

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Automatique

Spécialité : Automatique et Informatique Industrielle

Par : - BOUHASSANE Salah Eddine

- HARCHAOUI Bilal Abdessamed

Sujet

Contrôle visuel d'un robot aérien

Soutenu publiquement, le 26/09/2024 , devant le jury composé de :

Mme Kahouaji Salima
M Benyahia Boumediene
M Bensalah Choukri

MCA
Prof
MCB

Université de Tlemcen
Université de Tlemcen
Université de Tlemcen

Présidente
Examineur
Encadrant

Année universitaire : 2023 / 2024

Remerciement

Nous tenons à exprimer nos sincères remerciements à toutes les personnes qui ont contribué à la réalisation de ce projet de fin d'études.

Nous remercions tout particulièrement notre encadrant, Mr.BENSALAH Choukri, pour son accompagnement, ses conseils avisés et sa disponibilité tout au long de ce travail. Ses orientations et son expertise ont grandement contribué à la réussite de cette étude.

Nous souhaitons également exprimer notre gratitude à l'ensemble des enseignants et responsables de la faculté de technologie, dont les enseignements et le soutien nous ont permis d'acquérir les connaissances nécessaires pour mener à bien ce projet. Leur encadrement académique a joué un rôle fondamental dans notre parcours.

Nos remerciements s'adressent également aux membres du jury pour avoir pris le temps d'évaluer notre travail et pour leurs remarques constructives. Leur expertise et leurs retours seront précieux pour améliorer et approfondir nos connaissances dans le domaine.

Nos remerciements s'adressent également à nos collègues et camarades, qui nous ont soutenus et ont partagé avec nous leurs idées et leur expertise, enrichissant ainsi ce projet.

Enfin, nous exprimons toute notre reconnaissance à nos familles pour leur soutien inconditionnel, leur patience et leurs encouragements. Leur présence à nos côtés a été une source précieuse de motivation.

À toutes et à tous, nous vous remercions chaleureusement pour votre contribution à l'accomplissement de ce travail.

Table des matières

1	Introduction	1
1.1	Contexte général	2
1.2	Drone	2
1.3	Bras Manipulateurs	3
1.3.1	Composants et Fonctionnalités des Bras Manipulateurs	4
1.3.2	Applications des Bras Manipulateurs dans les Drones	4
1.3.3	Défis et Perspectives	4
1.4	Caméra	4
1.5	Notre robot	5
1.5.1	Conception et Assemblage	5
1.5.2	Système de Contrôle Visuel	6
1.6	Motivations	6
1.7	L'Etat de l'art	6
1.7.1	Avancées Technologiques	6
1.7.2	Défis et Limitations	7
1.7.3	Perspectives et Directions Futures	7
1.8	Conclusion	7
2	Modèle cinématique	8
2.1	Introduction	8
2.2	Modèle géométrique direct	8
2.3	Définition du repère de référence	8
2.3.1	Position du quadricoptère (P_b)	9
2.3.2	Orientation du Quadricoptère (R_b)	9
2.3.3	Analyse cinématique de position	10
2.4	Position de l'effecteur final (P_e)	10
2.4.1	Orientation de l'effecteur final (R_e)	10
2.5	Analyse de la cinématique directe	10
2.5.1	La vitesse angulaire de l'effecteur final	11
2.5.2	Vitesse généralisée de l'effecteur final par rapport à la base du drone	11
2.5.3	La vitesse généralisée de l'effecteur final par rapport au repère global	11
2.6	Le problème inverse	11
2.6.1	Position du châssis par rapport au repère global :	13
2.6.2	Drone vu virtuellement comme un manipulateur série :	13
2.6.3	Évitement de l'auto-collision	14
3	Asservissement Visuel 2D	15
3.1	Introduction	15
3.1.1	Paramètres Intrinsèques et Extrinsèques	15
3.1.2	Paramètres Extrinsèques de la Caméra	16
3.1.2.1	Système de coordonnées mondial (3D) :	16
3.1.2.2	Système de coordonnées objet/caméra (3D) :	16

3.1.3	Paramètres Intrinsèques de la Caméra	18
3.1.3.1	Système de coordonnées d'image (2D) [Modèle sténopé] :	18
3.1.3.2	Système de coordonnées de pixels (2D) :	20
3.1.4	Calibration de la Caméra et Correction de la Distorsion	24
3.2	Modèle du camera (Pinhol)	24
3.2.1	Principe du Modèle Sténopé	24
3.2.2	Équations du Modèle Sténopé	24
3.2.3	Matrice de Projection du Modèle Sténopé	24
3.3	Control visuel basé sur l'image	25
3.3.1	Génération de trajectoire basée sur le suivi visuel	26
3.3.2	Schema de commande visuel	27
4	Simulation	28
4.1	Introduction	28
4.2	Première simulateur : Gazebo	28
4.2.1	Gazebo	28
4.2.2	ROS (Robot Operating System) :	30
4.2.3	Difficultés Rencontrées :	32
4.3	Deuxième simulateur : CoppeliaSim	32
4.3.1	CoppeliaSim (V-REP) :	32
4.3.1.1	Configuration de la caméra :	33
4.3.1.2	Simulation du déplacement du robot	35
4.3.2	MATLAB :	36
4.3.2.1	Simulation sur matlab	37
5	Conclusion Général	42

Table des figures

1.1	Drone Quadrator	3
1.2	Bras Manipulateurs Pincherx	4
1.3	Camera robotique	5
1.4	Notre Robot	6
2.1	manipulateur aérien	8
2.2	Schéma avec les repères pertinents	9
2.3	drone comme manipulateur en série de 4 Dof (RPPP) et 3 Dof (RRR) du bras	14
2.4	Possibilité d'auto-collision	14
3.1	Controle du robot de manière a tendre vers l'image désirée	15
3.2	Les systèmes de coordonnées couramment utilisés dans Computer Vision	15
3.3	Point 3D dans un système de coordonnées mondiales	16
3.4	Matrice extrinsèque de la caméra	17
3.5	Point 3D dans le système de coordonnées de caméra	18
3.6	Plan d'image 2d de la caméra	18
3.7	Discrétisation du plan d'image 2D de la caméra	21
3.8	coordonnées pixels	21
3.9	Plan d'image asymétrique	23
3.10	Modele du camera Pinhole	24
3.11	Trajectoires des primitives visuelles dans l'image	26
3.12	Structure d'un asservissement visuel 2D.	27
4.1	Interface de GAZEBO.	29
4.2	Icon du blender	29
4.3	Interface du Blender	29
4.4	Icon du Sketchup	30
4.5	Interface du SketchUp	30
4.6	Explication de systeme ROS	30
4.7	Icon du ROS	31
4.8	Interface du ROS	31
4.9	Explication de communication entre ROS et Gazebo	31
4.10	Icon du CoppeliaSim	32
4.11	Interface du CoppeliaSim	32
4.12	Notre Robot sur CoppeliaSim	33
4.13	le réglage des parametre de la camera	34
4.14	lavisoin de la camera sur CoppeliaSim	34
4.15	détection des cercles	35
4.16	Déplacement du robot vers le point désiré	36
4.17	Icon du Matlab	37
4.18	Interface du Matlab	37
4.19	Résultats de l'asservissement visuel et de la cinématique inverse	38
4.20	Résultats du déplacement du repère caméra et l'évolution des points sur le plan d'image	39

4.21 l'erreur dans le plan image	40
4.22 Mouvement cartésien du drone (X Y Z)	41
4.23 Articulation de bras	41

Chapitre 1

Introduction

Au début du XXe siècle, la robotique avec vision a commencé à se développer en raison des recherches menées par des ingénieurs pour tester des hypothèses émises par des biologistes et des psychologues. En 1973, l'université de Waseda à Tokyo a créé le WABOT-1, l'un des premiers robots à utiliser la vision. Il avait la capacité de contrôler ses membres, possédait une vision et était capable de communiquer.

L'émergence des drones avec vision a insufflé une nouvelle vitalité au développement de divers domaines et industries. Ces dernières années, la recherche et le développement technologiques se sont multipliés autour des drones, et l'exploration de l'application de la vision industrielle dans les drones est une priorité majeure.

Les avancées significatives dans le domaine de la robotique aérienne ont été rendues possibles grâce au couplage des deux systèmes, la robotique et les drones avec vision. Par exemple, un projet de recherche intitulé "Couplage Apprentissage & Vision pour la robotique mobile" vise à améliorer l'autonomie des robots aériens en leur apprenant à faire des réglages de commande visuelle en plein vol ou atterrissage sans aucune intervention humaine.

Le contrôle visuel (visual servoing) est une technique qui permet de contrôler le mouvement d'un robot en utilisant les informations visuelles fournies par une ou plusieurs caméras. Cette technique présente de nombreux avantages, tels que la robustesse aux perturbations, la simplicité de mise en œuvre et la possibilité de réaliser des tâches complexes sans avoir besoin de connaître la position exacte du robot ou de l'objet désiré à manipuler [11].

Les manipulateurs aériens sont des systèmes robotiques composés d'un véhicule aérien sans pilote (Unmanned Aerial Vehicle) et d'un bras manipulateur. Ces systèmes offrent une grande mobilité et une capacité d'intervention dans des environnements difficiles et dangereux, tels que les opérations de maintenance, de réparation, de sauvetage etc [7]. Cependant, ils présentent aussi des défis importants, tels que la stabilité du vol, le couplage dynamique entre le drone et le bras, la limitation de la charge utile et de l'autonomie, ou encore la perception de l'environnement [13].

Dans ce contexte, plusieurs travaux sont réalisés pour concevoir des méthodes de contrôle visuel efficaces et fiables pour les UAV (Unmanned Aerial Vehicle) en environnements naturels. Cette recherche vise à améliorer les performances du contrôle visuel, à renforcer la stabilité et la robustesse du système, et à développer des techniques plus innovantes pour permettre aux manipulateurs aériens de réaliser des tâches de manipulation aérienne en toute sécurité et efficacité dans des environnements naturels complexes.

Cette étude se focalisera sur les différentes techniques de contrôle visuel pour les manipulateurs aériens autonomes. L'objectif est de fournir des connaissances théoriques et pratiques par simulation sur les techniques de contrôle visuel les plus récentes, ainsi que des suggestions pour les applications futures.

1.1 Contexte général

La manipulation aérienne est une discipline qui vise à utiliser un véhicule aérien sans pilote (UAV) équipé d'un bras manipulateur pour réaliser des tâches de manipulation dans des environnements inaccessibles ou dangereux. Tant sur le plan scientifique que socio-économique, cette discipline comporte de nombreuses motivations et enjeux. D'un point de vue scientifique, la manipulation aérienne présente des défis majeurs en ce qui concerne la modélisation, la perception, le contrôle et la coordination des systèmes robotiques complexes et couplés. Sur le plan socio-économique, la manipulation aérienne offre des perspectives prometteuses pour des applications variées, telles que la construction, la maintenance, l'inspection, le sauvetage, ou encore l'exploration.

En utilisant les informations visuelles fournies par une ou plusieurs caméras, on peut contrôler le mouvement d'un robot en utilisant le contrôle visuel. Cette technique présente plusieurs avantages, tels que la simplicité de mise en œuvre, la robustesse aux perturbations et la possibilité de réaliser des tâches complexes sans avoir besoin de connaître la position exacte du robot ou de l'objet à manipuler [12]. Cependant, elle pose aussi plusieurs défis, tels que le choix des caractéristiques visuelles à utiliser, la modélisation du système visuel, la conception de la loi de commande et la garantie de la stabilité et de la performance du système [1]. Il existe deux types principaux de contrôle visuel par asservissement : le contrôle visuel basé sur la position, qui utilise une estimation de la pose du robot ou de l'objet dans un repère tridimensionnel, et le contrôle visuel basé sur l'image, qui utilise directement les mesures extraites des images [3]. Chaque type a ses avantages et ses inconvénients, et il n'y a pas de solution universelle pour toutes les situations [9].

Les objectifs de ce mémoire sont de concevoir, de simuler et de valider une approche de contrôle visuel pour la manipulation aérienne en utilisant un UAV simplifié et un bras manipulateur série. L'approche proposée se compose de deux éléments principaux : (1) un système de détection et de suivi visuel, qui permet de guider le UAV vers l'objet à manipuler en utilisant les lignes présentes dans l'environnement, et (2) un système de contrôle basé sur la géométrie visuelle, qui permet de réguler le mouvement du bras manipulateur en fonction de la distance et de l'orientation de l'objet. La méthode recommandée ne dépend pas de l'estimation de la position ou de la vitesse du robot à partir d'un système de localisation externe, ni de l'emploi de marqueurs visuels prédéfinis. La complexité du système mécanique et de l'architecture de contrôle est également réduite grâce à la nature simplifiée du UAV, qui dispose de quatre degrés de liberté de contrôle, et du bras manipulateur, qui dispose de trois degrés de liberté. Une seule caméra est utilisée pour le contrôle visuel, ce qui réduit le coût et le poids du système. Des simulations seront menées sur des logiciels sous Windows, tels que Coppelia vrep, MATLAB, pour évaluer la faisabilité et l'efficacité de l'approche proposée.

1.2 Drone

Les drones, également connus sous le nom de véhicules aériens sans pilote (UAV) ou de systèmes aériens sans pilote (UAS), sont des engins sans équipage dont le pilotage est automatique ou télécommandé. Ils peuvent être volants, terrestres ou amphibies, et sont utilisés à des fins civiles ou militaires. Leur masse varie de quelques grammes à plusieurs tonnes, en fonction des capacités recherchées.

Au début du 20ème siècle, les premiers prototypes d'avions-cibles ont été conçus pour être utilisés comme leurres ou comme bombes volantes, marquant ainsi le début de l'histoire des drones. Ces premiers drones étaient rudimentaires et limités dans leurs capacités. Cependant, avec le temps et les progrès technologiques, ils ont évolué pour devenir des outils sophistiqués et polyvalents.

Aujourd'hui, les drones sont équipés de divers capteurs, caméras, missiles ou bras manipulateurs. Ces équipements leur permettent d'effectuer une grande variété de missions. Par exemple, ils peuvent être utilisés pour la reconnaissance, la surveillance, l'attaque, la recherche, le sauvetage, la cartographie, la livraison, la photographie ou la vidéographie. Leur autonomie peut atteindre jusqu'à plusieurs dizaines d'heures, ce qui les rend particulièrement utiles pour des missions de longue durée.

Il existe différentes sortes de drones, qui se distinguent par leur taille, leur vitesse, leur autonomie, leur fonction ou leur mode de propulsion. On distingue ainsi les micro et mini-drones, les drones tactiques, les drones MALE et HALE, les drones de combat, les drones de loisir, les drones civils, les drones à voilure fixe, les drones à voilure tournante, les drones hybrides, etc.

Les micro et mini-drones sont généralement utilisés pour des missions de reconnaissance ou de surveillance. Les drones tactiques sont utilisés pour des missions militaires spécifiques. Les drones MALE (Moyenne Altitude Longue Endurance) et HALE (Haute Altitude Longue Endurance) sont capables de voler à des altitudes élevées pendant de longues périodes.[2] [5] Les drones de combat sont équipés d'armes pour des missions d'attaque. Les drones de loisir sont utilisés pour le plaisir et peuvent être pilotés par n'importe qui. Les drones civils sont utilisés pour diverses applications civiles, comme la surveillance, la cartographie ou la livraison. [4] Les drones à voilure fixe ressemblent à des avions traditionnels, tandis que les drones à voilure tournante ressemblent à des hélicoptères. Enfin, les drones hybrides combinent les caractéristiques des drones à voilure fixe et à voilure tournante. [6] [10]



FIGURE 1.1 – Drone Quadrator

1.3 Bras Manipulateurs

Un bras manipulateur est indispensable dans le domaine de la robotique. C'est un dispositif habituellement programmable qui reproduit les fonctionnalités d'un bras humain. Les connexions de ce manipulateur sont fixées par des axes qui permettent soit un mouvement de rotation (comme dans un robot articulé) soit un mouvement de translation (linéaire). On peut utiliser ces bras pour effectuer une variété de tâches avec une grande précision. Qu'ils soient fixes ou mobiles (c'est-à-dire à roues), ils peuvent être conçus pour des applications industrielles.

En ce qui concerne les bras manipulateurs aériens, ils sont des systèmes robotiques aériens combinant l'agilité de leur base, sous forme de multicoptère, à la capacité de manipulation du ou des bras robotiques montés. Le robot aérien développé au Laas-CNRS est un exemple de cela, avec un bras léger et deux articulations. Voici un autre exemple : BrasVor, un petit robot muni de 6 servomoteurs de 9g et d'une carte Arduino Uno, ainsi que d'une carte contrôleur PWM/servo 16 canaux.

La sélection d'un bras manipulateur aérien est influencée par divers éléments. D'abord, il est important de prendre en considération la fonction pour laquelle le bras sera employé. Par exemple, si le bras est utilisé pour saisir des objets statiques, il serait peut-être préférable de choisir un bras qui possède une grande précision et une bonne capacité de manipulation. En outre, il convient également de prendre en compte la complexité mécanique et informatique du bras. Effectivement, un bras doté de plusieurs niveaux de liberté sera plus souple dans ses mouvements, mais également plus difficile à concevoir et à maîtriser. Enfin, il est également nécessaire de prendre en considération les contraintes particulières de l'environnement dans lequel le bras sera employé.

