

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Conception de la commande et de la téléopération d'un bras robotique
ultraléger intégré sur un drone**

FLORIAN COIFFARD

Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie mécanique

Décembre 2022

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Conception de la commande et de la téléopération d'un bras robotique
ultraléger intégré sur un drone**

présenté par **Florian COIFFARD**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Richard GOURDEAU, président

Lionel BIRGLEN, membre et directeur de recherche

Jean-Philippe ROBERGE, membre

REMERCIEMENTS

Tout d'abord, je tiens à remercier mes superviseurs, Michel Lambert et Bruno Monsarrat pour leur accueil et leur aide tout au long du projet au Centre de Technologies de Fabrication en Aérospatial (CTFA) du Conseil National de Recherches du Canada (CNRC). Je voulais aussi remercier Bruno Chapdelaine, Mathis Celce ainsi que tous les membres de l'équipe de robotique du CTFA.

Je tiens à remercier mon directeur de recherche Lionel Birglen, professeur au département de génie mécanique et directeur du laboratoire de robotique à l'université de Polytechnique Montréal, pour sa disponibilité et sa supervision pour l'écriture de l'article scientifique.

Enfin, je tiens à remercier ma soeur qui m'a beaucoup soutenu pendant ces deux années.

RÉSUMÉ

Ce mémoire présente la conception d'un bras robotique ultraléger à six axes téléopéré et embarqué sur un drone ayant des capacités d'accostage. Ce bras robotique est destiné à être capable d'effectuer des tâches locales de maintenance sur des zones difficiles d'accès. En effet, la maintenance d'infrastructures difficiles d'accès (ponts, paroi rocheuse, éolienne) nécessite un déploiement humain et matériel complexe pour garantir la sécurité. Ainsi des solutions de robotique aérienne existent et permettent de remplacer les tâches de l'opérateur dans des zones difficiles d'accès. Actuellement ces tâches de maintenance sont principalement liées à de l'inspection.

Dans un objectif d'opérer des tâches de contact (meulage, ponçage, perçage, etc.) dans des zones difficiles d'accès, la conception d'un bras robot six axes ultraléger a été développée. Ce mémoire s'inscrit dans la continuité des travaux réalisés sur la conception d'un bras robotique au sein du Centre des Technologies de Fabrication en Aérospatial du CNRC.

Le bras robotique est embarqué sur un drone ayant une capacité d'emport de charges lourdes. Le drone est équipé d'un module d'accostage par ventouses actives (BAM). Le BAM a la possibilité de s'accoster sur une surface dont la plage angulaire par rapport au sol est entre 0 à 180°. Une fois le drone accosté sur la structure, celui-ci peut arrêter ses moteurs puis déployer son bras robotique afin d'effectuer les tâches de maintenance souhaitées.

Précédemment, l'architecture mécanique du bras robotique a été proposée, celle-ci se basant sur une optimisation visant à maximiser la surface de travail pour une masse fixée et des contraintes de performances sur différents scénarios. Cette optimisation a amené à des choix de composants et à la création d'un modèle 3D numérique.

Par la suite, une architecture électrique globale du bras a été établie avec le choix de l'ordinateur de bord, des contrôleurs moteurs et des capteurs. Une architecture de contrôle a été présentée à partir de la dynamique du robot, ainsi différentes compensations ont été ajoutées afin d'obtenir un fonctionnement stable et précis du bras. Un système de téléopération visuelle et haptique ont été intégrés pour assister l'opérateur dans le contrôle du bras, lui permettant de définir les trajectoires à suivre par le robot.

Une programmation sous ROS (Robot Operating System) a été développée pour gérer la commande du robot. Celle-ci intègre notamment un planificateur de trajectoire et un système de vision permettant la génération de trajectoire et l'évitement d'obstacle. Deux modes de contrôle ont été intégrés au bras pour permettre à l'opérateur de choisir entre le mouvement

libre ou supervisé du robot.

Enfin des expériences ont été réalisées sur un modèle simplifié du bras et sur un manipulateur robotique UR10 afin de valider le fonctionnement de l'architecture de contrôle et d'évaluer ses performances.

ABSTRACT

This thesis presents the design of an ultralight six-axis remotely operated and mounted on a UAV with docking capacities. This robotic arm is designed to be able to perform local maintenance tasks in hard-to-reach areas. Indeed, the maintenance of hard-to-reach infrastructures (bridges, rock walls, wind turbines) requires a complex human and material deployment to guarantee safety. Aerial robotics solutions exist to replace the operator's tasks in hard-to-reach areas. Currently, these maintenance tasks are mainly related to inspection.

In order to perform contact tasks (grinding, sanding, drilling, ...) in difficult to access areas, the design of a six-axis ultralight robot arm was developed. This thesis is a continuation of the work accomplished on the design of a robotic arm at the NRC Aerospace Manufacturing Technology Centre.

The robotic arm is mounted on a drone with a capacity to carry heavy loads. The drone is equipped with an active suction cup docking module (BAM). The BAM is able to dock on a surface whose angular range with reference to the ground is between 0 and 180°. Once the drone is docked on the structure, it can shut down its motors and then deploy its robotic arm to perform the desired maintenance tasks. Previously, the mechanical architecture of the robotic arm has been proposed, based on an optimization to maximize the working area for a fixed mass and performance requirements on different scenarios. This optimization led to the choice of components and creation of a 3D digital model.

Thereafter, a complete electrical architecture of the arm is defined with the choice of the on-board computer, the motor controllers and the sensors. A control architecture is proposed based on the dynamics of the robot, and different compensations are applied in order to obtain a stable and precise operation of the arm. A visual and haptic teleoperation system is implemented to assist the operator in controlling the arm, allowing him to define the trajectories to be followed by the robot.

A programming under ROS (Robot Operating System) is developed to manage the robot control. It integrates a path planner and a vision system allowing the generation of trajectory and obstacle avoidance. Two control modes are built into the arm to allow the operator to choose between free or supervised movement of the robot.

Finally, experiments were carried out on a simplified model of the arm and on a UR10 robotic manipulator in order to validate the operation of the control architecture and to evaluate its performance.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
LISTE DES ANNEXES	xv
1 INTRODUCTION	1
1.1 Contexte du projet	1
1.2 Problématique	3
1.3 Objectifs de recherche	3
1.4 Plan du mémoire	4
2 REVUE DE LITTÉRATURE	5
2.1 Maintenance de structures , difficiles et dangereuses d'accès, par robots téléopérés	5
2.1.1 Robotique mobile	6
2.1.2 Robotique aérienne	8
2.2 Bras robotique léger	10
2.2.1 Architecture mécanique	10
2.2.2 Architecture de contrôle	11
2.3 Téléopération d'un bras robotique	13
2.3.1 Retour d'effort	15
2.3.2 Vision	16
3 DEMARCHE DE TRAVAIL	18
3.1 Résumé de la problématique	18
3.2 Objectif général	18

3.3	Objectifs spécifiques	18
4	ARTICLE 1 : DESIGN AND CONTROL OF A UAV-MOUNTED UL- TRALIGHTWEIGHT ROBOT ARM FOR CONTACT TASKS	20
4.1	Introduction	20
4.1.1	Motivation	20
4.1.2	State of the art in aerial manipulation	21
4.1.3	Methodology and contributions	23
4.2	Design Requirements	24
4.2.1	System Environment	24
4.2.2	Initial Choices	24
4.2.3	Motors and Gearheads	26
4.2.4	Modeling	27
4.3	Electronics Design	32
4.3.1	General Architecture	32
4.3.2	Power Supply	33
4.3.3	Sensors	33
4.4	Control and Software Design	33
4.4.1	Control of the embedded robotic arm	33
4.4.2	Operational Architecture	36
4.5	Experiments and Results	42
4.5.1	Free Motion Mode	42
4.5.2	Supervised Motion Mode	44
4.6	Conclusion	47
5	MÉTHODOLOGIE ET RÉSULTATS COMPLÉMENTAIRES	48
5.1	Implémentation de l'architecture de contrôle du bras	48
5.1.1	Partie locale (Opérateur)	48
5.1.2	Partie embarquée (Drone)	53
5.2	Améliorations envisagées	57
5.2.1	Implémentation d'un contrôle en admittance (capteur de force)	57
5.2.2	Implémentation d'un planificateur de trajectoire	58
5.2.3	Implémentation d'un modèle de rigidité du bras	58
6	DISCUSSION GENERALE	60
7	CONCLUSION	62

7.1 Synthèse des travaux	62
7.2 Limitations de la solution proposée	63
7.3 Améliorations futures	63
RÉFÉRENCES	64
ANNEXES	72

LISTE DES TABLEAUX

Tableau 2.1	Tableau des technologies de caméra utilisées pour des systèmes robotiques	17
Tableau 4.1	Summary of initial design choices	25
Tableau 4.2	Joints specifications	28
Tableau 4.3	Links specifications	28
Tableau 4.4	Denavit-Hartenberg parameters	30
Tableau 4.5	Comparison of two IK solvers (KDL, TRACK-IK) based on computation time and success rate for UR10 robotic manipulator and UAV_ARM	40
Tableau 5.1	Coefficient de frottement statique et dynamique	56
Tableau 5.2	Taux de réussite des planificateurs de trajectoire	58

LISTE DES FIGURES

Figure 1.1	Rendu 3D du drone avec son bras robotique accosté sur une pale d'éolienne à l'aide du BAM	2
Figure 2.1	Ensemble des procédés effectués par des techniciens cordistes en fonction de sa complexité et de la charge utile nécessaire. La ligne pointillée délimite les procédés réalisable actuellement par des solutions de robotique aérienne. Ainsi les deux zones représentent l'aptitude (en bleu) et l'inaptitude (en orange) des solutions de robotiques aériennes actuelles pour ces procédés spécifiques.	5
Figure 2.2	Classification des degrés de téléopération d'un système robotique selon la complexité de l'opération effectuée et la variabilité de l'environnement dans lequel il évolue	6
Figure 2.3	Visuel des trois systèmes robotiques présentés : 1) Robot de maintenance de pale d'éolienne (reproduit avec l'autorisation de Aeronex [1]) - 2) BR-8 (reproduit avec l'autorisation de Rope Robotics [2]) - 3) SCOMPI (photo extraite de [3])	8
Figure 2.4	Visuel des quatre systèmes de robotiques aériennes : 1) Compact AeRial MANipulator (photo extraite de [4]) 2) PD6B-AW-ARM (Reproduit avec l'autorisation de Prodrone) 3) HYFLIERS (photo extraite de [5]) 4) Manipulateur Delta intégré à un drone (photo extraite de [6]) . . .	9
Figure 2.5	Bras robotiques légers : 1) DLR LWR arm and hand (photo tirée de [7]) - 2) KINOVA GEN2 - 3) KINOVA GEN3 - 4) CAESAR (illustration tirée de [8]) - 5) LARAD (photo tirée de [9])	11
Figure 2.6	Schéma régulateur PID en position	12
Figure 2.7	Schéma de l'architecture de contrôle du manipulateur CAESAR . . .	13
Figure 2.8	Interfaces haptiques commercialisée	14
Figure 2.9	Schéma représentant les trois principaux modes de téléopération . . .	15
Figure 4.1	Examples of tasks performed by rope access technicians (wind turbine maintenance on the left, NDT of water pipes on the right), courtesy of Vertica (Brittany, France)	22
Figure 4.2	Payload $\times$ Reach as a function of weight for a selection of lightweight robots. The orange square represents the targeted performance	23
Figure 4.3	Geometries of the three docking scenarios considered in this paper (90 and 180 dockings are close to the examples shown in Fig. 4.1)	25

Figure 4.4	Mechanical model of the robot arm used for dynamic simulations . . .	26
Figure 4.5	Section views of the large Joint 1 (left) and the smaller Joint 2 (right)	27
Figure 4.6	Robotic arm model with articulation modules, links and a linkage system	29
Figure 4.7	Robotic arm design with joints modules and links (left) and Denavit-Hartenberg frames of reference and parameters (right).	30
Figure 4.8	Inverse dynamics implementation	31
Figure 4.9	Schematic of the electronic architecture of the robotic arm	32
Figure 4.10	Structure of the robot arm position controller with PD control and additional compensations	34
Figure 4.11	Control architecture diagram of robotic arm. Local workspace corresponds to elements used by the operator and remote workspace corresponds to elements used onboard the UAV.	35
Figure 4.12	Graphical user interface built on RViz : 1)Video stream from ZED2 camera 2)Virtual model of the robotic arm and its local environment 3)Display of the ROS computation graph (rqt-graph) 4)Robotic arm parameters (Default motion planner interface in Rviz)	36
Figure 4.13	Quanser High Definition Haptic Device HD^2 on the left and its Cartesian representation with a spring damper model on the right	37
Figure 4.14	State machine diagram of the task planner	39
Figure 4.15	Pilz Industrial motion planner trajectory passing through five points along the X axis. Red dash-dot vertical lines are the representation of the desired points and the error bars on the joint position represent the tolerance of the end effector to pass close to the desired points . .	41
Figure 4.16	Round-trip communication delay between haptic interface and UR10 in free motion mode	42
Figure 4.17	Feedback force received after a movement x_d of the haptic interface by the operator	43
Figure 4.18	Top left : virtual model of the UR10 arm during a simulated trajectory on RViz. Top right : virtual model of the UR10 during a free motion mode. Bottom : physical system.	44
Figure 4.19	Results of the angular position error during the execution of a trochoidal trajectory as shown in Fig. 4.18	45
Figure 4.20	RViz virtual model of the robotic arm embedded on the drone docked to a wind turbine blade. Description of the scene elements during the trajectory generation in the supervised motion mode.	46

Figure 4.21	Results of the absolute position error during the execution of a trochoidal trajectory as shown in Fig. 4.20	46
Figure 5.1	Chaîne de téléopération avec retour haptique et visuel	48
Figure 5.2	Schéma simplifié des relations entre les packages ROS	49
Figure 5.3	Schéma des connexions entre les parties constituantes du système . .	50
Figure 5.4	1) Trajectoire zig-zag 2) Trajectoire trochoïdale 3) Trajectoire spirale	52
Figure 5.5	Trajectoires zig-zag, trochoïdale et carrée simulées sur Rviz	53
Figure 5.6	Photo du système embarqué (Raspberry Pi, contrôleur Moteus, moteur, alimentation électrique)	54
Figure 5.7	Schéma simplifié de l'API développée pour interagir avec le contrôleur Moteus	54
Figure 5.8	Histogramme de résolution de l'encodeur en position et en vitesse . .	55
Figure 5.9	Graphique représentant le couple de maintien à vide en fonction de la vitesse	56
Figure 5.10	1)Schéma cinématique du mécanisme 3 axes planaire 2)Module de compensation de gravité sous Simulink 3)Prototype de bras 3 axes planaire	57
Figure 5.11	Schéma modulaire du bras conçu avec le module Simscape Multibody à gauche. Visuel simplifié du modèle du bras 3 axes	59

LISTE DES SIGLES ET ABRÉVIATIONS

BAM	Based Attachment Module
ROS	Robot Operating System
CTFA	Centre de Technologies de Fabrication en Aeronautique
CNRC	Conseil National de Recherches Canada
DDL	Degres de liberte
HD	Harmonic Drive
BLDC	Brushless DC
UR10	Universal Robots UR10
IK	Cinématique inverse
SRDF	Semantic Robot Description Format
HD2	Quanser High Definition Haptic Device
API	Interface de Programmation d'Application
PID	Régulateur Proportionnel, Intégral, Dérivé
CAESAR	Compliant Assistance and Exploration SpAce Robot
LARAD	Lightweight Advanced Robotic Arm Demonstrator

LISTE DES ANNEXES

Annexe A	Description de l'architecture du robot SRDF	72
----------	---	----

CHAPITRE 1 INTRODUCTION

Le travail de recherche ici présenté s’inscrit dans le développement des activités de robotique aérienne du Centre de Technologies de Fabrication en Aéronautique (CTFA) du Conseil National de Recherches du Canada (CNRC).

Les maintenances prédictives et correctives sont des actions importantes à mener pour prolonger les performances et la durée de vie des structures difficiles d’accès telles que les éoliennes, les ponts, etc. Cependant ces maintenances sur ces structures aériennes peuvent provoquer des risques pour la santé et la sécurité de l’opérateur [2]. C’est pourquoi ce projet a pour objectif le développement d’un bras robotique léger intégré à un drone. Cela afin d’effectuer des tâches de contact (ponçage, polissage, etc) et permettre à l’opérateur de téléopérer le bras robotique dans une zone sécurisée. La partie mécanique du bras robotique a déjà été étudiée [10] et a permis de ressortir une multitude de contraintes à satisfaire pour différents scénarios de déploiement. Parmi ces dernières, une maximisation de la surface de travail pour une masse faible du bras et des performances satisfaisantes pour des tâches de contact avec enlèvement de matière. À ces contraintes mécaniques se rajoutent des contraintes de contrôle du bras robotique à distance telles qu’une communication fiable, une bonne maniabilité pour l’opérateur et des performances satisfaisantes (position, vitesse, couple, délais de communication, vibration). Enfin le bras robotique doit pouvoir être téléopéré de manière optimale par l’opérateur dans ses différents modes de fonctionnement.

1.1 Contexte du projet

Les problématiques actuelles de l’utilisation de bras robotique intégré à un drone sont principalement de deux types.

Tout d’abord, la capacité de chargement du drone qui est limitée par la puissance de ses moteurs ou bien par la capacité de sa batterie qui influencent directement le temps de vol désiré. La masse actuelle du bras robotique intégré au drone est de 10kg. Celui-ci est intégré sur un drone pouvant embarquer des charges relativement lourdes de 15kg (Hercule 20 de DRONE VOLT).

Ensuite, les opérations effectuées par le bras robotique sont majoritairement liées au transport ou à de la manipulation d’objet en vol. Les opérations de maintenance avec contact de l’outil sont aussi des aspects importants pour l’utilisation du bras.

Le bras robotique effectue ses opérations de contact avec enlèvement de matière lorsque le

drone est accosté sur la structure à laquelle il opère. Un dispositif d'accostage (Base Attachment Module BAM) est intégré au drone afin de réaliser un accostage sur une structure légèrement irrégulière à l'aide de ventouses actives [11]. L'accostage du drone via le BAM permet d'opérer plus facilement dans un environnement fixe pour le bras robotique, il permet aussi de couper ses moteurs lors des phases d'opération du bras robotique et par conséquent réduit la consommation d'énergie qui influence directement la durée d'opération possible de la maintenance.

L'orientation de la surface d'accostage par rapport à la gravité étant propre à chaque structure nécessitant une maintenance, le BAM a la possibilité de s'accoster à des surfaces orientées entre 0 et 180° par rapport au sol. Cette orientation par rapport à la gravité engendre des cas de charge spécifiques sur certaines articulations, ce qui a amené l'étude précédente à concevoir de manière modulaire certaines parties du bras robotique. Afin d'obtenir une bonne dextérité lors de l'utilisation du bras robotique, celui-ci est composé de six degrés de liberté dont un poignet sphérique.



FIGURE 1.1 Rendu 3D du drone avec son bras robotique accosté sur une pale d'éolienne à l'aide du BAM

1.2 Problématique

La maintenance de structure aérienne est nécessaire pour le bon fonctionnement du système dans le temps. Une intervention humaine est nécessaire pour cette maintenance. Cependant, cela implique des risques pour l'opérateur qui doit accéder à des endroits difficiles d'accès, mais aussi un temps de maintenance relativement long afin d'assurer au mieux sa sécurité. L'utilisation d'un drone embarquant un bras robotique, afin de réaliser des tâches de contact avec la structure, est donc une réponse intéressante à la sécurisation de l'environnement de l'opérateur. Le drone a pour objectif de s'accoster dans la zone nécessitant une maintenance, puis de déployer son bras robotique afin d'effectuer les opérations de contact désirées par l'opérateur (ponçage, polissage, etc).

Le bras robotique relève plusieurs problématiques lors de son utilisation. Premièrement, la nécessité de le contrôler à distance. Celui-ci doit pouvoir satisfaire des critères de performances en position, en vitesse, en accélération, en effort et en stabilité. Il doit notamment être maniable lors du contrôle à distance par l'opérateur ce qui sera réalisé par une interface haptique. Cette maniabilité est désirée en terme de déplacement dans son espace de travail et dans un aspect temporel. Il est nécessaire d'obtenir un contrôle du bras se rapprochant du temps réel afin de rendre l'opération de maintenance plus immersive pour l'opérateur.

La seconde problématique est la nécessité d'intégrer une vision de l'environnement de travail, permettant d'adapter le comportement du bras dans son environnement pour générer la trajectoire de son outil et éviter les collisions avec des obstacles. Cela permet entre autres à l'opérateur de visualiser l'environnement de travail du bras et de rendre plus immersif son contrôle. Cette intégration d'une vision de l'environnement du bras nécessite un système de caméra permettant par la suite de modéliser un environnement numérique de travail du bras robotique.

Cette étude répond à une problématique générale qui est le contrôle et la téléopération d'un bras robotique dans un environnement inconnu afin d'effectuer des opérations de maintenance sur des structures difficiles d'accès et dangereuses pour un opérateur.

1.3 Objectifs de recherche

L'étude précédente a permis d'obtenir une architecture mécanique du bras spécifiquement optimisé pour les opérations de maintenance désirées (détailée dans le Chapitre 2), l'objectif étant maintenant de réaliser son architecture de contrôle du bras. Pour mener à bien cet objectif, celui-ci a été décomposé en plusieurs sous-objectifs :

1. Réaliser un asservissement des articulations du bras, en prenant en compte toutes les compensations nécessaires pour valider les performances requises (stabilité, position, vitesse, effort, etc) ;
2. Définir une architecture électrique des composants du système complet (bras, drone, espace opérateur) ;
3. Implémenter une téléopération par interface haptique ;
4. Implémenter un système de vision pour la génération de l'espace de travail ;
5. Définir différents modes de contrôle du bras (suivi de trajectoire générée, déplacement libre par l'opérateur, déplacement assisté) ;
6. Modéliser le bras robotique et ses interactions dans son environnement à partir du système de vision ;
7. Réaliser des expériences pour valider ses performances et le bon fonctionnement de l'architecture de contrôle ;

1.4 Plan du mémoire

Un état de l'art est présenté, au chapitre 2, sur les différentes thématiques du projet : robotique aérienne, opérations de maintenance sur des structures difficiles d'accès, contrôle d'un bras robotique, téléopération, interface haptique et vision par ordinateur. Le chapitre 3 présente un résumé de la problématique et des objectifs de recherche qui en découlent. Le chapitre 4 correspond à la présentation de l'architecture de contrôle réalisée et aux résultats obtenus, cela sous la forme d'un article soumis au journal Mechatronics de Elsevier. Dans le chapitre 5, l'architecture de contrôle est plus amplement détaillée avec des modes de contrôle et des résultats complémentaires. Le chapitre 6 permet d'ouvrir une discussion générale sur les travaux réalisés lors de cette étude pour enfin conclure au chapitre 7.

CHAPITRE 2 REVUE DE LITTÉRATURE

2.1 Maintenance de structures , difficiles et dangereuses d'accès, par robots téléopérés

La maintenance des structures aériennes est un point important dans le bon fonctionnement d'un système tout au long de sa durée de vie. Ces structures, dites aériennes, correspondent à des éoliennes, ponts, bâtiments industriels ou installations sur des parois rocheuses qui sont par nature difficiles et dangereuses d'accès pour un technicien de maintenance. Ces maintenances de structure nécessitent donc la mobilisation de techniciens cordistes afin d'accéder à la zone de travail. Ainsi cette mobilisation demande une réflexion en amont pour sécuriser l'environnement de travail et pour le bon déroulement de la mission dans ces zones difficiles d'accès. Cela peut aussi nécessiter l'installation de matériels de sécurisation (filets, système de communication, voies sécurisées, etc.) qui sont coûteux et impliquent des durées de mission et de mise en œuvre logistique importantes.

La maintenance nécessaire à ces structures aériennes peut être de différentes formes, une synthèse figure 2.1 rassemble les différents procédés proposés par des entreprises canadiennes (Ramaccess¹, Mistras²) pour de la maintenance effectuée par des techniciens cordistes.

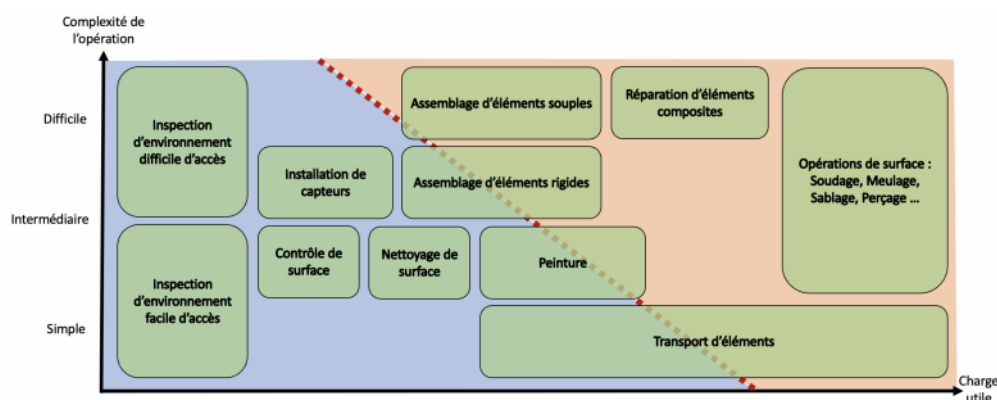


FIGURE 2.1 Ensemble des procédés effectués par des techniciens cordistes en fonction de sa complexité et de la charge utile nécessaire. La ligne pointillée délimite les procédés réalisable actuellement par des solutions de robotique aérienne. Ainsi les deux zones représentent l'aptitude (en bleu) et l'inaptitude (en orange) des solutions de robotiques aériennes actuelles pour ces procédés spécifiques.

1. <https://www.ramaccess.ca/ramsolutions>

2. <https://mistrasgroup.fr/>

Ces procédés sont disposés suivant deux critères, le premier étant la complexité de mise en œuvre et le second la charge à transporter. Un drone étant limité dans sa capacité d'emport, le bras robotique embarqué devra être optimisé selon un critère de masse ce qui peut influencer sa rigidité structurelle, ses dimensions, ses performances. De ce fait, la complexité des opérations à effectuer en est amoindrie. Ainsi ces deux critères, capacité d'emport et de complexité d'opérations, seront déterminants pour l'exécution des opérations à effectuer par le drone.

