



HAL
open science

La robotisation dans les entrepôts logistiques

Jean-François Brethé, François Guérin, Andréa Cynthia Santos, Imene Khames

► **To cite this version:**

Jean-François Brethé, François Guérin, Andréa Cynthia Santos, Imene Khames. La robotisation dans les entrepôts logistiques. Cahiers de la logistique, La robotisation dans les entrepôts logistiques (5), 2024. hal-04854528

HAL Id: hal-04854528

<https://normandie-univ.hal.science/hal-04854528v1>

Submitted on 23 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License



**La robotisation dans
les entrepôts
logistiques**

**PÔLE INGÉNIEUR
LOGISTIQUE
LE HAVRE NORMANDIE**

**CAHIERS DE LA
LOGISTIQUE**



**CAHIERS DE LA
LOGISTIQUE
CL2024 – N° 05**

La robotisation dans les entrepôts logistiques

Décembre, 2024
Copyright © 2024, ISEL



La présente publication est protégée par le code de la propriété intellectuelle et plus précisément ses articles relatifs au respect du droit d’auteur.

Les auteurs demeurent seuls responsables du contenu de leur œuvre. Ils garantissent disposer des autorisations nécessaires de la part des tierces parties titulaires de droits sur des œuvres partiellement ou globalement reproduites. Les droits moraux et patrimoniaux sur l’œuvre demeurent attachés à leurs auteurs.

Afin de faciliter le partage et l’utilisation de leur création, les auteurs autorisent les utilisateurs à :

- Télécharger et imprimer une copie de la présente publication à des fins d’études, de recherche, de diffusion de la culture scientifique et de partage des connaissances ;
- Distribuer gratuitement et sans aucune contrepartie l’URL identifiant la publication.

La présente publication, et l’URL associée ne peuvent en aucun cas être distribuées ou utilisées dans le cadre d’une activité à but lucratif ou à des fins commerciales. Aucune modification du contenu de l’œuvre n’est autorisée.

Préface

Le volume « **La robotisation dans les entrepôts logistiques** » des *Cahiers de la logistique* est consacré aux enjeux de l'automatisation et de la robotisation dans la transition que la logistique doit opérer pour intégrer les concepts de l'industrie 4.0, et au-delà.

Aujourd'hui, le secteur de la logistique est confronté à des défis majeurs :

- Il souffre d'une pénurie de main-d'œuvre due à des conditions de travail éprouvantes qui impactent la santé des travailleurs, et à une déshumanisation du travail, concept théorisé par Simone Weil.
- Il évolue dans un contexte de concurrence mondiale et doit donc faire preuve de productivité et d'efficacité.
- Il est au cœur des enjeux énergétiques mondiaux et incarne les solutions possibles pour une transition climatique.
- Il doit garantir une traçabilité irréprochable tout en assurant la fluidité du transit des marchandises sur les différents corridors logistiques.
- Il doit s'adapter aux nouveaux modes de consommation ainsi qu'aux réseaux de distribution qui en découlent.

Dans ce volume, François Guérin et Jean-François Brethé, tous deux enseignants-chercheurs à l'ISEL, présentent leurs travaux menés en collaboration avec des doctorants, des chercheurs, des laboratoires et des entreprises. Ces recherches s'inscrivent dans les projets FEDER XTERM (2015-2021), DRIVELOG et INTERREG COROT (2016-2022), ainsi que dans une thèse CIFRE réalisée avec FM *Logistics* (Aqib Khan, PhD, 2019). Le domaine de recherche exploré concerne la manutention automatisée dans les entrepôts logistiques.

Les trois premières contributions se concentrent sur les robots manipulateurs mobiles. Alors que les robots mobiles d'Amazon Robotics déplacent des étagères vers la personne en charge de saisir les objets (*Goods to Man*), les robots manipulateurs mobiles sont des robots mobiles autonomes (AMR, de l'anglais *Autonomous Mobile Robot*) équipés d'un bras manipulateur. Les principaux verrous scientifiques pour leur adoption à grande échelle incluent la localisation et la navigation autonome dans l'entrepôt, la sécurité des opérateurs humains (les systèmes de perception robotique doivent distinguer les opérateurs des obstacles statiques ou dynamiques non humains, Mourad Kenk, PhD, 2020), ainsi que la conception de systèmes mécatroniques capables de manipuler des colis de tailles et de masses variables, dans un espace restreint sous les racks et à une profondeur d'environ 1,50 m. Une synthèse des travaux réalisés lors de la thèse CIFRE avec FM *Logistics* soutenue en 2020 est présentée par le Dr Aqib Khan. Les problématiques d'identification et de localisation d'objets ainsi que les stratégies de

préhension ont bénéficié de l'introduction des techniques d'intelligence artificielle pour l'analyse d'images (Wafae Sebbata, PhD, 2021).

La quatrième contribution s'appuie sur la thèse de El Houssein Chouaib Harik (2016, *Air Ground Cooperation in Mobile Robotics*, dirigée par les Prs Frédéric Guinand et Jean-François Brethé et encadrée par François Guérin). Elle explore l'usage de drones pour réaliser des inventaires robotisés dans les entrepôts logistiques dans le cadre du projet DRIVELOG.

Les cinquième et sixième contributions adoptent une approche bio-inspirée des techniques de manipulation, fondée sur l'observation des comportements humains. Un torse humanoïde, conçu à partir de deux bras UR10, permet de manipuler des cartons de tailles variées (Khairidine Benali, PhD, 2019). Ce torse est ensuite utilisé pour une dépalettisation autonome à l'aide d'outils puissants d'IA visant à identifier, localiser et saisir divers cartons (Santheep Yesudasu, PhD, 2024).

Dans la septième contribution, Arnaud Belhomme, enseignant à l'ISEL, présente ses travaux sur l'allocation décentralisée de tâches dans des systèmes multi-robots, en collaboration avec Safran Nacelles (Gonfreville-l'Orcher). L'objectif est la gestion d'une flotte de robots mobiles terrestres, chargée d'apporter ou de retirer des caisses pleines ou vides sur les postes de travail des opérateurs responsables de l'assemblage des réacteurs de l'A320 NEO.

Ce volume met en lumière le savoir-faire des enseignants-chercheurs de l'ISEL dans le domaine de la robotique logistique (www.robotics-logistics.eu). Les membres de cette équipe de recherche sont ouverts à toute forme de collaboration et vous souhaitent des lectures studieuses et enrichissantes.

Jean-François Brethé

Professeur des Universités
GREAH, ISEL, ULHN

François Guérin

Maître de Conférences HDR
GREAH, ISEL, ULHN

Andréa Cynthia Santos

Professeure des Universités
LITIS, ISEL, ULHN
Directrice de l'ISEL

Imene Khames

Maître de Conférences
LMAH, ISEL, ULHN

FUSION MULTICAPTEURS POUR UNE NAVIGATION SÛRE DES ROBOTS MOBILES AUTONOMES DANS LES ENTREPÔTS LOGISTIQUES

Mourad A. Kenk¹, Jean-François Brethé², M. Hassaballah^{3,4}

¹Cybersecurity Department, College of Engineering and Information Technology
Buraydah Private College, Saudi Arabia

²GREAH, Université Le Havre Normandie, 75, Rue Bellot, 76058
Le Havre Cedex, France

³Department of Computer Science, Faculty of Computers and Information,
South Valley University, Qena, Egypt

⁴Department of Computer Science, College of Computer Engineering and Sciences,
Prince Sattam Bin Abdulaziz University, AlKharj, Saudi Arabia

RESUME

Dans la logistique 4.0, les tâches de manutention se robotisent. L'introduction des robots mobiles est une première étape qui permet de remplacer l'homme pour le transport et évite les pertes de temps liées aux déambulations dans les allées. Cependant ces robots mobiles qui partagent le même espace géographique que les opérateurs humains, doivent respecter des contraintes pour assurer la sécurité physique et psychique des opérateurs. Le robot, devenu cobot, doit apprendre à distinguer les obstacles humains des obstacles non humains. Dans cet article, nous nous intéressons à cette problématique. Nous montrons en effet que les LIDAR 2D ne suffisent pas pour gérer ces aspects et qu'il faut fusionner cette information avec des données issues d'autres capteurs comme les caméras RGB-D pour que le comportement du cobot soit sûr et socialement acceptable.