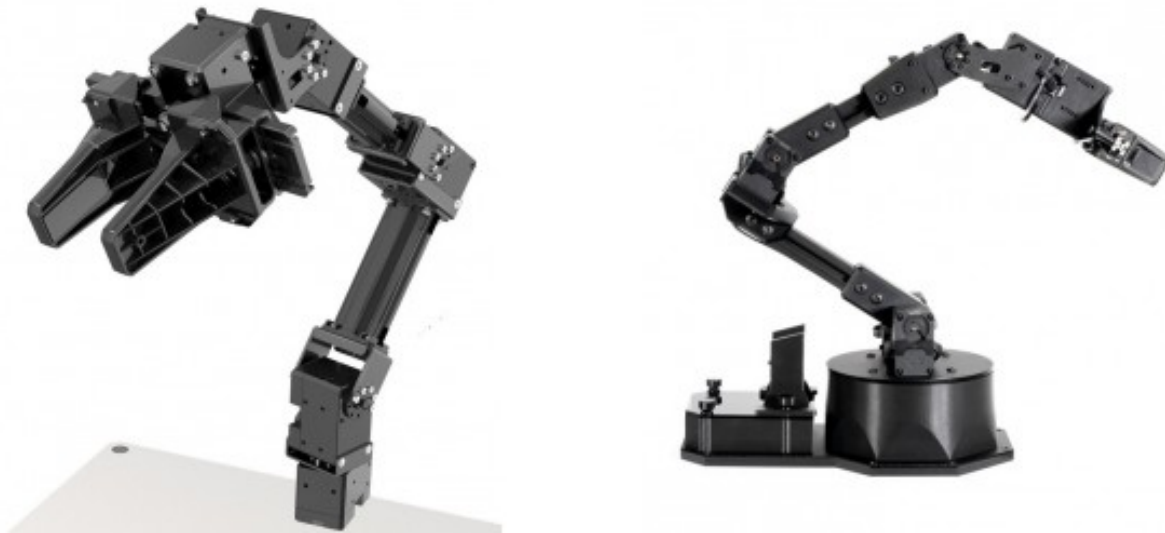


FIGURE 1.2 – Bras Manipulateurs Pincherx

1.3.1 Composants et Fonctionnalités des Bras Manipulateurs

Les bras manipulateurs sont des dispositifs programmables qui reproduisent les mouvements d'un bras humain. Ils sont formés de connexions reliées par des axes de rotation et/ou de translation. Grâce à ces axes, il est possible d'effectuer des mouvements précis et contrôlés, ce qui est crucial pour manipuler des objets en trois dimensions. Il est possible de monter les bras manipulateurs sur des drones, des robots industriels ou d'autres plateformes aériennes.

1.3.2 Applications des Bras Manipulateurs dans les Drones

Les bras manipulateurs offrent des possibilités passionnantes dans le cadre des drones quadricoptères. Il est possible pour le bras manipulateur de se déplacer vers l'objet, de l'attraper et de le transporter. Cela permet de développer des applications telles que la maintenance aérienne, la collecte d'échantillons, la livraison de petits colis et même sauvetage en cas de catastrophe.

1.3.3 Défis et Perspectives

Cependant, l'utilisation de bras manipulateurs dans les drones comporte des défis importants. La stabilité et la précision sont des préoccupations importantes. Les vibrations du drone peuvent altérer la performance du bras manipulateur, et il est crucial de gérer l'énergie pour augmenter l'autonomie du système. En outre, un autre défi majeur est la génération de trajectoires pour l'ensemble drone + bras manipulateur. La coordination entre le mouvement du drone et celui du bras manipulateur est essentielle pour réaliser des tâches précises. La génération de trajectoires efficaces et réalisables pour ce système couplé est un domaine de recherche actif et présente des défis uniques en termes de planification et de contrôle.

1.4 Caméra

La caméra joue un rôle crucial dans le domaine de la robotique aérienne. Sa fonction est similaire à celle des yeux humains, captant des images et des vidéos du monde qui nous entoure. Dans le cadre de votre recherche sur le contrôle par servoing visuel pour les manipulateurs aériens, examinons attentivement le rôle essentiel de la caméra dans le système de votre drone quadricoptère équipé d'un bras manipulateur.

L'appareil photo est fixé sur l'effecteur final. Son orientation vers l'environnement est essentielle pour la détection visuelle des objets. En utilisant cette technologie, le drone a la capacité de voir son environnement,

repérer des cibles et prendre des décisions en temps réel.

Différents types de caméras sont employés dans les drones. Les caméras de haute résolution, les caméras thermiques, les caméras infrarouges et les caméras multispectrales sont les plus fréquentes. Chacune de ces caméras présente ses propres bénéfices et désavantages. Par exemple, il est possible que les problèmes fréquents comprennent des mises à jour de firmware, une calibration incorrecte, des cartes SD endommagées, des rayures sur l'objectif et même des crashes de drones.

Heureusement, ces problèmes peuvent être résolus. Par exemple, afin de résoudre les problèmes de mise au point de la caméra, il est possible de vérifier si l'objectif présente des micro-rayures et de la poussière, des gouttelettes d'eau à l'intérieur de l'objectif de la caméra, ou si l'autofocus ne fonctionne pas. En outre, des avancées technologiques telles que les cellules de combustible à hydrogène peuvent contribuer à étendre la durée de vol des drones, ce qui permet d'accélérer la prise d'images.

Quant à la position de la caméra sur les drones, elle dépend du type de drone et de l'utilisation. Par exemple, dans le cas des drones de course FPV, la caméra est habituellement en position antérieure. En général, les drones destinés à la photographie et à la vidéographie sont montés sur un cardan situé sous le drone.

En utilisant des moteurs sans balais et des capteurs internes, le cardan permet de stabiliser la caméra du drone pendant le vol.[6] En diminuant les vibrations et les mouvements soudains, il améliore la photographie aérienne, offrant ainsi des images claires et fluides.[4][8][10]



FIGURE 1.3 – Camera robotique

1.5 Notre robot

1.5.1 Conception et Assemblage

L'assemblage de notre système robotique aérien représente une avancée significative dans le domaine de la robotique et de l'automatisation. Notre dispositif comprend un drone quadrateur qui est conçu pour être stable et agile, capable de manœuvres précises dans des espaces restreints. Le bras manipulateur, relié au drone, dispose de trois articulations qui lui offrent une liberté de mouvement essentielle pour atteindre et saisir une grande variété d'objets. La caméra est installée de manière à obtenir une vue dégagée de l'environnement, ce qui permet une détection efficace des objets à manipuler.

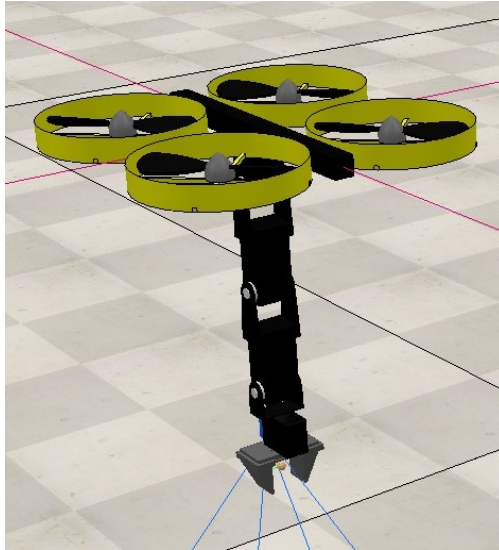


FIGURE 1.4 – Notre Robot

1.5.2 Système de Contrôle Visuel

Le système de contrôle visuel est au cœur de l'interaction entre le drone et son environnement. Grâce à des algorithmes avancés de traitement d'image, la caméra capte les caractéristiques visuelles des objets, les transforme en données exploitables et transmet ces informations au système de commande du drone. Ce dernier, en interprétant ces données, ajuste sa trajectoire et coordonne le mouvement du bras pour effectuer la saisie de l'objet avec précision.

1.6 Motivations

La motivation derrière le choix de mon sujet de recherche découle d'une fascination profonde pour la robotique avancée et ses applications innovantes. L'automatisation a le potentiel de transformer radicalement la manière dont nous interagissons avec notre environnement, en particulier dans des domaines nécessitant précision et réactivité.

L'étude du contrôle visuel pour les manipulateurs aériens s'inscrit dans cette vision, où l'intégration de la perception visuelle et de l'action mécanique ouvre la voie à des avancées significatives. Les drones équipés de bras manipulateurs et de systèmes de vision sont à l'avant-garde de cette révolution, capables de réaliser des tâches autonomes dans des environnements complexes et dynamiques.

Cette recherche est motivée par le désir de repousser les limites de ce qui est techniquement possible, en explorant comment ces systèmes peuvent non seulement améliorer les opérations dans des secteurs tels que l'inspection industrielle, la maintenance et le sauvetage, mais aussi comment ils peuvent contribuer à des avancées scientifiques fondamentales dans la compréhension de l'interaction homme-machine. C'est un domaine d'étude stimulant qui présente des défis techniques non triviaux et offre des applications pratiques immédiates, rendant la recherche à la fois passionnante et pertinente.

1.7 L'Etat de l'art

1.7.1 Avancées Technologiques

Les dernières décennies ont vu des progrès significatifs dans le développement de systèmes de contrôle visuel pour les manipulateurs aériens. Les recherches se sont concentrées sur l'amélioration de la précision et de la réactivité des UAVs (Unmanned Aerial Vehicles) dans des tâches de manipulation à haute altitude, réduisant ainsi les risques et les coûts associés au travail humain[7][8]. Les systèmes de localisation et de

cartographie simultanées (SLAM) et les algorithmes de vision par ordinateur ont joué un rôle clé dans ces avancées, permettant une meilleure compréhension et interaction avec l'environnement.

1.7.2 Défis et Limitations

Malgré ces avancées, plusieurs défis demeurent, notamment la dépendance à des systèmes de localisation externes et la nécessité de marqueurs visuels prédéfinis, qui ne sont pas toujours pratiques ou possibles dans tous les environnements[8]. De plus, la plupart des travaux antérieurs ont utilisé des UAVs sous-actionnés, ce qui complique la conception mécanique et le contrôle de l'effecteur final.

1.7.3 Perspectives et Directions Futures

Les recherches actuelles explorent l'utilisation de UAVs entièrement actionnés pour simplifier la conception mécanique et l'architecture de contrôle. L'intégration de systèmes de détection, ainsi que de systèmes de contrôle hybride, montre une capacité prometteuse à exécuter des tâches de suivi de mouvement et de maintien. Ces études ouvrent la voie à des applications pratiques telles que la maintenance de ponts et la réparation de turbines éoliennes, où la précision et la sécurité sont primordiales.

1.8 Conclusion

En guise de conclusion, l'étude du contrôle visuel pour les manipulateurs aériens représente un domaine de recherche dynamique et en pleine expansion. L'intégration de la vision par ordinateur dans les systèmes UAVs ouvre des possibilités sans précédent pour l'automatisation de tâches complexes et l'amélioration de l'efficacité opérationnelle dans divers secteurs.

La capacité de ces systèmes à naviguer et à interagir de manière autonome dans des environnements tridimensionnels complexe est particulièrement pertinente pour des applications telles que l'inspection d'infrastructures et les interventions d'urgence. Les progrès continus dans le domaines de la robotique promettent d'accroître encore davantage la précision et la fiabilité de ces systèmes.

En somme, le contrôle visuel pour les manipulateurs aériens est un champ de recherche prometteur qui continue d'évoluer, offrant une gamme de technologies et de méthodes pour relever les défis associés à la navigation et à la manipulation dans des environnements qui étaient auparavant inaccessibles ou trop risqués pour l'intervention humaine.

Chapitre 2

Modèle cinématique

2.1 Introduction

Ce chapitre se concentre sur les principes fondamentaux de la modélisation cinématique d'un drone quadrotor équipé d'un bras manipulateur à deux articulations.

La modélisation cinématique est essentielle pour comprendre le mouvement et les interactions complexes de ce système, et sert de base pour la conception et l'analyse ultérieures des systèmes de contrôle.

En d'autres termes, la modélisation cinématique permet d'analyser les performances du système et identifier les limitations potentielles.

La modélisation cinématique est donc un outil crucial pour le développement de drones quad-rotors avec bras manipulateurs performants et fiables.



FIGURE 2.1 – manipulateur aérien

2.2 Modèle géométrique direct

Le modèle géométrique direct est un outil mathématique utilisé pour prédire la position et l'orientation d'un point dans l'espace cartésien en fonction des valeurs des articulations d'un système mécanique.

Pour notre cas la géométrie directe d'un drone avec bras est le processus qui permet de déterminer la position et l'orientation de l'extrémité du bras manipulateur en fonction des angles des moteurs du drone.

2.3 Définition du repère de référence

Pour commencer, nous allons établir un système de coordonnées pour notre quadricoptère. Nous choisissons un référentiel attaché directement au quadricoptère lui-même, appelé référentiel fixe au corps, dont l'origine est placée au centre du quadricoptère. Ce référentiel est mobile avec le quadricoptère. De plus, on

utilise aussi un référentiel inertiel fixe par rapport au monde. Ce dernier reste stable et utilisé pour décrire la position et l'orientation du quadricoptère par rapport à un point de référence fixe. Pour l'effecteur final, on utilise également un autre référentiel.

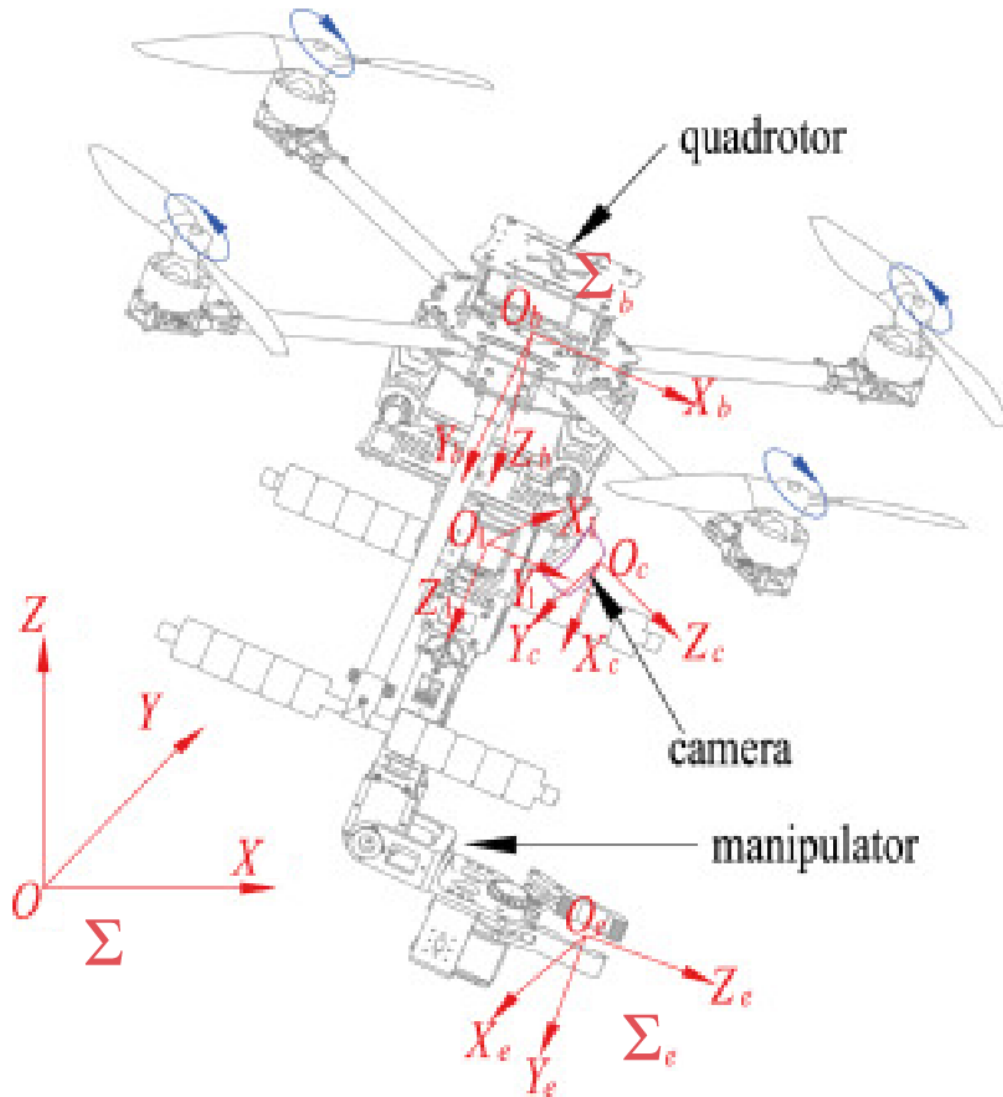


FIGURE 2.2 – Schéma avec les repères pertinents

2.3.1 Position du quadricoptère (P_b)

Il s'agit des coordonnées x, y et z du centre de masse du quadricoptère dans le repère inertiel. vecteur P_b :

$$\mathbf{P}_b = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

2.3.2 Orientation du Quadricoptère (R_b)

Le quadricoptère est défini par une matrice de rotation qui décrit son orientation dans l'espace. Elle est déterminée par ses angles de roulis ($\psi \theta \phi$). Cette matrice offre une représentation mathématique précise du

quadricoptère tourné dans l'espace de trois dimensions, permettant des calculs et des simulations précises de ses mouvements.

La matrice $R_b : R_b = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi)$

avec :

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$R_b = \begin{bmatrix} C_\psi C_\theta & S_\phi S_\theta C_\psi - S_\psi C_\phi & S_\psi S_\phi + C_\psi S_\theta C_\phi \\ S_\psi C_\theta & C_\psi C_\phi + S_\psi S_\theta S_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

2.3.3 Analyse cinématique de position

Nous examinons comment nous pouvons décrire la position de l'effecteur final du quadricoptère en termes de la vitesse de la position et de l'orientation. Nous utilisons la matrice de rotation R_b pour traduire la position de l'extrémité effectrice par rapport au repère fixe au corps en sa position par rapport au repère fixe au monde.

2.4 Position de l'effecteur final (P_e)

On a La position du référentiel attaché à l'effecteur final par rapport au référentiel inertiel fixé dans le monde, Nous obtenons la position de l'extrémité effectrice par rapport au monde en ajoutant la position du quadricoptère (P_b) à la rotation de la position de l'extrémité effectrice par rapport au quadricoptère (P_{eb}^b). Considérons le repère e , O_2 x_2 y_2 z_2 , attaché à l'effecteur final du manipulateur, comme illustré dans la figure(2.2) Ainsi, la position de l'effecteur final e est donnée par :

$$P_e = P_b + R_b P_{eb}^b \quad (2.1)$$

2.4.1 Orientation de l'effecteur final (R_e)

L'orientation de l'effecteur final peut être définie par la matrice de rotation obtenue en multipliant la matrice de rotation du quadricoptère (R_b) avec la matrice de rotation de l'effecteur final par rapport au quadricoptère (R_e^b).

$$R_e = R_b R_e^b \quad (2.2)$$

Où R_e^b décrit l'orientation de l'effecteur final du bras manipulateur par rapport au centre de gravité de drone.