Enfin la variabilité de l'environnement est un critère important lors de l'exécution d'opération de maintenance. La figure 2.2 montre la complexité des opérations réalisables selon la variabilité de l'environnement de travail et du degrés de téléopération du système robotique. Les opérations en extérieur, ayant un environnement très variable, nécessitent une téléopération active de l'opérateur pour remédier à des risques pour le système robotique. À l'inverse, un environnement de travail à faible variabilité pourra faire l'œuvre d'opération réalisée par un système robotique avec une téléopération assistée voire autonome.

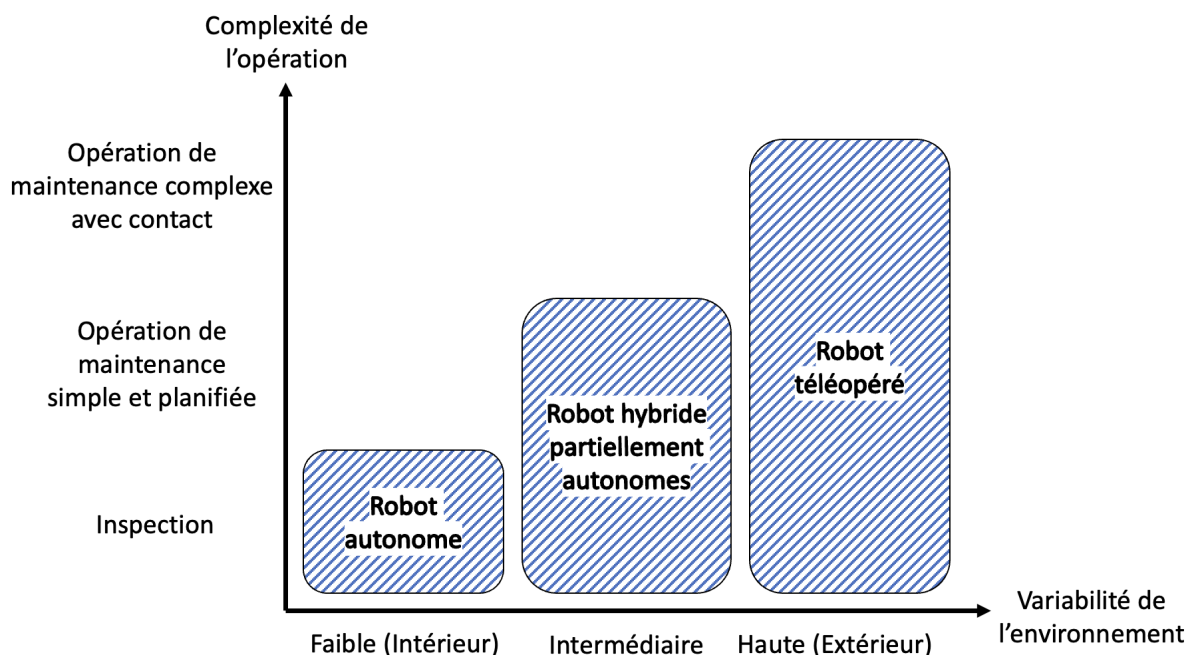


FIGURE 2.2 Classification des degrés de téléopération d'un système robotique selon la complexité de l'opération effectuée et la variabilité de l'environnement dans lequel il évolue

2.1.1 Robotique mobile

La robotique mobile est un aspect important dans la maintenance de structure par téléopération. Notamment les robots grimpeurs téléopérés qui ont la capacité d'accéder aux structures

aériennes [12].

De nombreuses applications de robots grimpeurs sont actuellement dédiées pour de la maintenance d'éolienne terrestre [13] ou en mer [14]. L'entreprise Aeronos commercialise un service de maintenance de pale d'éolienne à l'aide d'un robot grimpeur [1]. Ce robot est équipé d'outils spécifiques pour réaliser différents types d'opérations comme de l'inspection, du nettoyage ou de la réparation sur la pale.

Développé par l'Institut de recherche d'Hydro-Québec (IREQ), le robot SCOMPI est utilisé pour des applications d'intervention de meulage, de soudage et de martelage sur des turbines de centrales hydroélectriques [3]. Certains outils ont été développés spécialement pour fournir, à distance, des tâches de contact avec un effort modéré voire important. Le robot est muni d'un bras robotique avec cinq axes de rotation et sa base qui est guidée par une liaison glissière [15].

Le robot BR-8 a été développé par l'entreprise Rope Robotics [2] afin de mener des opérations de réparation au niveau du bord d'attaque des pales d'éoliennes. Le BR-8 a la capacité d'être téléopéré afin d'effectuer une inspection visuelle sur le bord d'attaque de la pale qui par la suite pourra nécessiter une réparation. Une fois la zone endommagée détectée, le robot effectue des opérations de polissage, de nettoyage puis de recouvrement. Le BR-8 est composé de deux parties, la première est sa plateforme mobile lui permettant de se déplacer et de se fixer sur la pale à l'aide d'un système de ventouse. La seconde partie est son manipulateur robotique UR10 provenant de la société Universal Robots [16], celui-ci à six degrés de liberté et est équipé d'une caméra pour faciliter la téléopération.

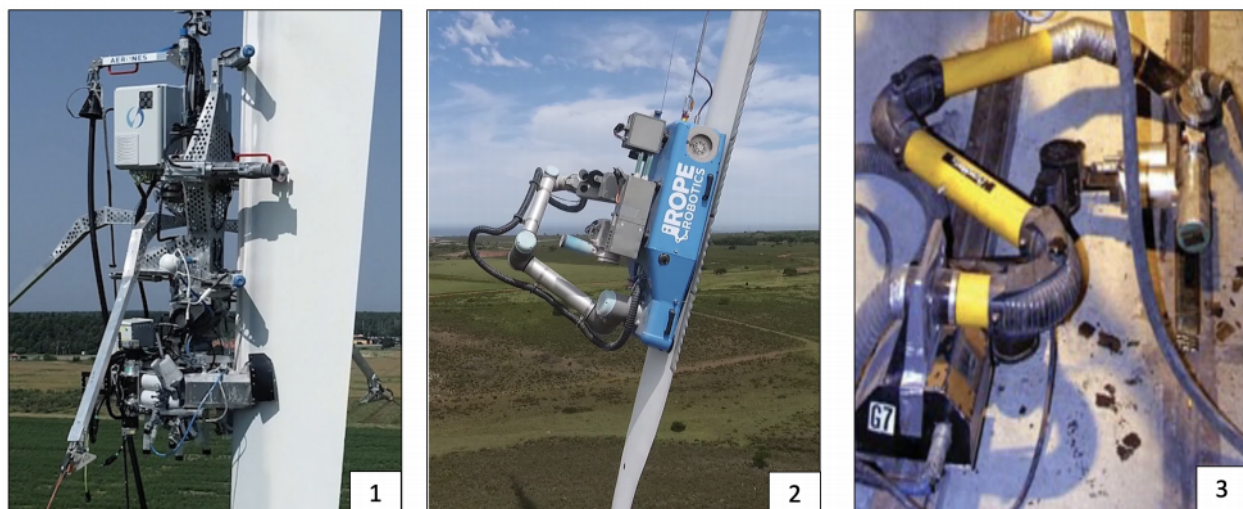


FIGURE 2.3 Visuel des trois systèmes robotiques présentés : 1) Robot de maintenance de pale d'éolienne (reproduit avec l'autorisation de Aeronex [1]) - 2) BR-8 (reproduit avec l'autorisation de Rope Robotics [2]) - 3) SCOMPI (photo extraite de [3])

La figure 2.3 donne un visuel des trois systèmes robotiques présentés au-dessus. On peut remarquer que ces solutions robotiques ne sont pas adaptables à une plateforme aérienne car leur masse est trop importante. Cependant leur structure, principalement 6 axes, est efficace pour les opérations de maintenance souhaitées.

2.1.2 Robotique aérienne

La robotique aérienne est un domaine de recherche relativement récent. Il s'agit, pour la majorité du temps, de l'utilisation d'un bras robotique, ayant plusieurs degrés de liberté, implémenté sur un drone volant [17] [18]. Les problématiques importantes dans le développement de projet de robotique aérienne sont l'interaction de l'ensemble du système robotique (drone volant + bras robotique) avec son environnement et la charge d'emport admise par le drone [19].

En effet, l'utilisation d'un bras robotique additionné au déplacement d'un drone ajoute de la complexité dans la pose de celui-ci, par rapport à une application de robotique mobile où le référentiel est fixe. De ce fait, les opérations de maintenance classique, nécessitant des interactions de contact avec la surface de travail, deviennent complexes à réaliser vis-à-vis d'un bras robotique fixe ou d'un opérateur humain [19]. Enfin, ces dernières années, des progrès de performances d'emport ont été effectués grâce notamment à l'amélioration des batteries, une structure plus légère et une propulsion plus performante [20].

Les solutions de robotiques aériennes actuelles sont principalement orientées sur du déplacement de charge de masse inférieure à 10 kg, l'installation d'éléments et l'inspection visuelle. Le projet CARMA (Compact AeRial MANipulator) [4] utilise un bras manipulateur, quatre axes, installé sur un drone ASCTEC NEO pour de l'inspection visuelle majoritairement.

L'entreprise Prodrone produit des drones PD6B-AW-ARM [21] capable de transporter des objets à partir de deux bras robotique, appelés griffe. Le système robotique est aussi utilisé pour de l'inspection et de la cartographie. A contrario de la masse d'emport des solutions de robotiques aériennes classiques, la masse emport maximale du système est de 20kg pour une autonomie entre 10 et 20 minutes.

Le projet HYFLIERS [5] permet l'inspection de tuyaux à partir d'un bras robotique redondant SAP (Snake Arm with pan and tilt Probe) [22]. Celui-ci a la possibilité d'accéder à des zones étroites pour de l'inspection.

L'utilisation d'un manipulateur Delta installé sur un drone volant [6] a démontré son efficacité pour des opérations de soudage et ponçage sur des surfaces métalliques courbes. La masse du manipulateur Delta avec son outil embarquée sur le drone est d'environ 3,5 kg. La figure 2.4 montre les quatre solutions de robotiques aériennes présentées précédemment.

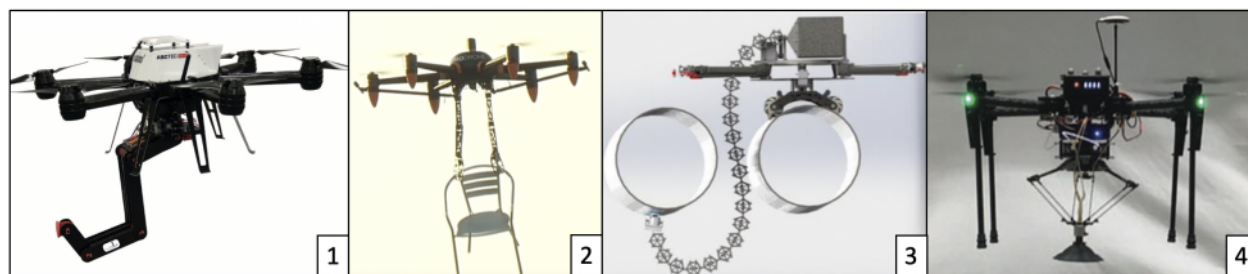


FIGURE 2.4 Visuel des quatre systèmes de robotiques aériennes : 1) Compact AeRial Manipulator (photo extraite de [4]) 2) PD6B-AW-ARM (Reproduit avec l'autorisation de Prodrone) 3) HYFLIERS (photo extraite de [5]) 4) Manipulateur Delta intégré à un drone (photo extraite de [6])

Nous pouvons noter que la principale utilisation des technologies de robotique aérienne est destinée à des applications nécessitant des efforts relativement faibles au niveau du bras robotique, telles que l'inspection visuelle et la manipulation de charge légère. Ainsi, des performances plus importantes sont nécessaires afin d'obtenir une plus grande diversité d'opération de maintenance.

2.2 Bras robotique léger

2.2.1 Architecture mécanique

Les bras robotiques légers voire ultralégers ont été développés à l'origine pour des applications dans le domaine spatial [23] [24] [25], cela afin d'obtenir les meilleures performances entre la masse du bras et sa charge utile. L'architecture mécanique des bras robotiques légers a subi des optimisations au cours du temps avec l'utilisation de technologie de réducteurs différents et plus performants, tels que les réducteurs harmoniques. Il en est de même pour la motorisation qui est passée sur l'utilisation de technologie de moteurs brushless plus performants.

Le projet de bras robotique LARAD [9] (Lightweight Advanced Robotic Arm Demonstrator) a pour objectif le développement d'une solution robotique permettant la récupération de d'échantillons martien à partir d'un bras robotique intégré à un rover. Un démonstrateur terrestre 6 DDL de 15kg a été conçu avec une portée de 2m de long avec une charge utile déployée allant jusqu'à 6kg. Les articulations sont conçues afin d'optimiser le rapport couple/masse. De plus, la conception est faite à partir de fabrication additive 3D (ALM) en alliage de titane. Du fait de sa conception modulaire, les articulations du bras LARAD peuvent parvenir à trois tailles de bras différentes.

Le bras robotique CAESAR [8] (Compliant Assistance and Exploration SpAce Robot) est un projet dans le domaine du spatial afin de procéder à des opérations d'assemblage, de maintenance, de réparation et d'enlèvement de débris en orbite. Ce bras fait suite aux travaux du centre de recherche allemande DLR sur le robot LWR [26](LightWeight Robot) et le développement de systèmes robotiques à commande par force/couple. Enfin le bras CAESAR est basé sur une conception modulaire avec des articulations identiques car l'absence de gravité en orbite implique une répartition plus uniforme des efforts sur les articulations. Cette modularité permet aussi d'obtenir plusieurs configurations de bras ainsi qu'une cinématique différente.

L'entreprise KINOVA commercialise deux modèles de bras robotique léger pour de l'utilisation cobotique. En effet, la faible masse des bras robotiques permet de les utiliser dans des applications de cobotique avec un aspect de sécurité vis-à-vis de l'utilisateur. Le premier modèle, le GEN2 [27] est un bras robotique avec une masse de 4,4 kg et une charge utile de 2,6 kg (sous la version six DDL). Il a une portée allant jusqu'à 985 mm, ce qui fait de lui un candidat intéressant pour une application de robotique aérienne. Cependant sa charge utile n'est pas assez importante pour entreprendre des opérations de maintenance sous charge. Le deuxième modèle commercialisé par KINOVA est le modèle GEN3 [28], est légèrement

différent du GEN2, avec une masse de 7,2kg et une charge utile de 4 kg (sous la version six DDL). Pour ces deux modèles un septième DDL peut être configuré pour obtenir une meilleure maniabilité mais ajoutant de la masse et réduisant la charge utile.

La figure 2.5 montre les cinq bras robotiques légers présentés précédemment.

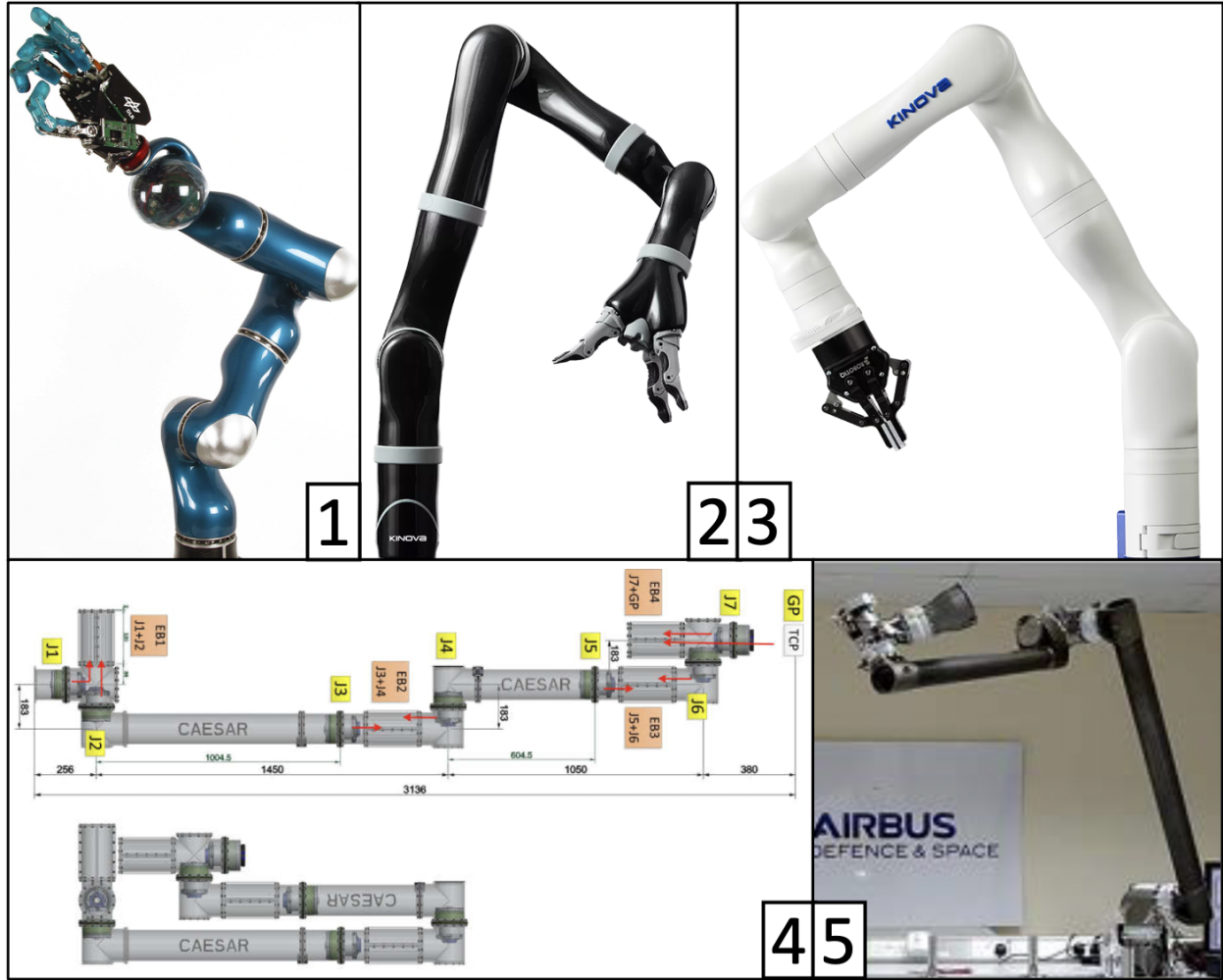


FIGURE 2.5 Bras robotiques légers : 1) DLR LWR arm and hand (photo tirée de [7]) - 2) KINOVA GEN2 - 3) KINOVA GEN3 - 4) CAESAR (illustration tirée de [8]) - 5) LARAD (photo tirée de [9])

2.2.2 Architecture de contrôle

L'architecture de contrôle d'un bras robotique léger est un aspect important dans le fonctionnement de celui-ci, les travaux menés dans le domaine spatial répondent en partie à des problématiques de contrôle à distance sur des structures légères et de faibles rigidités [24]. Les délais de communication entre l'opérateur et le système robotique ont aussi un effet im-

portant dans la commande d'un bras à distance, cela impliquant d'adapter l'architecture de contrôle pour obtenir une commande stable [29].

Le projet LARAD [9] intègre un microcontrôleur à la base de son bras pour permettre le contrôle du bras à distance via une communication par bus CAN. Chaque articulation est connectée à ce microcontrôleur ainsi que des capteurs de température, couple, position, courant et accélération, cela afin de surveiller l'ensemble du bras. L'architecture de contrôle intègre quatre modes de contrôle, une commande en boucle ouverte, un PID en position, en vitesse et en couple. Le système est scindé en deux parties avec une partie correspondant au système de contrôle au sol et l'autre au système évoluant dans l'espace (constitué de l'ordinateur embarqué et du bras robotique).

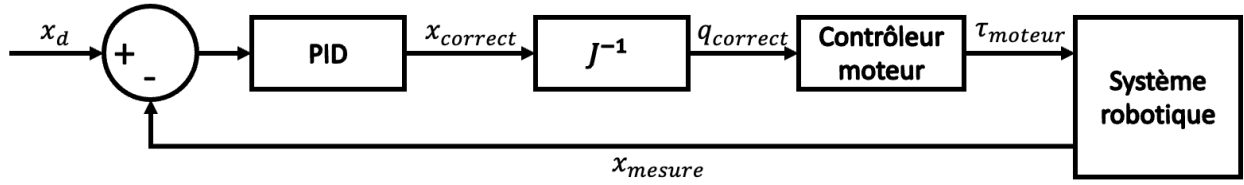


FIGURE 2.6 Schéma régulateur PID en position

Dans le projet [30], il est utilisé un robot BAXTER avec deux bras robotiques. L'architecture de contrôle est établie en deux espaces de travail, le premier est local par rapport à l'opérateur et correspond aux outils d'interfaçage homme-machine. Le second est à distance et correspond au système robotique avec sa partie asservissement moteur. Un PID est utilisé pour chaque bras afin de contrôler les articulations en position (figure 2.6 ou en orientation, ce choix de mode de PID est réalisé par l'opérateur).

Le projet SHERPA [31] est un bras robotique installé sur un rover mobile pour effectuer des opérations consistant à attraper des drones en vol. L'architecture de contrôle du bras utilise un environnement ROS (Robot Operating System) pour effectuer les interactions entre tous les éléments constitutifs du système (contrôleur moteur, capteurs, batterie, interfaces opérateur, etc...). Cette architecture présente l'avantage d'être modulaire en intégrant des modules de planification de trajectoire, de planification de tâche, de contrôle des articulations et d'utilisation de capteurs.

Le projet CAESAR [8] utilise une architecture de contrôle relativement similaire au projet LARAD. Celui-ci utilise des modules RobotDrive pour ses moteurs, ces modules sont contrôlés à une fréquence de 20kHz avec une couche intermédiaire de contrôle à 3kHz. Cela afin de compenser les effets de friction des réducteurs harmoniques/moteur et la flexibilité des articulations. Enfin une dernière couche de contrôle est effectuée pour mettre en relation

toutes les articulations du bras, ce contrôle est réalisé à partir d'un ordinateur externe au manipulateur et s'exécute à une fréquence de 500Hz. La figure 2.7 représente l'architecture de contrôle du manipulateur CAESAR.

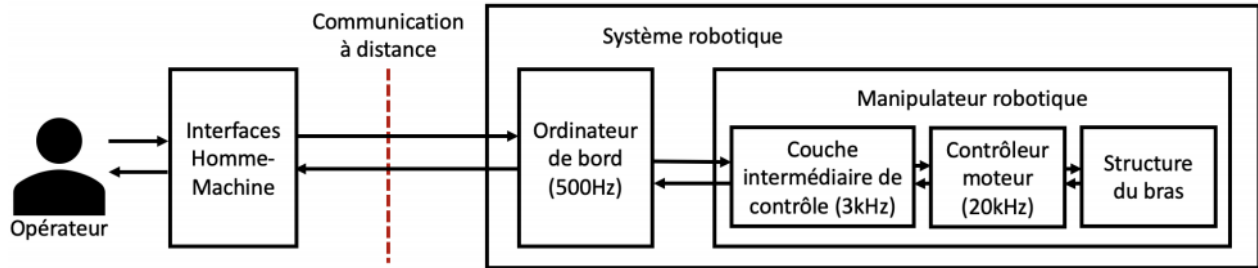


FIGURE 2.7 Schéma de l'architecture de contrôle du manipulateur CAESAR

À partir de cette architecture contrôle, deux modes de commandes sont configurés, la première est une commande en position prenant uniquement le retour en position des articulations du bras et la seconde est une commande cartésienne en impédance. La commande en impédance est une méthode de contrôle permettant de mettre en relation la force et la position du bras robotique. Elle est souvent utilisée par des systèmes robotiques ayant des interactions de contact avec leur environnement. En exemple, l'article [32] étudie les effets d'une téléopération haptique par une commande par impédance avec des forces générées par retour visuel. De plus, l'étude [33] propose un contrôleur d'impédance d'un robot industriel pour obtenir des mouvements stables en contact avec une surface.

Les différents travaux présentés permettent de répondre en partie à une problématique de conception d'une architecture de contrôle pour un bras robotique léger. Cependant, des éléments de contrôle restent à définir, on peut relever un manque d'information sur les performances de contrôle attendu des systèmes robotiques présentés, et les interactions entre le système robotique, sa commande à distance et l'environnement d'exécution.

2.3 Téléopération d'un bras robotique

La téléopération robotique (nommé aussi télérobotique) est un aspect important dans le contrôle de robots, celle-ci est utilisée dans de multiples domaines, notamment le contrôle de véhicules aériens, terrestres ou spatiaux [34] ainsi que la chirurgie à distance [35].

La téléopération robotique permet à l'opérateur de commander et contrôler un manipulateur robotique à distance et à travers une interface. Les mouvements désirés par l'opérateur sont envoyés pour être restitués le plus fidèlement possible sur le manipulateur robotique. Un

retour sur le mouvement du manipulateur est possible et est principalement lié à un retour d'effort. On définit ce type de téléopération comme bilatéral [36].

Les principales raisons d'utilisation de la télérobotique sont les suivantes [37] :

- Environnement dangereux et zone difficile voire impossible d'accès pour l'opérateur [38]. Comme exemples, nous pouvons prendre les zones aquatiques à profondeur importante, les zones soumises à la radioactivité ou les zones escarpées en hauteur (éolienne, pont, parois rocheuses) [34].
- Besoin de performances (ergonomie, force, réduction des tremblements, etc.) pour l'opérateur [39]. Un exemple est le domaine de la chirurgie [35] où il est nécessaire pour le chirurgien d'avoir un mouvement très précis et sans tremblement.

L'interface opérateur permettant la téléopération est nommée interface haptique, celle-ci permet d'acquérir le mouvement de l'opérateur dans l'espace. L'interface haptique est capable de fournir des efforts suivant plusieurs voire tous les axes de rotation et de translation de l'espace. Ces efforts proviennent des efforts captés par le système robotique ce qui permet à l'opérateur d'avoir un ressenti plus immersif du mouvement.

La figure 2.8, montre les différentes interfaces haptiques commercialisées et leur type d'architecture. Il existe deux types d'architectures, série et parallèle, ce qui permet d'obtenir des caractéristiques de performances en termes de nombre d'axes de mouvement, de précision, de rigidité et d'effort maximum aux axes.