MOTS CLÉS: Entrepôts logistiques, Robots Mobiles Autonomes, Cobotique, Evitement d'obstacles, Sécurité opérateurs

ABSTRACT

In Logistics 4.0, handling tasks are being robotised. The introduction of mobile robots is a first step towards replacing human for transport and avoiding the time wasted wandering through aisles. However, these mobile robots, which share the same geographical space as human operators, must respect certain constraints to ensure the physical and psychological safety of the operators. The robot, now becoming a cobot, must learn to distinguish between human and non-human obstacles. This paper focuses on this issue. We show that 2D LIDARs information is not sufficient to manage these issues and that this information must be merged with data from other sensors such as RGB-D cameras to ensure that the cobot's behaviour is safe and socially acceptable.

KEYWORDS: Logistics Warehouses, Autonomous Mobile Robot, Cobotics, Collision avoidance, Human safety.

1. Introduction

Dans le secteur logistique et plus spécifiquement dans les activités de préparation de commande, les opérateurs humains passent plus de 50% de leur temps à déambuler dans les allées de l'entrepôt. Le reste du temps de travail est dédié aux tâches de saisie des objets ou de traçabilité. Or les tâches de saisie sont encore difficiles à robotiser du fait que dans un entrepôt, les géométries, les textures et les masses des objets à saisir sont souvent très différentes. De plus persistent un certain nombre de verrous scientifiques (identification de l'objet, localisation précise, détermination de la position de saisie) qui commencent seulement à être levés grâce aux développements de l'intelligence artificielle.

Pour augmenter la productivité, les premières solutions robotisées ont consisté à transporter les objets à saisir vers l'opérateur. Les objets sont placés sur des étagères et les étagères sont transportées par des robots mobiles. Les robots mobiles et les opérateurs ne partagent pas le même espace. Ce sont des systèmes robotisés développés initialement par Kiva Systems, société rachetée par la suite par Amazon. L'organisation robotisée est connue sous la dénomination "goods to man". Depuis de nombreux autres concurrents proposent des produits similaires.

Une autre organisation consiste à permettre aux robots mobiles autonomes de partager le même espace de travail que les opérateurs. Des sociétés comme Locus Robotics proposent cette organisation avec des "robots assistants" qui se positionnent devant les étagères. Les opérateurs s'approchent des robots assistants et lisent sur l'interface homme-machine les objets à saisir. De ce fait, ces robots deviennent collaboratifs et sont dénommés cobots. Dans ce contexte, il est impératif d'assurer la sécurité des opérateurs et l'objectif poursuivi est double : d'une part assurer l'intégrité physique et éviter les collisions homme-cobot, d'autre part, faire en sorte que le comportement du cobot soit socialement acceptable et ne provoque pas de stress pour l'opérateur. La théorie de la proxémique nous indique qu'une distance de 1.20 m est nécessaire entre un opérateur et un cobot en mouvement, avec une vitesse d'évolution réduite pour le cobot, pour que l'opérateur ne se sente pas en danger.

Une autre alternative consiste à installer un bras manipulateur sur un robot mobile pour en faire un "manipulateur mobile", capable de saisir les objets et de remplacer l'homme dans les deux tâches de déambulation et de saisie. Ces manipulateurs mobiles sont aussi souvent amenés à partager le même espace géographique que les opérateurs et les règles de sécurité précédemment mentionnées s'appliquent. Un des verrous scientifiques rencontrés est la nécessité de doter le cobot de capacités de perception et de décision lui permettant d'identifier un obstacle humain d'un obstacle non-humain. En effet, le comportement du cobot sera fondamentalement différent en fonction de cette classification : si l'obstacle est un humain, le cobot doit cesser tout mouvement lorsque la distance est inférieure à 1.20 m et si l'obstacle est non-humain, il peut s'approcher de l'obstacle (typiquement la palette) et effectuer ses mouvements de manipulation pour saisir l'objet désiré.

Dans cet article, nous nous focalisons sur cette dernière problématique. Après une étude de l'état de l'art, nous proposons une approche qui utilise deux types de capteurs (LIDAR 2D et caméra RGB-D), fusionne les informations dans la carte de navigation. La méthode est testée en simulation puis dans des conditions réelles.

2. Détection et localisation d'un opérateur humain

2.1 Etat de l'art

La navigation autonome de robots mobiles dans un entrepôt utilise principalement des LIDARs qui permettent de dresser une carte 2D où apparaissent les contours des obstacles dans une section plane de la scène à quelques centimètres du sol. Des caméras de type RGB-D peuvent aussi être exploitées pour avoir une vision 3D de la scène. Krajník et al. (2017) ont proposé d'utiliser des caractéristiques visuelles hétérogènes telles que des points, des segments, des lignes, des plans et des points de fuite pour traiter une localisation et une cartographie visuelles simultanées (SLAM) ou pour utiliser des informations de profondeur à partir d'un capteur à faible coût (RGB-D) pour localiser les obstacles. Dans Yang et al. (2016), un contrôleur non linéaire est conçu pour réaliser le suivi de la cible et l'évitement des obstacles dans des environnements complexes. Dans Malone et al. (2017), des ensembles stochastiques atteignables sont utilisés pour générer un champ de potentiel artificiel précis pour les obstacles dynamiques afin de planifier la trajectoire en ligne. Dans Nardi and Stachniss (2017), une approche probabiliste est développée pour modéliser les trajectoires incertaines des entités mobiles qui partagent l'espace de travail avec le robot.

La plupart des travaux de recherche précédents considèrent les humains comme des obstacles dynamiques en utilisant différentes techniques et modalités de détection. Néanmoins, considérer les humains comme des obstacles dynamiques pour simplement les éviter est certainement suffisant du point de vue de la sécurité, mais peut causer des désagréments pour les humains. En effet, la proxémique, qui théorise l'interaction spatiale homme-robot (HRSI), conseille de sélectionner des distances d'interaction appropriées entre les robots et les humains. Ainsi pour Lindner and Eschenbach (2017), en phase de navigation, le robot doit garder une distance supérieure à 1.20 m par rapport à l'homme pour ne pas violer son espace personnel.

De ce fait, il apparaît clairement que la navigation des cobots dans des environnements complexes nécessite de mettre en œuvre des algorithmes évolués. Par exemple, Choi et al. (2017) ont proposé une génération de trajectoire par processus gaussien prédisant la trajectoire future de l'opérateur en utilisant deux caméras Microsoft Kinect. Dans Zimmermann et al. (2018), la planification de la trajectoire du robot est faite en estimant la pose 3D de l'homme dans les images RGB-D. Dans Song et al. (2017), le schéma de commande combine l'évitement actif des obstacles et la prédiction du mouvement de l'homme en utilisant un détecteur de jambe par capteur laser 2D.

Linder et al. (2016) ainsi que Song et al. (2017) mettent en œuvre un classificateur binaire exploitant l'intensité du signal laser réfléchi par les jambes de l'opérateur en la caractérisant à l'aide de quatorze paramètres prédéfinis tels que le nombre de points réfléchis, la largeur, la linéarité, la circularité et le rayon, comme décrit par Arras et al. (2007). Néanmoins, ces méthodes produisent un nombre élevé de fausses détections ce qui rend difficile leur application dans les entrepôts logistiques. Comme les détecteurs humains basés sur la vision 3D montrent des performances élevées en environnement industriel Munaro et al. (2016) et entrepôts intralogistiques Linder et al. (2018), nous avons dirigé nos travaux vers la fusion des informations LIDAR et caméra RGB-D pour résoudre la problématique posée.

2.2 Principe générale de la méthode de fusion capteurs LIDAR 2D et caméra 3D

La commande de cobot mobile proposée intègre les informations de capteurs LIDAR (2D) et de caméras RGB-D (3D) afin d'être capable de distinguer en temps réel les obstacles humains et non-humains en environnement dynamique et de respecter l'intégrité physique des opérateurs et leur sérénité sur la base des critères de la proxémique.