2.5 Analyse de la cinématique directe

Le problème de la cinématique consiste à établir les relations entre les variables articulaire et opérationnelles main en dérivée. $\chi_e = [\dot{x}_e, \dot{y}_e, \dot{z}_e, \dot{\psi}_e, \dot{\theta}_e, \dot{\phi}_e]^T$ en fonction des vitesse des coordonnées articulaires $\dot{q} = [\dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\theta}, \dot{\phi}, \dot{\theta}_1, \dot{\theta}_2]^T$.

Pour résoudre la cinématique avant, les entrées sont 8 variables, q , et la sortie est 6 variables, χ_e .

La position de l'effecteur final peut être trouvée à partir de l'équation (2.1).

Les angles d'Euler de l'effet terminal Φ_e peuvent être calculés à partir de la matrice de rotation R_e^b , en utilisant l'équation (2.2).

2.5.1 La vitesse angulaire de l'effecteur final

Dans la suite de cette section, l'analyse cinématique des vitesses sera discutée. La vitesse linéaire \dot{P}_e de l'effecteur final dans le monde fixe est obtenue par différenciation de l'équation (2.1).

$$\dot{P}_e = \dot{P}_b - \text{Skew}(R_b p_{eb}^b) \omega_b + R_b \dot{P}_{eb}^b \quad (2.3)$$

où $\text{Skew}(\cdot)$ est l'opérateur de la matrice antisymétrique (3×3), alors que ω_b est la vitesse angulaire du quadricoptère exprimée dans le référentiel inertiel fixé dans le monde. La vitesse angulaire ω_e de l'effecteur final est exprimée par cette equation :

$$\omega_e = \omega_b + R_b \omega_{eb}^b \quad (2.4)$$

La vitesse angulaire ω_{eb}^b représente la vitesse de l'effecteur final par rapport à Σ_b .

2.5.2 Vitesse généralisée de l'effecteur final par rapport à la base du drone

Considérons $\Theta = [\theta_1 \theta_2]^T$ comme le vecteur (2×1) des angles articulaires du manipulateur. Le vecteur généralisé de vitesse de (6×1) de l'effecteur final par rapport à Σ_b , $v_{b_{eb}} = [\dot{p}_{b_{eb}}^T \omega_{b_{eb}}^T]^T$, peut être exprimé en fonction des vitesses articulaires $\dot{\Theta}$ via le Jacobien du manipulateur, $J_{b_{eb}}$, Peut-être exprimée de la manière suivante :

$$v_{b_{eb}} = J_{b_{eb}}^b \dot{\Theta} \quad (2.5)$$

2.5.3 La vitesse généralisée de l'effecteur final par rapport au repère global

À partir des équations (2.3) et (2.4), la vitesse généralisée de l'effecteur final, $\mathbf{v}_e = [\dot{P}_e^T \omega_e^T]^T$, peut-être exprimée de la manière suivante :

$$\mathbf{v}_e = \mathbf{J}_b \mathbf{v}_b + \mathbf{J}_{eb} \dot{\Theta} \quad (2.6)$$

$$\text{où } v_b = [\dot{P}_b^T \omega_b^T]^T, J_b = \begin{pmatrix} I_3 & -\text{Skew}(R_b p_{bp}^{bb}) \\ \mathbf{0}_{3 \times 3} & I_3 \end{pmatrix}, J_{eb} = \begin{pmatrix} R_b & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & R_b \end{pmatrix} J_b$$

2.6 Le problème inverse

Il s'agit d'un problème de la géométrie inverse pour le système complet, c'est-à-dire trouver les paramètres de configuration du système (pose du drone et les deux articulations du bras) à partir de la position et de l'orientation souhaitées de l'effecteur final.

Pour résoudre ce problème, il est nécessaire de dériver six équations algébriques qui lient les paramètres de configuration (q) au résultat attendu (χ_e). Cependant, il est également nécessaire de trouver deux équations supplémentaires, appelées contraintes non holonomiques, qui sont des équations différentielles.

Ces équations supplémentaires sont nécessaires pour résoudre les huit variables de q à partir des six variables de χ_e . Une fois résolues, ces équations sont réduites à deux équations différentielles en fonction de la variable temps (σ_b).

Cependant, il est mentionné que la solution exacte de ces équations est très coûteuse en termes de calcul, ce qui rend sa mise en œuvre pratique impossible.

Enfin, il est supposé que les trajectoires souhaitées pour l'effecteur final sont délimitées, ce qui signifie que l'effecteur final doit rester dans une certaine zone géographique ou à une certaine distance.

La validation de cet algorithme sera vérifiée à la fin de cette section. Les six équations algébriques de cinématique inverse sont dérivées ensuite.

θ_1 et θ_2 sont les angles de rotations pour les articulations du bras robotique .

Définissons la forme générale de la matrice de rotation R_e en fonction des variables de l'effecteur final χ_e , comme suit :

$$R_e = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

En assimilant (2.2) et la matrice R_e , une expression pour les éléments de \mathbf{R}_e , r_{ij} ; $i, j = 1, 2, 3$, peut être trouvée.

Selon la structure de (2.2), la première étape consiste à effectuer l'orientation inverse, suivie de la position inverse. On peut calculer l'orientation inverse en se basant sur la valeur de θ_1 , qui peut être obtenue à partir de l'élément r_{33} .

$$C_\phi C_\theta C\theta_1 - S_\phi C_\theta S\theta_1 = r_{33} \quad (2.7)$$

En réorganisant (2.7) et en trouvant la solution de l'équation résultante pour θ_1 , alors

$$\theta_1 = 2 \arctan \left(\frac{-2b_1 \pm \sqrt{4b_1^2 - 4(-a_1 - r_{33})(a_1 - r_{33})}}{2(-a_1 - r_{33})} \right) \quad (2.8)$$

quand on a : $a_1 = C_\phi C_\theta \cos \theta_1 = -S_\phi C_\theta$

CAS 1 : $\theta_1 \neq 0$ et $\theta_1 \neq \pi$

En inspectant les éléments r_{13} et r_{23} , on peut trouver la valeur de ψ comme

$$r_{13} = a_2 S_\psi + b_2 C_\psi \quad (2.9)$$

$$r_{23} = c_2 S_\psi + d_2 C_\psi \quad (2.10)$$

quand : $a_2 = C_\phi S_{\theta_1} + S_\phi C_{\theta_1}$, $b_2 = C_\phi S_\theta C_{\theta_1} - S_\phi S_\theta S_{\theta_1}$, $c_2 = b^2$, $d_2 = -a^2$
on peut résoudre ces deux équations r_{13} et r_{23} on trouve :

$$S_\psi = \frac{r_{23}b_2 - r_{13}d_2}{-d_2a_2 + b_2c_2} \quad (2.11)$$

$$C_\psi = \frac{r_{23}a_2 - r_{13}c_2}{d_2a_2 - b_2c_2} \quad (2.12)$$

$$\psi = \arctan(S_\psi, C_\psi) \quad (2.13)$$

En inspectant r_{32} et r_{31} , θ_2 peut être trouvé comme

$$r_{32} = a_3 S_{\theta_2} + b_3 C_{\theta_2} \quad (2.14)$$

$$r_{31} = c_3 S_{\theta_2} + d_3 C_{\theta_2} \quad (2.15)$$

quand : $a_3 = C_\phi C_\theta S_{\theta_1} + C_{\theta_1} C_\phi C_{\theta_1}$, $b_3 = -S_\theta$, $c_3 = b_3$, $d_3 = -a_3$

$$S_{\theta_2} = \frac{r_{31}b_3 - r_{32}d_3}{-d_3a_3 + b_3c_3} \quad (2.16)$$

$$C_{\theta_2} = \frac{r_{31}a_3 - r_{32}c_3}{d_3a_3 - b_3c_3} \quad (2.17)$$

$$\theta_2 = \arctan(S_{\theta_2}, C_{\theta_2}) \quad (2.18)$$

CAS 2 : $\theta_1 = 0$

Si $\theta_1 = 0$, alors la somme $\theta_2 + \psi$ peut être déterminée. On peut prendre n'importe quelle valeur pour ψ et obtenir θ_2 . Il y a donc une infinité de solutions. En mettant $\psi = 0$, la valeur de θ_2 peut être déterminée de la manière suivante : En inspectant les éléments r_{11} et r_{12} , θ_2 peut être trouvé comme suit :

$$r_{11} = a_4 S_{\theta_2} + b_4 C_{\theta_2} \quad (2.19)$$

$$r_{12} = c_4 S_{\theta_2} + d_4 C_{\theta_2} \quad (2.20)$$

où : $a_4 = C_\theta$, $b_4 = -S_\phi S_\theta$, $c_4 = -b_4$, $d_4 = a_4$

Résoudre (2.19) et (2.20), puis

$$S_{\theta_2} = \frac{r_{11}b_4 - r_{11}d_4}{-d_4a_4 + b_4c_4} \quad (2.21)$$

$$C_{\theta_2} = \frac{r_{12}a_3 - r_{11}c_4}{d_4a_4 - b_4c_4} \quad (2.22)$$

$$\theta_2 = \arctan(S_{\theta_2}, C_{\theta_2}) \quad (2.23)$$

CAS 3 : $\theta_1 = \pi$

Puisque $S_{\theta_1} = 0$, ce cas est similaire au cas 2. Cependant, $\theta_2 - \psi$ peut être déterminé et en choisissant $\psi = 0$, une expression pour θ_2 peut être déterminée comme suit :

En inspectant les éléments r_{11} et r_{12} , θ_2 peut être trouvé comme

$$r_{11} = a_4S_{\theta_2} + b_4C_{\theta_2} \quad (2.24)$$

$$r_{12} = c_4S_{\theta_2} + d_4C_{\theta_2} \quad (2.25)$$

quand : $a_5 = a_4$, $b_5 = -b_4$, $c_5 = -c_4$, $d_5 = d_4$
 Résoudre (2.24) et (2.25), puis

$$S_{\theta_2} = \frac{r_{12}b_5 - r_{11}d_5}{-d_5a_5 + b_5c_5} \quad (2.26)$$

$$C_{\theta_2} = \frac{r_{12}a_5 - r_{11}c_5}{d_5a_5 - b_5c_5} \quad (2.27)$$

$$\theta_2 = \arctan(S_{\theta_2}, C_{\theta_2}) \quad (2.28)$$

2.6.1 Position du châssis par rapport au repère global :

Comme indiqué ci-dessus, il existe quatre solutions possibles pour le problème de cinématique inverse de rotation, à condition de poser $\psi = 0$, dans les cas 2.1 et 2.2. Lorsque nous commençons à appliquer l'algorithme décrit en temps réel, on sélectionne la solution de θ_1 qui coïncide avec la configuration donnée du quadrirotor système de manipulation. Après cela, nous continuons avec l'une des quatre solutions qui produisent ce θ_1 . Enfin, la position inverse est déterminée à partir de (2.1) comme suit :

$$P_b = P_e - R_b p_{eb}^b \quad (2.29)$$

2.6.2 Drone vu virtuellement comme un manipulateur série :

L'originalité de ce travail est de considérer la plate-forme aérienne comme un manipulateur sériel virtuel. Chaque degré de liberté est généré par une articulation. Les mouvements de translation et de rotation sont générés respectivement par les articulations prismatiques et rotoïdes voir la figure(2.3).

Joint	1	2	3	4	5	6	7
θ_i	θ_1	θ_2	$\frac{\pi}{2}$	0	θ_5	θ_6	θ_7
d_i	0	d_2	d_3	d_4	0	0	0
a_i	0	0	0	l_1	l_2	l_3	l_4
α_i	0	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	0	0	0

TABLE 2.1 – Paramètres D-H du bras robotique virtuel de 7 Dof

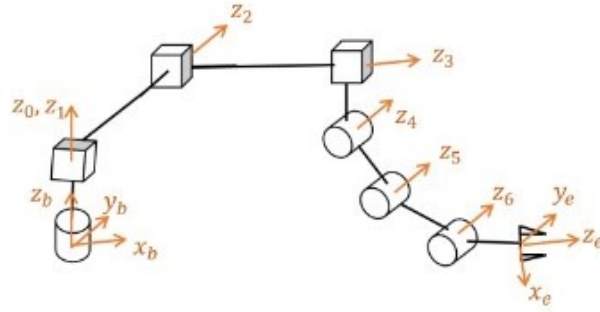


FIGURE 2.3 – drone comme manipulateur en série de 4 Dof (RPPPP) et 3 Dof (RRR) du bras

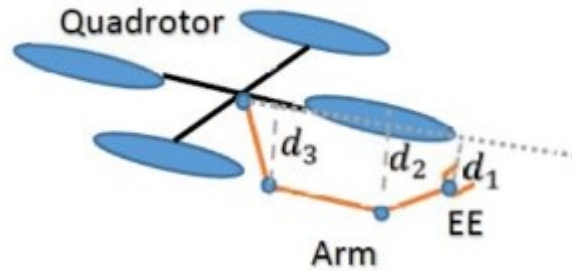


FIGURE 2.4 – Possibilité d’auto-collision

2.6.3 Évitement de l’auto-collision

L’objectif de cette tâche c’est pour éviter les collisions entre le bras et la plate-forme aérienne voir la figure (2.4). et pour faire cette tâche on utilise l’équation (2.30) pour maximiser la distance entre le bras et la plate-forme aérienne

$$\Phi = \frac{1}{2} \sum_{i=1}^3 h_i d_i^2 \quad (2.30)$$

avec h_i est un coefficient de pondération et d_i sont des distances présentées par Fig 2.4. Pour que cette tâche supplémentaire n’influence pas le mouvement de l’effecteur final, elle doit être projetée dans l’espace nul, ou le noyau, de la matrice jacobienne :

$$\dot{\theta} = J^+ \dot{x} + (I - J^+ J)(\alpha \nabla \Phi) \quad (2.31)$$

Dans ce contexte, étant donné que l’objectif est de maximiser les distances d_i , le paramètre α est sélectionné positif. En revanche, pour la minimisation d’un critère, ce coefficient est pris négatif.

Chapitre 3

Asservissement Visuel 2D

3.1 Introduction

comme il était mentionné auparavant, il existe différents types d'asservissement visuel. dans ce chapitre nous décrivons une méthode d'asservissement visuel 2d, est une technique qui permet de contrôler les mouvements d'un système dynamique en utilisant des informations visuelles acquises par une caméra. Dans ce contexte, la boucle de commande utilise des informations visuelles pour générer les mouvements de la caméra et atteindre des objectifs.

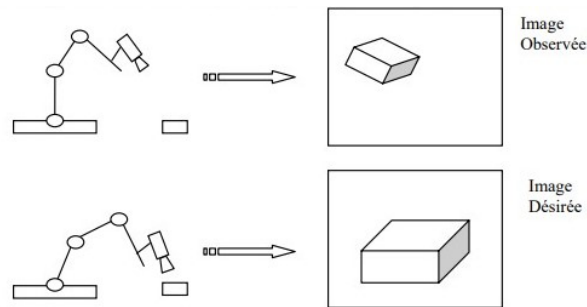


FIGURE 3.1 – Controle du robot de manière a tendre vers l'image désirée

3.1.1 Paramètres Intrinsèques et Extrinsèques

la visualisation par caméra dans le domaine de la vision par ordinateur et de la robotique est un sujet complexe qui englobe plusieurs aspects techniques essentiels.[8] Pour comprendre comment une caméra capture et interprète l'environnement, il est crucial de se familiariser avec ses paramètres intrinsèques et extrinsèques :

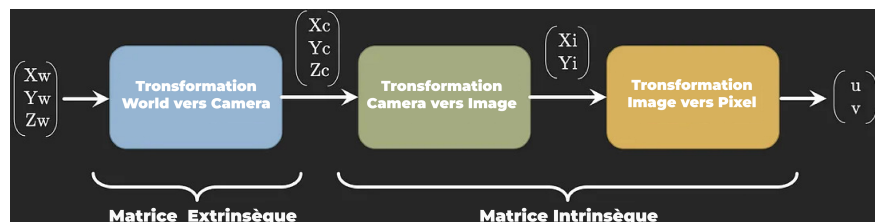


FIGURE 3.2 – Les systèmes de coordonnées couramment utilisés dans Computer Vision

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{f}{\rho_u} & 0 & c_x & 0 \\ 0 & \frac{f}{\rho_v} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Camera Intrinsic Matrix}} \cdot \underbrace{\begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1_{1 \times 1} \end{pmatrix}}_{\text{Camera Extrinsic Matrix}} \cdot \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} \quad (3.1)$$

3.1.2 Paramètres Extrinsèques de la Caméra

Les paramètres extrinsèques, quant à eux, définissent la position et l'orientation de la caméra dans l'espace par rapport à un système de coordonnées mondial. Ils sont représentés par une matrice de transformation qui convertit les points du système de coordonnées mondial au système de coordonnées de la caméra. Cette matrice change si l'emplacement physique ou l'orientation de la caméra est modifié, comme dans le cas d'une caméra montée sur un véhicule en mouvement.

3.1.2.1 Système de coordonnées mondial (3D) :

$[X_w, Y_w, Z_w]$: Il s'agit d'un système de coordonnées cartésiennes de base 3D avec une origine arbitraire. Par exemple un coin précis de la pièce. Un point dans ce système de coordonnées peut être noté $P_w = (X_w, Y_w, Z_w)$.