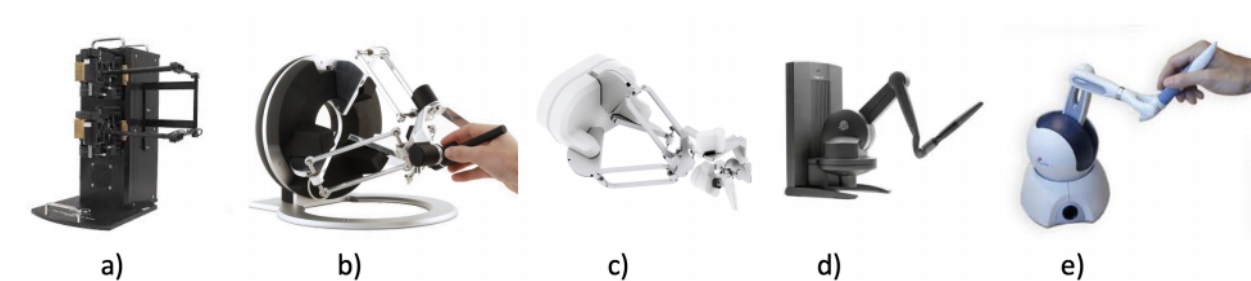


FIGURE 2.8 Interfaces haptiques commercialisées : Architecture parallèle : a)HD² High-Definition Haptic Device⁸ , b)Omega.6⁹, c)Lambda.7¹⁰ Architecture série : d)TouchX¹¹ e)TouchTM 12

La téléopération peut se caractériser sous trois principaux modes [40], ces derniers sont représentés dans la figure 2.9 :

- Téléopération directe, l'opérateur utilise une interface haptique pour contrôler directement le système robotique. Ce mode est utilisé pour des mouvements ne nécessitant pas une grande précision et répétabilité. En effet, les performances de ce mode sont variables et fonction de la dextérité de l'opérateur. Le retour haptique perçu par l'opé-

- rateur est basé sur les capteurs de position et de force intégrés au système robotique.
- Téléopération supervisée, l'opérateur utilise une interface haptique pour contrôler un système robotique physique. Cette téléopération se déroule à travers une modélisation numérique de l'environnement et du système robotique puis est retranscrite au système physique. Ce mode permet d'assister l'opérateur dans les tâches à effectuer en programmant des stratégies de déplacement, en étant plus précis et répétable dans les mouvements. Comme dans la téléopération directe, le retour haptique perçu provient des capteurs de position et de force intégrés au système robotique. À cela s'ajoute des informations générées par l'environnement numérique, par exemple un effort répulsif à l'approche d'un obstacle.
 - Téléopération virtuelle, l'opérateur utilise une interface haptique pour contrôler un système robotique virtuel dans un environnement virtuel. Ce mode est utilisé en phase d'apprentissage du système par l'opérateur puis pour valider ses compétences. Il permet aussi d'essayer de nouvelles stratégies de déplacement. En effet, à l'instar d'une téléopération directe, la téléopération virtuelle n'est pas soumise à des problèmes de répétabilité, de sensibilité à l'environnement et de dégradation du matériel. Le retour haptique perçu par l'opérateur est donc entièrement simulé, cette simulation se doit d'être le plus fidèle possible à la réalité.

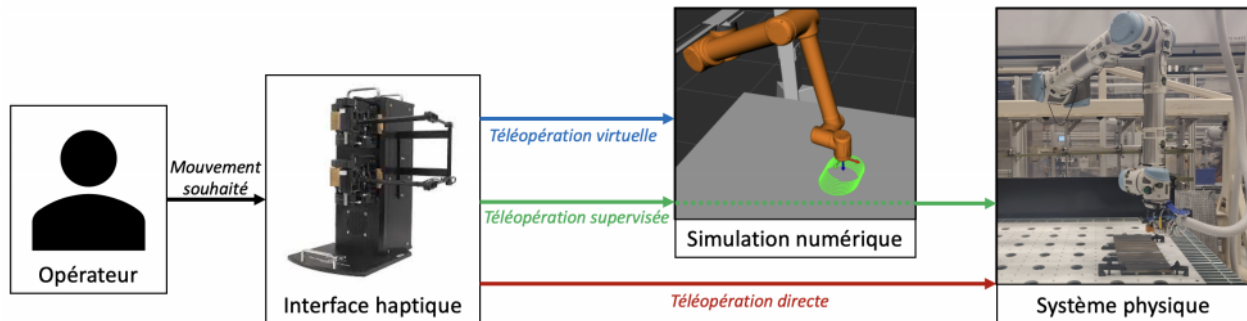


FIGURE 2.9 Schéma représentant les trois principaux modes de téléopération

2.3.1 Retour d'effort

La téléopération par retour d'effort est caractérisée par deux paramètres principaux :

- Sa stabilité, celle-ci est importante pour permettre de retranscrire le mouvement de l'interface haptique en mouvement pour le système robotique. Des instabilités dans le système sont présentes et sont liées principalement à du bruit dans les mesures, des approximations de modèle, des taux d'échantillonnage trop faible et des délais de communication trop important. La notion de passivité est introduite dans le paramètre

de stabilité, celle-ci correspond à la nécessité que le système mécanique soit dissipatif en termes d'énergie dans son environnement [41].

- Sa transparence, celle-ci se définit comme la capacité du système de téléopération à restituer à l'opérateur le plus fidèlement possible les sensations du mouvement du système robotique dans son environnement. La transparence est considérée comme la performance du rendu haptique de la téléopération [42].

Une modélisation d'un système de téléopération peut être réalisée à partir de la relation (2.1) décrite dans l'article de Hannaford [43]. Le système de téléopération se modélise par un réseau à deux ports qui relie la position/orientation X_{in} et la force/couple F_{in} de l'opérateur à la position/orientation X_{out} et la force/couple F_{out} de l'effecteur du robot :

$$\begin{bmatrix} F_{out} \\ X_{out} \end{bmatrix} = \begin{bmatrix} h11 & h12 \\ h21 & h22 \end{bmatrix} \begin{bmatrix} F_{in} \\ X_{in} \end{bmatrix} \quad (2.1)$$

$h11$ représente l'impédance mécanique de l'interface haptique et $h22$ l'admittance mécanique du système téléopéré ; $h12$ représente l'aptitude de transmettre les efforts de l'effecteur robotique à l'interface haptique ; $h21$ représente l'aptitude de la commande du système robotique à suivre les mouvements de l'interface haptique.

2.3.2 Vision

La vision par ordinateur est un aspect important pour une solution de téléopération de système robotique. Celle-ci permet d'obtenir en temps réel un visuel de l'environnement de travail du robot à l'opérateur, cela afin de prendre décisions, planifier et opérer des actions sur le système.

Les systèmes de vision/perception robotique ont deux principales fonctions, l'acquisition des données des capteurs et le traitement de ces données. À partir du système de perception robotique, des solutions de reconnaissance d'objet [44], de détection d'obstacles [45] [46], de reconnaissance de l'environnement et de suivi d'objet pourront être implémentées [47].

Les capteurs de vision sont principalement utilisés comme des capteurs extéroceptifs, c'est-à-dire qu'ils recueillent des données venant de l'environnement extérieur. À l'instar des capteurs proprioceptifs qui recueillent des données du système. Les capteurs de vision ont des applications dans de multiples domaines tels que le tri d'objet, l'inspection de matériel, le déplacement de système, la surveillance [48].

De nombreuses entreprises conçoivent des capteurs de vision intégrant la partie acquisition de l'image et la partie traitement de l'image avec des technologies de reconnaissance d'objets

et de génération d'environnement numérique. Le tableau 2.1 suivant résume les différentes technologies de caméra existantes avec leurs caractéristiques techniques et leurs utilisations.

TABLEAU 2.1 Tableau des technologies de caméra utilisées pour des systèmes robotiques

Nature du capteur	Camera 2D	Camera à lumière structurée	Camera Stéréo	Camera Infrarouge
Applications	Détection d'objet, surveillance.	Détection d'environnement 3D, reconnaissance d'objets/mouvements.	Déplacement dans un environnement inconnu, reconnaissance d'objets/mouvements, génération d'environnement numérique 3D	Reconnaissance d'objets dans des environnements à faible luminosité.
Avantages	Coût faible, intégration simple et bonne fiabilité.	Adapté à des environnements complexes et variables.	Comparable à la vision humaine. Fonctionnement en intérieur et extérieur.	Adapté à des environnements à faible luminosité, coût faible.
Inconvénients	Performance réduite pour une utilisation autonome.	Coûteux et nécessite un GPU pour le traitement du nuage de point.	Coûteux et nécessite des performances CPU. (2 caméras)	Sensible à l'exposition de lumière direct. (infrarouge et soleil)
Exemple de modèle commercialisé	Robotiq Wrist Camera 	Microsoft Kinect v2 3D 	Zed 2i 	SPOT CAM+ Infrarouge 
Résolution	640 x 480 , 30 FPS	Couleur : 1920 x 1080 Profondeur : 512 x 424 30 FPS	2560x720, 60 FPS	640 x 512, 7.5FPS
Masse	230g	1.4 kg	166 g	7,5 kg
Dimensions	22mmx88mmx75mm	66 x 249 x 67 mm	175 x 30 x 43 mm	334 x 227 x 357 mm
Alimentation	USB 2.0, 24V DC	USB 3.0, 12V DC	USB 3.0, 5V DC	USB 3.1, 12V DC

Récemment, de nombreux projets utilisent des systèmes de vision pour la téléopération de système robotique. Le projet Baxter [30], présenté précédemment, est constitué de deux bras robotiques afin d'effectuer des tâches de surface. Le robot est équipé de deux caméras qui permettent à l'opérateur de visualiser la surface de travail des bras. De plus, les données de vision permettent de générer un environnement numérique du système robotique.

Un second projet SHERPA [44] utilise aussi des solutions de vision pour son bras robotique. Le bras a pour objectif d'attraper, de manière autonome, des drones en vol stationnaire dans un environnement extérieur inconnu. Ainsi dans ce projet, une caméra monoculaire est installée sur le bras robotique et est utilisée pour localiser et estimer la pose du drone. Enfin un déplacement du bras est effectué pour attraper le drone.

CHAPITRE 3 DEMARCHE DE TRAVAIL

3.1 Résumé de la problématique

Dans une optique de réduire les risques humains et d'accéder à des endroits difficiles d'accès lors d'opération de maintenance, le développement de technologies robotiques spécifiques aux structures aériennes a été entrepris. Les applications robotiques aériennes actuelles se concentrent majoritairement sur des opérations nécessitant de faibles efforts et précisions ce qui permet d'obtenir une masse d'emport très faible pour le drone volant. A contrario des systèmes de robotique mobile qui peuvent opérer à des efforts plus importants mais ayant une structure plus massive et imposante, ce qui est incompatible avec des critères de robotique aérienne. Il est à mentionner que les bras robotiques, développés pour des applications dans le domaine spatial, ont des contraintes de maximisation des performances pour une masse réduite. Ces développements sont donc en accord avec les contraintes liées à notre système.

Ainsi, à la suite des travaux effectués précédemment sur l'architecture mécanique du bras, une problématique de contrôle à distance du bras robotique intervient. L'état de l'art actuel permet en partie de proposer une architecture de contrôle d'un bras robotique léger, à laquelle s'ajoute une téléopération via une interface haptique et un retour visuel, qui permet d'intégrer l'opérateur dans la boucle de contrôle. Enfin, l'intégration de tous ces éléments, pour la réalisation d'un système robotique complet, amènera des problématiques supplémentaires en termes d'intégration et de performance.

3.2 Objectif général

L'objectif principal de ce projet est de proposer une architecture de contrôle et la téléopération d'un bras robotique afin d'effectuer des opérations de maintenance sur une structure aérienne. Les travaux feront état de la conception d'une architecture électrique du bras avec le développement de la commande des moteurs. Une modélisation 3D sera incluse et permettra la simulation de différents systèmes implémentés, tels que le planificateur de trajectoire, le système de vision et d'interface haptique.

3.3 Objectifs spécifiques

Des objectifs spécifiques sont étudiés dans l'article au chapitre 4 avec des résultats complémentaires au chapitre 5 :

1. Modéliser le système
2. (a) Définir une architecture commande du système robotique.
(b) Modéliser la boucle de contrôle du bras robotique.
(c) Implémenter la commande des articulations du bras.
(d) Réaliser la liaison entre les différents composants du système robotique (bras, drone, espace opérateur).
3. Implémenter une téléopération.
(a) Modéliser l'interface haptique.
(b) Implémenter l'interface haptique dans la boucle de contrôle.
(c) Modéliser le bras robotique et ses interactions dans son environnement à partir du système de vision.
(d) Implémenter un système de vision.
4. Définir les modes de contrôle du bras.
(a) Modéliser les modes de contrôle.
(b) Implémenter un algorithme de génération des surfaces de travail.
(c) Implémenter les modes de contrôle (suivi de trajectoire générée, déplacement libre par l'opérateur, déplacement assisté).
5. Réaliser des expériences.
(a) Caractériser les paramètres mécaniques du système (friction, rigidité, etc).
(b) Évaluer les performances de l'architecture de contrôle.
(c) Valider le bon fonctionnement du système téléopéré.

CHAPITRE 4 ARTICLE 1 : DESIGN AND CONTROL OF A UAV-MOUNTED ULTRALIGHTWEIGHT ROBOT ARM FOR CONTACT TASKS

Le présent article a été soumis 8 Novembre 2022 pour publication dans la revue scientifique Mechatronics (Elsevier).

Florian Coiffard, Valentin Omé, Lionel Birglen et Jason Michel Lambert.

Abstract

This paper details the design and control of a novel ultralightweight six-axis serial robot arm remotely operated and mounted on a UAV with docking capabilities. This robot is intended to be capable of performing local grinding/polishing tasks on hard-to-reach areas. The characteristics and variety of targeted work scenarios are first discussed and led to the definition of a specific arm geometry. The associated 3D model of this arm was then used to design an appropriate controller. This controller aims at achieving high reliability during remote control while performing grinding/polishing tasks and can move the arm while automatically avoid both singularities and collisions. A visual and haptic teleoperation system is integrated to the controller in order to assist the operator of the system, allowing him/her to define trajectories to be followed by the robot. Experiments using this control system were carried out on a UR10 robotic manipulator to validate the use of the teleoperation interface and establish benchmark performance.

4.1 Introduction

4.1.1 Motivation

Maintenance of structures such as wind turbines, bridges or industrial plants is typically conducted by rope access technicians performing various tasks at heights. Such a deployment is time-consuming and can be dangerous, making it slow and tedious to perform even the most trivial tasks. Aerial [49] and climbing robots [50] have been proposed to address this problem while ultralightweight high performance robots are typically designed for space robotics projects [9, 51]. To the best of the authors' knowledge, a robot that is meeting an acceptable reach, a very low weight, can produce sufficient torques for material removal operations, is remotely controlled, and demonstrates compatibility with an aerial system has not yet been

developed. This paper aims at filling that gap.

Rope access technicians perform a wide range of tasks in diverse work environments. These tasks go from visual inspections to Non Destructive Testing (NDT), corrosion, dust or painting removal, bolts replacements, coating, welding, etc. Typical examples are illustrated in Fig. 4.1. Many of these applications do not require great mobility from the worker on the surface, but rather a good dexterity and accessibility on surfaces of all inclinations, typically corresponding to ground, wall, and roof operations. To replace a human operator with an unmanned aerial vehicle (UAV), the first challenge is securing the latter to the work surface. A solution to this issue consists in the use of a docking system, an example of which is the Based Attachment Module (BAM) developed at the Aerospace Research Centre of the National Research Council (NRC) of Canada [11]. Combined to a high-lifting capacity drone, and after an approach phase, this BAM provides attachment of the aerial platform to diverse surfaces, under variable docking angles, even when the UAV's motors are shut down. Complemented by the robotic arm introduced in this paper, the UAV can then reproduce most of the work conducted by rope access technicians.

4.1.2 State of the art in aerial manipulation

Previous works in climbing, aerial, space, and collaborative robotics were the inspiration for the development of the novel robot arm discussed in this work. For instance, aerial robotics projects [49] have demonstrated solutions for structural inspection, sensor installation, NDT [52–54], and can extend to structural assembly tasks [55], payload manipulation [56, 57], and transportation with two robotics arms [58]. In some of these cases, UAVs are mobile during operation which makes the deployment of a contact task very complex. Aerial systems are of interest for the application discussed in this paper nonetheless because of their very low weight, high compactness, and the constraints that are met in their design. Unmanned maintenance in renewable energy applications using climbing robots is also a related topic of interest, as they are leading to better efficiency and extended service life of complex structures [50, 59]. For example, the rope-deployed BR8 robot from Rope Robotics integrates a UR10 robot arm for mapping, cleaning, and coating tasks on wind turbines [50]. Hydro-Quebec Scompi robot performs grinding, polishing, hammering, and welding operations on hydropower equipment, often translating on a rail to access hard-to-reach areas [60, 61]. Grit blasting is also performed using heavy walking robots [62] or magnetic crawlers. Very agile solutions exist but with low payloads such as the BladeBug robot [50].

For a robotic process with a fixed base, i.e. moving from cited aerial and climbing robots to more classical robot design requirements, a spectrum of lightweight robots has been proposed



FIGURE 4.1 Examples of tasks performed by rope access technicians (wind turbine maintenance on the left, NDT of water pipes on the right), courtesy of Vertica (Brittany, France)

and can be classified through a measure of their $\text{Payload} \times \text{Reach}$ as illustrated in Fig. 4.2. For material removal, insufficient performances are generally obtained with robots under 10 kg, either due to low payloads preventing the use of heavy end-effectors (EE), or because of very small reaches. The German Aerospace Center DLR developed three generations of Light Weight Robots (LWR) specifically addressing these issues. The latest one, namely the LWR III, achieved a payload to weight ratio of 1 with a very compact design and demonstrated a 936 mm reach with 7 axes [51, 63]. Service robotics manufacturer Kinova also specializes in the design of ultralightweight robots [64] such as the Jaco² or Gen2 arms which both have a reach of about 900 mm, 6 axes and weight less than 5 kg. Similarly with a low weight but also focusing on maximal reach, the UK Space Agency proposed the LARAD 6-axis robot arm and a reach of two meters with a weight under 15 kg was achieved [9]. Such performances have been pioneered by the DLR focus on high torque density drive trains during the design of the LWR III [65] with internal developments on high torque frameless permanent magnets inrunner motors, low weight Harmonic Drive gearheads [66] and high rigidity / low weight composite links [9, 64].

Robotic teleoperation is an important aspect of robot control and is already used in many applications including the control of aerial vehicles, ground vehicles, or space vehicles [34] as well as remote surgery [35]. Teleoperation is typically used in areas that are difficult to access or dangerous for an operator [38] and can also solve ergonomic issues [39]. There are different types of teleoperation amongst which the direct mode, giving total control to the operator, or supervised mode, facilitating intended actions but with a level of autonomy [34].

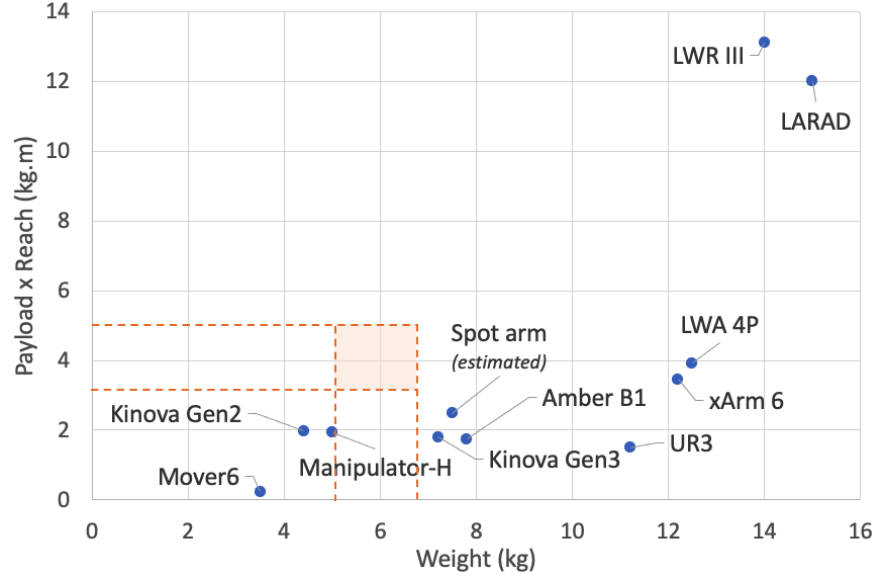


FIGURE 4.2 Payload $\times$ Reach as a function of weight for a selection of lightweight robots. The orange square represents the targeted performance

Haptic interfaces often comes in complement to teleoperation, in order to provide a better immersion for the operator by adding feedback using the sense of touch. Haptic interfacing is for instance used to provide critical learning information in flight simulators [67] or medical training [35].

The use of a robotic arm in an unknown environment to perform contact tasks requires a control architecture with a strong emphasis on safety. To this aim, a multi-level control approach is commonly used to plan and execute motions [34] [68]. Multi-level control can be implemented by a low-level approach [69] [70] with pre-registered tasks giving more reliability to the controller but not able to consider modifications in the working environment. A high level approach on the other hand, as demonstrated in [71], allows to plan more complex tasks in a sequential way while taking into account the information coming from sensors and the working environment, e.g. obstacles. The combination of the two approaches, namely low- and high-level, has been implemented on many robotic arms such as the Baxter dual-arm robot [31] and the SHERPA arm [30] both of which are used in mobile environments.

4.1.3 Methodology and contributions

In this paper, a 6-axis serial robot kinematic structure is selected to meet the specifications arising from the use of a robotic arm mounted on a UAV. For example, the arm must meet the requirement of being used with several docking orientations (illustrated in Fig. 4.3). A

selection of joints modules and links are then introduced from their technical specifications (torque, dimension, weight) which result in the global mechanical architecture of the arm. A modeling of the kinematics and dynamics of the arm then follows. After the general mechanical layout of the arm is achieved, an electronic controller for the joints will be subsequently proposed and combined with a dedicated control algorithm. A multi-level approach is used and two control modes are presented. A supervised mode allowing the generation and execution of tasks predefined by the operator and a manual mode giving full control of the arm movements to this operator. The global architecture of the control system includes : task planner, trajectory planner, scene plan, obstacle avoidance, haptic interface, communication, compensation, motor controller, and sensor interface. Control experiments with a 3D numerical model of the arm as well as a UR10 robot arm will also be presented and discussed.

4.2 Design Requirements

4.2.1 System Environment

The actual deployment of a robotic arm with an aerial platform requires multiple steps : the approach phase, docking on the work surface (using the BAM in our case), rotors shutdown, and finally, robot arm usage. Both the BAM and drone pitch control capabilities are required to perform multi-angle docking (see Fig. 4.3). Dronevolt Hercules 20 (heavy lift drone) is used in this paper as the UAV platform because of its high nominal payload of approx. 15 kg of which 9 kg are permissible for the robot and its EE. The work surface is assumed to be flat and identical to the docking surface. Targeted applications of the system are mainly polishing and grinding operations for corrosion, paint or dust removal. To this aim, joint speeds in the arm are low but in a tele-operated context, have to remain close to commercial manipulators' typical values. With the robot being lightweight by requirement, a flexible structure is anticipated and the integration of an active compliance module for force control is likely to be required [72]. Targeted EE therefore consists in tools such as the 11.5 mm stroke Ferrobotics compact Active Contact Flange XS, weighting under 1.4 kg, and coupled to a lightweight die grinder, weighting under 1.3 kg, mounted in a configuration where its abrasive contact point is aligned with the compliance module's axis (see Fig 4.4).

4.2.2 Initial Choices

Initial design choices pertaining to the robot arm architecture can be made based on the system environment and are listed in Tab. 4.1. The rationale for selecting a 6 DoF serial architecture is that it generally allows a good dexterity, compactness and large range of

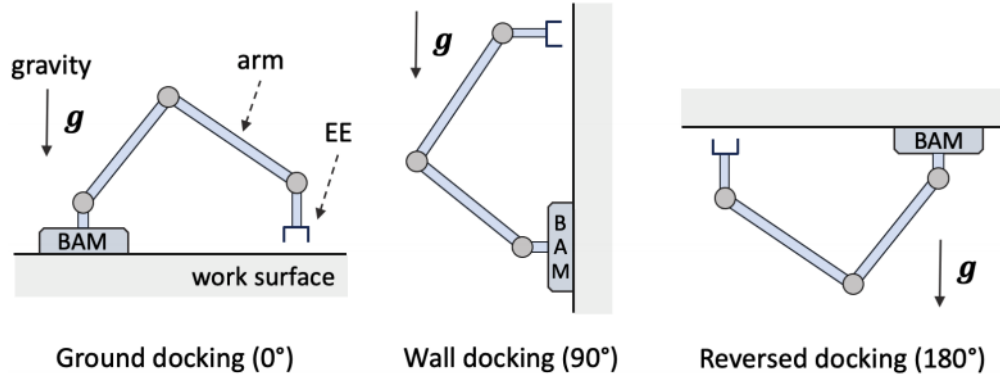


FIGURE 4.3 Geometries of the three docking scenarios considered in this paper (90 and 180 dockings are close to the examples shown in Fig. 4.1)

TABLEAU 4.1 Summary of initial design choices

Characteristic	Design choice
Mobilities	Serial robot with 6 Degrees of Freedom (DoF)
Configuration	Shoulder (2 DoF), elbow (1 DoF), wrist (3 DoF)
Wrist	3 DoF spherical wrist
Joint 1	Same large actuator for axes 1 to 3
Joint 2	Same small actuator for axes 4 to 6
Motors	Robodrive BLDC Inrunners ILM-series
Gearboxes	Harmonic Drive CPL, CSD series
Links	High stiffness carbon fiber tube
Drive	In-joint CAN-FD compatible controller
Power Supply	24V Lithium batteries
End Effector	Active Contact Flange + Die grinder + Flap disk

motion with reasonable weight, and indeed, shoulder-elbow-wrist anthropomorphic arms are commonly found in the literature dealing with lightweight robot designs [9, 51, 64]. Using a spherical wrist allows to decouple the inverse kinematic positioning and orientation problems which significantly simplifies its resolution. The design in this paper focuses on only two different joint modules, separating the 6 DoF between larger load axes (1 to 3) at the base and smaller load axes in the wrist (4 to 6). The axes 2, 3 and 4 are connected by high stiffness commercial carbon fiber tubes. This choice also allows for a modular assembly with easy replacement of components (tubes of different lengths, joint modules of different output torques and weights) if needed from one deployment to another. Selected motors are powered with DC 24 V voltage. CAN-FD compatible in-joint controllers (Moteus controller¹), each

1. <https://github.com/mjbots/moteus>

integrating a field oriented control driver and an absolute magnetic encoder, are used to control the brushless motors which are coupled to high reduction Harmonic Drive gearheads, required for accuracy, to increase output torques. Standard thin section roller bearings are combined with needle roller bearings on motor axes while IKO slim and ultraslim crossed roller bearing series² are used on output shafts for compactness and high load capacity with low weight. The general architecture of the robot arm is presented in Fig. 4.4.