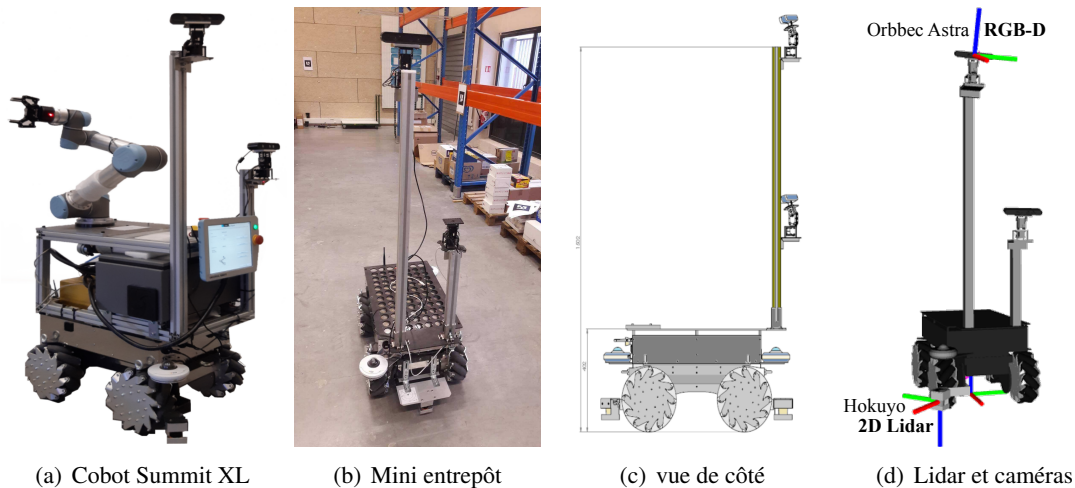


FIGURE 1 – Caractéristiques du cobot et de l'environnement

La Figure 1 montre les caractéristiques du robot mobile Robotnik utilisé pour les expériences. Ce cobot est commandé via le Middleware ROS (Robot Operating System) disposant d'un logiciel de visualisation (Rviz) et de simulation dynamique (Gazebo). ROS possède en outre de nombreuses fonctionnalités avancées (Toolboxes) comme le module navigation autonome¹. Les signaux des LIDAR 2D permettent de dresser une carte des obstacles en temps réel dans un plan 2D parallèle au sol. Le module navigation dispose d'un planificateur global qui va donner les grandes directions de déplacement et d'un planificateur local qui permet d'éviter les obstacles qui pourraient se trouver sur la trajectoire générée et qui ne figurent pas dans la carte initiale. Il est donc important de bien déterminer les paramètres de ce module. Parmi ces paramètres, il faut ajuster une distance de sécurité par rapport aux obstacles pour construire la zone possible d'évolution du robot. C'est à ce point qu'il est nécessaire de savoir si l'obstacle est humain ou non-humain : pour un obstacle non-humain comme une palette, la distance sera fixée à 5 cm pour pouvoir s'approcher de la palette afin de saisir l'objet alors que pour un obstacle humain, nous fixerons la distance à 1.20 m. Chaque fois qu'un opérateur est détecté par la caméra 3D, il faudra utiliser cette information pour intégrer un obstacle correspondant à un disque virtuel de rayon 1.20 m dans la carte 2D LIDAR utilisée pour la navigation : il s'agit donc d'une fusion multicapteur.

2.3 Détection et localisation d'obstacles humains vs. non-humains

Pour la détection d'obstacles humains, le schéma fonctionnel est détaillé sur la Figure 2. Les signaux issus de la caméra utilisée de type RGB-D (Red-Green-Blue Depth) sont récupérés via la bibliothèque OpenNI et aboutissent à la construction d'un nuage de points 3D permettant de visualiser la surface de l'obstacle exposée au rayonnement infrarouge ou laser de la caméra. Il est nécessaire de travailler sur ces données pour réussir à classifier l'obstacle (humain vs. non-humain).

1. ROS navigation, <http://wiki.ros.org/navigation>.

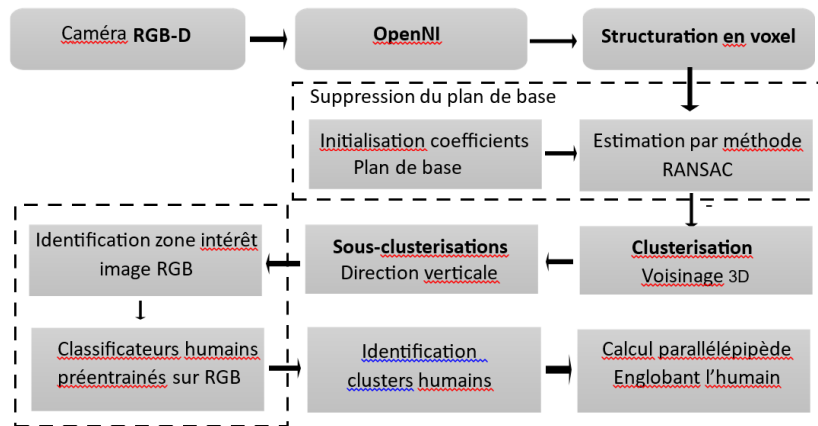


FIGURE 2 – Schéma fonctionnel de la méthode d'identification et de localisation d'un obstacle humain.

La bibliothèque logicielle PCL (Point Cloud Library) permet de passer d'une représentation des points 3D à une représentation en voxel (plus petit élément d'une image en 3D, équivalent du pixel en 2D) de type cube d'arête 6 cm. A ce stade, les éléments correspondant au plan de base (le sol) sont éliminés de l'image 3D par un algorithme de filtrage de type RANSAC (Random Sample Consensus) et méthode des moindres carrés. Les voxels restants sont agrégés dans des parallélépipèdes rectangles en fonction de leurs distances euclidiennes respectives. Pour chaque parallélépipède, une segmentation est effectuée sur l'axe vertical pour associer aux différentes positions verticales les volumes détectés. Cette distribution statistique est ensuite testée dans un détecteur 2D de caractéristiques de distribution type d'humains afin d'éliminer les fausses détections et de garder les vrais positifs. La position d'une tête par exemple peut être détectée.

À ce stade, deux méthodes sont étudiées :

- Une méthode habituelle d'apprentissage automatique de détection d'un humain basée sur l'histogramme de gradient orienté, (HOG) Histogram of Oriented Gradients avec les configurations d'entraînement recommandées par Dalal and Triggs (2005).
- Une méthode de réseau neuronal profond basée sur le calcul de renforcement mis en œuvre dans le CPU ou le GPU : Détecteur humain basé sur YOLO par Redmon and Farhadi (2018) dans le cadre open source Darknet avec 53 couches convolutives pour l'extraction de caractéristiques.

Les différentes étapes détaillées précédemment sont illustrées sur la Figure 3 : sur la première ligne, de gauche à droite, l'image de profondeur issue de OpenNI, puis l'agrégation des voxels dans des parallélépipèdes rectangles, la segmentation en sous-ensembles ; sur la seconde ligne, la détection d'humains sur l'image RGB (rectangle rouge pour le détecteur YOLO et rectangle jaune pour le détecteur HOG) et enfin la visualisation des régions 3D correspondant à des obstacles 3D humains visualisées dans l'environnement associées à un indice de confiance via une publication sur un ROS topic.

2.4 Création de la zone de sécurité autour de l'humain dans la carte LIDAR

Une fois les obstacles humains identifiés et localisés par le processus précédent, la théorie de la proxémie nous indique qu'une distance minimale de sécurité égale à 1.20 m doit être

2. Darknet, <https://pjreddie.com/darknet/>

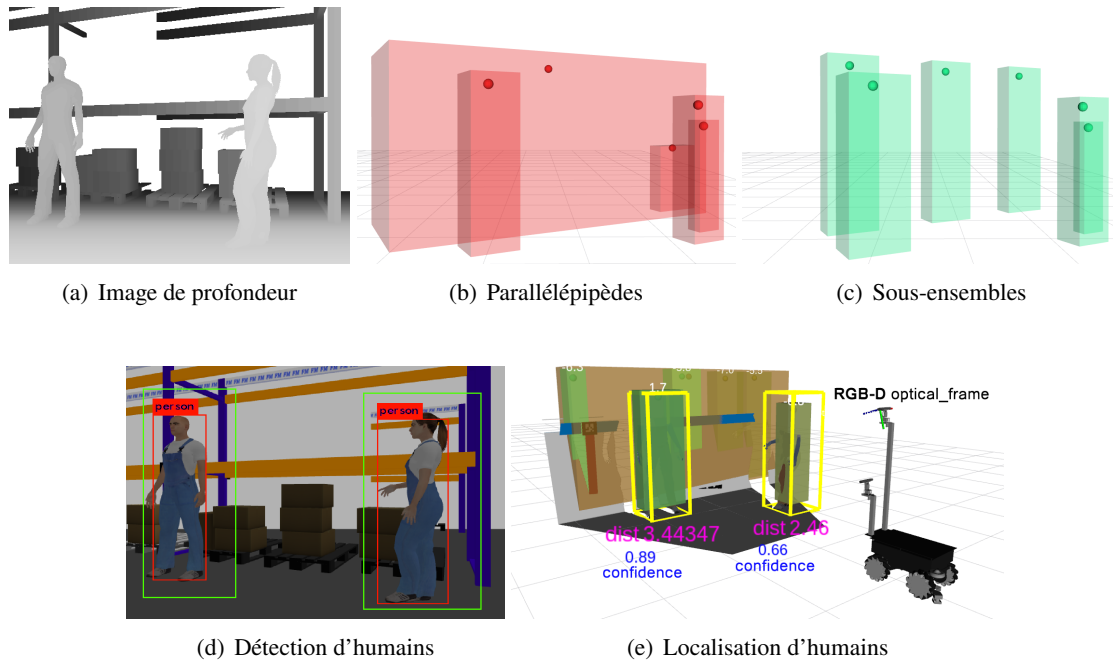


FIGURE 3 – Les différentes étapes de l’identification et localisation d’un humain