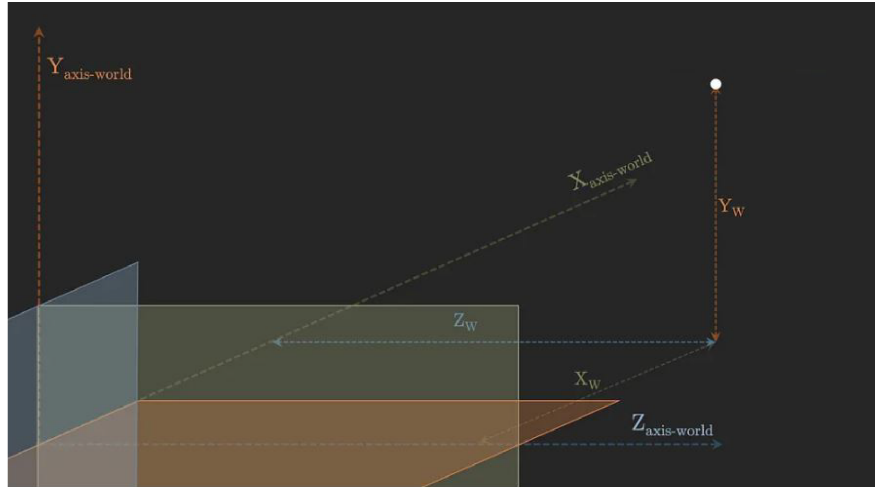


FIGURE 3.3 – Point 3D dans un système de coordonnées mondiales

3.1.2.2 Système de coordonnées objet/caméra (3D) :

$[X_c, Y_c, Z_c]$: C'est le système de coordonnées qui mesure par rapport à l'origine/orientation de l'objet/de la caméra. L'axe z du système de coordonnées de la caméra est généralement orienté vers l'extérieur ou l'intérieur par rapport à l'objectif de la caméra (axe principal de la caméra), comme indiqué dans l'image ci-dessus (axe z orienté vers l'intérieur par rapport à l'objectif de la caméra). Il est envisageable de transférer des coordonnées du système de coordonnées monde vers le système de coordonnées objet (et vice-versa) en utilisant des opérations de rotation et de translation.

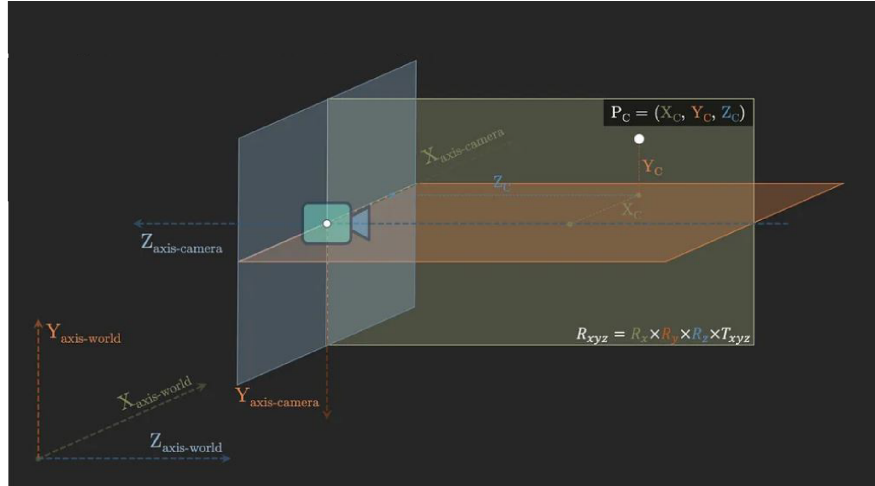


FIGURE 3.4 – Matrice extrinsèque de la caméra

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

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

$$T_{xyz} = \begin{pmatrix} 0 & 0 & 0 & t_x \\ 0 & 0 & 0 & t_y \\ 0 & 0 & 0 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matrice de transformation 4x4 qui convertit les points du système de coordonnées mondial en système de coordonnées de la caméra est connue sous le nom de matrice extrinsèque de la caméra. Si l'emplacement ou l'orientation physique de la caméra est modifié (par exemple, caméra sur une voiture en mouvement), la matrice extrinsèque de la caméra subit des changements.

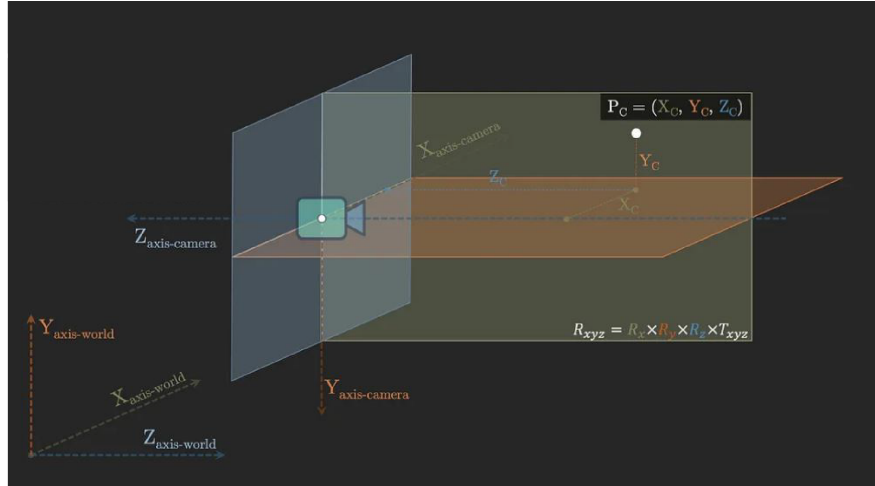


FIGURE 3.5 – Point 3D dans le système de coordonnées de caméra

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1_{1 \times 1} \end{pmatrix}}_{\text{Camera Extrinsic Matrix (4x4)}} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} \quad (3.2)$$

3.1.3 Paramètres Intrinsèques de la Caméra

Les paramètres intrinsèques d'une caméra décrivent ses caractéristiques internes, telles que la longueur focale, l'ouverture, le champ de vision et la résolution. Ces paramètres sont fondamentaux pour comprendre comment une caméra convertit les points tridimensionnels du monde réel en une image bidimensionnelle. La matrice intrinsèque de la caméra, qui est une transformation de la coordonnée de la caméra au système de coordonnées des pixels, est définie par ces paramètres et est unique à chaque caméra.

3.1.3.1 Système de coordonnées d'image (2D) [Modèle sténopé] :

$[X_i, Y_i]$: un système de coordonnées 2D dans lequel les points 3D du système de coordonnées de la caméra sont projetés sur un plan 2D (généralement normal à l'axe z du système de coordonnées de la caméra représenté par un plan jaune dans les figures ci-dessous) d'un appareil photo avec un modèle sténopé.

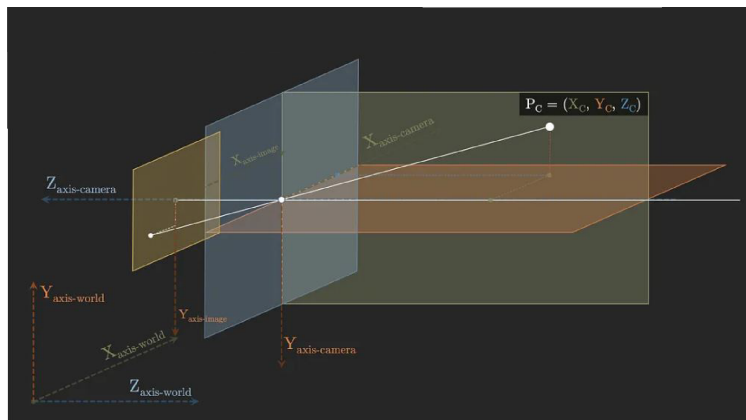
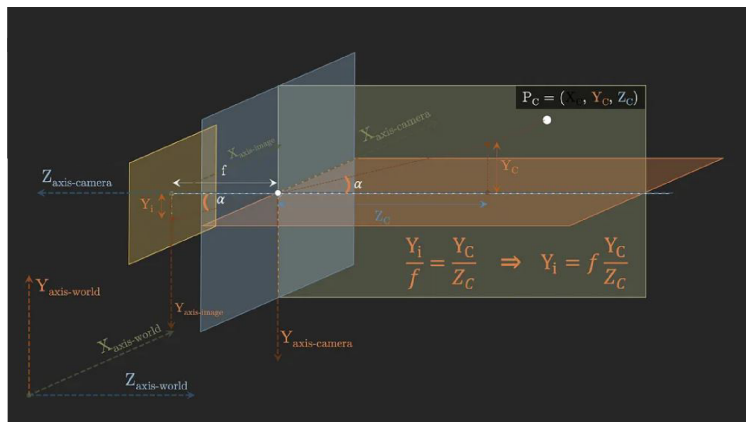
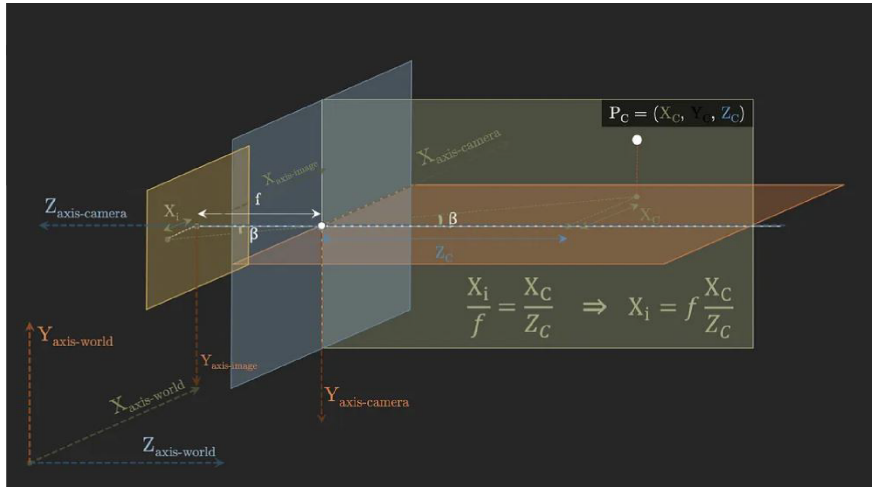
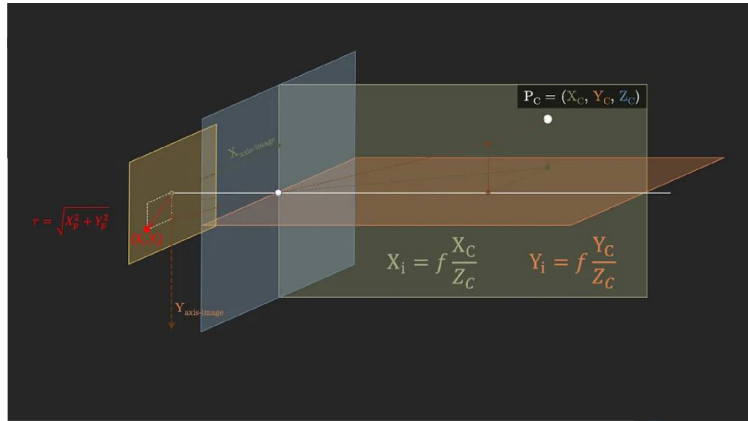


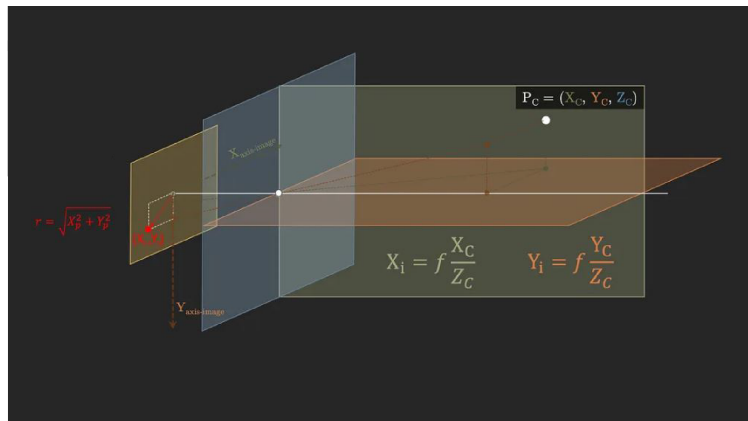
FIGURE 3.6 – Plan d'image 2d de la caméra

Les rayons traversent le centre de l'ouverture de la caméra et sont projetés sur le plan 2D à l'extrémité opposée. La caméra capture le plan 2D sous forme d'images. Il s'agit d'une transformation avec perte, ce qui signifie que la projection des points du système de coordonnées de la caméra vers le plan 2D ne peut pas être inversée (les informations de profondeur sont perdues). C'est pourquoi, en regardant une image capturée par une caméra, nous ne pouvons pas déterminer la profondeur réelle. des points). Les coordonnées X et Y des points sont projetées sur le plan 2D. Le plan 2D est à une distance f (focale) de la caméra. La projection X_i , Y_i peut être trouvée par la loi des triangles similaires (le rayon entrant et sortant du centre de la caméra a le même angle avec les axes x et y, respectivement alpha et bêta).





Ainsi sous forme matricielle, nous avons la matrice de transformation suivante du système de coordonnées de la caméra au système de coordonnées de l'image.



$$\begin{pmatrix} X_i \\ Y_i \\ Z_c \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \quad (3.3)$$

Cette transformation (de la caméra au système de coordonnées de l'image) est la première partie de la matrice intrinsèque de la caméra .

3.1.3.2 Système de coordonnées de pixels (2D) :

[u, v] : Ceci représente les valeurs entières en discrétisant les points dans le système de coordonnées de l'image. Les coordonnées en pixels d'une image sont des valeurs discrètes dans une plage qui peut être obtenue en divisant les coordonnées de l'image par la largeur et la hauteur des pixels (paramètres de la caméra — unités : mètre/pixel).

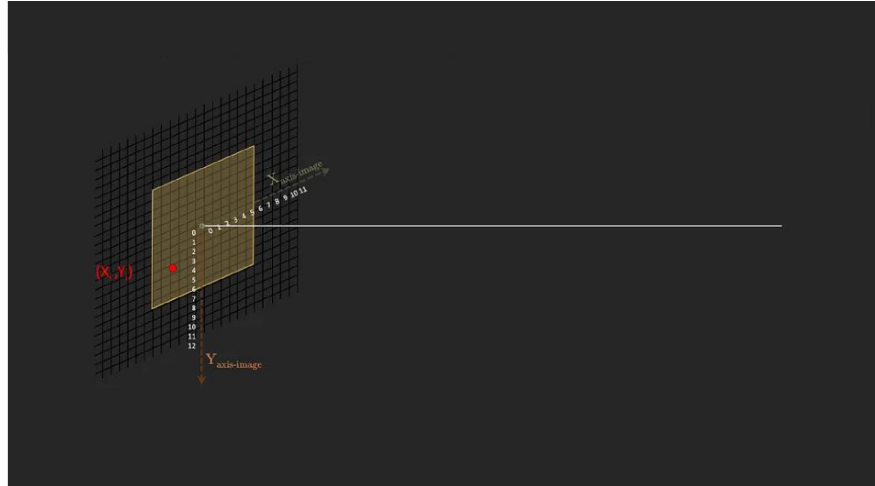


FIGURE 3.7 – Discrétisation du plan d'image 2D de la caméra

$$X_i = f \frac{X_c}{Z_c} \quad (3.4)$$

$$Y_i = f \frac{Y_c}{Z_c} \quad (3.5)$$

Chaque pixel mesure $\rho_u \times \rho_v$ mètres de largeur (mètre/pixel)

$$X_i = \frac{1}{\rho_u} f \frac{X_c}{Z_c} \quad (3.6)$$

$$Y_i = \frac{1}{\rho_v} f \frac{Y_c}{Z_c} \quad (3.7)$$

Le système de coordonnées de pixels a l'origine dans le coin supérieur gauche, donc un opérateur de traduction (c_x, c_y) est également requis parallèlement à la discrétisation.

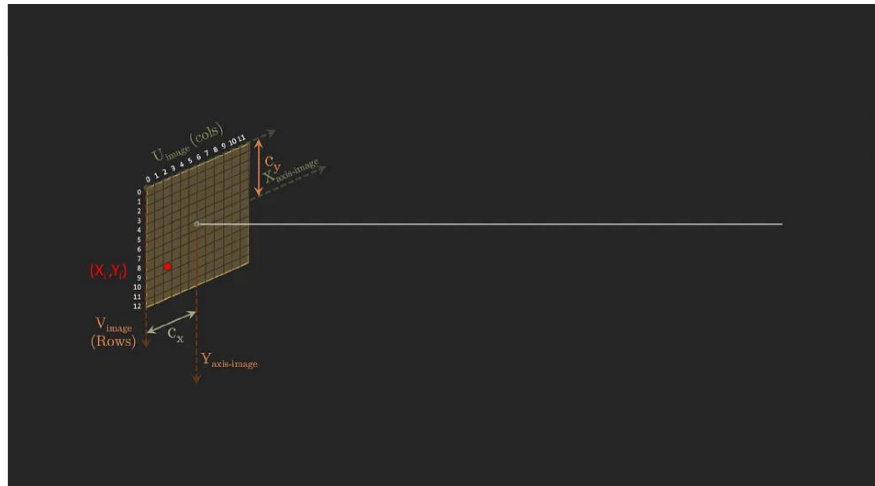


FIGURE 3.8 – coordonnées pixels

$$x_i = \frac{1}{\rho_u} f \frac{X_c}{Z_c} \quad (3.8)$$

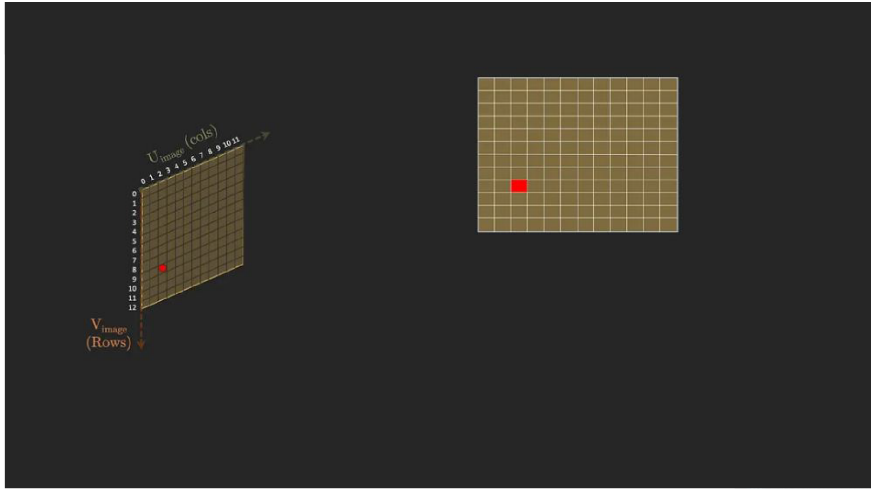
$$y_i = \frac{1}{\rho_v} f \frac{Y_c}{Z_c} \quad (3.9)$$

Le système de coordonnées des pixels a son origine dans le coin supérieur gauche de l'image.

$$u = \frac{1}{\rho_u} f \frac{X_c}{Z_c} + c_x \quad (3.10)$$

$$v = \frac{1}{\rho_v} f \frac{Y_c}{Z_c} + c_y \quad (3.11)$$

La transformation complète du système de coordonnées d'image en système de coordonnées de pixels peut être affichée sous forme matricielle comme ci-dessous



$$u = \frac{1}{\rho_u} f \frac{X_c}{Z_c} + c_x \quad (3.12)$$

$$v = \frac{1}{\rho_v} f \frac{Y_c}{Z_c} + c_y \quad (3.13)$$

Dans l'exemple ci-dessus, $(u, v) = (2, 9)$ et la taille de l'image est 12×11

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & c_x \\ 0 & \frac{1}{\rho_v} & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \quad (3.14)$$

Il arrive parfois que le plan de l'image 2D ne soit pas un rectangle, mais qu'il soit plutôt incliné, ce qui signifie que l'angle entre les axes X et Y n'est pas de 90 degrés. Pour passer du plan rectangulaire au plan asymétrique, il est nécessaire d'effectuer une autre transformation avant de convertir l'image en repère pixel. Dans le cas où l'angle entre les axes x et y est θ , il est possible de trouver la transformation qui transforme les points du plan rectangulaire idéal en plan asymétrique, comme indiqué ci-dessous.