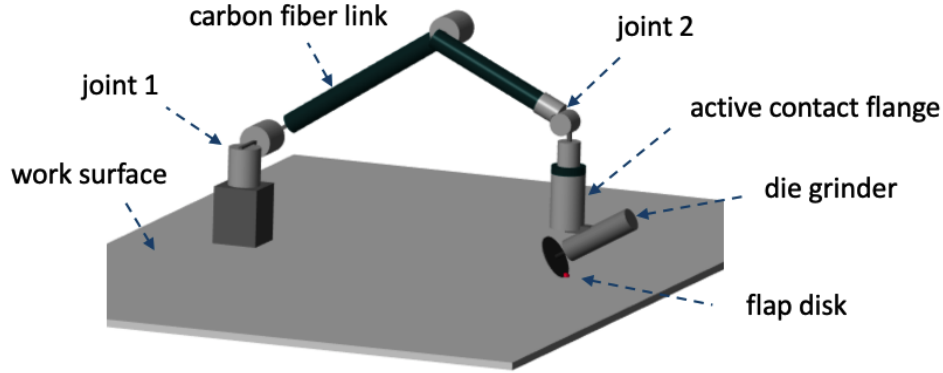


FIGURE 4.4 Mechanical model of the robot arm used for dynamic simulations

4.2.3 Motors and Gearheads

Frameless Brushless DC (BLDC) motors are commonly used to produce very high torque densities in robot joint modules [65, 73]. Additional machined parts are however required for their assembly but typically the weight that is saved using this type of actuators compared to the integration of housed commercial motors remains significant. BLDC outrunners are widely used in UAV rotors or quasi-direct drive actuation and were initially considered here for maximizing nominal torques under low weight [73]. However, the available range of such motors for lightweight robotics is narrow and their assembly with Harmonic Drive gearheads is difficult. Additionally, better performances are available using for example the Robodrive ILM series, initially developed for space robotics [65]. ILM series motors are therefore selected here and used with Harmonic Drive gearheads with a suitable range of rated torques, starting from 0.032 Nm to 2.9 Nm with a weight under 700 g. T-Motor RI series³ also is a valid competitor with high torque densities and cheaper actuators but this series is not considered in this paper because of its nominal speeds that are far below Robodrive's. It should be pointed out that the choice of using high reduction Harmonic Drives is common in many robotics

2. <https://www.ikont.com/rotary-bearings/crossed-roller-bearings/>

3. <https://store.tmotor.com/category.php?id=92>

applications as they offer great torque densities in a compact embodiment as required for the space industry [66]. The Harmonic Drive series considered in this paper are the lightweight HD series CPL⁴ (CSF-LW for North America), and the flat HD series CSD.

A choice of the components of the joints and the links is carried out. The specifications of these are presented in Tab. 4.2 and Tab. 4.3. Joints 1 and 2 are modeled in Fig 4.5 and integrate the Harmonic Drive gearhead, the BLDC motor as well as its controller. Each joint has a link system to the frame and to the output of the Harmonic Drive gearhead to allow connection to another joint or link. A model of the different components of the arm is presented in Fig 4.6. A total mass of about 6.4 kg without EE is obtained reaching 9.1 kg with the tool.

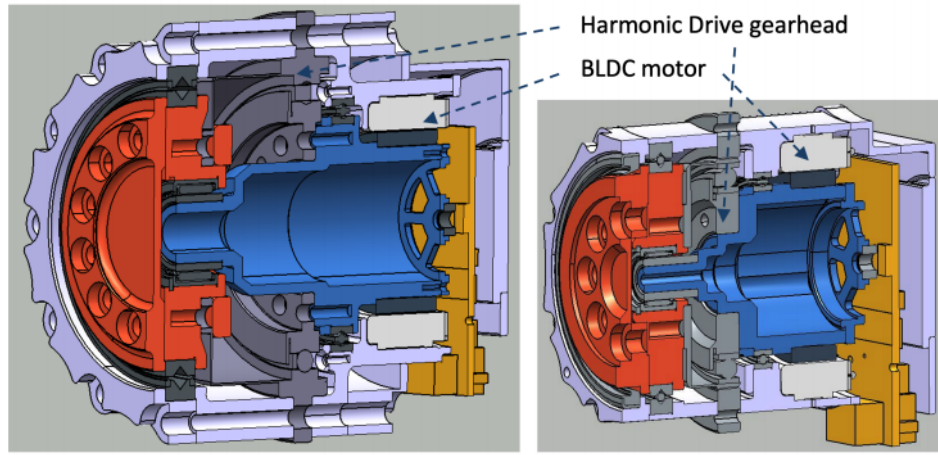


FIGURE 4.5 Section views of the large Joint 1 (left) and the smaller Joint 2 (right)

4.2.4 Modeling

Kinematics

The kinematics of a 6-axis serial robot is usually defined using the Denavit-Hartenberg (DH) convention. Accordingly, reference frames and DH parameters of the robot discussed in this work are presented in Tab. 4.4 and Fig. 4.7. Parameters a_2 and d_4 are mainly the lengths of the carbon fiber tubes (including a small offset for attachment), d_1 is function of the joint 1 design, and d_6 combines the joint 5, joint 6 and tool lengths.

Transformation matrices, e.g. ${}^{i-1}\mathbf{T}_i$, are used to shift from one joint frame to the next using

4. <https://harmonicdrive.de/en/product/gear-component-sets/cpl-2a>

TABLEAU 4.2 Joints specifications

Specification	Joint 1	Joint 2
BLDC motor	Robodrive ILM-50x14	Robodrive ILM-50x18
Harmonic Drive gearhead	Harmonic Drive CPL-25-2A	Harmonic Drive CSD-17-2A
Nominal torque	67 N.m	16 N.m
Nominal speed	3.9 rad/s	11.3 rad/s
Mass	1169 g	499 g
Length	97.4 mm	69 mm
Diameter	85 mm	60 mm
Inertia	0.353 kg.cm ²	0.109 kg.cm ²

TABLEAU 4.3 Links specifications

Specification	Link 1	Link 2
Length	310 mm	465 mm
Outside diameter	52.3 mm	52.3 mm
Inside diameter	47.6 mm	47.6 mm
Mass	182 g	274 g

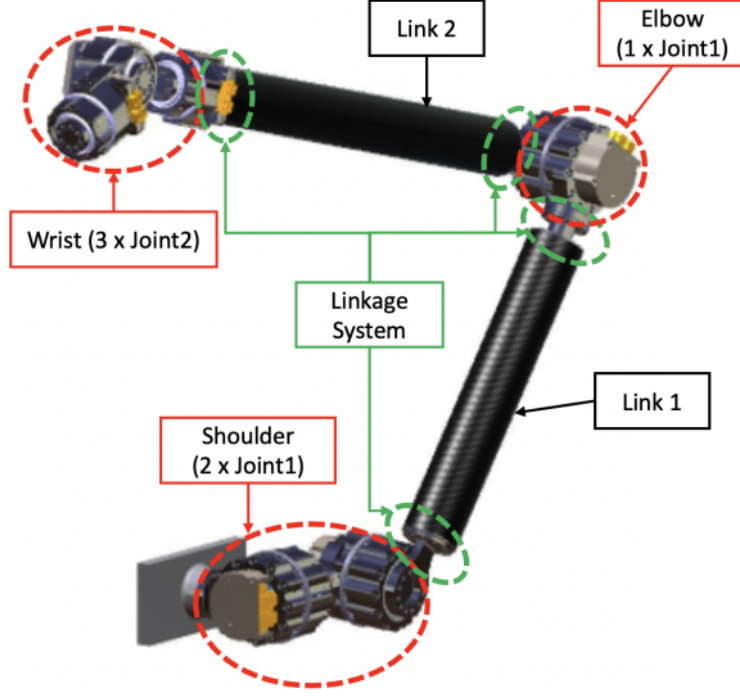


FIGURE 4.6 Robotic arm model with articulation modules, links and a linkage system

DH convention. These matrices can be expressed as :

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & -C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

where $C\theta_i$ is a shorthand notation for $\cos \theta_i$, $S\alpha_i$ a notation for $\sin \alpha_i$ and so on. The forward kinematics matrix $\mathbf{T}$ of the complete robot is obtained as the product of the six transformation matrices defined with Eq. (4.1), namely :

$$\mathbf{T} = \prod_{i=1}^6 {}^{i-1}\mathbf{T}_i = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (4.2)$$

where $\mathbf{R}$ is the 3×3 orientation matrix and $\mathbf{p}$ the 3×1 translation vector of the EE. Establishing $\mathbf{T}$ knowing the joint angles θ_i is trivial and constitutes solving the direct kinematics of the robot. The solution to the inverse kinematics problem is obtained in this paper analytically using decoupled positioning and orientation sub-problems [74]. The Jacobian matrix $\mathbf{J}$ of the arm is obtained from the differentiation of Eq. (4.2) considering small displacements in

TABLEAU 4.4 Denavit-Hartenberg parameters

Frame	θ ($^\circ$)	d (mm)	a (mm)	α ($^\circ$)
1	θ_{1v}	$d_1 = 137$	0	-90
2	θ_{2v}	0	$a_2 = 425$	180
3	$\theta_{3v} - 90$	0	0	-90
4	θ_{4v}	$d_4 = 625$	0	90
5	θ_{5v}	0	0	-90
6	θ_{6v}	$d_6 = 110$	0	-180

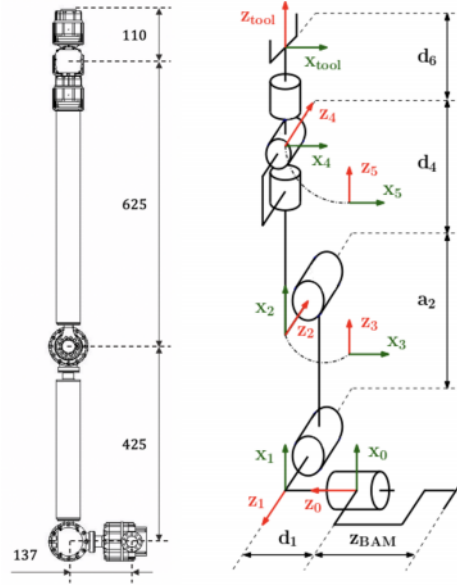


FIGURE 4.7 Robotic arm design with joints modules and links (left) and Denavit-Hartengerg frames of reference and parameters (right).

the operational field and a kinematically achievable path. With $\mathbf{v}$, $\boldsymbol{\omega}$ the velocity and angular velocity vectors, $\mathbf{a}$ and $\boldsymbol{\alpha}$ the associated acceleration and angular acceleration vectors, and $\dot{\mathbf{q}}$ the angular rate vector of the joints, one has :

$$\mathbf{q} = \mathbf{q}_0 + \int_{t_0}^t (J^{-1}(q_\tau) \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}) d\tau \quad (4.3)$$

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J} \dot{\mathbf{q}}, \quad (4.4)$$

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1} \left(\begin{bmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \end{bmatrix} - \dot{\mathbf{J}} \mathbf{J}^{-1} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \right). \quad (4.5)$$

Dynamics

A rigid body dynamic model is used as a first approximation to relate the EE motion defined with Eqs. (4.4) and (4.5) to the joint torques. The vector of joint torques $\boldsymbol{\tau}$ can be obtained using the general dynamic equation of a manipulator, namely :

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\mathbf{f}_{\text{tool}} \quad (4.6)$$

where $\mathbf{M}(\mathbf{q})$ the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ a vector of velocity terms (e.g. Coriolis), $\mathbf{g}(\mathbf{q})$ the gravity vector, $\mathbf{J}^T(\mathbf{q})$ the transposed Jacobian matrix and $\mathbf{f}_{\text{tool}}$ the vector of external forces and torques acting on the EE. Eq. (4.6) is numerically solved in this paper using Matlab Simscape Multibody library. The robot is approximated using simple shapes (as shown in Fig. 4.4) to compute the inertias of the links in parallel with the Transformation matrices (see Fig. 4.8 for an illustration of the method). The values of the torques on the motor shafts

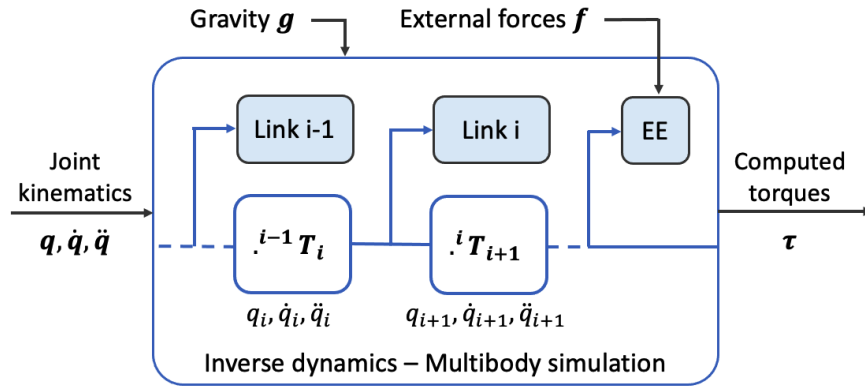


FIGURE 4.8 Inverse dynamics implementation

are approximated using a first order model from the axis angular accelerations established by Eq. (4.5), the axis torques from Eq. (4.6), and the motors and gearheads characteristics, i.e. :

$$\boldsymbol{\tau}_{\text{mot}} = (\mathbf{J}_{\text{m}} + \mathbf{J}_{\text{HD}})\ddot{\mathbf{q}} + (\mathbf{H}_{\text{HD}}\mathbf{N})^{-1}\boldsymbol{\tau} \quad (4.7)$$

where $\mathbf{J}_{\text{m}}$, $\mathbf{J}_{\text{HD}}$, $\mathbf{H}_{\text{HD}}$ and $\mathbf{N}$ are diagonal matrices containing the motors' and gearheads' inertias, gearheads efficiency, and their reduction ratios respectively.

4.3 Electronics Design

4.3.1 General Architecture

The electronic architecture of the embedded robotic arm has been designed with a modular structure around a central control system that communicates synchronously with the different subsystems of the arm. The general layout of the associated electronic architecture is presented in Fig. 4.9 which also shows the communication and power connections. The

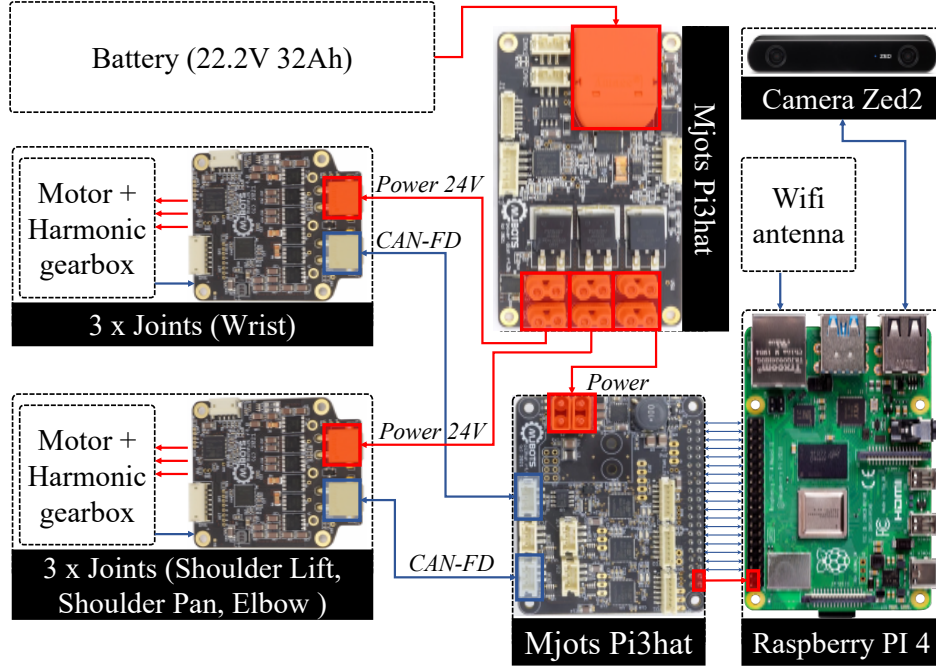


FIGURE 4.9 Schematic of the electronic architecture of the robotic arm

central control system relies on a Raspberry Pi 4 board with 8 Gb allowing easy development and integration between the Raspberry Pi OS and Robot Operating System (ROS). This board is also used during the docking of the drone and communicates with the remote operator by TCP/IP through WiFi. Each joint of the arm embeds a Moteus controller to drive the selected BLDC motors using field-oriented control at a PWM rate of 40kHz. The Moteus controller is designed around a 32 bit microcontroller able to achieve a 1.5kHz sample rate for the PID position control loop⁵. Each controller also includes an absolute magnetic encoder with 14-bit resolution to measure the position at the rotor and communicate with the microcontroller via I²C bus. The communication with the motor controllers and the RaspberryPI relies on a Controller Area Network Flexible Data-Rate (CAN-FD) bus with a 1Mbit bitrate.

5. <https://github.com/mjbots/moteus/blob/9ed36d8d47bc8d0071c5df0253210919b4b12707/docs/reference.md>

A RaspberryPI board extension is used (Mjbots Pi3hat⁶) to power the Raspberry Pi and add communication interfaces for joint control. This extension board provides five independent 5Mbps CAN-FD buses and a 5V supply for the Raspberry Pi.

4.3.2 Power Supply

The power needed to operate the robotic arm, and all associated devices, is provided by the drone's main battery which has a 32Ah capacity and can deliver a nominal 22.2V. This battery is connected to a power distribution board (Mjbots Power Dist⁷) providing energy monitoring, pre-charging, and over-current protection for each joint. This distribution board is also connected to the Raspberry Pi with the CAN-FD bus.

4.3.3 Sensors

A ZED2⁸ camera (3840×1080 resolution at 30 FPS) is mounted at the fifth joint of the arm in order to provide realtime video feedback of the environment. This camera is equipped with an object detection and depth sensing module. Once processed, a list of geometric objects as well as a point cloud of the environment are sent and integrated into the scene plane which will then help to generate optimal trajectories along the work surface.

4.4 Control and Software Design

4.4.1 Control of the embedded robotic arm

The position control of the robot arm is achieved by a Proportional-Derivative (PD) controller with the addition of gravity and friction feedforward compensations and an elastic model as illustrated in Fig. 4.10. since the robot arm is lightweight and is expected to have significant elasticity in its joints, it is necessary to use an elastic model for accurately controlling the arm [75]. Such an elastic model allows to calculate the deformed angular positions of the joints $\Delta \mathbf{q}$ based on the identified robot stiffness, the small displacements (i.e. the Jacobin transpose is calculable) and the filtered and corrected (taking into account gravity at the tool) information from a load sensor, i.e. :

$$\Delta \mathbf{q} = \mathbf{K}_q^{-1} (\mathbf{J}^T \mathbf{f}_{\text{tool}} - \mathbf{g}_{\text{comp}}(\mathbf{q})) \quad (4.8)$$

6. <https://github.com/mjbots/pi3hat/blob/master/docs/reference.md>

7. https://github.com/mjbots/power_dist/blob/3f985a925e2f8bbc1a654fbb4692995da48ee7de/docs/reference.md

8. <https://www.stereolabs.com/zed-2/>

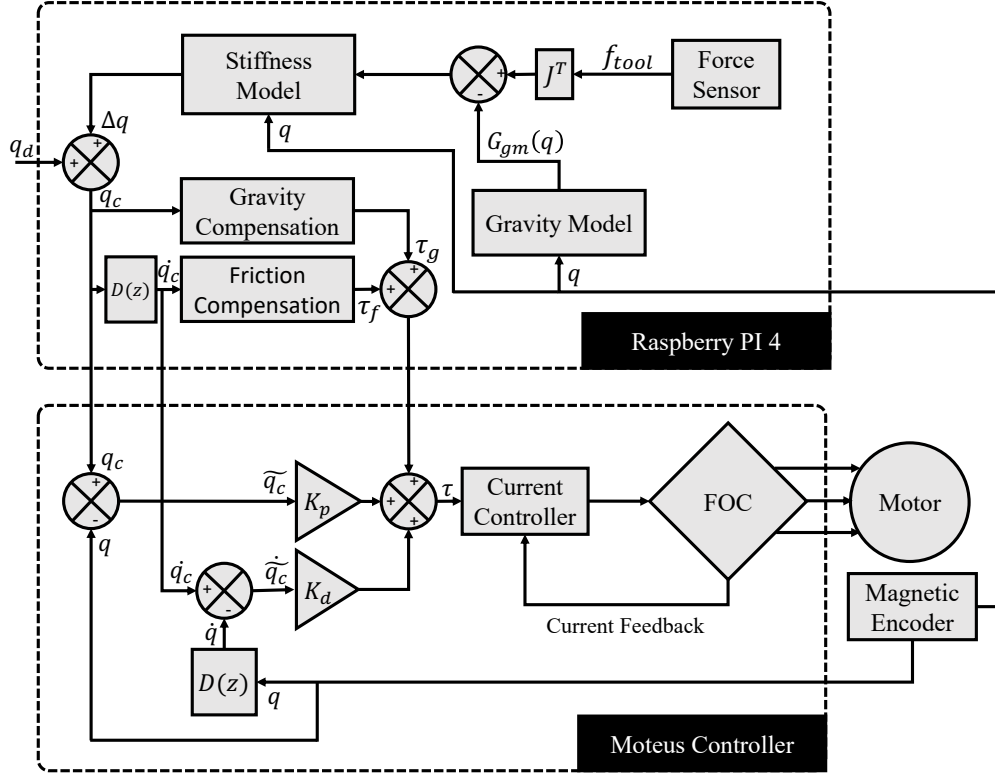


FIGURE 4.10 Structure of the robot arm position controller with PD control and additional compensations

where $\mathbf{K}_q$ is the 6×6 joint stiffness matrix, $\mathbf{q}$ is the joint angular position measured by the magnetic encoder, $\Delta \mathbf{q}$ is the deformed position vector at the joints, and finally, $\mathbf{g}_{\text{comp}}(\mathbf{q})$ is the vector corresponding to the tool gravity model. The actual stiffness matrix of the arm is complex to estimate because it considers both the stiffness of the links and the joints and thus, requires to know precisely the stiffness of the robot $\mathbf{K}_q$. In order to simplify the model, only the elasticity of the joints is considered as a first approximation, which yields :

$$\Delta \mathbf{q} = \mathbf{K}_q^{-1} \boldsymbol{\tau} \quad (4.9)$$

Where $\mathbf{K}_q = \text{diag}([k_1, k_2, k_3, k_4, k_5, k_6])$ is the 6×6 diagonal matrix of the joint stiffnesses and k_i is the stiffness of the i^{th} joint counting from the base. The resulting deformation vector $\Delta \mathbf{q}$ is then used to calculate the corrected angular position to be sent to the PD controller using :

$$\mathbf{q}_c = \mathbf{q}_d + \Delta \mathbf{q} \quad (4.10)$$

where $\mathbf{q}_c$ is the corrected angular position vector and $\mathbf{q}_d$ is the desired angular position vector sent by the path planner.

A friction compensation model is also added taking into account both Coulomb and viscous effects, namely :

$$\boldsymbol{\tau}_f = \mu_v \dot{\mathbf{q}}_c + \mu_s \text{sign}(\dot{\mathbf{q}}_c) \quad (4.11)$$

where μ_v is the kinetic friction coefficient, μ_s is the static friction coefficient, and $\boldsymbol{\tau}_f$ is the resulting joint torque vector due to friction. $\dot{\mathbf{q}}_c$ is the filtered numerical derivative of $\mathbf{q}_c$:

$$\dot{q}(z) = D(z)q(z) = \frac{1 - z^{-1}}{T_e} H(z)q(z) \quad (4.12)$$

where T_e is the sampling period and $H(z)$ is a 2nd order low pass digital filter.

From Eqs. (4.10) and (5.9), the final PD control algorithm can be expressed as :

$$\boldsymbol{\tau} = \mathbf{K}_p \widetilde{\mathbf{q}}_c + \mathbf{K}_d \dot{\widetilde{\mathbf{q}}}_c + \mathbf{G}(\mathbf{q}_c, \mathbf{g}_0) + \boldsymbol{\tau}_f \text{ with } \widetilde{\mathbf{q}}_c = \mathbf{q}_c - \mathbf{q} \quad (4.13)$$

where $\mathbf{K}_p$, $\mathbf{K}_d$ are positive diagonal matrices, corresponding to the proportional and derivative gains for each axis respectively, $\boldsymbol{\tau}$ is the torque vector fed to the motors and regulated with the current controller of the Moteus boards, $\widetilde{\mathbf{q}}_c$ is the error vector of the angular positions, $\mathbf{G}(\mathbf{q}_c, \mathbf{g}_0)$ accounts for the effect of gravity on the mechanical arm and $\mathbf{g}_0$ is the gravity vector at initialization, which depends on the orientation of the BAM. The computed control input is divided into two parts. The first corresponds to the PD part of the control law which is directly processed by the Moteus controller at 1.5kHz. The second part, corresponding to the feedforward compensations, which are more computationally intensive is handled by the Raspberry Pi at 500Hz.

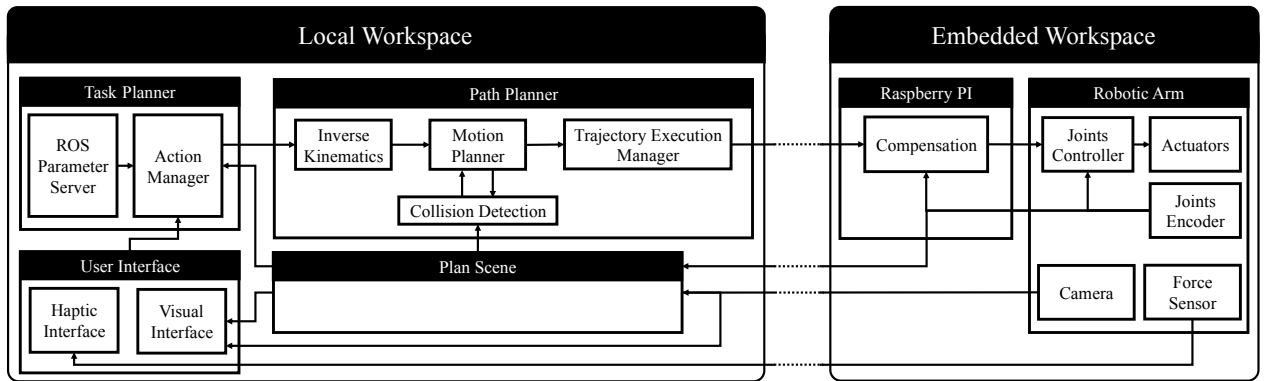


FIGURE 4.11 Control architecture diagram of robotic arm. Local workspace corresponds to elements used by the operator and remote workspace corresponds to elements used onboard the UAV.