respectée entre l’homme et le cobot. Comme le cobot mobile navigue à partir de la carte LIDAR 2D de son environnement, il convient donc d’ajouter une zone plus large autour des obstacles humains pour que la présence d’un humain se traduise par un obstacle correspondant à un disque centré sur l’homme de rayon 1.20 m dans la carte LIDAR 2D du cobot mobile. Ce processus nécessite un traitement mathématique que nous avons mis en œuvre et détaillé dans Kenk et al. (2019) et qui est résumé sur la figure 4 : le centre du parallélépipède contenant l’homme est noté (x_c, y_c) ; L est le centre du LIDAR. Dans le secteur angulaire $\widehat{N_iLP_i}$, il convient de déterminer les points d’intersection de tous les impacts du laser avec l’arc de cercle $\widehat{N_iP_i}$ et d’intégrer ces points dans la carte 2D de l’environnement du cobot. Les calculs doivent être effectués pour chaque humain identifié. Dans notre cas, les LIDAR sont de type Hokuyo UST-20L placés à 12 cm du sol avec un rayon de détection de 20 mètres maximum et couvrent un secteur angulaire de 270° . Le pas de discrétisation des angles couverts par le LASER est de $\Delta = 0.25^\circ$.

Le processus de fusion consiste à injecter dans la carte LIDAR 2D les impacts du rayon Laser avec le disque fictif de rayon 1.20 m centré sur (x_c, y_c) . Ceci est illustré sur la Figure 5 : à gauche, les points noirs correspondent aux impacts des rayons laser avec un obstacle statique représenté par une palette et les deux jambes D_1 et D_2 d’un humain en déplacement dynamique ; à droite, les impacts sur les deux jambes sont remplacés par les impacts sur le disque fictif ; la carte est mise à jour en temps réel et est utilisée ensuite par le cobot mobile pour sa navigation.

3. Résultats expérimentaux

Des expériences ont été menées dans la halle robotique de l’ISEL (PIL) au sein d’un mini entrepôt logistique. Le cobot mobile utilisé est un Summit XL de la société Robotnik qui dispose de deux capteurs laser aux extrémités Nord-Ouest et Sud-Est de la plate-forme permettant ainsi de couvrir l’intégralité du plan autour du cobot. Deux caméras Orbbec Astra de type RGB-D sont montées sur deux mâts permettant une vue 3D dans l’espace environnant. L’ensemble du dispositif est détaillé

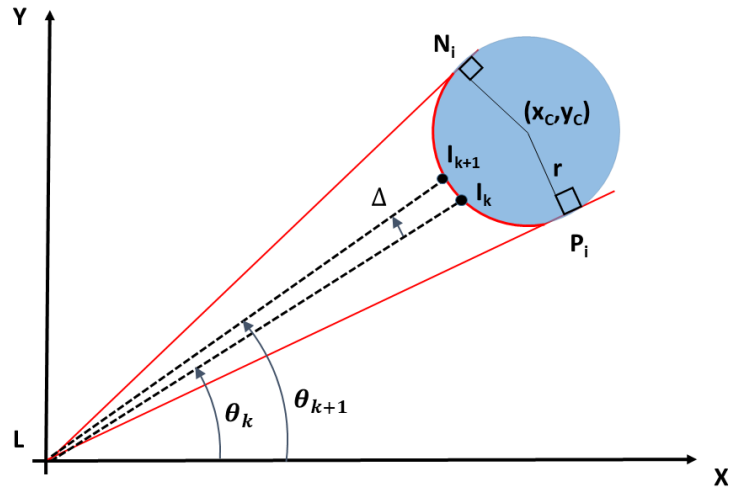


FIGURE 4 – Calcul dynamique dans la carte LIDAR 2D de la zone de sécurité de l'humain

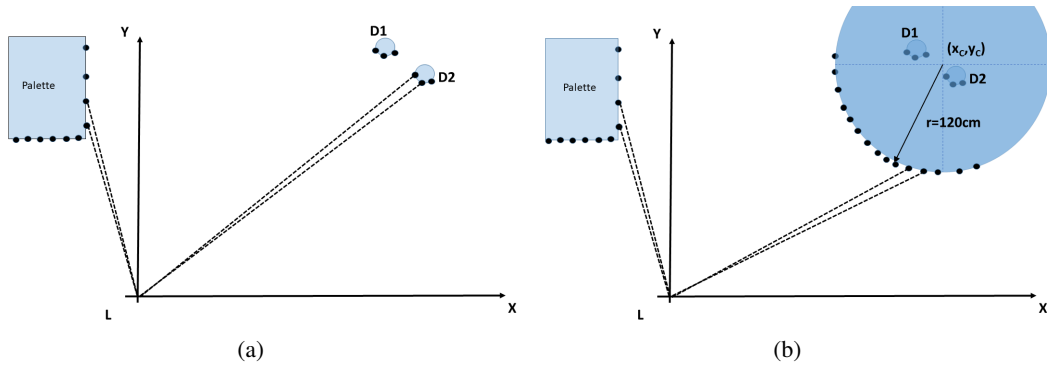


FIGURE 5 – Fusion dynamique des informations dans la carte LIDAR 2D

sur la Figure 1.

3.1 Performances des méthodes de détection d'un humain

Pour évaluer les performances de la détection d'un humain dans l'environnement du cobot mobile, nous suivons la méthodologie proposée par Goutte and Gaussier (2005). La métrique définie est construite à partir de quatre catégories : humain détecté et présent (Vrai Positif VP), humain détecté et absent (Faux Positif FP), humain non détecté et présent (Faux Négatif FN), humain non détecté et absent (Vrai Négatif VN). Plusieurs indicateurs de performance sont couramment estimés à partir des occurrences de ces catégories :

- $I_P = VP / (VP + FP)$ est le rapport des détections exactes d'un humain sur l'ensemble des détections qui prédisent la présence d'un humain ("Precision" en anglais)
- $I_R = VP / (VP + FN)$ donne le rapport des détections correctes sur l'ensemble des cas où l'humain est présent ("Recall").
- $I_A = (VP + VN) / (VP + VN + FP + FN)$ donne le rapport des détections ou non détections exactes sur l'ensemble des détections ("Accuracy").
- $I_{FP} = FP / (FP + VN)$ donne le rapport des fausses détections sur l'ensemble des cas où l'humain est absent ("FP-Rate").

Les deux méthodes proposées précédemment pour la détection d'un humain à partir

TABLE 1 – Performances de la méthode de détection d’un humain à partir d’une caméra RGB-D dans l’entrepôt.

Méthode	GPU	dét (sec)	Hz	I_A	I_P	I_R	I_{FP}
HOG	Off	0.039	18.82	0.718	0.664	0.619	0.154
HOG	On	0.021	34.57	0.741	0.761	0.672	0.136
YOLO	Off	1.961	5.29	0.966	0.906	0.813	0.021
YOLO	On	0.008	53.80	0.974	0.912	0.830	0.018

de l’utilisation des données de la caméra RGB-D (HOG et YOLO) sont mises en œuvre et les performances de ces méthodes sont comparées. La mise en œuvre de chaque méthode est testée sous deux configurations de calcul : utilisation de la CPU uniquement ou utilisation CPU/GPU. Nous disposons à cet effet d’un PC Intel Core i7-6700HQ CPU 2.60 GHz, 32 GB de RAM et Nvidia Quadro GPU M3000M avec 4 GB de mémoire et 1024 coeurs CUDA sous Ubuntu 16.04 (Xenial) avec ROS Kinetic.

