude approfondie des bases théoriques de la robotique, en mettant en lumière les différents types de robots et les principes de fonctionnement des manipulateurs articulés. Par la suite, une conception mécanique détaillée a été réalisée, accompagnée d'un choix judicieux des matériaux pour garantir à la fois robustesse et légèreté. La modélisation cinématique a permis de définir avec précision les mouvements du robot, grâce à la convention de Denavit-Hartenberg, et de poser les bases nécessaires à la commande.

Sur le plan logiciel, deux approches de commande ont été explorées. La première, basée sur les équations de la modélisation, offre une vision analytique et permet une compréhension approfondie du fonctionnement du bras. La seconde, plus pragmatique, repose sur l'intégration du bras dans l'environnement ROS, avec l'utilisation de MoveIt pour la planification de trajectoires et de Gazebo pour la simulation. Ce double choix permet d'avoir un système hybride, mêlant rigueur théorique et flexibilité opérationnelle.

En somme, ce projet a permis de mettre en œuvre un large éventail de compétences : conception assistée par ordinateur (CAO), calculs mécaniques, programmation embarquée, simulation robotique, et intégration logicielle. Il constitue une base solide pour l'approfondissement des connaissances dans le domaine de la robotique industrielle, et un exemple concret de la manière dont la théorie peut être traduite en une application fonctionnelle.

Ce travail, au-delà de sa portée académique, reflète également une passion pour la robotique et une volonté de contribuer à l'évolution de cette discipline en constante mutation.

Bibliographie

- [1] Joseph F. Engelberger. *Robotics in Practice : Management and Applications of Industrial Robots*. Springer, 1989.
- [2] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Springer, 2016.
- [3] John J. Craig. *Introduction to Robotics : Mechanics and Control*. Pearson Education, 2005.
- [4] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros : an open-source robot operating system. *ICRA Workshop on Open Source Software*, 3(3.2) :5, 2009.
- [5] Isaac Asimov. *Runaround*. Gnome Press, 1942.
- [6] Robot Institute of America. Robot definitions, 1979. Published by RIA, USA.
- [7] Japan Industrial Robot Association. Definition of a robot, 1980. JIRA publication.
- [8] Iso 8373 :2012 — robots and robotic devices – vocabulary, 2012. <https://www.iso.org/standard/55890.html>.
- [9] Intuitive Surgical. da vinci surgical system, 2020. <https://www.intuitive.com/en-us/products-and-services/da-vinci>.
- [10] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![motion planning framework for robots]. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [11] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2149–2154, 2004.
- [12] Dassault Systèmes. Solidworks 2020 - user guide, 2020. <https://www.solidworks.com>.
- [13] Martin Philip Bendsoe and Ole Sigmund. *Topology Optimization : Theory, Methods, and Applications*. Springer Science & Business Media, 2003.
- [14] Ole Sigmund and Kurt Maute. Topology optimization approaches. *Structural and Multi-disciplinary Optimization*, 48(6) :1031–1055, 2013.
- [15] Lorna J. Gibson and Michael F. Ashby. *Cellular Solids : Structure and Properties*. Cambridge University Press, 2 edition, 2010.
- [16] Michael F. Ashby. *Materials Selection in Mechanical Design*. Butterworth-Heinemann, 4 edition, 2010.
- [17] Takashi Kenjo and Akira Sugawara. *Stepping Motors and Their Microprocessor Controls*. Oxford University Press, 1994.
- [18] Robert L. Norton. *Design of Machinery : An Introduction to the Synthesis and Analysis of Mechanisms and Machines*. McGraw-Hill Education, 4 edition, 2006.
- [19] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics : Modeling, Planning and Control*. Springer, London, 2010.
- [20] John J. Craig. *Introduction to Robotics : Mechanics and Control*. Pearson Education, Upper Saddle River, NJ, 3rd edition, 2005.
- [21] Mike McCauley. Accelstepper library for arduino. <http://www.airspayce.com/mikem/arduino/AccelStepper/>. Consulté en 2025.

- [22] Ros (robot operating system). <https://www.ros.org/>. Consulté en 2025.
- [23] Moveit. <https://moveit.picknik.ai/>. Consulté en 2025.
- [24] Sw2urdf exporter plugin. https://github.com/ros/solidworks_urdf_exporter. Consulté en 2025.
- [25] Gazebo simulator. <https://gazebo.org/>. Consulté en 2025.
- [26] roserial : Ros communication protocol for microcontrollers. <http://wiki.ros.org/roserial>. Consulté en juin 2025.