4.4.2 Operational Architecture

This section discusses the control architecture of the local (operator) workspace, detailing the four main parts depicted in the left-hand side of Fig. 4.11.

User Interface

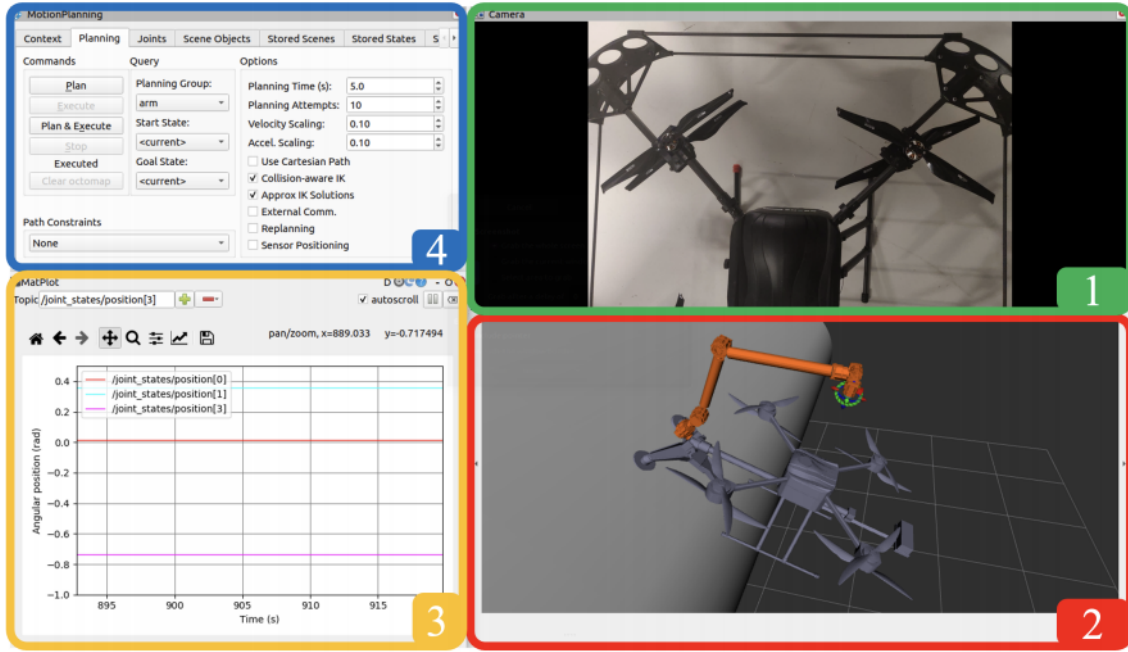


FIGURE 4.12 Graphical user interface built on RViz : 1)Video stream from ZED2 camera 2)Virtual model of the robotic arm and its local environment 3)Display of the ROS computation graph (rqt-graph) 4)Robotic arm parameters (Default motion planner interface in Rviz)

Visual feedback is necessary for the operator to correctly locate the robotic arm in its environment and to be able to perform the operations at the desired location. A graphical user interface (GUI) was developed and based on the ROS visualization package (RViz). This graphical user interface consists of four parts, see Fig. 4.12 :

- The graphical user interface (GUI) from the video feedback of the ZED2 camera placed on the robotic arm at the fifth joint. This camera allows to obtain a visual return in real time of the operations of the robotic arm for the operator, which allows greater immersion for visual validation of the operations' progress of the operations carried out by the arm on the contact surface.
- The second part is a virtual model of the robotic arm and its local environment. This modeling allows the visualization of the actual and planned trajectories, and the

display of the possible collisions of the arm with itself and its surroundings during its movement. This virtual model allows the operator to have a global vision of the current scene (compared to the GUI) by seeing the drone, the robotic arm and the various objects constituting its environment. Finally, this 3D model of the arm provides interaction capabilities to the operator, allowing him to easily position the effector when the haptic interface is not used.

- The third part is a graphical display of the arm's state in real time (applied forces, position, position error, velocity, etc.)
- Finally, the fourth part represents the movement parameters of the arm (path planner, maximum velocity of the joints, trajectory generation time). These parameters can be modified during an operation for example in order to reduce the velocity of the arm or to change the type of trajectory generation desired (linear, circular, etc.).

The Quanser High Definition Haptic Device HD^2 (see Fig.4.13) is used to remotely control the position and orientation of the robotic arm end effector. This haptic interface has 6 DOFs for its movement and can provide force feedback for 5 DOFs. The rotation about the axis of the handle is measured but not actuated.

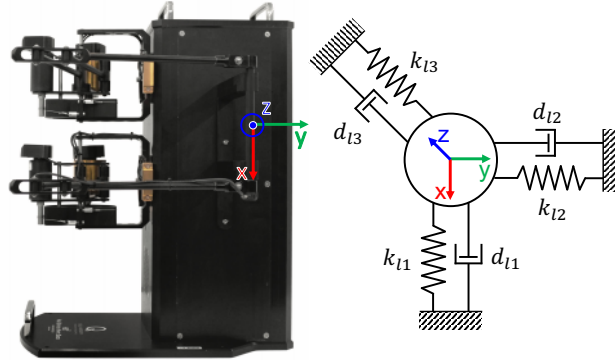


FIGURE 4.13 Quanser High Definition Haptic Device HD^2 on the left and its Cartesian representation with a spring damper model on the right

Two control modes have been implemented for the direct teleoperation of the robot. First, a position/orientation control allows to move from the Cartesian workspace of the haptic interface (W_h) to the Cartesian workspace of the end effector (W_e), via an affine transformation, namely :

$$\mathbf{f} : W_h \rightarrow W_e, \quad \mathbf{p}_e = \mathbf{f}(\mathbf{p}_h) = \mathbf{A}\mathbf{p}_h + \mathbf{b} \quad (4.14)$$

where $\mathbf{A}$ is a linear transformation matrix of W_h (scaling factor), $\mathbf{b}$ is a translation vector in space W_e , $\mathbf{p}_e$ is the end effector position vector in space W_e and $\mathbf{p}_h$ is the haptic interface position vector in space W_h . Then, a velocity controller allows to move from the Cartesian

position workspace of the haptic interface (W_h) to the Cartesian velocity workspace of the EE (W_{e_v}) with a second linear transformation :

$$\mathbf{g} : W_h \rightarrow W_{e_v}, \quad \dot{\mathbf{p}}_e = \mathbf{g}(\mathbf{p}_h) = \mathbf{A}_v \mathbf{p}_h \quad (4.15)$$

where $\mathbf{A}_v$ is a linear transformation matrix of W_h corresponding to a scaling factor. A force feedback control has also been implemented based on the position and velocity error at the arm's EE. These measurements are filtered using a second order low-pass filter to reduce noise while maintaining a low communication delay for teleoperation. The force feedback controller follows a spring-damper model [30] along each Cartesian motion axis, see Fig. 4.13. The resulting force $\mathbf{f}_{fb}$ corresponding to the force vector applied to the haptic interface is then :

$$\mathbf{f}_{fb} = \mathbf{K}_1 \mathbf{e}_p + \mathbf{D}_1 \dot{\mathbf{e}}_p \quad (4.16)$$

where $\mathbf{K}_1 = [k_{l1}, k_{l2}, k_{l3}]$ and $\mathbf{D}_1 = [d_{l1}, d_{l2}, d_{l3}]$ are the diagonal stiffness and damping matrices respectively. Vector $\mathbf{e}_p = [x_d - x, y_d - y, z_d - z]$ is the position error vector and $\dot{\mathbf{e}}_p$ its derivative. This force-feedback controller stabilizes the overall system by closing the loop around the local and remote devices while providing a more immersive control of the arm by the operator.

Task Planner

A task planner is used to manage the sequential execution of the different actions performed by the arm. It continuously receives information from the user interface (mainly the haptic interface) and from the scene (environment, robot state) in order to make decisions on the current and future actions for operation and safety. The state machine diagram of the task planner is presented in Fig. 4.14. An initialization step is first performed in which the task planner is linked to the parameter server in order to obtain configuration parameters of the arm, such as robot architecture, dimensions, initial position, joint limits, collision information between all objects, parameters of the path-planner, and the inverse kinematics algorithm. A scan of the robot's working environment is performed via the camera installed on the arm. The resulting point cloud is sent to the plan scene, which will then allow to locate potential obstacles during motion. Once the environment is scanned, the arm moves to its initial preconfigured working position.

Two modes of operation are available in the task planner, the free motion mode which allows the operator to directly control the movement of the robotic arm via the haptic interface and the supervised motion mode which allows to configure beforehand a path of the arm which will be subsequently performed.

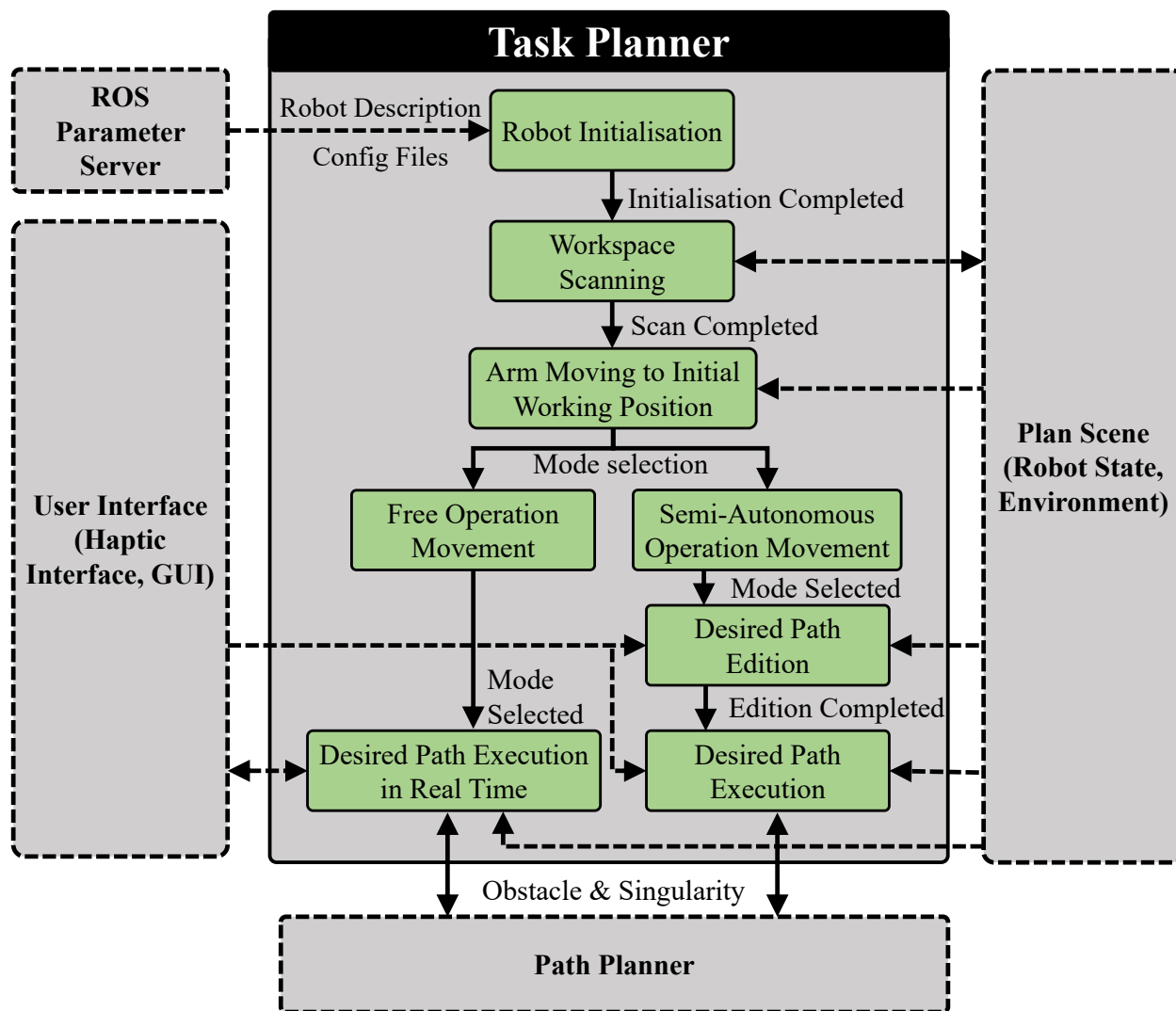


FIGURE 4.14 State machine diagram of the task planner

Free Motion Mode : this mode continuously receives position and orientation information from the haptic interface. Two movement controllers are configured, one in position and the second in velocity. Collisions with the environment are avoided during the movement and force feedback is transmitted to the haptic interface.

Supervised Motion Mode : this mode starts by editing a predefined path pattern on the work surface (circle, ellipse, etc). This pattern is configurable from the haptic interface (starting position, final position, orientation, etc.) and adapts to the work surface. The desired path is continuously displayed on the GUI allowing the operator to visualize the operations that will be performed by the arm. The path is only performed after the operator's approval.

Although it has a collision avoidance feature, the operator can take back control of the current path at any time in case of an unexpected event.

Path Planner

From the information sent by the Task Planner, the trajectory planner transforms the Cartesian pose objective of the arm into an articular path which is then sent to the motor controllers. This transformation must take into consideration obstacle avoidance, potential kinematic singularities, limiting parameters (maximum velocity, maximum acceleration, etc.). This section discusses the design choices of this Path Planner.

The common method to this aim is to use the inverse Jacobian in the Kinematics and Dynamics Library (KDL) with ROS to determine the joint pose of a robot as a function of Cartesian coordinates. However, this method is not optimal because of the potential proximity of joint limits and kinematic singularities [76]. Therefore, the ROS TRACK-IK package is rather used to solve the inverse kinematics of the full path. This method is based on the implementation of two algorithms for solving the inverse kinematics (an improvement of KDL and a nonlinear optimization approach) [77]. TRACK-IK returns the solution of the fastest converging algorithm.

A comparison of the two IK solvers (KDL, TRACK-IK) was performed on both the Universal Robots UR10 and the designed robot arm embedded on the UAV (UAV_ARM), based on their success rate and computation time (see Table.4.5). This comparison follows the same methodology formulated in [76] with 10,000 random Cartesian positions with a position and orientation error of 1^{-5} m. A maximum convergence time was set to 5 ms before the method is assumed to have failed. The average convergence time to find a solution for our 6-DOF robotic arm is 0.54 ms with a success rate of 99,87%.

Once the inverse kinematics solution has been found, the motion planner will aim to find

TABLEAU 4.5 Comparison of two IK solvers (KDL, TRACK-IK) based on computation time and success rate for UR10 robotic manipulator and UAV_ARM

Kinematics Chain		IK Solver			
		KDL		TRAC-IK	
Robot	DOFs	Success Rate (%)	Avg Time (ms)	Success Rate (%)	Avg Time (ms)
UR10	6	86.3	0.26	99.23	0.42
UAV_ARM	6	82.32	0.37	99.87	0.54

the path from the current joint pose to the joint pose found by the inverse kinematics. The

Pilz Industrial motion planner [78] has been implemented for the supervised motion mode. It converts a sequence of joint poses into a continuous trajectory with a trapezoidal velocity profile between each desired point (Fig. 4.15). In addition, constraints of position, speed, acceleration along the movement are considered.

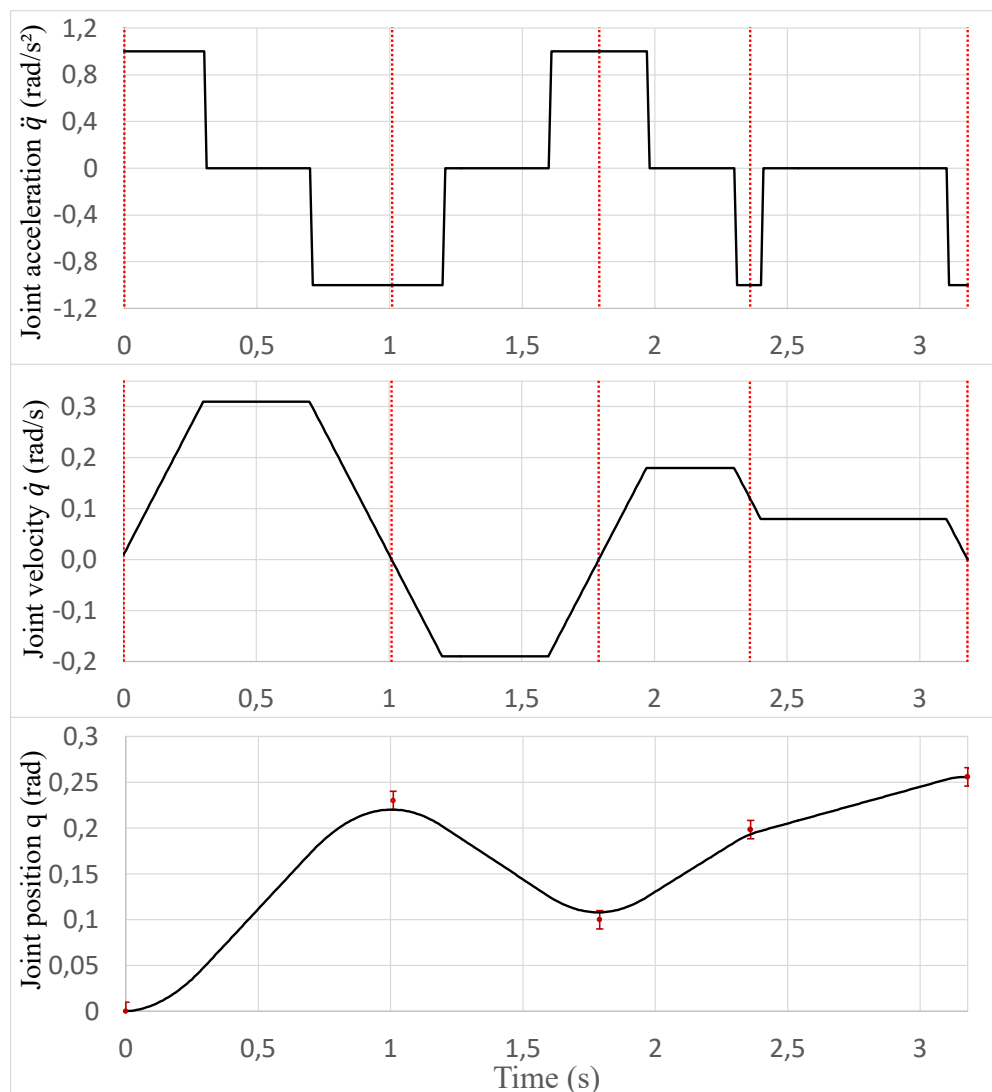


FIGURE 4.15 Pilz Industrial motion planner trajectory passing through five points along the X axis. Red dash-dot vertical lines are the representation of the desired points and the error bars on the joint position represent the tolerance of the end effector to pass close to the desired points

During the generation of the trajectory by the motion planner an algorithm observes the potential collisions that the robot could undergo during the movement. When a collision is detected along a trajectory, the motion planner iterates a new path until it stops if no path is

feasible. When the arm is in motion, the algorithm observes at each update of the scene plan if the robot is close to a collision with its environment to prevent the latter from happening. The trajectory generated by the motion planner is then sent to the motor controllers integrated in the arm through the Trajectory Execution Manager. The Trajectory Execution Manager transmits this planning in desired joint position and velocity via a ROS node that will be read by the onboard motor controllers.

4.5 Experiments and Results

This section presents experiments and their results on the two control modes with a virtual and physical UR10 arm and a virtual model of the proposed novel robotic arm embedded on the UAV. Due to difficulties in sourcing components for the novel robotic arm, it could not be built, so it was considered to use a UR10 arm. The communication interface with the UR10 controller runs at a rate of 125Hz similarly to the virtual model of the new arm.

4.5.1 Free Motion Mode

The latency between the haptic interface and the UR10 robotic manipulator was first measured, see Fig. 4.16. This delay is 15ms on average during standby and rises to 35ms on average when running the free motion mode. However, this delay can be affected by the WiFi remote connection stability but remains low enough to maintain the stability of the system and allows for a responsive user interface during the movement of the arm by the operator.

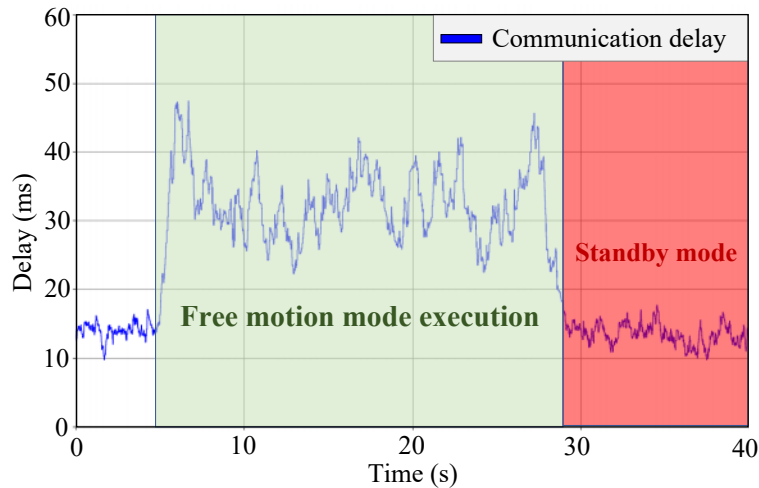


FIGURE 4.16 Round-trip communication delay between haptic interface and UR10 in free motion mode

A first experiment on force feedback with the haptic interface has been conducted, see Fig. 4.17, during which the operator moves the haptic interface about 45mm along the x-axis ($x_d = 45mm$) and then, returns to the original position ($x_d = 0mm$). The error vector feedback in arm position $\mathbf{e}_p$ and velocity $\dot{\mathbf{e}}_p$ are filtered with a 2nd order Butterworth low pass filter with 50 rad/s of cutoff frequency to reduce noise. The stiffness and damping parameters from Eq. (4.16) are set to $\mathbf{K}_1 = [70, 70, 70]$ and $\mathbf{D}_1 = [30, 30, 30]$ and the calculated feedback force is in the range of $[-5, 5]N$ to avoid damaging the haptic interface. The feedback force is applied in opposition to the motion generated by the operator, it is possible to adjust $\mathbf{K}_1$ and $\mathbf{D}_1$ to provide a better experience according to the operator's profile. Moreover, when the haptic interface is configured to control the velocity of the robotic arm, the interaction with the operator becomes more complicated. It has been decided to restrict the control in orientation in order to perform the process more easily.

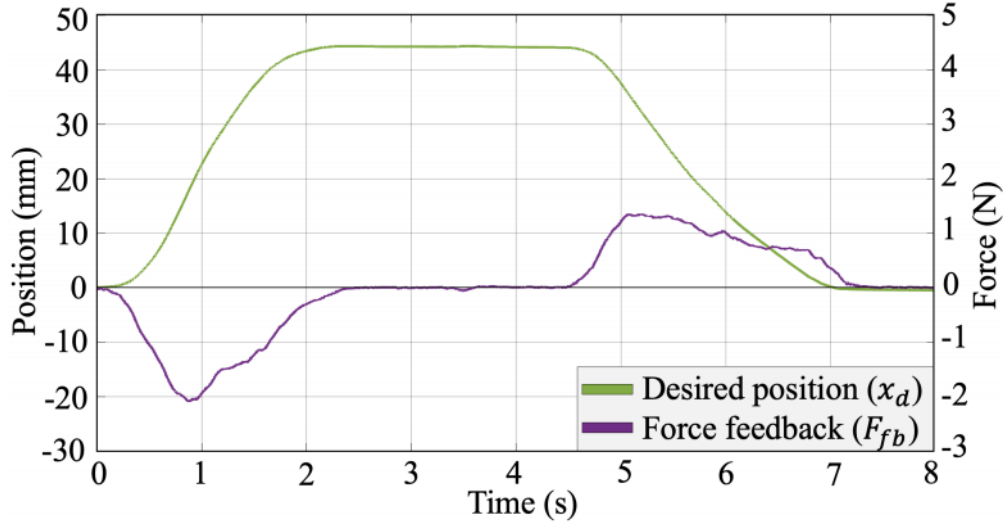


FIGURE 4.17 Feedback force received after a movement x_d of the haptic interface by the operator

A second experiment showed the obstacle avoidance behavior in the free motion mode where the arm slows down when approaching an obstacle at less than 20mm distance and stops when the distance is lower than 10mm. The picture on the top right of the Fig. 4.18 presents the trajectory performed to avoid the obstacle using the haptic interface.

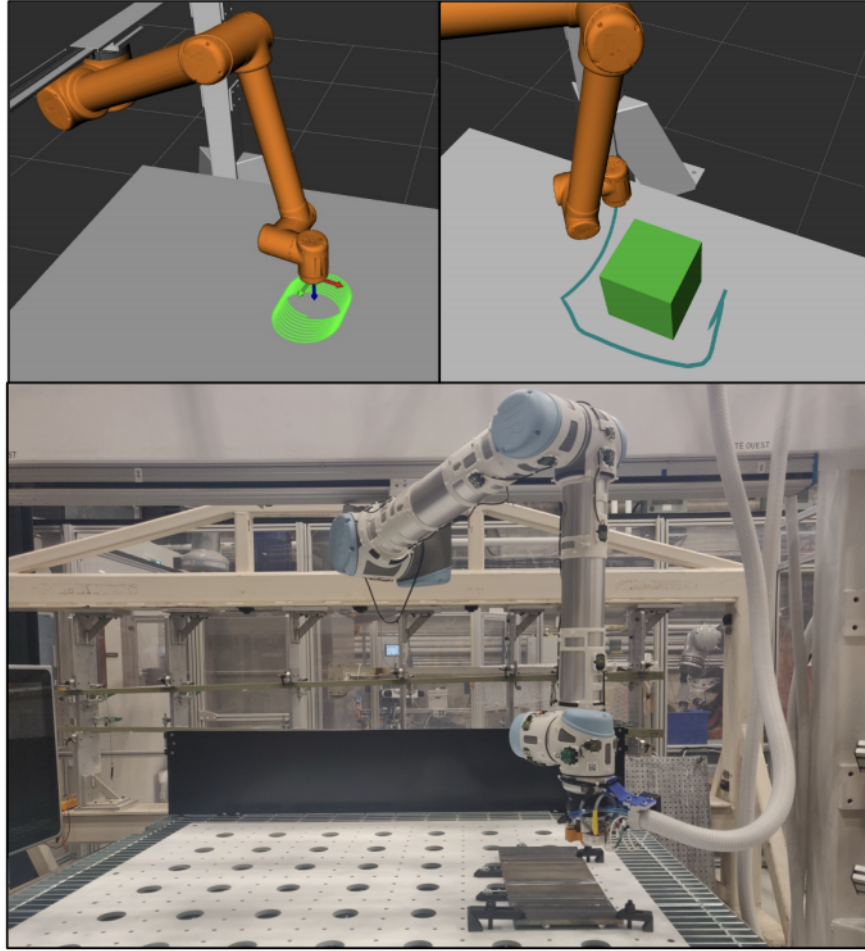


FIGURE 4.18 Top left : virtual model of the UR10 arm during a simulated trajectory on RViz. Top right : virtual model of the UR10 during a free motion mode. Bottom : physical system.