Les résultats expérimentaux figurent dans le tableau 1. Il apparaît que globalement la méthode YOLO est nettement plus performante que la méthode HOG. Cette conclusion doit cependant être relativisée et pondérée par l’étude de la durée de détection (en seconde). Si la mise en œuvre de la méthode YOLO doit se faire sur CPU uniquement, la durée de détection de 1.961 s est incompatible avec un fonctionnement temps réel du cobot mobile car elle ne peut assurer le maintien de la distance minimale de confort entre le cobot et l’humain. La méthode doit donc être implémentée sur GPU/CPU ce qui restreint sa diffusion dans le cadre contraint de systèmes embarqués. Dans ce dernier cas, la méthode YOLO est très performante avec une durée de détection extrêmement rapide (8 ms). Mise en œuvre avec ou sans GPU, la méthode HOG assure la détection avec une fréquence compatible avec les contraintes de la proxémique. Cependant ses résultats sont moins fiables que la méthode YOLO.

3.2 Navigation en environnement simulé

La simulation de l’environnement de l’entrepôt logistique est conçue sous le simulateur Rviz où des racks parallèles standard sont installés à des distances de 3.0 m les uns des autres. Pour les expériences de cartographie et de simulation, nous avons créé une carte SLAM par ROS Hector Kohlbrecher et al. (2011) et créé deux jeux de paramètres de coût pour la navigation globale et locale. Pour la navigation locale, la distance minimale aux obstacles est fixée à 5 cm après intégration de la fusion des obstacles humains. L’évitement d’obstacles est géré à l’aide des messages initiaux des LIDAR 2D enrichi par la fusion comme l’illustre la Figure 6.

La Figure 7 montre comment le robot mobile évite les humains présents dans l’entrepôt logistique simulé. La carte utilisée pour la planification locale est en gris clair ; la trajectoire verte est celle prévue par le planificateur global ; la trajectoire bleue est celle du planificateur local ; en vert le cercle de rayon 1.20 m délimitant la distance de sécurité.

3. Hector SLAM, http://wiki.ros.org/hector_slam

4. Vidéo, <https://vimeo.com/325696517>

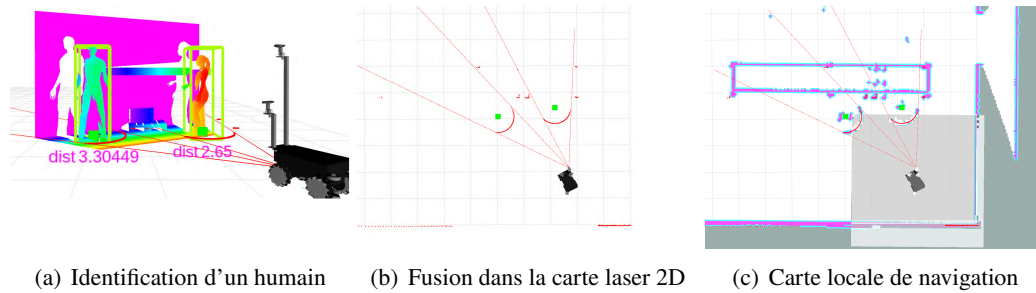


FIGURE 6 – Visualisation des processus d'identification 3D et de fusion laser 2D dans l'environnement simulé.

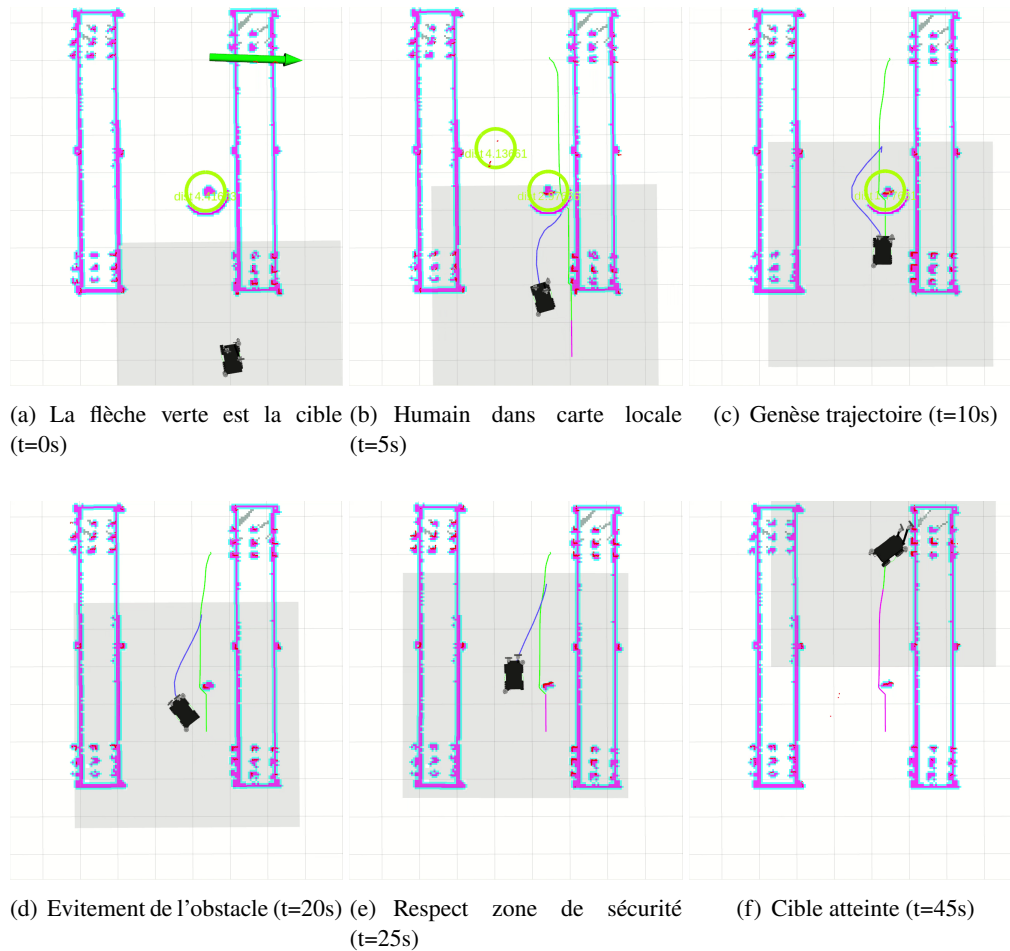


FIGURE 7 – Etapes clés de la navigation du cobot rencontrant un humain sur sa trajectoire dans l'environnement RViz.

3.3 Navigation en conditions réelles

Nous disposons d'un mini entrepôt logistique reconstitué avec un rack aux dimensions réelles dans une salle de longueur 20.5 m par 8.5 m. Le robot Summit XL précédemment cité est utilisé et navigue de manière autonome grâce au module Gmapping de ROS. Du fait que le cobot mobile ne dispose pas de GPU, nous sommes contraints de mettre en œuvre la méthode HOG pour la détection d'humain à partir des informations fournies par la caméra RGB-D. La Figure 8 démontre schématiquement la pertinence et l'efficacité du processus : en noir les obstacles repérés par le cobot dans la carte initiale ; en bleu la trajectoire planifiée initialement par le cobot pour

atteindre sa cible (target) ; en rouge, la trajectoire d'un humain qui se déplace de manière simultanée dans l'entrepôt ; en vert, la trajectoire d'évitement suivie par le cobot en respectant une distance de 1.20 m avec l'humain générée par le planificateur local.

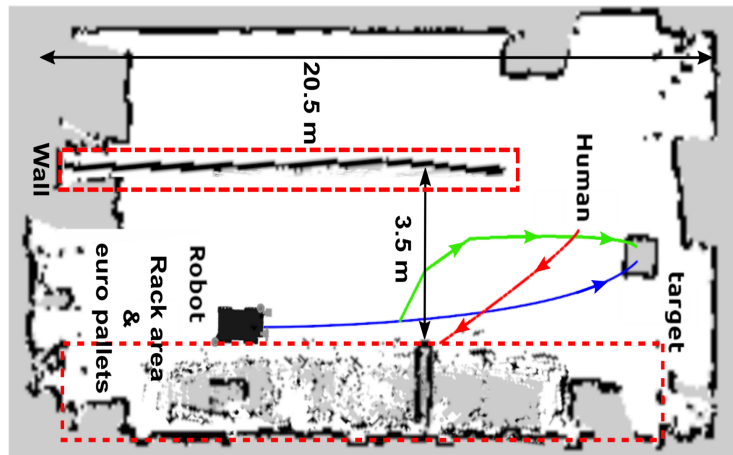


FIGURE 8 – Trajectoires du cobot (initiale en bleu et replanifiée en local en vert) et de l'humain (en rouge) dans le mini entrepôt.