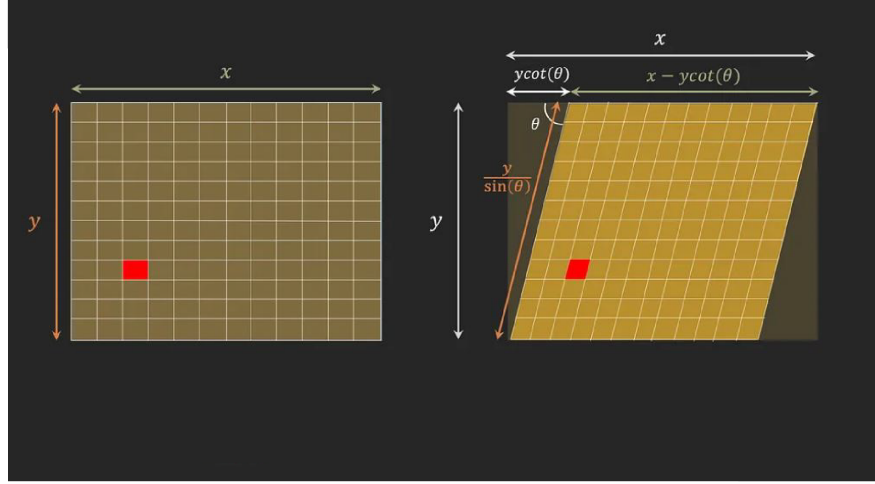


FIGURE 3.9 – Plan d'image asymétrique

Ces deux matrices de transformation, c'est-à-dire la transformation du système de coordonnées d'image rectangulaire en système de coordonnées d'image asymétrique et du système de coordonnées d'image asymétrique en système de coordonnées de pixels, constituent la deuxième partie de la matrice intrinsèque de la caméra.

La matrice extrinsèque de la caméra est obtenue en combinant les trois matrices de transformation, comme spécifié ci-dessous.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \frac{f}{\rho_u} & 0 & cx & 0 \\ 0 & \frac{f}{\rho_v} & cy & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -\cos(\theta) & 0 & 0 \\ 0 & 0 & \frac{1}{\sin(\theta)} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1_{1 \times 1} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} \quad (3.15)$$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \frac{f}{\rho_u} & -\frac{f}{\rho_v} \cos(\theta) & cx & 0 \\ 0 & \frac{f}{\rho_v} \sin(\theta) & cy & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1_{1 \times 1} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} \quad (3.16)$$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha & -\alpha \cos(\theta) & cx & 0 \\ 0 & \beta & cy & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Camera Intrinsic Matrix}} \underbrace{\begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1_{1 \times 1} \end{pmatrix}}_{\text{Camera Extrinsic Matrix}} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} \quad (3.17)$$

3.1.4 Calibration de la Caméra et Correction de la Distorsion

La calibration de la caméra est le processus par lequel on estime les paramètres intrinsèques et extrinsèques et corrige les distorsions optiques introduites par l'objectif de la caméra. Les distorsions radiales et tangentielles sont les deux principaux types de distorsion qui affectent la qualité de l'image. La distorsion radiale fait apparaître des lignes droites comme courbées, tandis que la distorsion tangentielle est due à un alignement imparfait de l'objectif par rapport au plan d'image. La calibration est généralement réalisée en utilisant des images contenant un motif de calibration connu, permettant ainsi d'estimer les coefficients de distorsion et d'ajuster les images en conséquence.

En résumé, une compréhension approfondie de ces paramètres est indispensable pour toute application impliquant la visualisation et l'interprétation précises de l'environnement par une caméra. Cela est particulièrement vrai dans les domaines de la robotique et de l'automatisation, où la précision de la perception visuelle est cruciale pour le succès des opérations autonomes.

3.2 Modèle du camera (Pinhol)

Le modèle de caméra sténopé, ou "pinhole", est un concept fondamental en vision par ordinateur qui permet de comprendre comment une image est formée sans l'utilisation de lentilles.

3.2.1 Principe du Modèle Sténopé

Le modèle sténopé repose sur l'idée qu'une image peut être formée en laissant passer la lumière à travers un petit trou. Dans ce modèle, la lumière provenant d'un point de l'espace traverse le sténopé et se projette sur un plan, créant ainsi une image inversée. Ce processus est décrit par la géométrie de projection perspective.

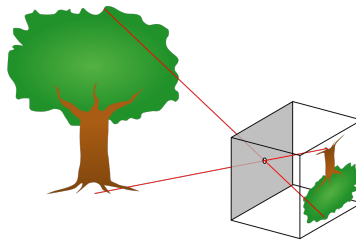


FIGURE 3.10 – Modèle du camera Pinhole

3.2.2 Équations du Modèle Sténopé

Dans le modèle sténopé, la relation entre un point dans l'espace tridimensionnel et sa projection sur le plan d'image est donnée par les équations suivantes :

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}$$

où (X, Y, Z) sont les coordonnées d'un point dans l'espace, (x, y) sont les coordonnées de ce point sur le plan d'image, et (f) est la distance focale du sténopé au plan d'image.

3.2.3 Matrice de Projection du Modèle Sténopé

La matrice de projection du modèle sténopé, qui transforme les coordonnées spatiales en coordonnées d'image, est représentée comme suit :

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Cette matrice prend en compte la distance focale (f) et les coordonnées du point dans l'espace pour calculer la position du point projeté sur le plan d'image.

3.3 Control visuel basé sur l'image

Asservissement visuel basé sur l'image (IBVS) est un contrôleur essentiellement basé sur des données visuelles extraites d'une caméra. Cette approche est issue de l'approche de rétroaction orientée tâche dans le but de contrôler les manipulateurs de robots. C'est ce qu'on appelle Eye-in-hand ou Eye-to-hand, en fonction de l'emplacement de la caméra par rapport au robot. Dans Vision Servoing, la stratégie de contrôle est basée sur la minimisation de la fonction de coût. Un coût de fonction de tâche nous pouvons prendre :

$$e_i(t) = s_i(t) - s_i^* \quad (3.18)$$

où $s_i(t)$ et s_i^* sont les mesures d'image et la configuration souhaitée dans le plan image, respectivement. Ces mesures sont définies en pixels $s_i = (p_{u_i}, p_{v_i})^T$ et sont utilisées pour déterminer le point correspondant dans le cadre de l'image, noté par $p_i = (x_i, y_i)$. La relation entre ces points est définie par l'application de la matrice intrinsèque de la caméra :

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_{xx} & f_{xx}\alpha & x_0 \\ 0 & f_{yy} & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{x_i} \\ p_{y_i} \\ 1 \end{bmatrix} \quad (3.19)$$

Dans l'approche "Eye-in-hand" adoptée dans ce travail, la relation entre la mesure de projection dans le cadre de l'image p_i et le point cible $P_i = (X_{c_i}, Y_{c_i}, Z_{c_i})^T$ dans le cadre caméra est dérivée du modèle de caméra Pinhole :

$$\begin{cases} x_i = \frac{X_{c_i}}{Z_{c_i}} \\ y_i = \frac{Y_{c_i}}{Z_{c_i}} \end{cases} \quad (3.20)$$

Lorsque la caméra se déplace avec une vitesse $V_c = [v_c, \omega_c]^T = [u_x, u_y, u_z, \omega_x, \omega_y, \omega_z]^T$, la dynamique de P_i est donnée en appliquant la formule suivante :

$$\dot{P}_i = v_c + \omega_c \times P_i \quad (3.21)$$

où \times est le produit vectoriel. La dynamique du point dans le plan image est tirée de la dérivée de l'Eq.(3.20) :

$$\begin{cases} \dot{x}_i = \frac{\dot{X}_{c_i} Z_{c_i} - X_{c_i} \dot{Z}_{c_i}}{Z_{c_i}^2} \\ \dot{y}_i = \frac{\dot{Y}_{c_i} Z_{c_i} - Y_{c_i} \dot{Z}_{c_i}}{Z_{c_i}^2} \end{cases} \quad (3.22)$$

En utilisant les Eqs. (3.21) et (3.22), les résultats suivants sont obtenus :

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = L \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \quad (3.23)$$

où L est la matrice d'interaction :

$$L = \begin{bmatrix} \frac{1}{Z_{c_i}} & 0 & -x_i/Z_{c_i} & -x_i y_i & (1 + x_i^2) & -y_i \\ 0 & \frac{1}{Z_{c_i}} & -(y_i/Z_{c_i}) & -(1 + y_i^2) & x_i y_i & x_i \end{bmatrix} \quad (3.24)$$

Il est clair que, pour calculer tous les composants de la vitesse de la caméra, nous avons besoin de plus de trois points avec des profondeurs différentes. Si les points caractéristiques à suivre sont coplanaires, nous avons besoin d'au moins quatre points.

3.3.1 Génération de trajectoire basée sur le suivi visuel

Le composant clé du contrôleur d'asservissement visuel est la matrice d'interaction. L'approche commune considérée dans les problèmes de contrôle IBVS est basée sur le choix d'une décroissance exponentielle de la fonction de coût :

$$\dot{e}_i = -\delta e_i(t) \quad (3.25)$$

où le taux de décroissance δ est une constante positive. En supposant que la configuration désirée est constante et en utilisant l'Eq.(3.23), la dérivée de l'erreur est :

$$\dot{s}(t) = -\delta e(t) = LV_c \quad (3.26)$$

Cela conduit à la vitesse de la caméra :

$$V_c = -\delta L^+ e(t) \quad (3.27)$$

où $L^+ \in \mathbb{R}^{6 \times 2k}$ est une pseudo-inverse avec $k \geq 4$ comme le nombre de points d'image à suivre. Si une seule caméra est utilisée, il n'est pas possible de déduire exactement la profondeur Z_{c_i} . Par conséquent, la matrice d'interaction n'est pas bien définie. Pour contourner ce problème, cette matrice peut être estimée ou approximée en fixant Z_c à sa valeur de la configuration désirée.

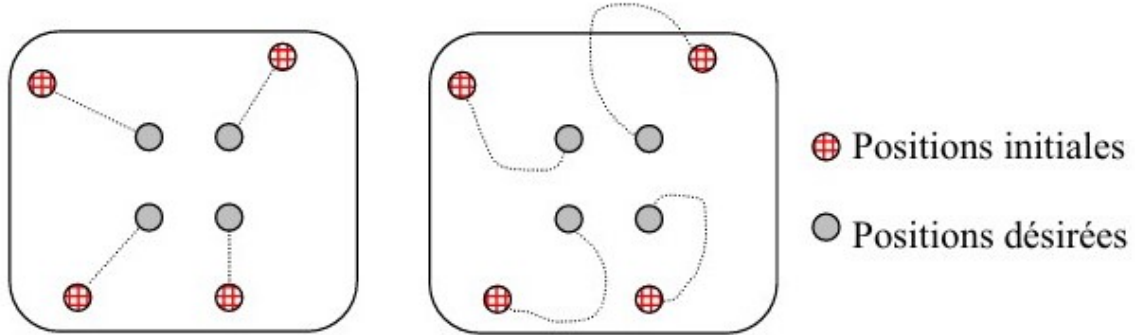


FIGURE 3.11 – Trajectoires des primitives visuelles dans l'image

3.3.2 Schema de commande visuel

Dans un asservissement 2D, les primitives visuelles dans l'image représentent le signal de référence. Une primitive est une forme géométrique élémentaire telle qu'un point, un segment de droite ou une portion d'ellipse. Son rôle est de représenter la projection d'un objet dans le plan de l'image.

Les primitives constituées de points sont utilisées dans la plupart des travaux liés à l'asservissement 2D. Il est possible que ces points soient situés à l'intersection de segments dans l'image, ou bien qu'ils soient extraits du centre de gravité de la projection de disques. En réalité, les cibles les plus fréquemment rencontrées sont simplement formées de plusieurs disques coplanaires. Ces cibles présentent l'avantage de demander un temps de traitement d'image faible.

Il n'est pas indispensable de connaître un modèle de la cible pour bénéficier de l'asservissement 2D, ce qui en fait l'un des principaux avantages. Dans le cas où les primitives sont des points, si on dispose d'une estimation de la profondeur de la cible (c'est à dire son éloignement par rapport à la caméra suivant l'axe optique), les données de l'image sont suffisantes pour déterminer le déplacement de la cible par rapport à la caméra.

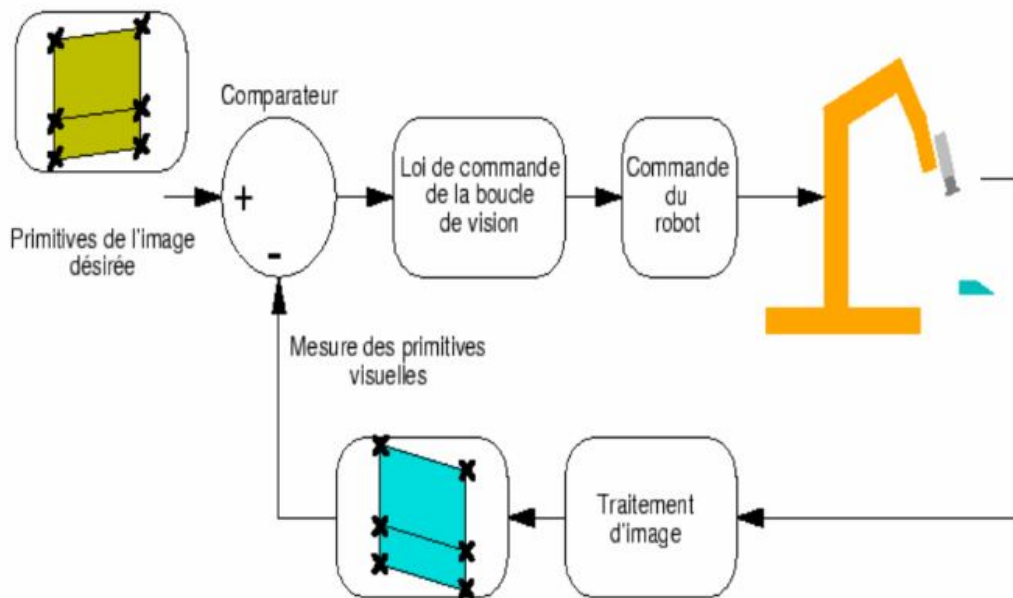


FIGURE 3.12 – Structure d'un asservissement visuel 2D.

Chapitre 4

Simulation

4.1 Introduction

La simulation est un outil essentiel pour le développement et la validation de systèmes robotiques. Dans le contexte de notre projet, qui implique un drone quadratique relié par un bras et équipé d'une caméra, elle joue un rôle crucial. Elle permet de tester et d'optimiser nos algorithmes de contrôle sans avoir besoin d'un véritable bras aérien physique. Dans ce chapitre, nous explorerons l'utilisation de plusieurs logiciels de simulation utilisés dans notre recherche : Gazebo, ROS et CoppeliaSim, un simulateur de robotique puissant et polyvalent.

4.2 Première simulateur : Gazebo

DUne fois le modèle cinématique inverse et la commande visuelle développés, il est essentiel de simuler l'ensemble des approches dans un simulateur physique. Nous avons d'abord opté pour le simulateur Gazebo en raison de son large usage et de sa puissance dans la simulation des systèmes physiques. Un autre atout majeur de ce logiciel est sa compatibilité avec ROS, l'interface incontournable pour les systèmes robotiques.

4.2.1 Gazebo

Gazebo est un simulateur open-source qui permet de créer des environnements 3D réalistes et d'y intégrer des robots, capteurs et objets dynamiques. Utilisé dans la recherche et l'industrie, il simule avec précision les lois physiques telles que la gravité, les collisions, et la friction, fournissant ainsi un environnement d'expérimentation pour tester des algorithmes de contrôle ou des systèmes robotiques avant leur mise en œuvre réelle.

L'un des principaux avantages de Gazebo est sa capacité à intégrer des capteurs et à simuler des environnements riches en détail avec des interactions physiques précises. Il permet également de générer des scénarios complexes, y compris des terrains accidentés et des obstacles mobiles, rendant les tests de simulation particulièrement fiables. Cela en fait un outil de choix pour les chercheurs et ingénieurs en robotique cherchant à valider des algorithmes de contrôle dans des conditions réalistes sans avoir besoin de matériel physique.