4.5.2 Supervised Motion Mode

The supervised motion mode demonstrates the ease for the operator of achieving a complex trajectory compared to the free motion mode. A teleoperation experiment was conducted on the virtual model of the UR10 arm controlling the physical model, see Fig. 4.18. The ROS graphical interface RViz allows the operator to visualize and move the manipulator. However, there are some limitations to this approach, specifically it is impossible for the operator to modify the trajectory during its execution because the trajectory generation does not support real time updating. The angular position errors of the joints during the motion are less than 1 mrad, see Fig. 4.19, with an EE position error less than 1 mm.

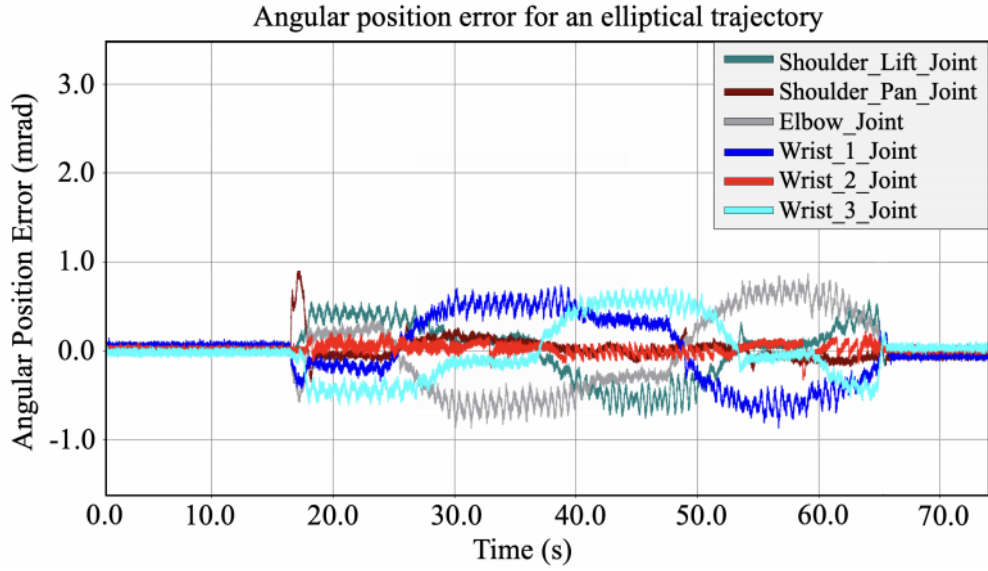


FIGURE 4.19 Results of the angular position error during the execution of a trochoidal trajectory as shown in Fig. 4.18

Another experiment in supervised motion was conducted on the virtual model of the robotic arm embedded on the UAV. The trajectory follows the shape of the wind turbine blade from the point cloud provided by the camera. In the situation shown in Fig. 4.20 the generation time of the trajectory is 300 ms. The absolute position error of the EE during the motion is around 2 mm, see Fig. 4.21. This error is reasonably close to the physical UR10 model. However, this depends on the location and orientation of the arm but remains at about 2mm. The physical model of the arm embedded on the UAV is expected to have a similar behavior to the simulation with the generation of a trajectory on the turbine blade from the point cloud of the camera attached to the arm. However, gravity effects depends on the orientation of the BAM on the workspace and it will be necessary to take into consideration a variable gravity vector in the gravity compensation model.

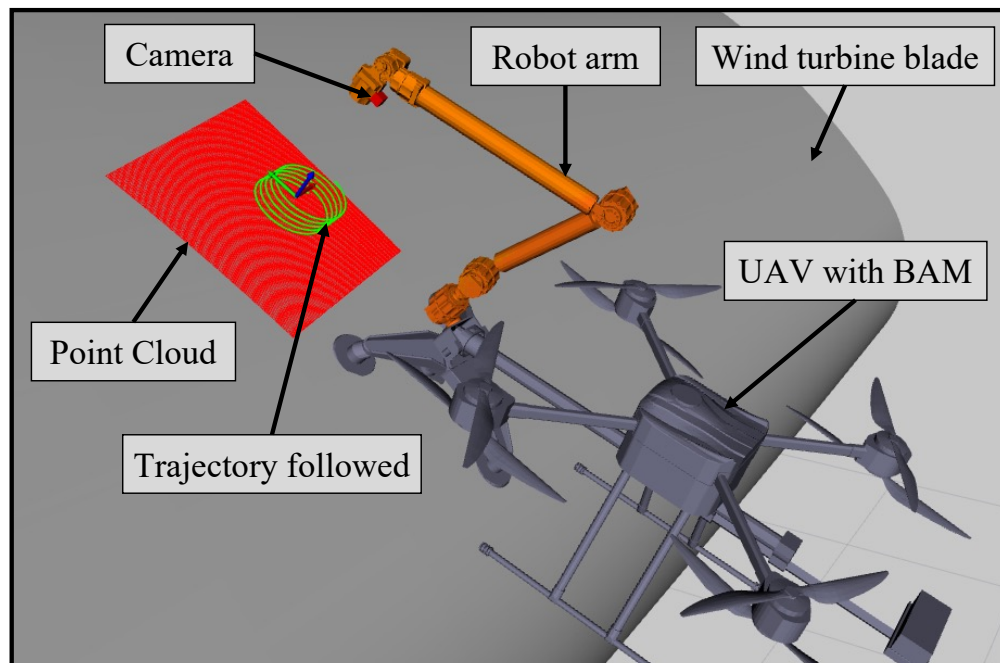


FIGURE 4.20 RViz virtual model of the robotic arm embedded on the drone docked to a wind turbine blade. Description of the scene elements during the trajectory generation in the supervised motion mode.

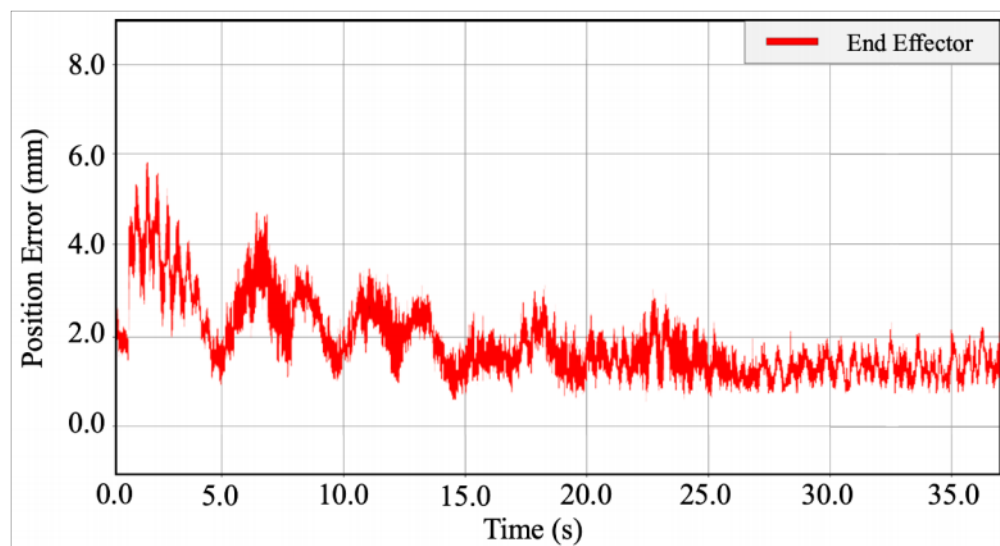


FIGURE 4.21 Results of the absolute position error during the execution of a trochoidal trajectory as shown in Fig. 4.20

4.6 Conclusion

The design of an ultra-lightweight robot arm has been shown in this paper first through the selection of joint module and link components. As a result of this mechanical design, a teleoperated control architecture of the arm has been proposed with several feedforward compensations. Virtual and physical experiments have demonstrated its operation in a 3D virtual environment of the robotic arm and its workspace based on the ROS framework allowing the operator to perform the trajectories in an immersive environment. A haptic interface has been set up to control the robotic arm with two modes of movement and force feedback, experiments have also shown its operation. Finally, a GUI was proposed with a video feedback, a visualization of the robotic arm state variables and various configuration and execution parameters of the arm. Future work will focus on the realization of the prototype robotic arm and its implementation on the UAV. In order to improve the visual feedback and the immersive sensation of the operator, a virtual reality approach could be proposed, or an operator to visualize and situate him/herself more precisely.

Acknowledgments

The authors would like to acknowledge the financial support of the NRC's Integrated Aerial Mobility program⁹.

9. <https://nrc.canada.ca/en/research-development/research-collaboration/programs/integrated-aerial-mobility-program>

CHAPITRE 5 MÉTHODOLOGIE ET RÉSULTATS COMPLÉMENTAIRES

L'article présenté dans le chapitre précédent étant restreint en termes de détails, ce chapitre présente plus précisément les développements effectués sur le bras robotique, ainsi que des résultats complémentaires obtenus.

5.1 Implémentation de l'architecture de contrôle du bras

L'architecture de contrôle du bras a été conçue en deux parties, la figure 5.1 représente l'architecture de téléopération avec le retour haptique et visuel. La première partie de l'architecture du bras est locale par rapport à l'opérateur et la seconde est embarquée sur le drone. Ces deux parties de contrôle doivent être configurées et pouvoir communiquer entre elles afin de permettre le bon fonctionnement.



FIGURE 5.1 Chaîne de téléopération avec retour haptique et visuel

5.1.1 Partie locale (Opérateur)

Configuration du système

Un ordinateur sous Ubuntu 20.04 est utilisé pour l'exécution de l'architecture de contrôle locale. La communication entre la partie locale et la partie embarquée sur le drone est réalisée à partir d'une connexion TCP/IP via WiFi. La plateforme de développement logicielle ROS (Robot Operating System) sous la version Noetic Ninjemys est installée, celle-ci permet une simplification du développement de logiciel dédié à la robotique.

Le principal outil utilisé pour le développement du bras robotique est le logiciel MoveIt! intégré à ROS. Celui-ci permet de facilement ajouter et retirer des packages qui vont s'exécuter et échanger autour d'un nœud principal (*move_group*) généré par MoveIt! . Le schéma simplifié figure 5.2 montre les relations entre les différentes parties de l'architecture locale.

Les éléments inclus dans la zone rouge sont des packages liés à MoveIt! en relation directe avec le nœud *move_group*.

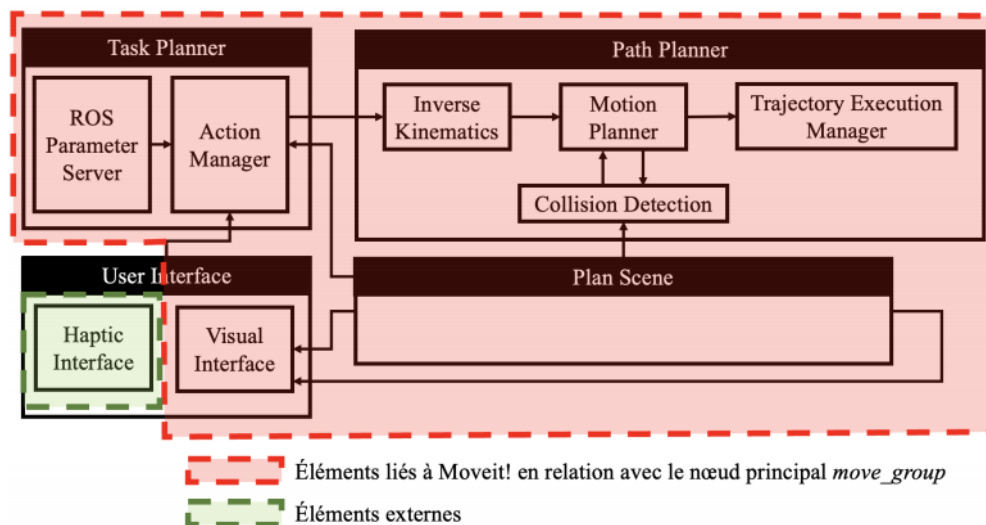


FIGURE 5.2 Schéma simplifié des relations entre les packages ROS

Implémentation du modèle numérique du bras

La description du modèle numérique du bras a été générée sous la forme d'un modèle URDF (Unified Robot Description Format) compatible avec les packages utilisés par MoveIt! Celui-ci utilise deux types de paramètres principalement.

Premièrement les paramètres « Links » qui décrivent physiquement les liens entre les différentes parties du robot, tels que son inertie, ses dimensions, son positionnement et sa forme.

Deuxièmement les paramètres « Joints » décrivant les propriétés de chaque articulation entre elles, leur degré de liberté, leurs limites angulaires, leur orientation. Ce descriptif permet de créer l'arborescence du bras robotique et de toutes les relations qui existent entre les différentes parties de celui-ci. À cette description du modèle numérique s'ajoute un descriptif de la configuration initiale du bras robotique sous la forme d'un fichier SRDF (Semantic Robot Description Format)(Annexe A), celui-ci permet aussi de décrire les relations de collision possible entre chaque « Link » notamment avec son environnement.

Enfin une multitude de fichiers de configuration sont utilisés lors de l'exécution de la simulation du modèle numérique, tels que le planificateur de trajectoire, l'algorithme de cinématique inverse, la génération et la visualisation de l'environnement du bras robotique. Il en est de même pour le modèle numérique du manipulateur robotique UR10. Cependant, les

contrôleurs des articulations utilisés pour la simulation viennent d'un package¹ fourni par le constructeur.

Implémentation de l'interface haptique

L'interface haptique utilisée pour le projet est le Quanser High Definition Haptic Device (HD_2), celle-ci fonctionne sur un ordinateur dédié exécutant un processus Simulink et connecté en réseau local à l'ordinateur principal. Le schéma figure 5.3 décrit les connexions entre les différents éléments constituant la partie locale.

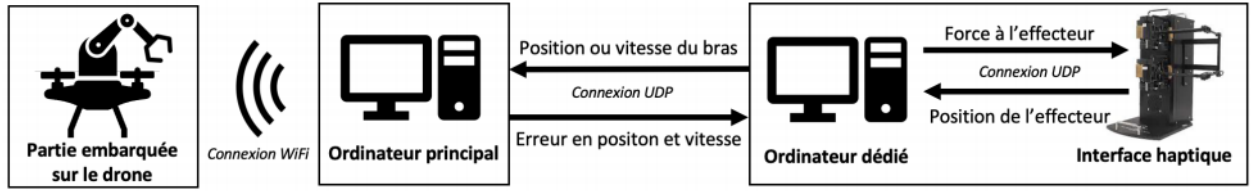


FIGURE 5.3 Schéma des connexions entre les parties constituant le système

Le processus Simulink s'exécute à une fréquence de 1 kHz. L'ordinateur dédié, par l'intermédiaire du processus Simulink, reçoit et filtre le vecteur position de l'effecteur de l'interface haptique $\mathbf{P}_h = [x_h, y_h, z_h, u_h, v_h, w_h]^t$. Comme présenté dans le Chapitre 4, un mode de contrôle en vitesse et en position est implémenté, celui-ci permet de transformer le vecteur position de l'effecteur de l'interface haptique $\mathbf{P}_h$ en vecteur position ou vitesse désirée de l'effecteur du bras $\mathbf{P}_e$ par la relation :

$$\begin{cases} \text{Mode Position} & \mathbf{P}_e = \mathbf{A}\mathbf{P}_h + \mathbf{B} \\ \text{Mode Vitesse} & : \dot{\mathbf{P}}_e = \mathbf{A}_v\mathbf{P}_h \end{cases} \quad (5.1)$$

Le vecteur $\mathbf{A}$ et $\mathbf{A}_v$ permettent d'appliquer un facteur d'échelle à la commande. Elles ont été définies expérimentalement à la suite de plusieurs essais :

$$\mathbf{A} = \text{diag}([20, 100, 100, 100, 100, 100]) \quad \text{et} \quad \mathbf{A}_v = \text{diag}([60, 100, 100, 0, 0, 0]) \quad (5.2)$$

La matrice $\mathbf{B}$ correspond à la position de l'effecteur du bras $\mathbf{P}_{e,t=0}$ lors de l'initialisation de la commande. La partie orientation de la matrice $\mathbf{A}_v$ a été mis à 0 car il a été observé qu'il était très difficile de contrôler en vitesse l'orientation de l'effecteur du bras. De plus une fonction continue f_{near_0} a été ajoutée pour imposer une valeur nulle pour les coordonnées proche de 0.

1. https://github.com/UniversalRobots/Universal_Robots_ROS_Driver

Cela permet d'enlever la légère dérive occasionnée lorsque l'effecteur de l'interface haptique est remis très proche de sa position initiale.

$$f_{near_0}(x) = \begin{cases} x - 10 & \text{si } x > 10 \\ 0 & \text{si } x \in [-10, 10] \\ x + 10 & \text{si } x < -10 \end{cases} \quad (5.3)$$

Le mode de contrôle en position semble le plus performant et le plus intuitif pour l'opérateur. Cependant celui-ci reste borné à la plage de déplacement de l'interface haptique, ce qui n'est pas le cas pour le mode de contrôle en vitesse où l'opérateur peut potentiellement se déplacer dans tout l'environnement de travail du bras.

La variable $\mathbf{P}_e$ ou $\dot{\mathbf{P}}_e$, selon le mode, est envoyé à l'ordinateur principal à travers une connexion UDP pour être réceptionnée par le package Spacenv². Celui-ci permet de convertir $\mathbf{P}_e$ ou $\dot{\mathbf{P}}_e$ en position angulaire des articulations puis de l'envoyer au planificateur de mouvement.

Le retour d'effort se calcule à partir des erreurs en position et en vitesse de l'effecteur du bras, qui est ensuite envoyé à l'ordinateur dédié à travers la connexion UDP. Une fois calculé, le retour d'effort est filtré par un filtre de Butterworth 5.4 passe bas d'ordre $n = 6$ avec une fréquence de coupure de $\omega_c = 50 \text{ rad/s}$:

$$G_n(\omega) = \frac{1}{\sqrt{1 + (\frac{\omega}{\omega_c})^{2n}}} \quad (5.4)$$

Le signal filtré de retour d'effort est envoyé à l'interface haptique à travers un module compatible Simulink, convertissant l'effort à l'effecteur en couple moteur.

Dans le mode de contrôle en vitesse, un retour élastique est implémenté, ce qui permet de repositionner automatiquement l'effecteur en position initiale.

$$F_{fb_{init}} = -0.3\mathbf{P}_h + [0, 0, 0.54] \quad (5.5)$$

Le vecteur $[0, 0, 0.54]$ N correspond à la compensation de la gravité sur l'effecteur de l'interface haptique suivant l'axe Z.

2. <http://wiki.ros.org/spacenv>

Génération de trajectoire en mode supervisé

Le mode supervisé, tel que décrit dans le chapitre 4, permet d'assister l'opérateur dans le déplacement du bras en générant une trajectoire le long de la surface de travail. L'effecteur fait un suivi de trajectoire avec une orientation orthogonale à la surface de contact. Pour cela, la caméra ZED2, montée sur la cinquième articulation, génère un nuage de points de la surface de contact. À l'initialisation du mode supervisé, l'opérateur choisit l'emplacement du début de la trajectoire $\mathbf{P}_{e0} = [x_0, y_0, z_0]$. Le point $\mathbf{P}_{cloud0} = [x_{cloud0}, y_{cloud0}, z_{cloud0}]$ appartenant au nuage de points et le plus proche de $\mathbf{P}_{e0}$. Ainsi $\mathbf{P}_{e0}$ le point donné et $\mathbf{P}_{cloud}$ le nuage de points, on peut trouver le point $\mathbf{P}_{cloud0} \in \mathbf{P}_{cloud}$ le plus proche de $\mathbf{P}_{e0}$ en minimisant la fonction de distance $D(\mathbf{P}_{e0}, \mathbf{P}_{cloud0})$:

$$\begin{cases} \min_{(\mathbf{P}_{cloud0} \in \mathbf{P}_{cloud})} (D(\mathbf{P}_{e0}, \mathbf{P}_{cloud0})) \\ D(\mathbf{P}_{e0}, \mathbf{P}_{cloud0}) = \|\mathbf{P}_{e0} - \mathbf{P}_{cloud0}\| \\ = \sqrt{(x_0 - x_{cloud0})^2 + (y_0 - y_{cloud0})^2 + (z_0 - z_{cloud0})^2} \end{cases} \quad (5.6)$$

Le vecteur $\vec{w}$, orthonormé au plan est défini comme $\vec{w} = \frac{\mathbf{P}_{e0} - \mathbf{P}_{cloud0}}{\|\mathbf{P}_{e0} - \mathbf{P}_{cloud0}\|}$, ce qui nous permet de créer une base orthonormée $(\mathbf{P}_{e0}, \vec{u}, \vec{v}, \vec{w})$ avec :

$$\begin{cases} \vec{u} = \frac{\vec{w} \wedge [-\vec{w}_z, \vec{w}_x, \vec{w}_y]}{\|\vec{w} \wedge [-\vec{w}_z, \vec{w}_x, \vec{w}_y]\|} \\ \vec{v} = \vec{w} \wedge \vec{u} \end{cases} \quad (5.7)$$

Trois types de trajectoires désirées sont présentés à la figure 5.4

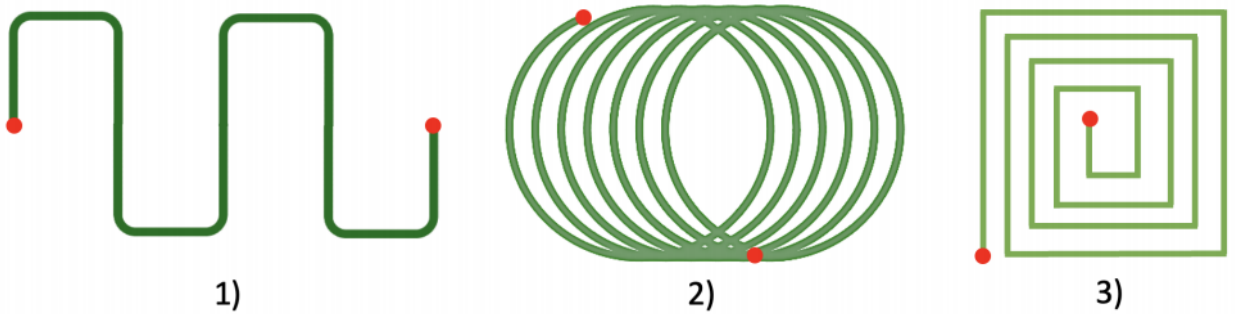


FIGURE 5.4 1) Trajectoire zig-zag 2) Trajectoire trochoïdale 3) Trajectoire spirale

Description mathématique du modèle de trajectoire trochoïdale :

$$P_{et} = P_{e0} + R\sin(t)\vec{u} + \frac{at}{2\pi}\vec{u} + R(\cos(t) - 1)\vec{v} \quad (5.8)$$

Avec a l'avance de l'outil ; t l'angle décrivant la rotation du cercle et R le rayon du cercle. À partir de P_{et} il est possible de trouver le point P_{cloudt} appartenant au nuage de points et le plus proche de P_{e0} en restant orthogonal à la surface de contact.

La figure 5.5 montre les différentes trajectoires désirées et effectuées par une simulation du bras robotique sous Rviz.

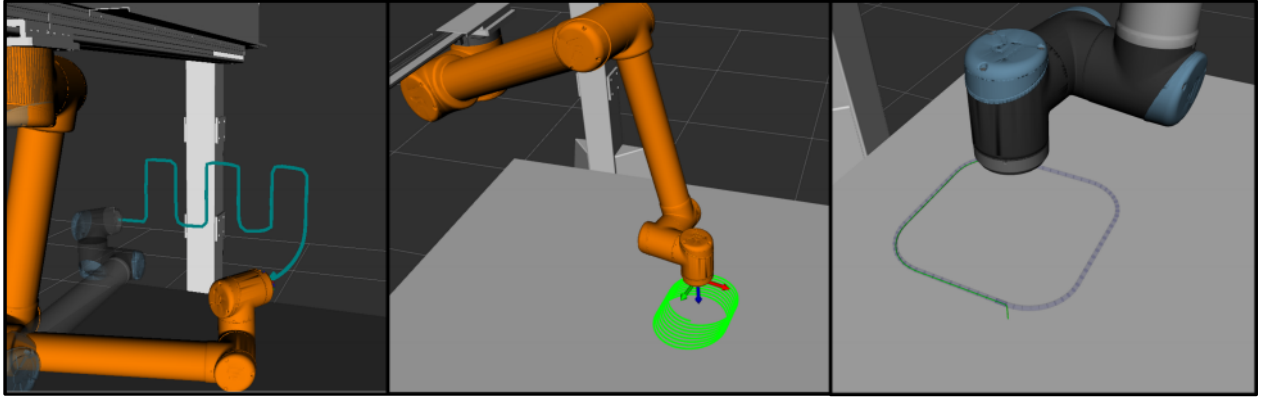


FIGURE 5.5 Trajectoires zig-zag, trochoïdale et carrée simulées sur Rviz

5.1.2 Partie embarquée (Drone)

L'ordinateur de bord embarqué sur le drone, tel que décrit dans la figure 4.9, est un Raspberry PI 4 (RPI) avec 8Go de RAM et le système d'exploitation Raspberry Pi OS ainsi que les outils de développement ROS. La figure 5.6 montre le RPI et les différents composants constitutifs de la partie embarquée.