4. Conclusions

Dans cet article, nous avons cherché à résoudre la problématique de classification des obstacles perçus par un cobot suivant qu'ils correspondent à un opérateur humain ou non-humain. Cette problématique prend tout son sens lors de l'introduction de cobots mobiles dans les entrepôts logistiques dans la mesure où le cobot doit pouvoir s'approcher à quelques centimètres d'un obstacle non-humain alors qu'il doit maintenir une distance minimale de 1.20 m avec un humain afin de ne pas le perturber. Nous avons montré que l'utilisation des seuls LIDAR 2D étant insuffisante, il était possible d'utiliser des caméras RGB-D et nous avons testé deux méthodes (HOG et YOLO). Les performances de ces deux méthodes ont été comparées : la méthode YOLO s'est révélée meilleure mais nécessitant l'utilisation d'une GPU pour une application temps réel. Une fois l'humain identifié et localisé sous la forme d'un parallépipède rectangle de centre connu, nous utilisons le centre de ce pavé pour construire un disque de rayon 1.20 m. Les points d'impact virtuel du faisceau laser avec ce disque sont ensuite calculés et fusionnés avec la carte LIDAR 2D pour mettre à jour la carte que le cobot mobile utilise pour la navigation. Des tests ont été menés en simulation pour vérifier la pertinence de la méthode globale, puis en environnement réel dans un mini entrepôt. La phase suivante consistera à améliorer les performances des méthodes de détection 3D en particulier pour une mise en œuvre en temps réel.

Remerciements

Les auteurs remercient l'Université le Havre Normandie, la Région Haute-Normandie et le FEDER (Fonds Européen de Développement Régional) qui ont financé les travaux dans le cadre du projet XTERM : Usine du Futur.

Références

- Arras, K. O., Mozos, O. M., and Burgard, W. (2007). Using boosted features for the detection of people in 2d range data. In *IEEE International Conference on Robotics and Automation*, pages 3402–3407. IEEE.
- Choi, S., Kim, E., Lee, K., and Oh, S. (2017). Real-time nonparametric reactive navigation of mobile robots in dynamic environments. *Robotics and Autonomous Systems*, 91 :11–24.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision & Pattern Recognition*, volume 1, pages 886–893. IEEE.
- Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359, Berlin, Heidelberg. Springer, Springer Berlin Heidelberg.
- Kenk, M. A., Hassaballah, M., and Brethé, J.-F. (2019). Human-aware robot navigation in logistics warehouses. In *International Conference on Informatics in Control, Automation and Robotics*.
- Kohlbrecher, S., Von Stryk, O., Meyer, J., and Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 155–160. IEEE.
- Krajník, T., Cristoforis, P., Kusumam, K., Neubert, P., and Duckett, T. (2017). Image features for visual teach and repeat navigation in changing environments. *Robotics and Autonomous Systems*, 88 :127–141.
- Linder, T., Breuers, S., Leibe, B., and Arras, K. O. (2016). On multi-modal people tracking from mobile platforms in very crowded and dynamic environments. In *IEEE International Conference on Robotics and Automation*, pages 5512–5519. IEEE.
- Linder, T., Griebner, D., Vaskevicius, N., Arras, K. O., and Gmbh, R. B. (2018). Towards accurate 3d person detection and localization from rgb-d in cluttered environments.
- Lindner, F. and Eschenbach, C. (2017). An affordance-based conceptual framework for spatial behavior of social robots. In *Sociality and Normativity for Robots*, pages 137–158. Springer.
- Malone, N., Chiang, H.-T., Lesser, K., Oishi, M., and Tapia, L. (2017). Hybrid dynamic moving obstacle avoidance using a stochastic reachable set based potential field. *IEEE Transactions on Robotics*, 33(5) :1124–1138.
- Munaro, M., Lewis, C., Chambers, D., Hvass, P., and Menegatti, E. (2016). RGB-D human detection and tracking for industrial environments. In *Intelligent Autonomous Systems (IAS-13)*, pages 1655–1668. Springer.
- Nardi, L. and Stachniss, C. (2017). User preferred behaviors for robot navigation exploiting previous experiences. *Robotics and Autonomous Systems*, 97 :204–216.
- Redmon, J. and Farhadi, A. (2018). Yolov3 : An incremental improvement. *arXiv preprint arXiv :1804.02767*.

- Song, K.-T., Jiang, S.-Y., and Wu, S.-Y. (2017). Safe guidance for a walking-assistant robot using gait estimation and obstacle avoidance. *IEEE/ASME Transactions on Mechatronics*, 22(5) :2070–2078.
- Yang, H., Fan, X., Shi, P., and Hua, C. (2016). Nonlinear control for tracking and obstacle avoidance of a wheeled mobile robot with nonholonomic constraint. *IEEE Transactions on Control Systems Technology*, 24(2) :741–746.
- Zimmermann, C., Welschehold, T., Dornhege, C., Burgard, W., and Brox, T. (2018). 3D human pose estimation in RGBD images for robotic task learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1986–1992. IEEE.

Biographie des auteurs



Mourad A. Kenk est professeur assistant à la Faculté d'informatique et d'intelligence artificielle de l'Université de la Vallée du Sud, Hurghada, Égypte. En 2017, il a rejoint le Groupe de Recherche en Electrotechnique et Automatique du Havre (GREA) en Normandie, France, en tant que chercheur. Pendant deux ans, il a travaillé sur la robotique logistique pour développer un système de commande sûre des cobots mobiles dans le contexte de la préparation de commandes dans un entrepôt logistique. Ses recherches portent sur la vision par ordinateur et la robotique. En particulier, la détection et le suivi d'objets, l'estimation de la pose et la navigation visuelle. Il a soutenu une thèse de doctorat en 2020 sur cette thématique. Depuis 2024, il est professeur assistant au Cybersecurity Department, College of Engineering and Information Technology, Buraydah Private College, Saudi Arabia.



Jean-François Brethé est Professeur des Universités en robotique à l'Université Le Havre Normandie, Institut Supérieur d'Etudes Logistiques (ISEL) et membre du GREA (Groupe de Recherche en Electrotechnique et Automatique du Havre). Ses travaux de recherche portent initialement sur la modélisation, la conception et l'étude des performances en précision des robots manipulateurs industriels, puis se focalisent aujourd'hui sur les verrous scientifiques posés par l'introduction de cobots dans les entrepôts logistiques (préparation de commandes robotisée par robot manipulateur mobile, saisie de colis de taille variable par torse humanoïde).



M. Hassaballah a obtenu une licence en mathématiques en 1997 et une maîtrise en informatique en 2003 à l'université de South Valley en Égypte, puis un doctorat en informatique à l'université d'Ehime, au Japon, en 2011. Il a été chercheur invité au département d'informatique et de sciences de la communication de l'université de Wakayama (Japon) en 2013 et au laboratoire GREA de l'université Le Havre Normandie (France) en 2019. Il est actuellement professeur d'informatique au département d'informatique du College of Computer Engineering and Sciences de l'université Prince Sattam Bin Abdulaziz, en Arabie saoudite. Il est également professeur d'informatique au département d'informatique de la faculté d'informatique et d'information de l'université de South Valley, en Égypte. Il est membre du comité de rédaction de Pattern Analysis and Applications, Real-Time Image Processing, IET Image Processing, Imaging Science. Il a publié 5 livres et plus de 100 communications internationales.

AUTOMATION OF ORDER PICKING IN WAREHOUSES

Aqib Khan¹, Francois Guérin¹, Jean-François Brethé¹

¹Laboratoire GREAH, Université Le Havre Normandie
jean-francois.brethe@univ-lehavre.fr

RESUME

Les plates-formes logistiques actuelles sont conçues pour l'entreposage de marchandises, pour des opérations de cross-docking et pour des tâches de production. Dans un tel environnement, les nombreuses tâches de manutention peuvent se révéler pénibles à effectuer et exposer les opérateurs aux troubles musculo-squelettiques. L'automatisation du processus de préparation des commandes est une solution pertinente pour pallier ce problème. L'article présente les travaux de conception, développement et mise en œuvre d'un robot manipulateur mobile dans la halle robotique-logistique de l'ISEL. Ces travaux menés en partenariat avec la société FM Logistic ont permis de déterminer les verrous technologiques et scientifiques à résoudre pour effectuer ces tâches de manutention robotisée en environnement réel. Après avoir dressé l'état de l'art, les auteurs présentent leur méthodologie et les résultats de leur travaux.