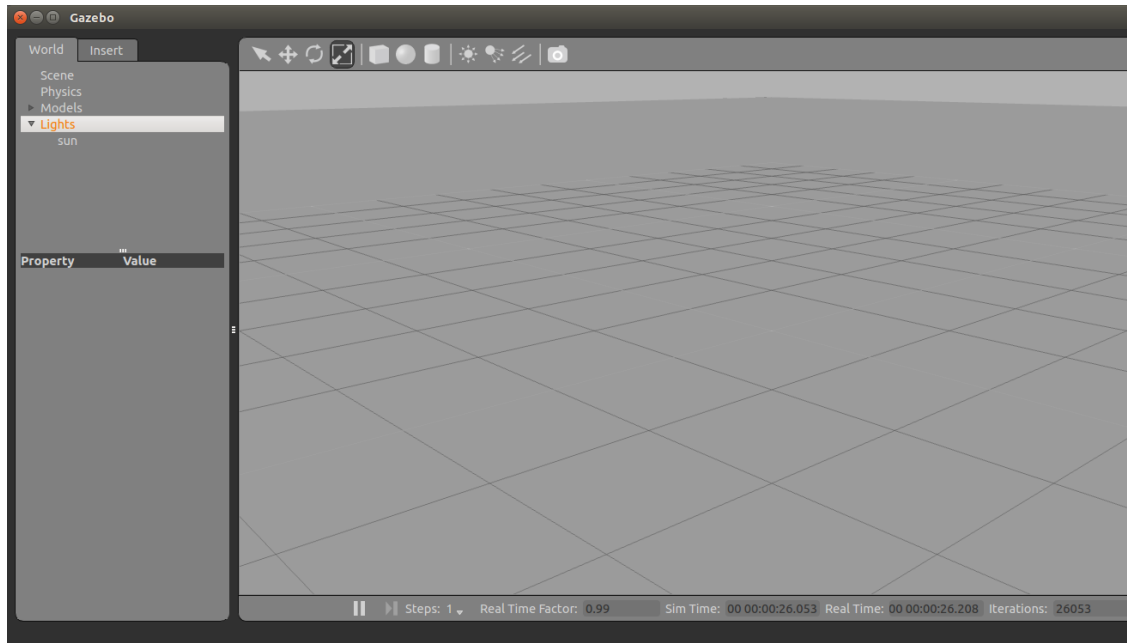


FIGURE 4.1 – Interface de GAZEBO.

Dans notre projet, Gazebo aurait dû permettre de simuler le comportement du drone et de son bras articulé dans un environnement virtuel. Toutefois, une limitation majeure est apparue rapidement : le modèle de drone que nous souhaitions utiliser n'était pas disponible dans la bibliothèque de modèles de Gazebo. Cette absence nous a conduits à envisager la création manuelle de notre propre modèle, une tâche que nous avons tenté de réaliser avec d'autres outils de modélisation 3D comme Blender et SketchUp. Cependant, cette démarche s'est révélée particulièrement difficile et chronophage, notamment en ce qui concerne l'ajout de propriétés dynamiques et de joints pour chaque composant du drone. Le processus d'intégration de ces modèles dans Gazebo a nécessité des ajustements fins pour garantir des interactions physiques réalistes, mais a pris beaucoup plus de temps que prévu.



FIGURE 4.2 – Icon du blender

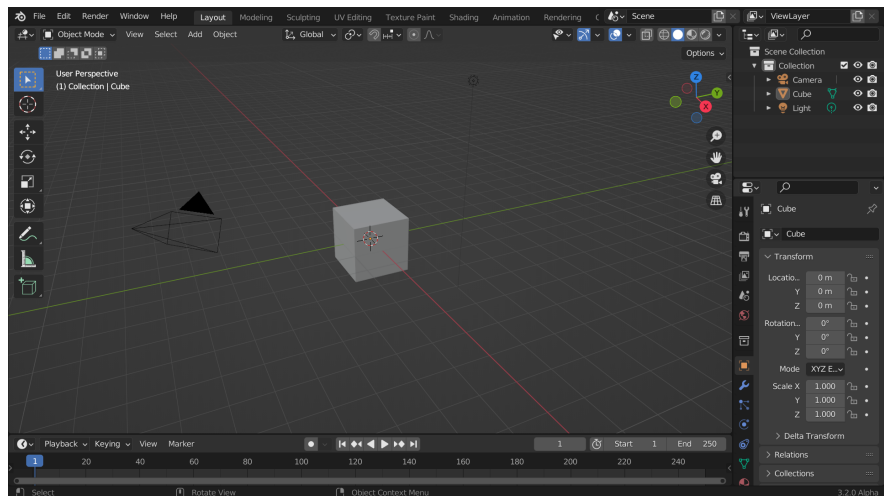


FIGURE 4.3 – Interface du Blender



FIGURE 4.4 – Icon du Sketchup

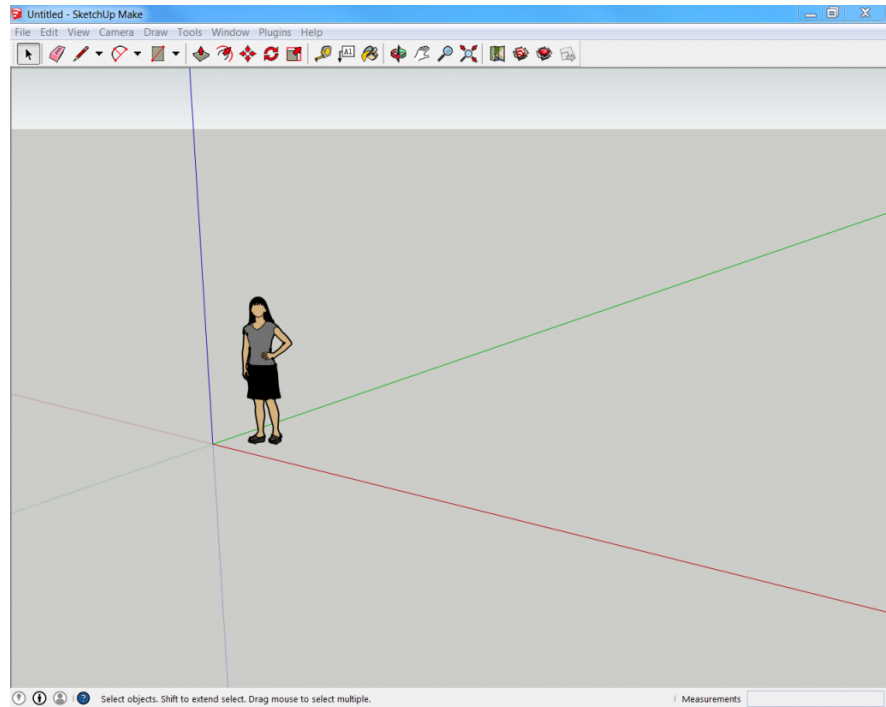


FIGURE 4.5 – Interface du SketchUp

4.2.2 ROS (Robot Operating System) :

ROS (Robot Operating System) est un cadre logiciel conçu pour la programmation et le contrôle de robots. Plutôt qu'un véritable système d'exploitation, ROS fonctionne comme un middleware, permettant à différents composants logiciels (appelés "nœuds") de communiquer entre eux de manière asynchrone ou synchronisée via des protocoles de messages standardisés. Cela en fait un outil extrêmement modulaire et flexible pour les systèmes robotiques complexes, facilitant l'intégration de capteurs, de modules de contrôle et d'autres composants.

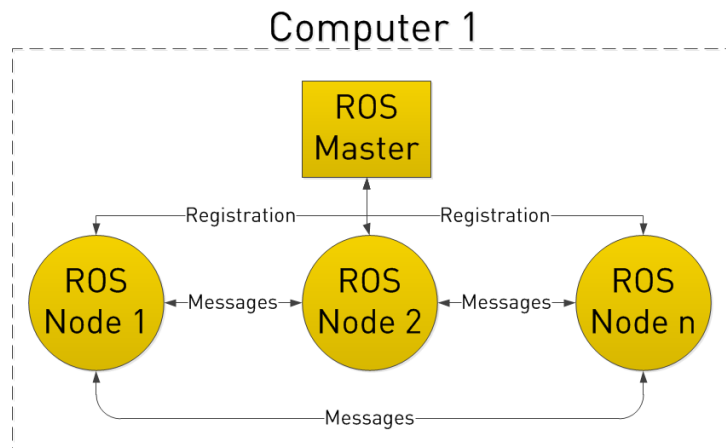


FIGURE 4.6 – Explication de systeme ROS

L'un des atouts majeurs de ROS est sa vaste collection de bibliothèques et de packages dédiés à divers types de robots, qu'il s'agisse de bras robotiques, de drones, ou de véhicules autonomes. Ces packages incluent des implémentations d'algorithmes avancés de navigation, de perception, de manipulation et de contrôle, ce qui permet de gagner un temps précieux dans le développement de solutions robotiques.



FIGURE 4.7 – Icon du ROS

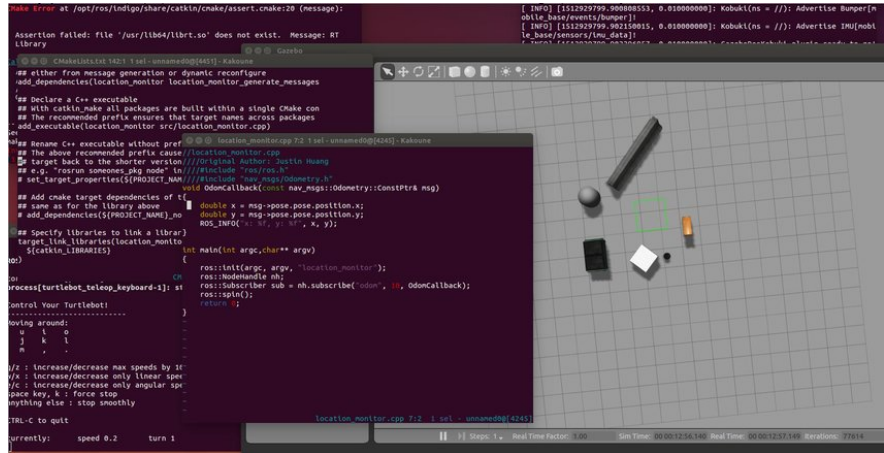


FIGURE 4.8 – Interface du ROS

Dans le cadre de notre simulation, ROS a joué un rôle central pour coordonner les différents composants du système, en particulier la caméra embarquée sur le drone et le bras articulé. Cependant, comme avec Gazebo, nous avons rencontré des défis significatifs lors de l'intégration du modèle du drone. En l'absence d'un modèle compatible dans les bibliothèques ROS, il nous a été nécessaire de développer ce modèle à partir de zéro, une tâche qui s'est avérée laborieuse, d'autant plus que l'intégration dynamique et l'ajustement des joints mécaniques exigent une fine expertise et un temps de développement conséquent.

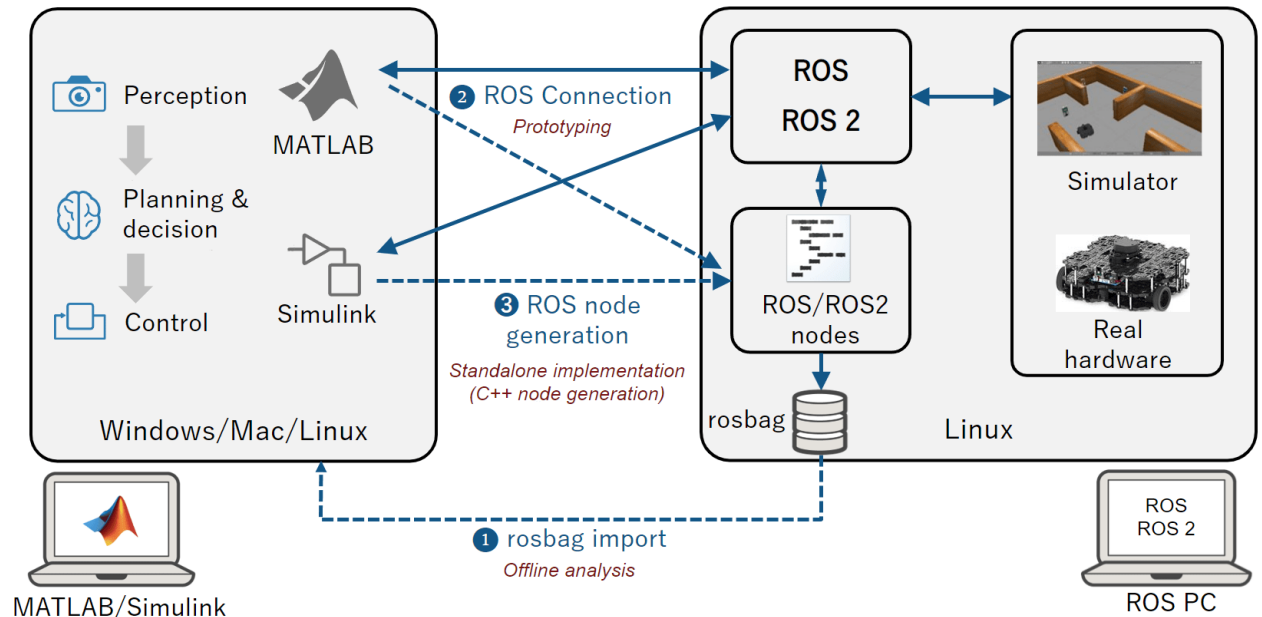


FIGURE 4.9 – Explication de communication entre ROS et Gazebo

4.2.3 Difficultés Rencontrées :

L'absence d'un modèle préexistant du drone nous a poussé à recourir à des outils comme Blender et SketchUp pour créer manuellement les différents composants du drone. Blender, un logiciel de modélisation 3D, offre des fonctionnalités avancées pour concevoir des modèles complexes, tandis que SketchUp est reconnu pour son accessibilité et sa simplicité dans la création de structures géométriques de base.

Cependant, la création du modèle s'est avérée chronophage, en particulier lorsqu'il s'agissait d'ajouter une étude dynamique réaliste aux joints de chaque pièce du drone. Cette complexité nous a ralenti considérablement, et la tâche d'intégrer ces modèles dans Gazebo avec des propriétés physiques et cinématiques appropriées a nécessité bien plus de temps que prévu. L'ajustement des joints et des paramètres dynamiques pour assurer un comportement cohérent du drone dans un environnement simulé a été particulièrement complexe.

Ainsi, bien que Gazebo et ROS soient adaptés à la simulation de systèmes robotiques, les limitations en termes de modèles préexistants et la nécessité de créer manuellement des composants spécifiques ont été un frein majeur dans cette approche.

4.3 Deuxième simulateur : CoppeliaSim

Pour contourner les limitations rencontrées avec Gazebo et ROS, nous avons décidé de passer à une autre approche de simulation en utilisant CoppeliaSim (anciennement V-REP) conjointement avec MATLAB. Cette combinaison nous a permis d'intégrer un modèle de drone avec un bras robotique et d'implanter une caméra sur l'effecteur final du bras pour traiter et contrôler le mécanisme du robot.

4.3.1 CoppeliaSim (V-REP) :

CoppeliaSim, anciennement appelé V-REP, est un simulateur de robots 3D polyvalent utilisé dans de nombreux domaines, tels que la robotique, la recherche, et l'éducation. Il permet une simulation temps réel avec de multiples options de contrôle, allant des scripts intégrés aux interfaces avec des logiciels externes, tels que MATLAB, ROS, et Python.



FIGURE 4.10 – Icon du CoppeliaSim

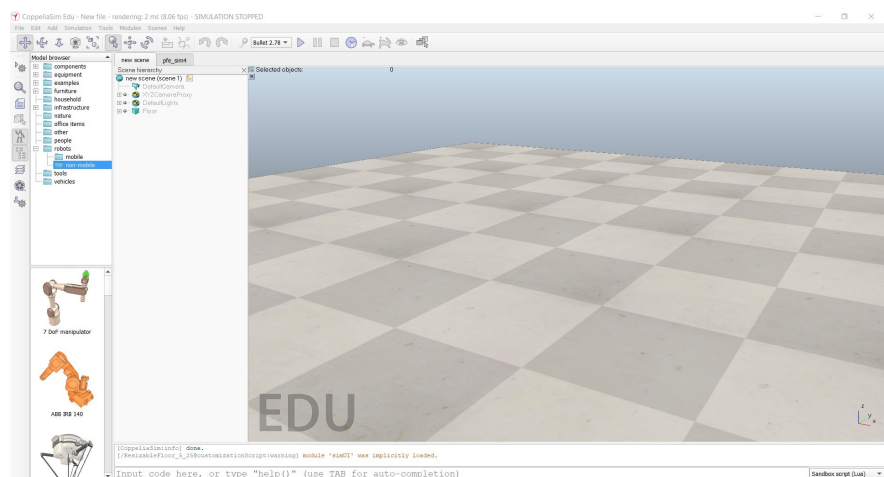


FIGURE 4.11 – Interface du CoppeliaSim

L'un des principaux avantages de CoppeliaSim est sa vaste bibliothèque de modèles de robots et de composants mécaniques, ce qui nous a permis de trouver rapidement un modèle de drone avec un bras robotique compatible avec notre projet. Le simulateur intègre également des mécanismes de simulation physique très réalistes (gestion des collisions, des dynamiques des corps rigides, etc.), essentiels pour tester des scénarios complexes.

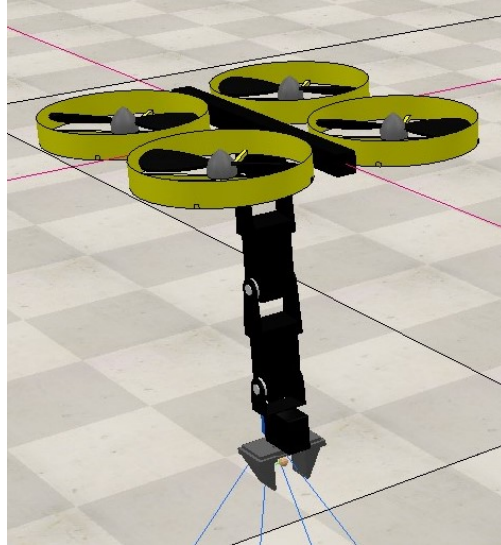


FIGURE 4.12 – Notre Robot sur CoppeliaSim

Dans notre simulation, la caméra a été placée sur l'effecteur final du bras robotique, ce qui nous a permis de capturer des images et d'analyser les données pour la navigation et le contrôle en temps réel. Cette configuration a facilité l'étude des interactions entre le drone, le bras et les objets environnants dans des environnements simulés réalistes.

Les avantages de *CoppeliaSim* incluent :

- Une interface graphique intuitive facilitant la création et la manipulation des modèles.
- La possibilité de contrôler des robots via des scripts *LUA* ou de se connecter à des logiciels externes comme *MATLAB* pour un contrôle plus avancé.
- Une grande bibliothèque de robots et composants prêts à l'emploi, comme le modèle de drone avec bras robotique utilisé dans ce projet.