Contrôleurs Moteurs

Les contrôleurs moteurs brushless (Moteurs) sont des produits open source provenant du constructeur Mjbots³. Ils ont été choisis précédemment pour leur bonne performance en termes de fréquence de contrôle et de leur encodeur. De plus, ils sont facilement paramétrables et leur dimension convient à une installation articulaire. Cependant, ces contrôleurs sont encore récents et ils ne sont pas directement compatibles avec ROS.

3. <https://mjbots.com/>

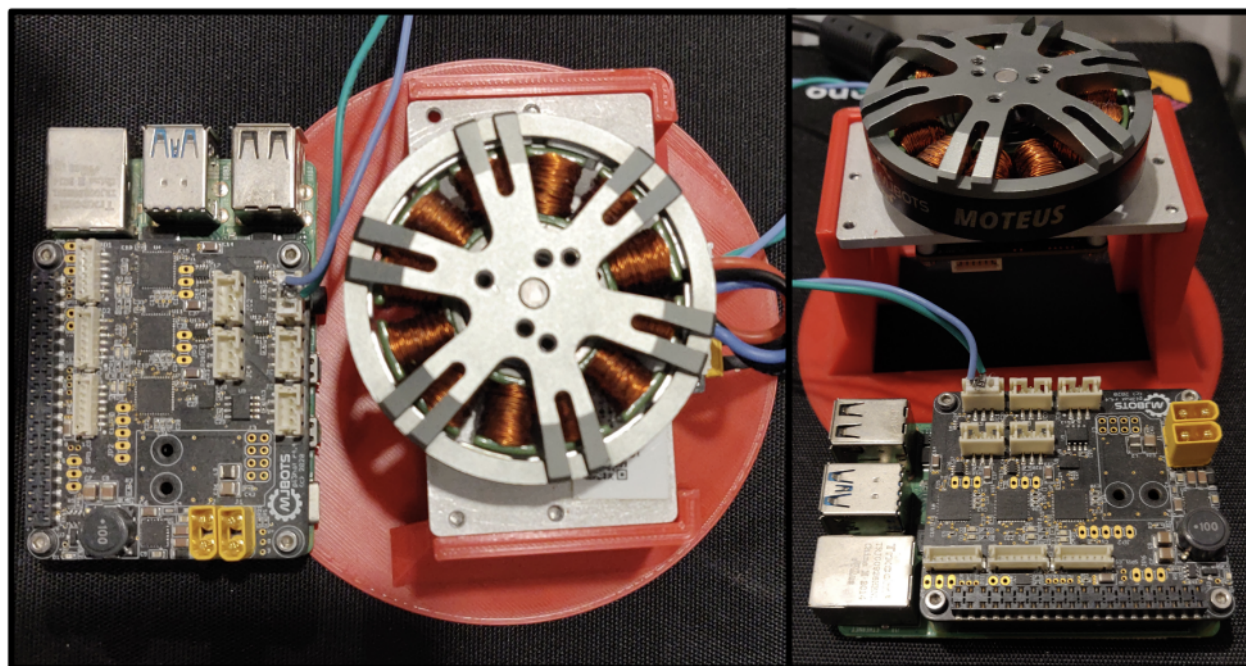


FIGURE 5.6 Photo du système embarqué (Raspberry Pi, contrôleur Moteus, moteur, alimentation électrique)

De ce fait, il a été développé une API (Interface de Programmation d'Application) pour les rendre compatible à ROS figure 5.7, avec pour principales requêtes le contrôle de la position, de la vitesse, du couple moteur, ainsi que le retour d'information de l'encodeur magnétique. Des éléments de sécurité ont été placés pour restreindre certaines requêtes qui pourraient endommager le moteur (Vitesse et accélération trop importantes, position invalide).

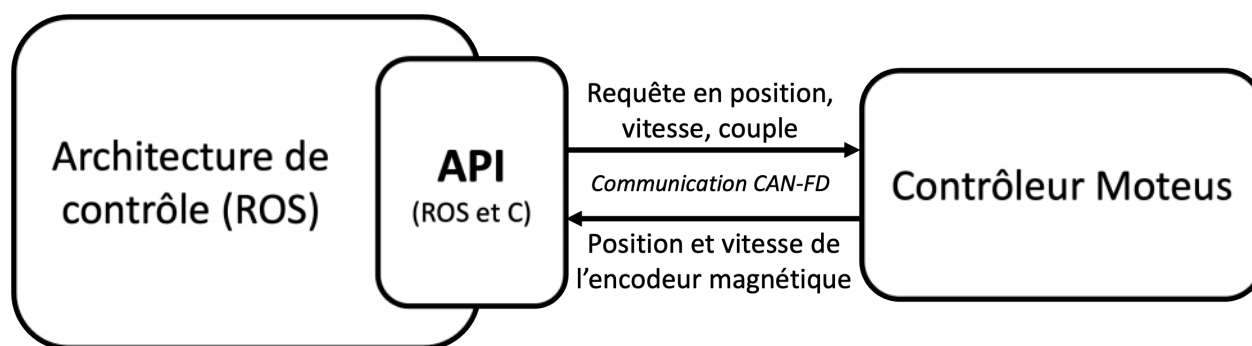


FIGURE 5.7 Schéma simplifié de l'API développée pour interagir avec le contrôleur Moteus

Le bruit du signal en position et en vitesse de l'encodeur magnétique a été caractérisé (position et vitesse non contrôlées) figure 5.8, puis filtré à l'aide d'un filtre passe bas de Butterworth

5.4 d'ordre $n = 4$ et de fréquence de coupure $w_c = 50\text{rad/s}$.

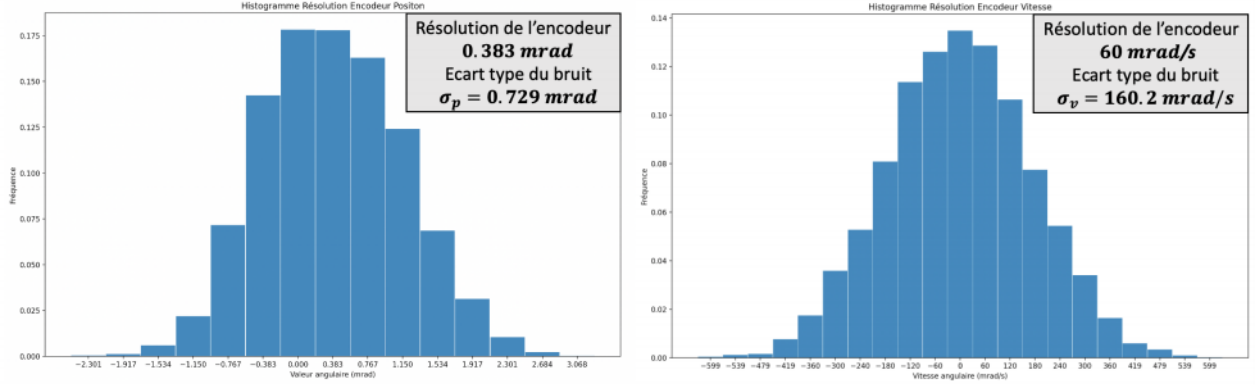


FIGURE 5.8 Histogramme de résolution de l'encodeur en position et en vitesse

Implémentation des compensations

Les compensations détaillées dans la figure 4.10 sont exécutées dans un programme Simulink, ce choix a été fait pour la simplicité de configuration de filtre passe bas et du module de compensation de gravité ainsi que sa compatibilité avec ROS.

La compensation de friction se calcule à partir du modèle de Coulomb + Visqueux 5.9 :

$$\tau_f = \mu_v \dot{\mathbf{q}} + \mu_s \text{sign}(\dot{\mathbf{q}}) \quad (5.9)$$

Avec $\dot{q}$ la vitesse articulaire, μ_s et μ_v les coefficients de frottement statique et dynamique respectivement. Une mesure de ces coefficients a été effectuée sur les contrôleurs Moteus avec les moteurs installés par défaut⁴, dans un second temps les mesures devront être faites sur le système complet (contrôleur + moteur + réducteur harmonique).

La mesure de couple a été menée, en régime établi, sur une plage de vitesse angulaire entre $[-10, 10]$ rad/s avec un pas de 0,1 rad/s entre chaque mesure. À l'approche de la vitesse nulle un plus grand nombre de mesures a été effectué. Nous obtenons la figure 5.9 pour le premier système (contrôleur + moteur) :

Nous pouvons en déduire la valeur des coefficients de frottement statique et dynamique pour le système (contrôleur + moteur), ces résultats sont résumés dans le tableau 5.1 :

4. https://mjbots.com/products/mj5208?pr_prod_strat=copurchase&pr_rec_id=861e09d8a&pr_rec_pid=5945253626017&pr_ref_pid=7358414749857&pr_seq=uniform

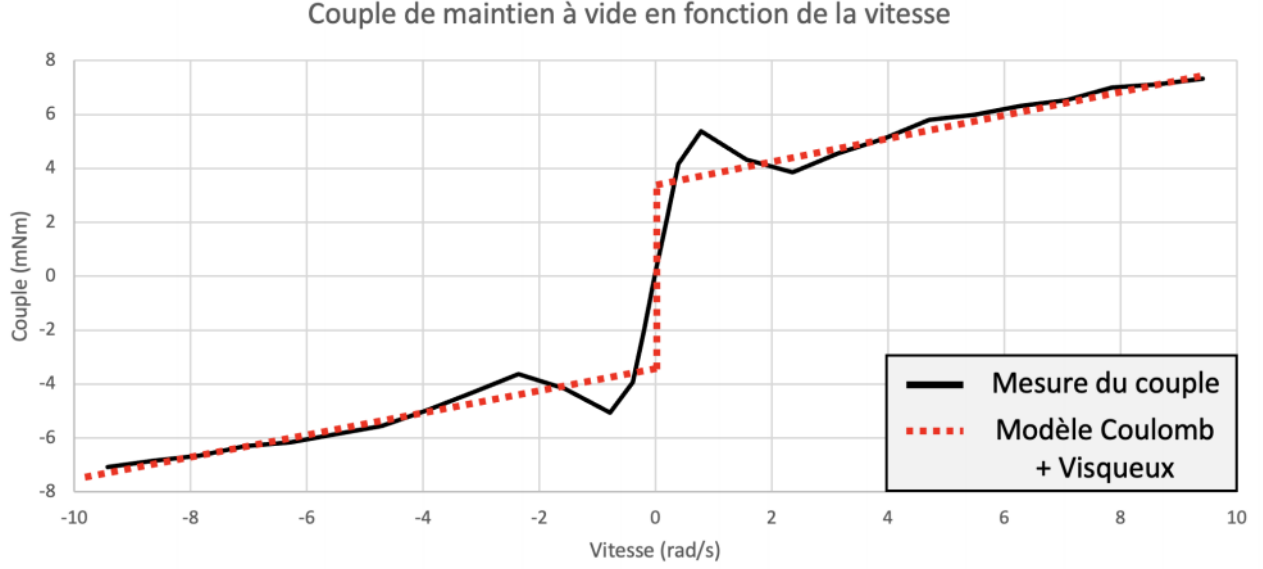


FIGURE 5.9 Graphique représentant le couple de maintien à vide en fonction de la vitesse

TABLEAU 5.1 Coefficient de frottement statique et dynamique

	Moteur 1	Moteur 2	Moteur 3
$\mu_s(mNm)$	3.25	3.55	4.22
$\mu_v(mNm.s/rad)$	0.45	0.50	0.54

La compensation de gravité est réalisée à partir d'un module directement intégré à Simulink (figure 5.10.2), celui-ci permet de calculer le couple de compensation aux articulations à partir de la description du robot et du vecteur gravité appliqué. Ce vecteur gravité est initialisé au début de l'exécution du programme en fonction de l'orientation du BAM.

Dans un premier temps, un prototype de bras planaire à 3 degrés de liberté a été conçu (figure 5.10.3) afin de valider le fonctionnement de l'architecture de contrôle, tel que les délais de communications, les compensations (gravité, friction, élasticité articulaire), les limites angulaires (position, vitesse, accélération), etc. Après validation du fonctionnement de l'architecture de contrôle sur le prototype, le modèle d'articulation développé précédemment [10] devait être fabriqué. À cause de problème de disponibilité et de délais de livraison des réducteurs harmoniques, il n'a pas été possible de fabriquer ces articulations. Nous avons donc continué nos tests sur le prototype et le manipulateur robotique UR10.

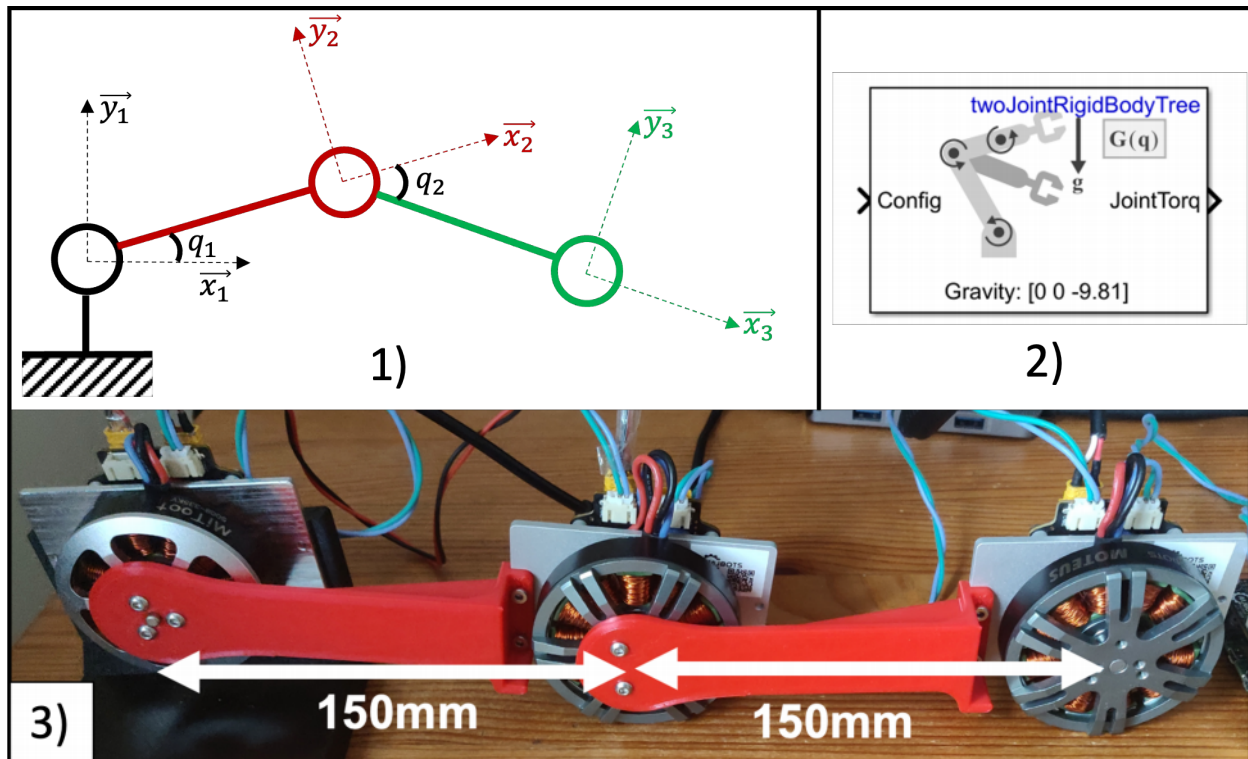


FIGURE 5.10 1)Schéma cinématique du mécanisme 3 axes planaire 2)Module de compensation de gravité sous Simulink 3)Prototype de bras 3 axes planaire

5.2 Améliorations envisagées

Cette partie présente les améliorations qui ont été envisagées sur l'architecture du contrôle du bras robotique mais qui, après différents tests de validation, ne se sont pas révélées intéressantes.

5.2.1 Implémentation d'un contrôle en admittance (capteur de force)

Un contrôle en admittance a d'abord été envisagé pour le contrôle du bras robotique [79]. Il nécessite de mesurer la force à l'effecteur du bras. Cependant le fonctionnement de celui-ci n'a pas pu être réalisé sur le manipulateur robotique UR10. En effet, le capteur de force en bout d'effecteur du UR10 est le FT 300-S⁵, celui-ci a une fréquence de fonctionnement de 100Hz ce qui limite son utilisation dans ce contexte de contrôle en admittance.

5. https://robotiq.com/products/ft-300-force-torque-sensor?ref=nav_product_new_button

5.2.2 Implémentation d'un planificateur de trajectoire

Le planificateur de trajectoire utilisé par défaut pour ce projet est le planificateur pilz-industrial-motion (PILZ) [78]. Afin d'optimiser au mieux le déplacement du robot, il a été envisagé d'utiliser le planificateur STOMP [80]. Ce planificateur produit des trajectoires fluides avec un évitement d'obstacle. Il utilise un processus itératif avec une minimisation d'une fonction coût. Dans notre cas, la fonction coût utilisée est la minimisation de l'énergie utilisée durant le déplacement avec un évitement d'obstacle (distance supérieure à 10mm entre le bras et l'obstacle).

Cependant, des limitations ont été observées lors de son utilisation sur le modèle virtuel du bras robotique. En effet, une expérience a été menée sur le bras robotique avec 200 trajectoires prisent aléatoirement comprenant un obstacle. Le tableau 5.2 résume le taux de réussite des deux planificateur de trajectoire. On obtient un taux de réussite du planificateur STOMP plus faible que le planificateur par défaut PILZ. Cette différence peut s'expliquer par le modèle du robot utilisé, car le manipulateur UR10 obtient des taux de réussite comparable entre le planificateur STOMP et PILZ. Ainsi le planificateur STOMP n'a pas été retenu.

TABLEAU 5.2 Taux de réussite des planificateurs de trajectoire

	STOMP	PILZ
Bras embarqué sur le drone	83%	95%
Manipulateur UR10	96%	97%

5.2.3 Implémentation d'un modèle de rigidité du bras

Un modèle de rigidité du bras a été implémenté dans le programme des compensations sur Simulink. Celui-ci utilise le module Simscape Multibody⁶ qui permet de modéliser de manière simplifiée le caractère élastique des membrures et des articulations. Dans un premier temps, un modèle élastique a été construit pour le prototype de bras 3 axes planaire figure 5.11.

Les paramètres liés à l'élasticité devront être déterminés par des méthodes expérimentales [81]. Cependant, lors de l'utilisation de ce module, certaines parties nécessitent une compilation sous un environnement Windows et non Linux, ce qui rend incompatible son utilisation dans le cas de l'utilisation d'une Raspberry PI. De plus, les calculs de la déflexion des membrures impliquent une utilisation importante du CPU de la Raspberry PI. Après expérience sur le prototype de bras 3 axes planaire, celui-ci indique une fréquence de contrôle d'environ

6. <https://fr.mathworks.com/products/simscape-multibody.html>

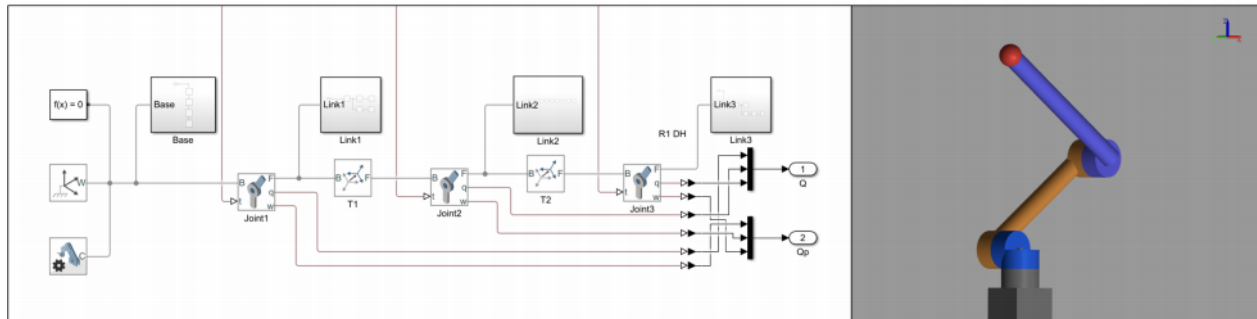


FIGURE 5.11 Schéma modulaire du bras conçu avec le module Simscape Multibody à gauche. Visuel simplifié du modèle du bras 3 axes

10Hz pour les performances , ce qui n'est pas acceptable pour le contrôle du bras qui se veut proche du temps réel.

CHAPITRE 6 DISCUSSION GENERALE

Le projet de maîtrise a eu pour objectifs principaux la conception et l'intégration d'une téléopération d'un bras robotique. Ces objectifs ont pu en partie être réalisés et des avancées techniques ont pu être proposées. Ainsi, la section présentée résume les différents travaux effectués tout au long de ce projet au regard de la revue de la littérature et aux contraintes imposées.

La revue de la littérature nous a permis de caractériser les différentes technologies existantes dans le domaine de la robotique aérienne pour des opérations de maintenance afin de s'en inspirer. De nombreuses contraintes ont été soulevées, notamment au niveau de la masse d'emport, de l'architecture mécanique du bras et des procédés de maintenance réalisables. Par la suite, différentes architectures de contrôle ont pu être détaillées avec leur application robotique actuelle. Ces architectures de contrôle ont pour la grande partie intégrée une fonctionnalité de téléopération. Cette téléopération se caractérise suivant différents modes de contrôle qui ont pu, par la suite, être abordés.

Une architecture de contrôle pour le déplacement du bras a été proposée et expérimentée sur un modèle numérique six axes puis un modèle physique trois axes. Cette architecture de contrôle permet l'ajout de compensation à différents niveaux. Ainsi des compensations ont été ajoutées et paramétrées pour corriger les erreurs de position dû à la gravité, la friction et à la rigidité des articulations. Des expérimentations ont pu être menées sur le bras physique trois axes pour déterminer les paramètres du modèle des compensations. La téléopération a pu être réalisée à partir d'une interface haptique dont les contraintes étaient de plusieurs natures, ergonomique, réactivité et immersion. Le retour d'effort est, quant à lui, paramétrable selon le profil de l'utilisateur et des opérations à effectuer. La partie vision, représentée par l'intégration d'une caméra stéréoscopique sur une articulation du bras, permet à l'utilisateur de repérer le bras dans son environnement. De plus, cette partie vision, couplée avec un planificateur de trajectoire, nous a permis de générer rapidement des trajectoires souhaitées sur la surface de travail.

Des améliorations d'architectures sont à envisager sur plusieurs points. Premièrement, des hypothèses simplificatrices, au niveau de la rigidité, ont été prises dans un contexte de performance. Un modèle de rigidité plus complet de bras devra être implémenté pour mieux prendre en compte les effets de rigidité des membrures, dû aux critères de faible masse.

Deuxièmement, des améliorations pourront être effectuées au niveau de la performance des encodeurs utilisés. Leur résolution actuelle est de 12 bits, ce qui peut devenir limitant pour

le contrôle du bras. De plus, les encodeurs sont placés au niveau de l'axe moteur, ce qui ne permet pas de connaître avec exactitude la rotation de l'articulation au niveau du réducteur harmonique. La solution actuellement utilisée est de prendre en compte la rigidité du réducteur harmonique pour la compenser. Une solution plus performante serait d'implémenter un encodeur magnétique au niveau du réducteur.

Enfin, les tâches avec contact n'ont pas pu être expérimentées sur le modèle trois axes. Dans une optique d'expérimentations de tâches avec contact, des essais supplémentaires pourront être réalisés sur le modèle UR10.

CHAPITRE 7 CONCLUSION

7.1 Synthèse des travaux

Dans ce mémoire, une présentation complète de la conception et la téléopération d'un bras robotique ultraléger a pu être menée. Une revue de la littérature dans le domaine de la robotique aérienne et de la téléopération de bras robotique, nous a permis d'orienter les choix de conception de notre bras robotique. De plus, ces choix ont dû se faire dans la continuité des précédents développements du bras robotique au sein du Centre des Technologies de Fabrication en Aérospatial du CNRC.

Le bras robotique a été modélisé dans un environnement numérique à partir des précédents développements menés sur son optimisation. Ce qui a permis dans un premier temps, d'effectuer diverses expériences de contrôle à partir de l'interface haptique sans réellement mettre en danger le bras robotique et son environnement. Une architecture électrique a été proposée permettant de mettre en relation tous les composants constituant le bras robotique. Par la suite, une architecture de contrôle a pu être proposée avec un détail des différentes compensations intégrées à la commande et le choix d'un planificateur de trajectoire. Une solution de vision par ordinateur a été implémentée, afin de faciliter l'opérateur lors du mouvement du bras robotique dans son environnement de travail. Cette solution se complète avec l'ajout d'une interface haptique avec retour d'effort. Une interface utilisateur a été présentée ayant pour objectif d'assister au mieux l'opérateur et cela en rassemblant la vision du bras robotique dans son environnement, sa modélisation numérique et différents paramètres de contrôle. Deux modes de contrôle ont été intégrés, le premier mode libre (Free Motion Mode) permettant de retranscrire les mouvements effectués par l'opérateur via l'interface haptique. Et le second mode supervisé (Supervised Motion Mode) permettant d'assister l'opérateur avec des profils de déplacement prédéfinis (carré, trochoïdale, spiral, etc.) le long de la surface de travail.

Après cette présentation détaillée du bras robotique, des expériences ont pu être menées de manière itérative afin de valider le bon fonctionnement de l'architecture de contrôle tout au long de sa conception. Des mesures de délais de communication, d'erreur en position, d'évitement d'obstacle et de force de retour à l'interface haptique ont pu être effectuées.

7.2 Limitations de la solution proposée

Les solutions proposées pour la conception de ce bras robotique comportent des limitations, notamment vis-à-vis des tâches de contact de l'outil avec la surface de travail. Celles-ci n'ont pas pu être expérimentées réellement avec le bras UR10. De plus, la stabilité du bras n'a pas été étudiée en prenant en compte l'interface d'accroche entre la zone de travail et les ventouses permettant d'accoster le drone. En effet, le contact du bras sur la surface de travail pourrait agir sur l'accroche des ventouses et vice versa. Des limitations de performance du bras sont à étudier pour différents scénarios, tel que le comportement de celui-ci lors d'une mauvaise réception ou un signal de mauvaise qualité.

7.3 Améliorations futures

Les prochains travaux à réaliser seront la fabrication du prototype et son intégration sur le drone. La fabrication ayant déjà été retardée à cause de délais d'approvisionnement de certaines pièces voire l'impossibilité de fourniture, il sera intéressant de considérer l'utilisation d'un bras robotique léger commercial (tel que le Kinova Gen3¹). Ce bras robotique commercial pourra être adapté au drone pour tester différents scénarios sur des tâches de maintenance. Ceci permettra par la suite de quantifier et valider les performances de la commande et de la téléopération du bras robotique intégré au drone.