MOTS CLÉS: Manipulation mobile, automatisation des entrepôts, navigation, commande non linéaire, systèmes à événements discrets.

ABSTRACT

Today's logistics platforms are designed for warehousing storing, cross-docking operations and production tasks. In such an environment, the numerous handling tasks can be painful and expose operators to musculoskeletal disorders. Automating the order-picking process is a relevant solution to this problem. This article describes the design, development and implementation of a mobile manipulator robot in the ISEL robotics-logistics hall. This work, carried out in partnership with FM Logistic, identified the technological and scientific obstacles to be overcome in order to carry out these robotic handling tasks in a real environment. After outlining the state of the art, the authors present their methodology and the results of their research work.

KEYWORDS: Mobile handling, warehouse automation, navigation, non-linear control, discrete event systems.

1. Introduction

1.1 Order preparation

Warehouse operations in logistics, such as receiving, sorting, storing, picking, repacking, and shipping goods, are being automated with robotics to enhance efficiency. One of the key activities, order picking, involves collecting goods from different locations in the warehouse, packing them, and shipping to fulfill customer orders. This process can be time-consuming, de Koster et al. (1999), and physically demanding for workers, leading to health issues, Calzavara et al. (2016), and increased sick leaves. The logistics industry is currently facing a labor shortage, exacerbated by the rapid growth of e-commerce, Ecommerce-Foundation (2015). This has created a gap between the availability of skilled workers and the operational needs of the logistics sector. Consequently, there is a growing demand to increase productivity and efficiency in warehouses.

1.2 Warehouse automation

Generally a typical warehouse surface is 6000 m², having multiple sections based on the type of activity. These large sections further contain long aisles of containment racks for storage of goods as seen in Figure 1. The structured environment of fixed racks yields perceptual aliasing, since all the racks are of the same type, i.e. dimensions, size and shape. These racks create 3 m wide and 100 m long alleys, and extend up to 12 m in height. Items are picked from ground-level pallets, which are placed by forklifts based on space availability. This dynamic environment is shared by workers and vehicles.

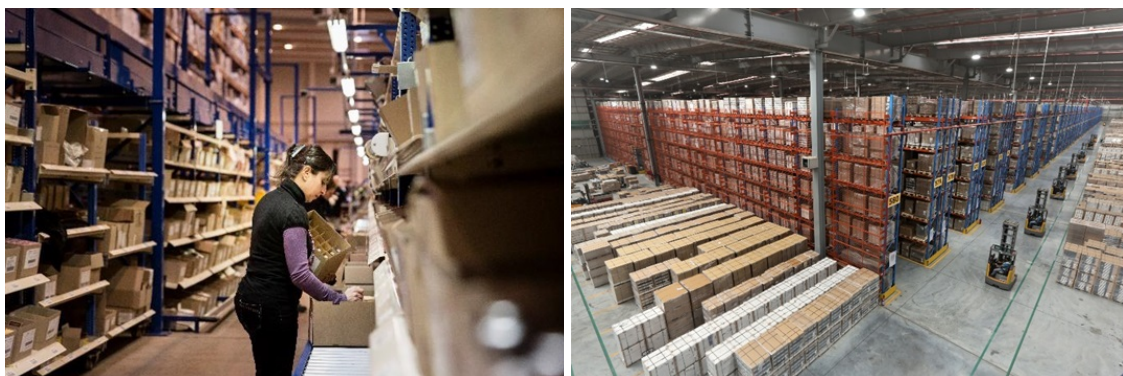


FIGURE 1 – Picking activity in a typical warehouse.

Taking into consideration the aforementioned issues, automation is used in warehouses to assist workers and expedite the logistics processes. This includes sorting systems that use stationary pickers, human or robotic, to handle items from a conveyor system. However, these systems require significant infrastructure changes, affecting the warehouse layout and introducing new safety zones. They also involve capital investment and limit flexibility for strategy changes. Order picking is often based on short-term contracts, requiring a balance of productivity, competition, and return on investment.

1.3 Collaborative robots

The more recent development and new trends in robotics, Peshkin et al. (2001), have brought collaborative robots to the factory floors and warehouses. As compared to the conventional industrial robots, this new generation of robots has been designed based on the concepts of shared environment and collaborative help. To provide more flexibility in terms of cutting down the time taken to walk and deliver, the new strategy is to use mobile picking units that can pick items and deliver.

2. State of the art and deadlocks

2.1 State of the art

Mobile picking solutions have been explored to address fatigue, time, and placement issues in warehouse operations. The scientific community has been studying mobile manipulation, with some focusing on cooperative object picking with humans, Hirata et al. (2005), while others on coordinated manipulation using multiple mobile manipulators, Khatib (1999). Over the past decade, several groups under the European Union Horizon 2020 initiative have aimed to bring autonomy to warehouses. They are been developing safe, flexible, hybrid robotic solutions that can integrate with existing facilities, handle pick-and-place tasks, and collaborate with human workers. Their focus areas included improving intralogistics (ILIAD), collaborative parts handling (ColRobot), shelf refilling in stores (REFILLS), developing manipulation strategies for variable object handling (PickPlace), and assisting warehouse technicians through dexterous manipulation and learning (SecondHands). The mentioned above projects and our experience of the work initiated in 2016, leads to the fact that the use of traditional industrial robots has limitations, including dexterity and mobility along with ensuring the safety of the robot and worker. To address these issues, companies like Amazon, GreyOrange, Scalog and Swisslog use a "goods to man" approach where mobile robots bring racks to workers for item picking. This method leverages human dexterity but does not utilize a versatile picking unit.

2.2 Scientific deadlocks in picking

European projects such as ILIAD and PickPlace were working on the challenge of picking by developing a versatile mobile picker. Their approach was cost-effective and flexible as it leveraged existing technology and infrastructure producing a readily deployable solution for the next application. However, adding mobility to robots introduced new challenges. The workspace of the robot expanded beyond a confined cell, requiring safety measures for co-working with humans in an open area. Additionally, these mobile robots now needed to navigate within warehouses, which imposed a requirement for localization, Ronzoni et al. (2011). This is more complex than outdoor localization because GPS is not available inside warehouses due to metal roofing and beams' structure. Warehouse racks are uniform, making localization challenging since an aisle cannot be distinguished from the other. Further, picking is complicated due to product variation of size and shape, requiring versatile grippers. The design of a universal gripper is still an open question. Given these constraints, it is crucial to decide which tasks are assigned to robots and which to workers.

3. Goals and methodology

3.1 Objective

Our goal was to create a convenient solution for picking that meets industrial objectives like return of investment and labor shortages. We used readily available components for easy integration and minimal infrastructure change. Given the challenge of picking a variety of products, we followed a balance between embedded intelligence in the mobile manipulator and the infrastructure. We followed the approach of the European projects to leverage existing infrastructure or integrate novel robotic technology. In collaboration with the industrial partner, we chose to use a mobile robot with grasping technology, i.e., a mobile manipulator. Our industrial partner, a logistics company with European clients, aimed to do mobile picking for both box and unit picking. With over 15,000 SKUs of varying sizes, shapes, and weights, a multidimensional strategy was required. All the products come in variable size, shape and weight and they could not be grasped with just one type of strategy. However, to simplify the approach, we constrained the product range. Figure 2 displays the data used to determine the number of references considering the weight variability and the boxes' dimensions.

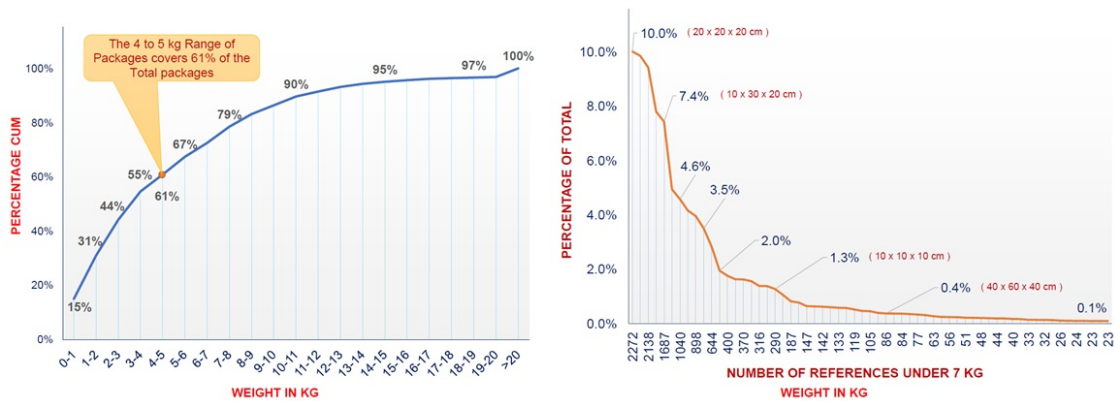


FIGURE 2 – Product Range with 5 kg limit (left) and weight vs. dimensions (right).