4.3.1.1 Configuration de la caméra :

Dans la simulation réalisée avec CoppeliaSim, la configuration de la caméra a été une étape essentielle pour capturer les données visuelles nécessaires à l'analyse et au contrôle du système robotique. Nous avons utilisé un capteur de vision, comme illustré dans les captures d'écran ci-jointes, pour suivre les objets dans l'environnement simulé. La caméra a été positionnée sur l'effecteur final du bras robotique du drone, permettant ainsi d'obtenir des vues dynamiques et adaptées aux mouvements du bras.

Le capteur de vision a été configuré avec une résolution de 256x256 pixels, ajustable selon les besoins de la simulation. Le champ de vision a été défini à 60 degrés pour offrir une perspective optimale sur la scène simulée. L'onglet des propriétés de l'objet permet d'ajuster divers paramètres de la caméra, tels que le plan de découpage (near/far), le mode de rendu et la taille de l'image perçue. Le mode de rendu choisi était "Legacy OpenGL", garantissant une performance optimale pendant la simulation.

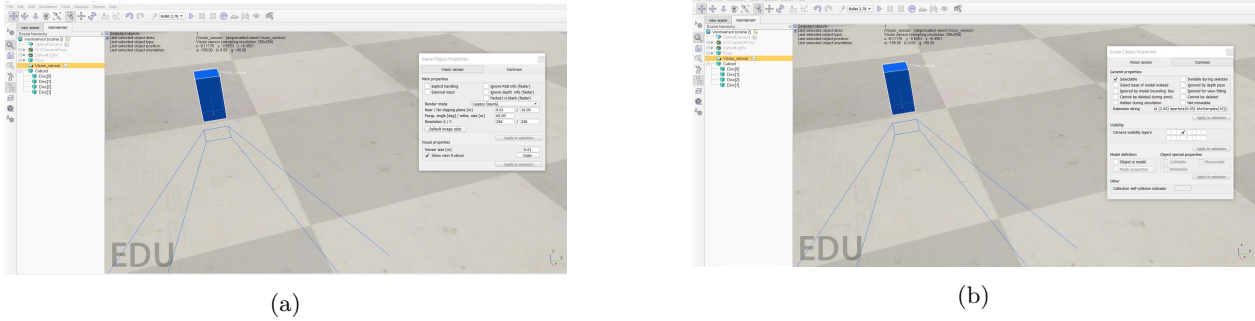


FIGURE 4.13 – le réglage des paramètre de la camera

Nous avons également activé le frustum visuel, qui permet de visualiser le champ de vision de la caméra directement dans l'interface. Cette fonctionnalité a grandement facilité l'ajustement de la position et de l'orientation de la caméra, assurant ainsi une couverture complète des objets d'intérêt pendant les tests. La caméra, avec son capteur de vision, a été configurée pour ignorer certaines informations, afin de minimiser la charge de calcul et optimiser les performances de la simulation.

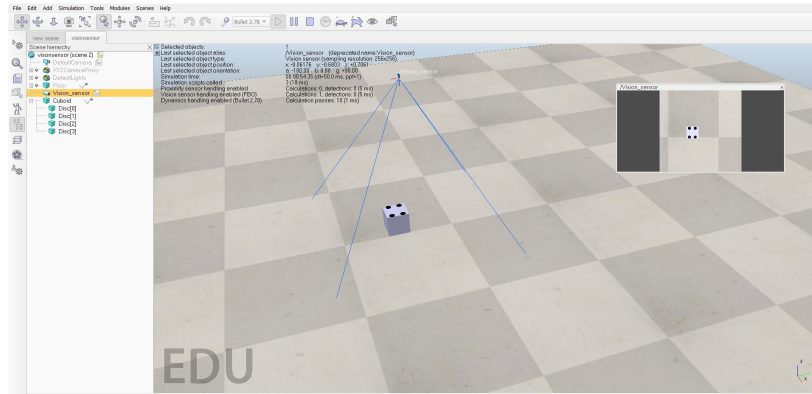


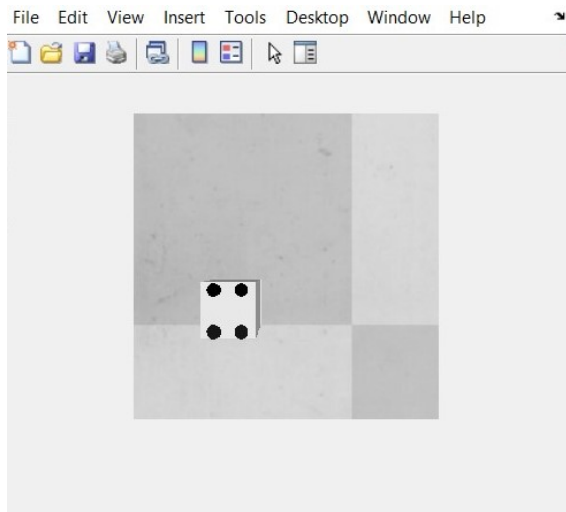
FIGURE 4.14 – la vision de la camera sur CoppeliaSim

Grâce à cette configuration, la caméra a pu capturer et transmettre les informations visuelles en temps réel à MATLAB, où elles ont été traitées pour piloter les actions du drone et du bras robotique.

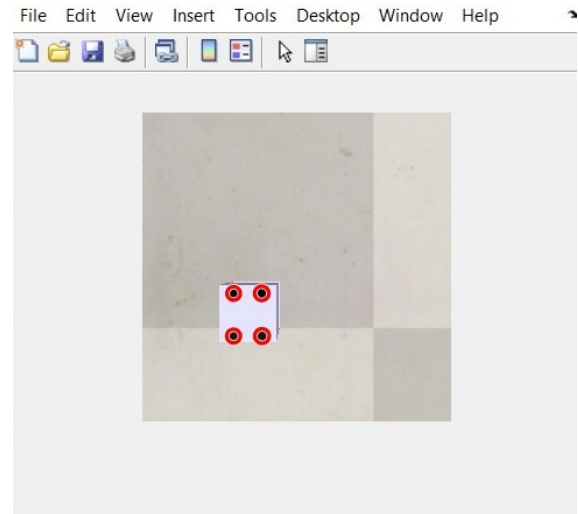
Lorsqu'un objet est détecté, le capteur de vision se concentre sur les cercles présents sur l'objet. Ces cercles sont utilisés comme points de référence pour positionner l'objet avec précision.

L'image capturée par la caméra est traitée en temps réel pour identifier les cercles à l'aide d'algorithmes de traitement d'image. Une fois les cercles localisés, des calculs sont effectués pour déterminer la position exacte de l'objet par rapport à l'axe du drone. Cela permet de manipuler l'objet de manière optimale, en tenant compte de la position relative des cercles. Le bras robotique ajuste ensuite son orientation pour saisir l'objet et le placer dans une position définie en fonction de la configuration souhaitée.

Ce processus de détection et de manipulation de l'objet est essentiel pour assurer une précision dans les tâches robotiques telles que la saisie et la relocalisation d'objets complexes. Il démontre la capacité du drone à interagir de manière intelligente avec son environnement grâce à l'intégration de la vision par ordinateur et du contrôle en boucle fermée dans la simulation.



(a)



(b)

FIGURE 4.15 – détection des cercles

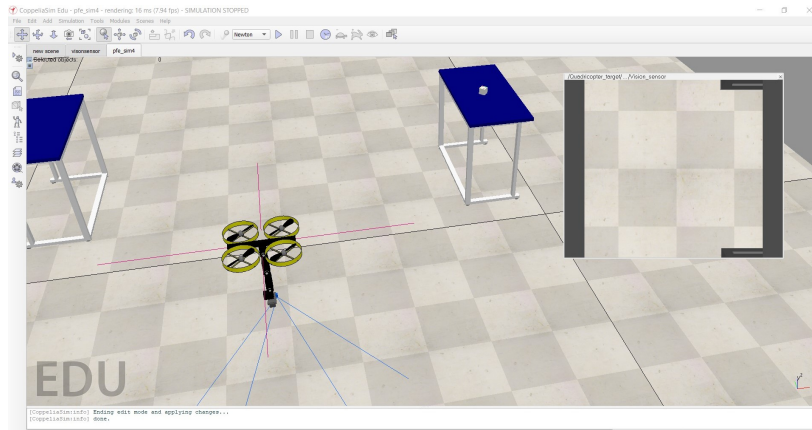
4.3.1.2 Simulation du déplacement du robot

La simulation réalisée dans Coppeliasim implique le déplacement d'un drone équipé d'un bras robotique vers des objets ciblés. Comme illustré dans les images ci-jointes, le drone est initialement positionné à une certaine distance d'un objet, avec la caméra montée sur l'effecteur final du bras robotique. Cette configuration permet au système de capturer en temps réel l'environnement visuel pour guider les actions du drone.

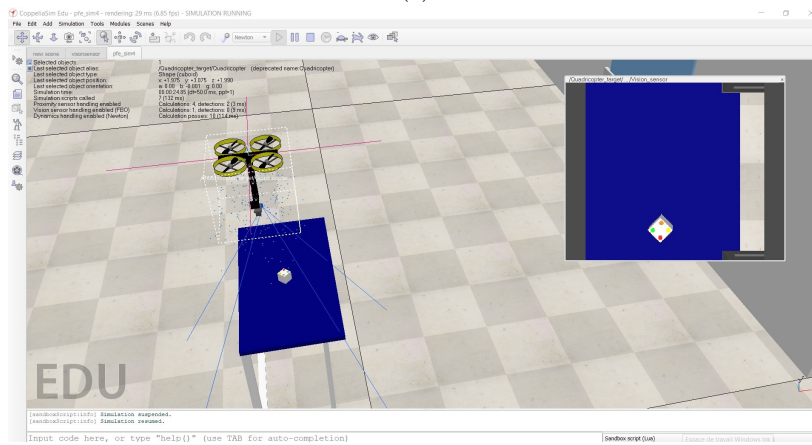
Le drone se déplace d'un point à un autre, en suivant des instructions programmées dans MATLAB. Le capteur de vision intégré permet de détecter les objets dans la scène, et la caméra fournit des images qui sont traitées en temps réel pour ajuster la trajectoire du drone. Une fois que le drone s'approche suffisamment de l'objet cible, le bras robotique est activé pour saisir l'objet.

L'image montre le moment où le drone s'approche d'un cube coloré, détecté par le capteur de vision, qui est visible dans la fenêtre de visualisation en temps réel sur la droite. Le script de simulation en Lua permet de contrôler la dynamique du drone ainsi que l'interaction avec les objets détectés. La combinaison de MATLAB et de Coppeliasim offre une synchronisation parfaite entre le contrôle de la trajectoire et la capture des objets dans l'environnement simulé.

Par contre, nous avons rencontré des problèmes lors de cette simulation en terme de traitement des données et la communication entre les deux logiciels après avoir des resultat ce qui nous a pousser a confirmer notre méthode et nos resultats par une autre simulation sur MATLAB, ainsi que d'autres difficultés liées à la récupération des paramètres intrinsèques et de la focale de la caméra.



(a)



(b)

FIGURE 4.16 – Déplacement du robot vers le point désiré

4.3.2 MATLAB :

MATLAB est un environnement de calcul numérique et de programmation largement utilisé dans l'ingénierie, la robotique, et les sciences. Grâce à ses outils de traitement de données, d'analyse, et de contrôle, MATLAB est idéal pour effectuer des calculs complexes et interagir avec des simulations robotiques.

Dans notre simulation, MATLAB a été utilisé pour lire les données capturées par la caméra installée sur le bras robotique et traiter ces informations pour effectuer des tâches comme la détection d'objets, le suivi, et le contrôle en boucle fermée du mécanisme robotique. L'intégration de MATLAB avec CoppeliaSim nous a permis de concevoir des algorithmes sophistiqués de contrôle et de traitement des données dans un environnement simulé.

Les avantages de l'utilisation de *MATLAB* dans notre projet incluent :

- Une grande flexibilité pour le traitement des données, grâce aux nombreuses bibliothèques intégrées.
- Une interface facile à configurer pour la communication avec des simulateurs comme *CoppeliaSim*.
- Une capacité à tester des algorithmes de contrôle en simulant le comportement en temps réel du drone et du bras robotique.



FIGURE 4.17 – Icon du Matlab

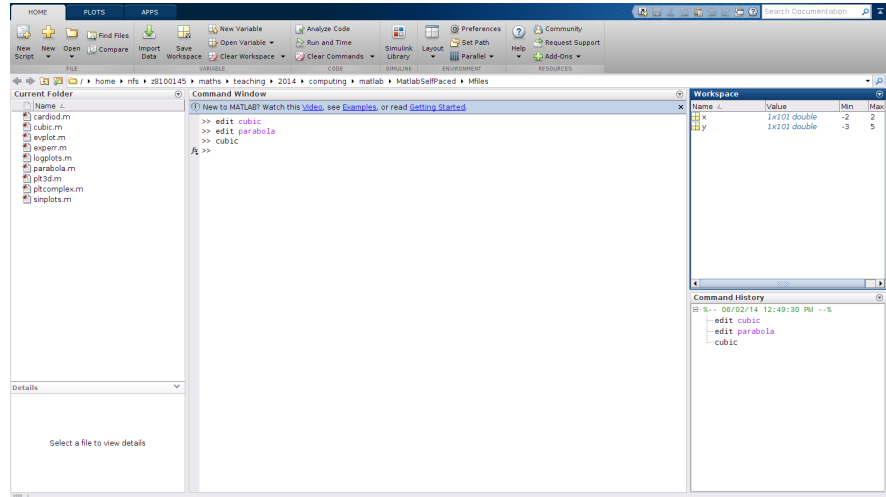


FIGURE 4.18 – Interface du Matlab

En combinant MATLAB avec CoppeliaSim, nous avons pu mettre en place une chaîne de contrôle fluide entre la simulation et le traitement des données capturées par la caméra, garantissant une simulation plus robuste et proche des conditions réelles.

4.3.2.1 Simulation sur matlab

Pour réaliser la simulation sur matlab on crée une caméra et on configure avec ces paramètres :

- La hauteur en pixels des images prises par la caméra : 480.
- La Largeur en pixels des images prises par la caméra : 640.
- Les coordonnées du point principal (le centre optique de la caméra) :

$$\begin{bmatrix} 303 & 243 \end{bmatrix}$$

- Les distances focales :

$$\begin{bmatrix} 657.4 \\ 657.8 \end{bmatrix}$$

- La matrice des paramètres intrinsèques de la caméra :

$$\begin{pmatrix} 657.4 & 0 & 303 \\ 0 & 657.8 & 243 \\ 0 & 0 & 1 \end{pmatrix}$$

La figure (4.19) montre une vue 3D de la simulation où le drone et son bras robotique sont représentés dans un espace tridimensionnel défini par les axes X, Y et Z. Le drone, symbolisé par une structure quadrilatérale équipée de quatre hélices, est capable de se déplacer dans cet espace tout en maintenant un contrôle précis de son orientation et de sa position.

La caméra, montée sur l'effecteur final du bras robotique, permet de capturer et traiter les images nécessaires à la navigation et à la manipulation. Le bras robotique est représenté par une structure articulée qui permet à l'effecteur de se déplacer dans différentes directions pour accomplir une tâche définie.

Dans cette simulation, le drone se déplace d'un point à un autre selon la trajectoire générée par la caméra par l'application de l'asservissement visuel et la cinématique inverse. Le mécanisme de contrôle inclut à la fois le mouvement du drone et celui du bras robotique, assurant une précision dans le placement de l'effecteur. Cette simulation illustre le processus de déplacement du drone en temps réel tout en ajustant le bras robotique, grâce à l'analyse visuelle des données capturées par la caméra.