Une amélioration à apporter au bras robotique est l'ajout d'une seconde camera pour rendre la téléopération plus immersive. En effet, cet ajout permet d'obtenir un angle de vue différent de la première camera, dont le champ de vision pourrait être limité, permettant le bon fonctionnement du module de génération de trajectoire et d'évitement d'obstacle.

1. <https://www.kinovarobotics.com/product/gen3-robots>

RÉFÉRENCES

- [1] “Systèmes robotisés d’entretien des éoliennes | Aeronos.” [En ligne]. Disponible : <https://aeronos.com/fr/>
- [2] M. H. Bjerger, “Leading Edge Repair.” [En ligne]. Disponible : <https://roperobotics.com/index.php/leading-edge-repair/>
- [3] B. Hazel, J. Côté, Y. Laroche et P. Mongenot, “Field repair and construction of large hydropower equipment with a portable robot,” *Journal of Field Robotics*, vol. 29, n°. 1, p. 102–122, 2012, _eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20427>. [En ligne]. Disponible : <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20427>
- [4] D. Wuthier, D. Kominiaik, C. Kanellakis, G. Andrikopoulos, M. Fumagalli, G. Schipper et G. Nikolakopoulos, “On the design, modeling and control of a novel compact aerial manipulator,” dans *2016 24th Mediterranean Conference on Control and Automation (MED)*. Athens, Greece : IEEE, juin 2016, p. 665–670. [En ligne]. Disponible : <http://ieeexplore.ieee.org/document/7536029/>
- [5] A. Suarez, A. Caballero, A. Garofano, P. J. Sanchez-Cuevas, G. Heredia et A. Ollero, “Aerial Manipulator With Rolling Base for Inspection of Pipe Arrays,” *IEEE Access*, vol. 8, p. 162 516–162 532, 2020. [En ligne]. Disponible : <https://ieeexplore.ieee.org/document/9184867/>
- [6] P. Chermprayong, K. Zhang, F. Xiao et M. Kovac, “An Integrated Delta Manipulator for Aerial Repair : A New Aerial Robotic System,” *IEEE Robotics & Automation Magazine*, vol. 26, n°. 1, p. 54–66, mars 2019. [En ligne]. Disponible : <https://ieeexplore.ieee.org/document/8641488/>
- [7] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck et G. Hirzinger, “The DLR lightweight robot : design and control concepts for robots in human environments,” *Industrial Robot : An International Journal*, vol. 34, n°. 5, p. 376–385, août 2007. [En ligne]. Disponible : <https://www.emerald.com/insight/content/doi/10.1108/01439910710774386/full/html>
- [8] A. Beyer, G. Grunwald, M. Heumos, M. Schedl, R. Bayer, W. Bertleff, B. Brunner, R. Burger, J. Butterfaß, R. Gruber, T. Gumpert, F. Hacker, E. Krämer, M. Maier, S. Moser, J. Reill, M. A. Roa, H.-J. Sedlmayr, N. Seitz et A. Albu-Schäffer, “CAESAR : Space Robotics Technology for Assembly, Maintenance, and Repair,” oct. 2018.
- [9] E. Allouis, R. Marc, M. Ruciński, M. Garland, D. Lachat, M. Shilton, G. Gajoch, A. Kisdi, W. Tubby, J. Hampton, M. Woods, M. Malinowski, S. Flitcroft, R. Durman et

- L. Tyler, “Larad – architecture and implementation of a lightweight advanced robotic arm demonstrator for future space and planetary applications,” juin 2017.
- [10] V. Omé, “Conception et optimisation d’un bras robotique ultraléger pour le déploiement de tâches de contact par drone,” p. 114.
- [11] B. Monsarrat, C. Vidal, M. De Montigny et I. Mantegh, “Base Attachment Module for Small Aerial Vehicles,” Brevet WO2020079668, avr. 2020.
- [12] R. Bogue, “Climbing robots : recent research and emerging applications,” vol. 46, n°. 6, p. 721–727, 2019, num Pages : 7 Place : Bedford, United Kingdom Publisher : Emerald Group Publishing Limited. [En ligne]. Disponible : <https://www.proquest.com/docview/2314535143/abstract/C00BC0F0E0BA46EDPQ/1>
- [13] J. Iqbal, A. Al-Zahrani, S. A. Alharbi et A. Hashmi, “Robotics Inspired Renewable Energy Developments : Prospective Opportunities and Challenges,” *IEEE Access*, vol. 7, p. 174898–174923, 2019, conference Name : IEEE Access.
- [14] T. P. Sattar, H. Leon Rodriguez et B. Bridge, “Climbing ring robot for inspection of offshore wind turbines,” *Industrial Robot : An International Journal*, vol. 36, n°. 4, p. 326–330, juin 2009. [En ligne]. Disponible : <https://www.emerald.com/insight/content/doi/10.1108/01439910910957075/full/html>
- [15] B. Hazel, J. Côté, Y. Laroche et P. Mongenot, “A portable, multiprocess, track-based robot for in situ work on hydropower equipment,” *Journal of Field Robotics*, vol. 29, n°. 1, p. 69–101, 2012, _eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20425>. [En ligne]. Disponible : <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20425>
- [16] “UR10e Collaborative industrial robot - Payload up to 12.5 kg.” [En ligne]. Disponible : <https://www.universal-robots.com/products/ur10-robot/>
- [17] R. Ladig, H. Paul, R. Miyazaki, K. Shimonomura et Ritsumeikan University 1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan, “Aerial Manipulation Using Multirotor UAV : A Review from the Aspect of Operating Space and Force,” *Journal of Robotics and Mechatronics*, vol. 33, n°. 2, p. 196–204, avr. 2021. [En ligne]. Disponible : <https://www.fujipress.jp/jrm/rb/robot003300020196>
- [18] X. Ding, P. Guo, K. Xu et Y. Yu, “A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems,” *Chinese Journal of Aeronautics*, vol. 32, n°. 1, p. 200–214, janv. 2019. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1000936118301894>
- [19] F. Ruggiero, V. Lippiello et A. Ollero, “Aerial Manipulation : A Literature Review,” *IEEE Robotics and Automation Letters*, vol. 3, n°. 3, p. 1957–1964, juill. 2018. [En ligne]. Disponible : <https://ieeexplore.ieee.org/document/8299552/>

- [20] M. N. Boukoherine, Z. Zhou et M. Benbouzid, “A critical review on unmanned aerial vehicles power supply and energy management : Solutions, strategies, and prospects,” *Applied Energy*, vol. 255, p. 113823, déc. 2019. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0306261919315107>
- [21] “Products | Prodrone | Revolutionary Drones for Professionals.” [En ligne]. Disponible : <https://www.prodrone.com/products/>
- [22] “HYbrid FLying-rolIng with-snakeE-aRm robot for contact inSpection | HYFLIERS Project | Fact Sheet | H2020.” [En ligne]. Disponible : <https://cordis.europa.eu/project/id/779411>
- [23] E. Allouis, R. Marc, M. Ruciński, M. Garland, D. Lachat, M. Shilton, G. Gajoch, Kisdi, W. Tubby, J. Hampton, M. Woods, M. Malinowski, S. Flitcroft et L. Tyler, “LARAD – ARCHITECTURE AND IMPLEMENTATION OF A LIGHTWEIGHT ADVANCED ROBOTIC ARM DEMONSTRATOR FOR FUTURE SPACE AND PLANETARY APPLICATIONS,” p. 9.
- [24] A. Flores-Abad, O. Ma, K. Pham et S. Ulrich, “A review of space robotics technologies for on-orbit servicing,” *Progress in Aerospace Sciences*, vol. 68, p. 1–26, juill. 2014. [En ligne]. Disponible : <https://linkinghub.elsevier.com/retrieve/pii/S0376042114000347>
- [25] R. Boumans et C. Heemskerk, “The European Robotic Arm for the International Space Station,” *Robotics and Autonomous Systems*, vol. 23, n°. 1-2, p. 17–27, mars 1998. [En ligne]. Disponible : <https://linkinghub.elsevier.com/retrieve/pii/S0921889097000547>
- [26] R. Bischoff et J. Kurth, “The KUKA-DLR Lightweight Robot Arm - a New Reference Platform for Robotics Research and Manufacturing,” p. 8.
- [27] “Discover our Gen2 robots.” [En ligne]. Disponible : <https://www.kinovarobotics.com/product/gen2-robots>
- [28] “Discover our Gen3 robots.” [En ligne]. Disponible : <https://www.kinovarobotics.com/product/gen3-robots>
- [29] K. Boman, M. Olofsson, J. Forsberg et S.-A. Boström, “Remote-Controlled Robotic Arm for Real-Time Echocardiography : The Diagnostic Future for Patients in Rural Areas?” *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, vol. 15, p. 142–7, avr. 2009.
- [30] V. Girbés-Juan, V. Schettino, Y. Demiris et J. Tornero, “Haptic and Visual Feedback Assistance for Dual-Arm Robot Teleoperation in Surface Conditioning Tasks,” *IEEE Transactions on Haptics*, vol. 14, n°. 1, p. 44–56, janv. 2021, conference Name : IEEE Transactions on Haptics.

- [31] G. Barbieri, “Control architecture for docking UAVs with a 7-DOF manipulator,” *Robotics and Mechatronics*, p. 47.
- [32] K. Wen, D. Neculescu et J. Sasiadek, “Haptic force control based on impedance/admittance control aided by visual feedback,” *Multimedia Tools and Applications*, vol. 37, n^o. 1, p. 39–52, mars 2008. [En ligne]. Disponible : <http://link.springer.com/10.1007/s11042-007-0172-1>
- [33] H.-C. Song, Y.-L. Kim et J.-B. Song, “Guidance algorithm for complex-shape peg-in-hole strategy based on geometrical information and force control,” *Advanced Robotics*, vol. 30, n^o. 8, p. 552–563, avr. 2016, publisher : Taylor & Francis _eprint : <https://doi.org/10.1080/01691864.2015.1130172>. [En ligne]. Disponible : <https://doi.org/10.1080/01691864.2015.1130172>
- [34] M. Moniruzzaman, A. Rassau, D. Chai et S. M. S. Islam, “Teleoperation methods and enhancement techniques for mobile robots : A comprehensive survey,” *Robotics and Autonomous Systems*, vol. 150, p. 103973, avr. 2022. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0921889021002414>
- [35] D. Escobar-Castillejos, J. Noguez, F. Bello, L. Neri, A. Magana et B. Benes, “A Review of Training and Guidance Systems in Medical Surgery,” *Applied Sciences*, vol. 10, p. 5752, août 2020.
- [36] P. F. Hokayem et M. W. Spong, “Bilateral teleoperation : An historical survey,” *Automatica*, vol. 42, n^o. 12, p. 2035–2057, déc. 2006. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0005109806002871>
- [37] J. Cui, S. Tosunoglu, R. Roberts, C. Moore et D. W. Repperger, “A REVIEW OF TELEOPERATION SYSTEM CONTROL,” p. 12.
- [38] F. Driewer, M. Sauer et K. Schilling, “MIXED REALITY FOR TELEOPERATION OF MOBILE ROBOTS IN SEARCH AND RESCUE SCENARIOS,” *IFAC Proceedings Volumes*, vol. 39, n^o. 3, p. 267–272, janv. 2006. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1474667015358687>
- [39] S. Gholami, M. Lorenzini, E. De Momi et A. Ajoudani, “Quantitative Physical Ergonomics Assessment of Teleoperation Interfaces,” *IEEE Transactions on Human-Machine Systems*, vol. 52, n^o. 2, p. 169–180, avr. 2022, conference Name : IEEE Transactions on Human-Machine Systems.
- [40] A. Bolopion, G. Millet, C. Pacoret et S. Régnier, “Haptic Feedback in Teleoperation in Micro- and Nanoworlds,” *Reviews of Human Factors and Ergonomics*, vol. 9, n^o. 1, p. 57–93, nov. 2013. [En ligne]. Disponible : <http://journals.sagepub.com/doi/10.1177/1557234X13503293>

- [41] G. Niemeyer et J.-J. Slotine, “Stable adaptive teleoperation,” *IEEE Journal of Oceanic Engineering*, vol. 16, n^o. 1, p. 152–162, janv. 1991. [En ligne]. Disponible : <http://ieeexplore.ieee.org/document/64895/>
- [42] B. Hannaford, “Stability and performance tradeoffs in bi-lateral telemanipulation,” dans *Proceedings, 1989 International Conference on Robotics and Automation*. Scottsdale, AZ, USA : IEEE Comput. Soc. Press, 1989, p. 1764–1767. [En ligne]. Disponible : <http://ieeexplore.ieee.org/document/100230/>
- [43] —, “A design framework for teleoperators with kinesthetic feedback,” *IEEE Transactions on Robotics and Automation*, vol. 5, n^o. 4, p. 426–434, août 1989. [En ligne]. Disponible : <http://ieeexplore.ieee.org/document/88057/>
- [44] R. Meijering, “Vision-based control of the SHERPA arm,” *Robotics and Mechatronics*, p. 67.
- [45] H. Nascimento, M. Mujica et M. Benoussaad, “Collision Avoidance in Human-Robot Interaction Using Kinect Vision System Combined With Robot’s Model and Data,” dans *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, oct. 2020, p. 10 293–10 298, iSSN : 2153-0866.
- [46] A. Winkler et J. Suchý, “Vision Based Collision Avoidance of Industrial Robots,” *IFAC Proceedings Volumes*, vol. 44, n^o. 1, p. 9452–9457, janv. 2011. [En ligne]. Disponible : <https://linkinghub.elsevier.com/retrieve/pii/S147466701645130X>
- [47] A. Singh, V. Kalaichelvi et R. Karthikeyan, “A survey on vision guided robotic systems with intelligent control strategies for autonomous tasks,” *Cogent Engineering*, vol. 9, n^o. 1, p. 2050020, déc. 2022. [En ligne]. Disponible : <https://www.tandfonline.com/doi/full/10.1080/23311916.2022.2050020>
- [48] V. Nandini, R. D. Vishal et C. A. P. a. S. Aishwarya, “A Review on Applications of Machine Vision Systems in Industries,” *Indian Journal of Science and Technology*, vol. 9, n^o. 48, p. 1–5, mai 2016, publisher : The Indian Society of Education and Environment. [En ligne]. Disponible : <https://indjst.org/>
- [49] F. Ruggiero, V. Lippiello et A. Ollero, “Aerial Manipulation : A Literature Review,” *IEEE Robotics and Automation Letters*, vol. 3, n^o. 3, p. 1957–1964, juill. 2018.
- [50] R. Bogue, “Climbing robots : recent research and emerging applications,” *Industrial Robot : the international journal of robotics research and application*, vol. 46, n^o. 6, p. 721–727, janv. 2019.
- [51] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck et G. Hirzinger, “The DLR Lightweight Robot – Design and Control Concepts for Robots in Human Environments,” *Industrial Robot*, vol. 34, p. 376–385, août 2007.

- [52] M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A. Q. Keemink, S. Stramigioli, R. Carloni et L. Marconi, “Developing an Aerial Manipulator Prototype : Physical Interaction with the Environment,” *IEEE Robotics Automation Magazine*, vol. 21, n^o. 3, p. 41–50, sept. 2014.
- [53] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. Martinez-de Dios, F. Pierri, J. Cortes, A. Santamaria-Navarro, M. A. Trujillo Soto, R. Balachandran, J. Andrade-Cetto et A. Rodriguez, “The AEROARMS Project : Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance,” *IEEE Robotics Automation Magazine*, vol. 25, n^o. 4, p. 12–23, déc. 2018.
- [54] M. Trujillo, J. R. Martínez-de Dios, C. Martín, A. Viguria et A. Ollero, “Novel Aerial Manipulator for Accurate and Robust Industrial NDT Contact Inspection : A New Tool for the Oil and Gas Inspection Industry,” *Sensors*, vol. 19, n^o. 6, p. 1305, janv. 2019.
- [55] R. Cano, C. Perez, F. Pruaño, A. Ollero et G. Heredia, “Mechanical Design of a 6-DOF Aerial Manipulator for assembling bar structures using UAVs,” 2014.
- [56] C. D. Bellicoso, L. R. Buonocore, V. Lippiello et B. Siciliano, “Design, modeling and control of a 5-DoF light-weight robot arm for aerial manipulation,” dans *2015 23rd Mediterranean Conference on Control and Automation (MED)*, juin 2015, p. 853–858.
- [57] D. Wuthier, D. Kominiak, C. Kanellakis, G. Andrikopoulos, M. Fumagalli, G. Schipper et G. Nikolakopoulos, “On the design, modeling and control of a novel compact aerial manipulator,” dans *2016 24th Mediterranean Conference on Control and Automation (MED)*, juin 2016, p. 665–670.
- [58] A. Suarez, A. E. Jimenez-Cano, V. M. Vega, G. Heredia, A. Rodriguez-Castaño et A. Ollero, “Design of a lightweight dual arm system for aerial manipulation,” *Mechatronics*, vol. 50, p. 30–44, avr. 2018.
- [59] J. Iqbal, A. Al-Zahrani, S. Alharbi et A. Hashmi, “Robotics Inspired Renewable Energy Developments : Prospective Opportunities and Challenges,” *IEEE Access*, vol. 7, p. 174 898–174 923, déc. 2019.
- [60] B. Hazel, J. Côté, Y. Laroche et P. Mongenot, “A portable, multiprocess, track-based robot for in situ work on hydropower equipment,” *Journal of Field Robotics*, vol. 29, p. 69–101, janv. 2012.
- [61] —, “Field repair and construction of large hydropower equipment with a portable robot,” *Journal of Field Robotics*, vol. 29, p. 102–122, janv. 2012.
- [62] D. Souto, A. Faiña, A. Deibe, F. Lopez-Peña et R. J. Duro, “A Robot for the Unsupervised Grit-Blasting of Ship Hulls,” *International Journal of Advanced Robotic Systems*, vol. 9, n^o. 3, p. 82, sept. 2012, publisher : SAGE Publications.

- [63] G. Hirzinger, N. Sporer, A. Albu-Schaffer, M. Hahnle, R. Krenn, A. Pascucci et M. Schedl, “DLR’s torque-controlled light weight robot III—are we reaching the technological limits now?” dans *2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, mai 2002, p. 1710–1716 vol.2.
- [64] A. Campeau-Lecours, H. Lamontagne, S. Latour, P. Fauteux, V. Maheu, F. Boucher, C. Deguire et L.-J. L’Ecuyer, “Kinova Modular Robot Arms for Service Robotics Applications,” *International Journal of Robotics Applications and Technologies*, vol. 5, p. 49–71, juill. 2017.
- [65] A. Wedler, M. Chalon, K. Landzettel, M. Görner, E. Krämer, R. Gruber, A. Beyer, H.-J. Sedlmayr, B. Willberg, W. Bertleff, J. Reill, M. Grebenstein, M. Schedl, A. Albu-Schäffer et G. Hirzinger, “DLR’s Dynamic Actuator Modules for Robotic Space Applications,” dans *Proceedings 41st Aerospace Mechanisms Symposium*, mai 2012.
- [66] K. Ueura et R. Slatter, “Development of the harmonic drive gear for space applications,” 1999.
- [67] J. Miller, M. Braun, J. Bilz, S. Matich, C. Neupert, W. Kunert et A. Kirschniak, “Impact of haptic feedback on applied intracorporeal forces using a novel surgical robotic system—a randomized cross-over study with novices in an experimental setup,” *Surgical Endoscopy*, vol. 35, n°. 7, p. 3554–3563, 2021. [En ligne]. Disponible : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8195941/>
- [68] Y. Xiong, S. Li et M. Xie, “Predictive display and interaction of telerobots based on augmented reality,” *Robotica*, vol. 24, n°. 4, p. 447–453, juill. 2006. [En ligne]. Disponible : https://www.cambridge.org/core/product/identifier/S0263574705002444/type/journal_article
- [69] N. Xi, T.-J. Tarn et A. Bejczy, “Intelligent planning and control for multirobot coordination : An event-based approach,” *IEEE Transactions on Robotics and Automation*, vol. 12, n°. 3, p. 439–452, juin 1996, conference Name : IEEE Transactions on Robotics and Automation.
- [70] J. Broenink et G. Hilderink, “A structured approach to embedded control systems implementation,” dans *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA’01) (Cat. No.01CH37204)*, sept. 2001, p. 761–766.
- [71] V. Lippiello, G. Fontanelli et F. Ruggiero, “Image-Based Visual-Impedance Control of a Dual-Arm Aerial Manipulator,” *IEEE Robotics and Automation Letters*, vol. 3, p. 1856–1863, juill. 2018.
- [72] R. Naderer, P. Ferrara et A. Rohrhofer, “Active handling apparatus and method for contact tasks,” US Brevet US9 375 840B2, juin 2016.

- [73] J. W. Sensinger, S. D. Clark et J. F. Schorsch, “Exterior vs. interior rotors in robotic brushless motors,” dans *2011 IEEE International Conference on Robotics and Automation*, mai 2011, p. 2764–2770.
- [74] K. Waldron et J. Schmiedeler, “Kinematics,” dans *Springer Handbook of Robotics*, B. Siciliano et O. Khatib, édit. Springer, Berlin, Heidelberg, 2008.
- [75] H. Zhang, J. Wang, G. Zhang, Z. Gan, Z. Pan, H. Cui et Z. Zhu, “Machining with flexible manipulator : Toward improving robotic machining performance,” vol. 2, août 2005, p. 1127–1132.
- [76] P. Beeson et B. Ames, “TRAC-IK : An open-source library for improved solving of generic inverse kinematics,” dans *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, nov. 2015, p. 928–935.
- [77] “traclabs / trac_ik — Bitbucket.” [En ligne]. Disponible : https://bitbucket.org/traclabs/trac_ik/src/master/
- [78] “Package : pilz_industrial_motion,” mai 2022, original-date : 2018-09-07T10 :22 :30Z. [En ligne]. Disponible : https://github.com/PilzDE/pilz_industrial_motion
- [79] A. Q. Keemink, H. van der Kooij et A. H. Stienen, “Admittance control for physical human–robot interaction,” *The International Journal of Robotics Research*, vol. 37, n°. 11, p. 1421–1444, sept. 2018, publisher : SAGE Publications Ltd STM. [En ligne]. Disponible : <https://doi.org/10.1177/0278364918768950>
- [80] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor et S. Schaal, “STOMP : Stochastic trajectory optimization for motion planning,” dans *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China : IEEE, mai 2011, p. 4569–4574. [En ligne]. Disponible : <http://ieeexplore.ieee.org/document/5980280/>
- [81] T. Hardeman, R. Aarts et B. Jonker, “Modelling and Identification of Robots with Joint and Drive Flexibilities,” dans *Solid Mechanics and its Applications*, janv. 2006, vol. 130, p. 173–182, journal Abbreviation : Solid Mechanics and its Applications.

ANNEXE A DESCRIPTION DE L'ARCHITECTURE DU ROBOT SRDF

```

<?xml version="1.0" encoding="UTF-8"?>
  <!--SUBGROUPS: Groups can also be formed by referencing to already defined group names-->
  <group name="arm">
    <joint name="virtual_joint"/>
    <joint name="Joint_1"/>
    <joint name="Joint_2"/>
    <joint name="Joint_3"/>
    <joint name="Joint_4"/>
    <joint name="Joint_5"/>
    <joint name="Joint_6"/>
  </group>
  <!--GROUP STATES: Purpose: Define a named state for a particular group, in terms of joint values.
  This is useful to define states like 'folded arms'-->
  <group_state name="home" group="arm">
    <joint name="Joint_5" value="-0.8722"/>
    <joint name="Joint_1" value="0.1046"/>
    <joint name="Joint_2" value="-0.8722"/>
    <joint name="Joint_3" value="1.57"/>
    <joint name="Joint_4" value="-0.785"/>
    <joint name="Joint_6" value="0"/>
  </group_state>
  <group_state name="work" group="arm">
    <joint name="Joint_5" value="2.26"/>
    <joint name="Joint_1" value="0.1046"/>
    <joint name="Joint_2" value="0.8722"/>
    <joint name="Joint_3" value="2.5"/>
    <joint name="Joint_4" value="-0.785"/>
    <joint name="Joint_6" value="0"/>
  </group_state>
  <!--END EFFECTOR: Purpose: Represent information about an end effector.-->
  <end_effector name="hand" parent_link="Link_7" group="arm"/>
  <!--VIRTUAL JOINT: Purpose: this element defines a virtual joint between a robot link and an
  external frame of reference (considered fixed with respect to the robot)-->
  <virtual_joint name="virtual_joint" type="fixed" parent_frame="world" child_link="Link_1"/>
  <!--DISABLE COLLISIONS: By default it is assumed that any link of the robot could potentially come
  into collision with any other link in the robot. -->
  <disable_collisions link1="Link_1" link2="Link_2" reason="Adjacent"/>
  <disable_collisions link1="Link_1" link2="UAV_BAM" reason="Adjacent"/>
  <disable_collisions link1="UAV_BAM" link2="Blade" reason="Adjacent"/>
  <disable_collisions link1="Link_2" link2="Link_3" reason="Adjacent"/>
  <disable_collisions link1="Link_2" link2="Link_5" reason="Never"/>
  <disable_collisions link1="Link_2" link2="Link_6" reason="Never"/>
  <disable_collisions link1="Link_2" link2="Link_7" reason="Never"/>
  <disable_collisions link1="Link_3" link2="Link_4" reason="Adjacent"/>
  <disable_collisions link1="Link_3" link2="Link_5" reason="Never"/>
  <disable_collisions link1="Link_3" link2="Link_6" reason="Never"/>
  <disable_collisions link1="Link_3" link2="Link_7" reason="Never"/>
  <disable_collisions link1="Link_4" link2="Link_5" reason="Adjacent"/>
  <disable_collisions link1="Link_5" link2="Link_6" reason="Adjacent"/>
  <disable_collisions link1="Link_5" link2="Link_7" reason="Never"/>
  <disable_collisions link1="Link_6" link2="Link_7" reason="Adjacent"/>
  <disable_collisions link1="Link_6" link2="camera_link" reason="Adjacent"/>
  <disable_collisions link1="camera_link" link2="Link_7" reason="Adjacent"/>
  <disable_collisions link1="camera_link" link2="Link_5" reason="Adjacent"/>
  <disable_collisions link1="Link_6" link2="camera_frame" reason="Adjacent"/>
  <disable_collisions link1="camera_frame" link2="Link_7" reason="Adjacent"/>
  <disable_collisions link1="camera_frame" link2="Link_5" reason="Adjacent"/>
  <disable_collisions link1="camera_frame" link2="camera_link" reason="Adjacent"/>
</robot>

```