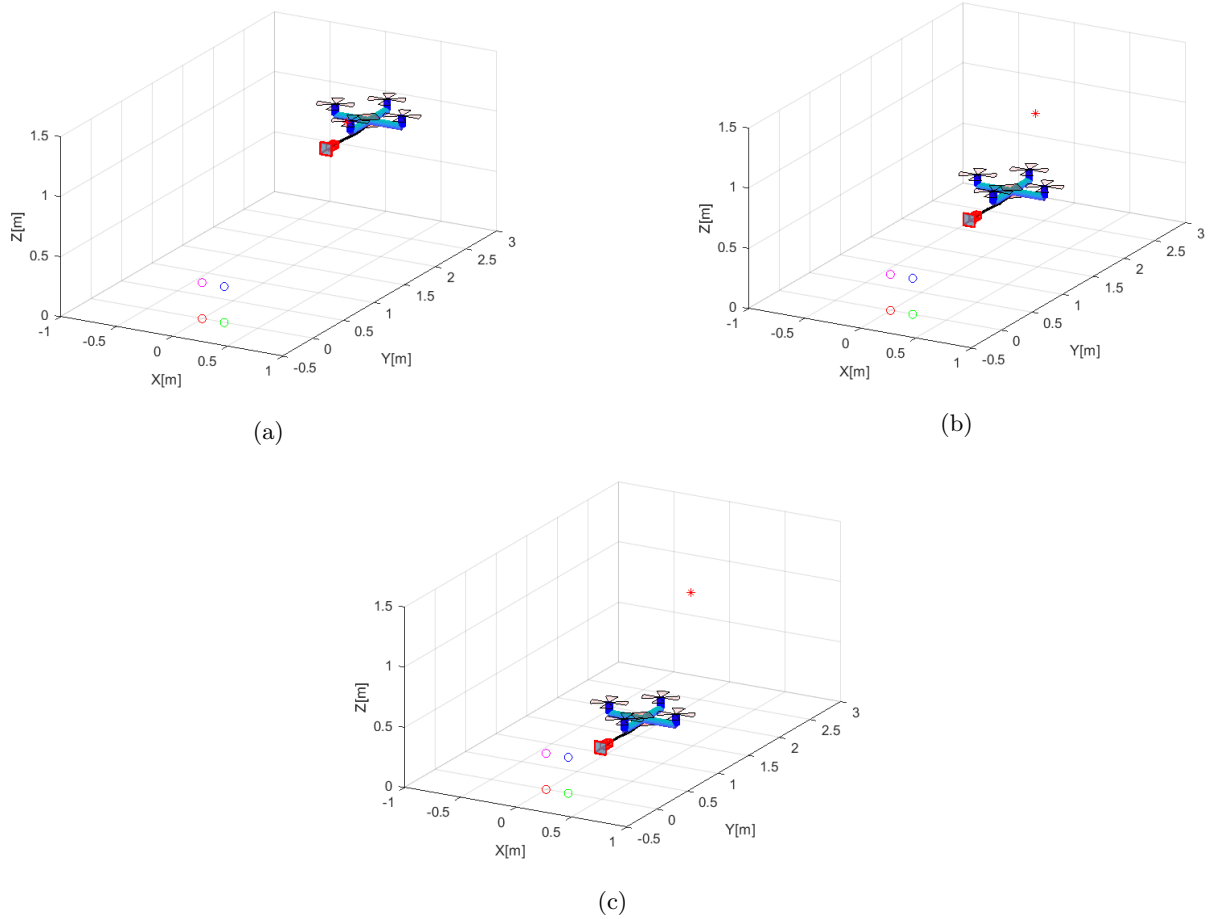
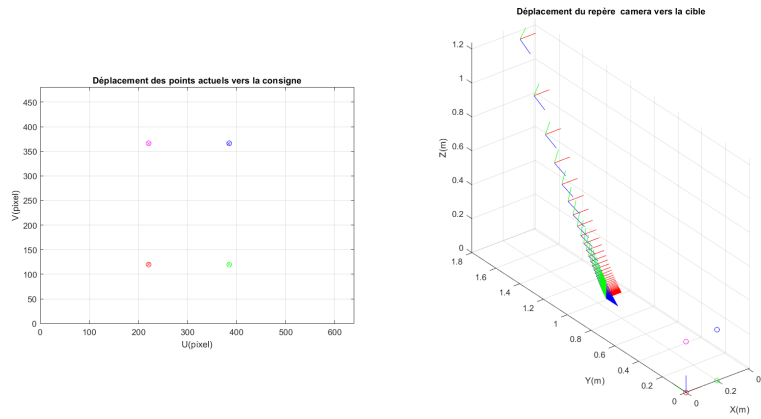
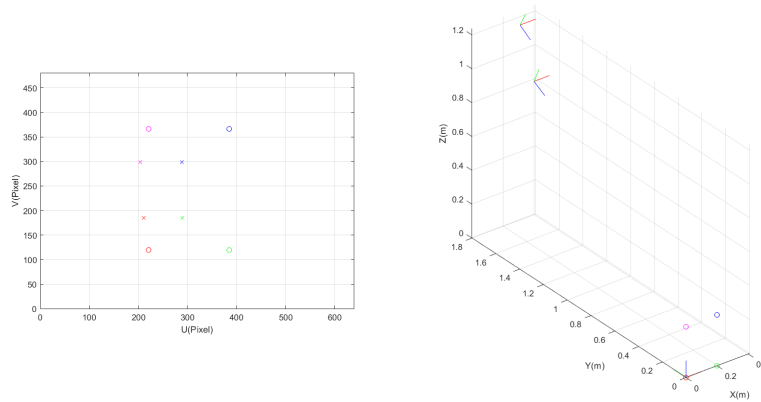


FIGURE 4.19 – Résultats de l'asservissement visuel et de la cinématique inverse

La figure (4.20) présente deux représentations graphiques du déplacement de la caméra. À gauche, nous avons une vue en 2D qui illustre la progression des points de départ jusqu'au point cible sous la forme des cercles. La disposition des points colorés sur le graphique représente les positions relatives des points d'intérêt détectés sur l'objet cible par la caméra. Chacun des points est représenté par un symbole distinct, permettant de visualiser la précision avec laquelle la caméra se superpose aux points prévus.



(a)



(b)

FIGURE 4.20 – Résultats du déplacement du repère caméra et l'évolution des points sur le plan d'image

Sur le graphique de droite, nous avons une vue en 3D qui montre l'évolution des orientations et des positions de la caméra lors de son déplacement. Les vecteurs colorés dans cette représentation 3D montrent les ajustements effectués par la caméra dans les trois dimensions de l'espace, ce qui permet d'améliorer la précision du contrôle visuel. L'objectif ici est d'amener l'effecteur final au-dessus de l'objet et de s'assurer que les points détectés par la caméra sont en adéquation avec les points d'intérêt calculés.

La simulation a été conçue pour reproduire les conditions d'un déplacement réel d'une caméra montée sur un bras robotique. L'algorithme utilisé exploite une boucle de rétroaction visuelle, où les informations captées par la caméra sont traitées en temps réel pour ajuster la position et l'orientation de l'effecteur final. Ce type de rétroaction permet de garantir une haute précision dans la manipulation des objets et une minimisation des erreurs de superposition.

Les résultats obtenus montrent que la caméra parvient à identifier les points clés de l'objet et à les superposer de manière satisfaisante. En effet, comme illustré dans les figures, la convergence des points colorés suggère que l'effecteur est bien aligné avec l'objet cible, prêt à effectuer l'opération de saisie. Cette approche est particulièrement efficace pour des applications nécessitant une manipulation fine, telle que la préhension d'objets complexes.

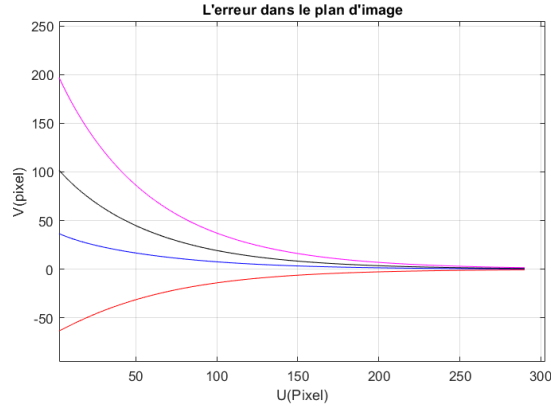


FIGURE 4.21 – l’erreur dans le plan image

Le graphe dans la figure (4.21) illustre l’évolution de l’erreur dans le plan image au fil du temps, issue de la simulation du contrôle de la caméra montée sur l’effecteur final du bras robotique. Cette erreur est mesurée en termes de déviation des points détectés par la caméra par rapport aux points cibles sur l’objet à saisir.

La tendance générale observée est une décroissance continue des erreurs, jusqu’à ce qu’elles atteignent zéro ou s’en approchent après un certain nombre d’itérations. Cela suggère que le système de contrôle basé sur la caméra parvient à ajuster la position de l’effecteur pour corriger les écarts et amener la caméra au-dessus des points d’intérêt détectés sur l’objet.

Le graphique montre que, malgré une erreur initiale significative dans certaines composantes, le mécanisme de rétroaction utilisé dans le contrôle de la caméra réussit à minimiser cette erreur de manière efficace. Le fait que toutes les courbes convergent vers zéro démontre la robustesse et la précision du système dans la réduction des écarts entre les points captés et les points cibles. Ce résultat est essentiel pour garantir que la caméra se positionne correctement sur les points d’intérêt, permettant ainsi à l’effecteur final de saisir avec précision l’objet désiré.

Le graphique de la figure (4.22) représente le mouvement cartésien du drone au cours de la simulation. Les résultats de cette simulation montrent que le drone est capable de se déplacer de manière fluide et contrôlée dans les trois dimensions spatiales. Les variations observées dans les graphes sont cohérentes avec les mouvements attendus du robot. Ces résultats confirment l’efficacité du système de contrôle et de traitement des données implémenté dans MATLAB.

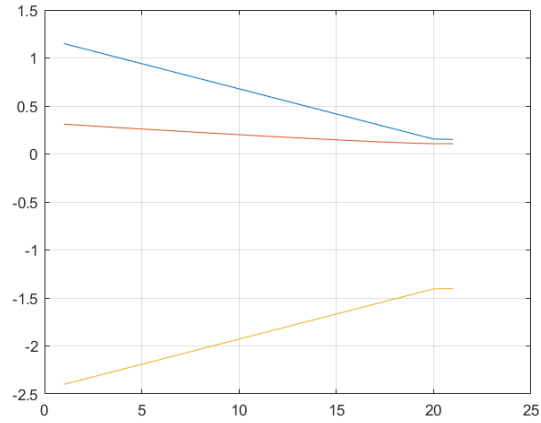


FIGURE 4.22 – Mouvement cartésien du drone (X Y Z)

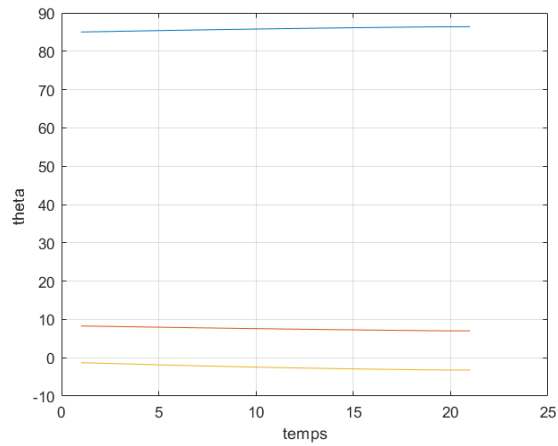


FIGURE 4.23 – Articulation de bras

Dans le graph de la figure (4.23) qui représente le mouvement des articulations du bras robotique, on conclut que dans notre cas le mouvement du bras est très léger d'où il n'a pas changé pour les 3 articulations car son position était pas très different de la position désirée pour attraper l'objet.

Chapitre 5

Conclusion Général

Ce travail a porté sur la conception, la modélisation et la simulation d'un système de manipulation aérienne autonome, composé d'un drone quadricoptère équipé d'un bras manipulateur et d'un système de contrôle visuel. Nous avons exploré diverses solutions pour permettre au système d'effectuer des tâches complexes de manipulation dans des environnements difficiles ou inaccessibles, en assurant la stabilité et la précision de ses mouvements.

Les résultats obtenus confirment la faisabilité de l'utilisation de drones pour la manipulation aérienne autonome dans des environnements complexes. Grâce à la coordination entre le drone et le bras robotisé, nous avons démontré la robustesse et la précision du contrôle visuel, ainsi que la capacité du système à traiter des images en temps réel et à ajuster sa trajectoire en conséquence. Ces résultats ouvrent la voie à une large gamme d'applications, telles que l'inspection d'infrastructures, la maintenance, ou encore les opérations de sauvetage dans des zones à risque.

Cependant, plusieurs défis restent à surmonter, notamment l'amélioration de l'autonomie énergétique du drone, la gestion des contraintes environnementales réelles (telles que le vent ou les variations de température) et la précision du contrôle en conditions réelles. Ces aspects seront cruciaux pour permettre une adoption industrielle plus large de ces systèmes.

Perspectives d'avenir

Dans un avenir proche, il serait pertinent d'approfondir les recherches sur plusieurs axes :

Optimisation du système de contrôle visuel

Une des priorités sera d'améliorer les algorithmes de contrôle visuel pour une meilleure détection des objets en mouvement, ainsi que pour des environnements plus variés. L'intégration d'algorithmes de traitement d'image plus sophistiqués, combinés à des systèmes d'intelligence artificielle, pourrait permettre au drone de s'adapter à des situations encore plus complexes et dynamiques.

Augmentation de l'autonomie et des performances

L'une des principales limitations identifiées dans notre travail est l'autonomie énergétique du drone. Dans l'optique de futures applications industrielles ou de sauvetage, il serait essentiel de développer des solutions permettant d'augmenter la durée de vol, telles que l'utilisation de nouvelles sources d'énergie comme les piles à hydrogène ou l'amélioration de la gestion énergétique.

Validation expérimentale

Bien que nos résultats en simulation soient prometteurs, il est impératif de tester et de valider ces approches dans des environnements réels. Cela inclut la construction d'un prototype physique basé sur les modèles développés en simulation, afin de vérifier la robustesse du système dans des conditions réelles. Ce travail expérimental permettra d'identifier des défis imprévus liés à la mécanique et à l'intégration des différents systèmes.

Exploration de nouveaux domaines d'application

Les systèmes de manipulation aérienne peuvent potentiellement être utilisés dans de nombreux secteurs. Des études approfondies sur les applications industrielles (par exemple, la maintenance des infrastructures ou la réparation en milieu difficile d'accès) ou dans des scénarios de gestion des catastrophes (recherche et sauvetage) pourraient révéler de nouvelles opportunités pour ces technologies.

En somme, cette étude constitue une étape importante vers la réalisation de drones manipulateurs autonomes. Avec l'intégration de systèmes d'intelligence artificielle et l'optimisation des composants mécaniques et énergétiques, ces systèmes pourraient révolutionner un large éventail d'industries, augmentant leur efficacité et réduisant les risques humains.

Bibliographie

- [1] Les techniques d’asservissement visuel - robotique ; commande par la vision.
- [2] Contributeurs aux projets Wikimedia. Drone Wikipédia, 2024.
- [3] Savoy Control. Contrôle visuel, méthode de contrôle non destructif (cnd) - savoy control, 1 2019.
- [4] R. Drone. Histoire du drone, May 2024. Référence Drone.
- [5] L. T. Dronez. L’histoire du drone, July 2023. Le Webzine Spécialiste Des Drones En France — Dronez.
- [6] Gerri. L’histoire et l’évolution des drones : Qui les a inventés ? <https://commentouvrir.com/info/lhistoire-et-levolution-des-drones-qui-les-a-inventes/>.
- [7] Guanqi He, Yash Jangir, Junyi Geng, Mohammadreza Mousaei, Dongwei Bai, and Sebastian Scherer. Image-based visual servo control for aerial manipulation using a fully-actuated uav. *arXiv preprint arXiv :2306.16530*, 6 2023. Accepted by 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [8] Guanqi He, Yash Jangir, Junyi Geng, Mohammadreza Mousaei, Dongwei Bai, and Sebastian Scherer. Image-based visual servo control for aerial manipulation using a fully-actuated uav. *arXiv.org*, June 2023.
- [9] Alexandre Krupa. Asservissement visuel par imagerie médicale, 4 2019.
- [10] Léa. Tout ce que vous devez savoir sur l’histoire du drone. <https://www.amusons-nous.fr/lhistoire-du-drone/>, March 2023.
- [11] Fengyu Quan, Haoyao Chen, Yanjie Li, Yunjiang Lou, Jian Chen, and Yunhui Liu. Singularity-robust hybrid visual servoing control for aerial manipulator. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 12 2018.
- [12] Fabien Spindler and Christian Blonz. Asservissement visuel, 2005.
- [13] Hang Zhong, Zhiqiang Miao, Yaonan Wang, Jianxu Mao, and Ling Li. A practical visual servo control for aerial manipulation using a spherical projection model. *IEEE Journals Magazine*, 12 2020.

ملخص

يتناول هذا البحث تصميم ومحاكاة نظام تحكم جوي مستقل يتكون من طائرة رباعية المراوح مجهزة بذراع آلية وكاميرا للرؤية الحاسوبية. الهدف الرئيسي من هذا المشروع هو تمكين الطائرة من أداء مهام المناولة بدقة في بيئات يصعب الوصول إليها، وذلك باستخدام نظام تحكم يعتمد على الرؤية.

لقد تم استكشاف نهجين للمحاكاة. الأول باستخدام برنامجي غازيبو و روس، وقد أظهرت هذه الطريقة بعض القيود بسبب عدم توافق نماذج الطائرات. أما النهج الثاني، والذي اعتمد على برنامجي كوبيليا و ماطلاب، فقد مكّن من التغلب على تلك الصعوبات ومحاكاة نموذج لطائرة مزودة بذراع آلية تعمل بشكل فعال.

النتائج المحققة أظهرت كفاءة واعدة للنظام في تنفيذ مهام المناولة المستقلة، مع تطبيقات محتملة في مجالات مثل التفيتش، الصيانة، والإقناذ. كما يقدم هذا البحث مقترحات لتحسين النظام من خلال زيادة استقلالية الطائرة وتحسين أدائها في البيئات الحقيقية.

الكلمات المفتاحية : التحكم البصري، طائرة بدون طيار رباعية المراوح، الروبوتات الجوية، نظام التحكم البصري، الكينماتيكا، التوجيه البصري، التطبيقات الصناعية.

Résumé

Ce mémoire traite de la conception et de la simulation d'un système de manipulation aérienne autonome, composé d'un drone quadricoptère équipé d'un bras manipulateur et d'une caméra pour le contrôle visuel. L'objectif principal de ce projet est de permettre au drone d'exécuter des tâches de manipulation précises dans des environnements difficiles d'accès, grâce à un asservissement visuel.

Deux approches ont été explorées pour la simulation. La première, utilisant les logiciels Gazebo et ROS, a révélé des limitations en raison de l'incompatibilité des modèles de drone. La deuxième approche, basée sur les logiciels CoppeliaSim et MATLAB, a permis de surmonter ces difficultés et de simuler un modèle de drone avec un bras robotique fonctionnel.

Les résultats obtenus montrent une efficacité prometteuse du système dans des tâches de manipulation autonomes, avec des applications potentielles dans des secteurs tels que l'inspection, la maintenance et le sauvetage. Cette étude propose également des pistes d'amélioration, notamment pour accroître l'autonomie du drone et optimiser ses performances dans des environnements réels.

Mots clés : Contrôle visuel, Drone quadricoptère, Robotique aérienne, Système de contrôle visuel, Cinématique, Asservissement visuel, Applications industrielles.

Abstract

This thesis focuses on the design and simulation of an autonomous aerial manipulation system, consisting of a quadcopter drone equipped with a manipulator arm and a camera for visual control. The primary objective of this project is to enable the drone to perform precise manipulation tasks in hard-to-reach environments through visual servoing.

Two approaches were explored for the simulation. The first, using Gazebo and ROS software, revealed limitations due to the incompatibility of drone models. The second approach, based on CoppeliaSim and MATLAB software, overcame these difficulties and successfully simulated a drone model with a functional robotic arm.

The results obtained demonstrate promising efficiency of the system in autonomous manipulation tasks, with potential applications in fields such as inspection, maintenance, and rescue. This study also proposes avenues for improvement, notably to enhance the drone's autonomy and optimize its performance in real-world environments.

Keywords : Visual control, Quadcopter drone, Aerial robotics, Visual control system, Kinematics, Visual servoing, Industrial applications.