Our strategy covered 61 % of warehouse products and this also lead to the conclusion that weight and shape are crucial for gripper design. By extracting the dimensions of products under 5 kg, it was found that the products are mostly parallelepiped (box-shaped), in which the max and min dimensions helped to infer the gripper dimensions. Moreover, surface material like cardboard or plastic, affects the grip since products were often covered with cellophane, providing less friction to grasp and hold. For objects requiring two-sided grip, a clamp or finger gripper could be used. For wide surfaces, a vacuum gripper is preferred. We focused on vacuum gripping.

3.2 Application of process

To implement autonomous picking using mobile manipulation, we first developed a simulated environment using the Robot Operating System (ROS), Quigley et al. (2009). This included a complete warehouse environment and a mobile manipulator. The robot in the simulation in Figure 3 was a replica of the actual platform. For the picking process the robot's operations involve localization within the warehouse, path planning to the goal location, navigation execution,

and product manipulation. After developing and testing these components in simulation, the next phase consisted of testing the simulated framework in a real environment with an actual platform.

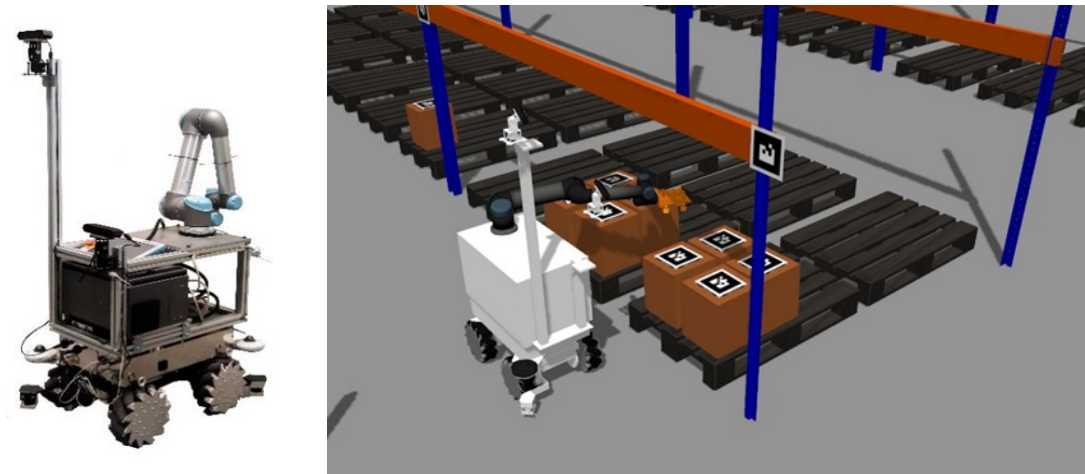


FIGURE 3 – Mobile manipulator and the simulated environment.

3.3 Development of platform

For mobile picking, we used a platform combining a 6 Degrees Of Freedom (DOF) UR5 arm and an omnidirectional mobile base (Robotnik Summit XL). The base was equipped with cameras and two 2D LIDARs (LIght Detection And Ranging) for obstacle's distance measuring. The integration of the platform was done in-house with the available off the shelf components and sensors. The arm, integrated with a vacuum pump, was mounted on the base for manipulation tasks. We used ROS for control and programming, a custom navigation controller utilizing the Hough Transform (HT) for robot navigation, and the Moveit motion planning framework for controlling the arm movements.

3.3.1 Localization and navigation

Localization is essential for a robot to identify its starting point and reach its goal. This process requires a reference, which can be a pre-existing map, an online-generated map using Simultaneous Localization And Mapping algorithm (SLAM), Wolf and Sukhatme (2004), or artificial indicators. Localization in a warehouse is more complex than outdoor localization with GPS due to the long aisles and infrastructure inside. To aid in localization without altering the layout, we used Augmented Reality (AR) markers placed on racks and aisles. These markers help differentiate one part of the environment from another and ensure reliable localization. Figure 3 right shows the placement of markers on box and racks in a simulated environment.

3.3.2 Path planning

For the motion inside the warehouse the robot requires to plan and execute its locomotion towards the goal. For this aspect a custom path planner using the Distance Transform Method algorithm (DTM), Xu et al. (2022), was developed and then a nonlinear control law was used to track this path while moving in the global space of the warehouse. The DTM algorithm considers

the task of path planning from the goal location to the start. This method uses a tessellated grid representation of the map and propagates distances from the goal cell to all the other cells, filling all of the space around the obstacles, similar to Dijkstra and A*. It employs a wave-front expansion from the goal cell spreading outwards while assigning each cell its distance from the goal. The CAD (Computer Aided Design) map of a warehouse was converted to a 2D occupancy grid tessellated representation. The occupied and free cells were then assigned respective values. The goal cell was assigned a null value and a 2D mask was used to propagate distances of the free cells incrementing the distances in all directions. The mask can either use 4 or 8 cells depending on the specified resolution of the grid.

The metric for distance propagation from cell to cell is the Euclidean distance. Once all the distance values have been assigned, the path to the goal is found by steepest gradient algorithm on the distance values of the cells. A snapshot of the 2D mask and the resulting map produced by DTM is displayed in Figure 4, where (x, y) represents the map matrix and $d(x, y)$ gives the distance matrix values to increment in the neighborhood cells of the center cell $C(x, y)$ using the mask as shown. The order propagation algorithm runs in a loop with each step processing the local neighborhood in order to increment the distance by one with respect to the current cell $C(x, y)$ at the center of the mask window. After the generation of a feasible path the robot follows the path accurately by a nonlinear control law. For navigating inside the racks, a custom nonlinear motion controller was developed utilizing the conventional HT applied to the distance data acquired during the scans.

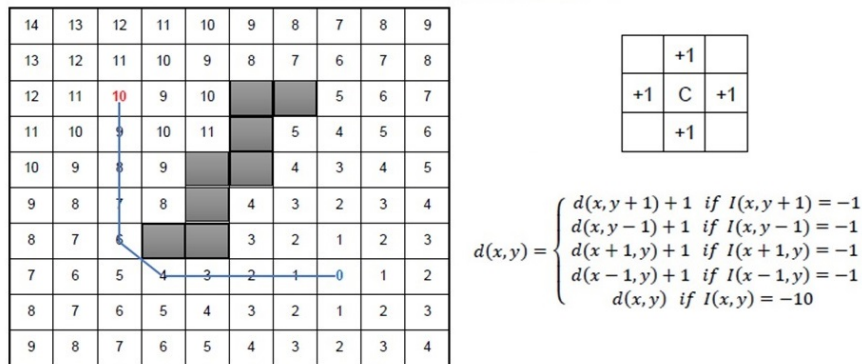


FIGURE 4 – Algorithm and the resulting map produced by the distance transform method.

3.3.3 Hough navigation

The pick location for items is usually inside the racks. Once the robot reaches this location using the path planner produced by the DTM, it then needs to navigate within the rack with a precision motion. This involves avoiding obstacles, aligning with the rack edge, and dealing with limited space. The typical width of an aisle formed by two racks is 3 to 3.5 meters, which restricts maneuverability. If two robots are operating in the same rack, they must follow a straight path along the rack edge to avoid collisions. To execute a precise motion inside a rack, it is visualized as a corridor-following problem and tracking a reference edge such as a wall in real time. The robot moves linearly while maintaining a fixed offset from the sides to avoid collisions. The racks, stocked with pallets, are considered as obstacles. A reference extraction and tracking algorithm is used to provide input for the robot's motion control. LIDAR data is used to extract the

reference of the rack, and the conventional HT is employed to convert the 2D LIDAR data points into lines or edges.

Duda and Hart (1972) proposed to use the conventional HT to convert the input data to a parametric representation in order to extract the parameters of the hypersurface (feature) to locate in the given signal. The input space is termed as the feature space and the transformed space is known as the parametric space. In the current scenario all the LIDAR points that lies on the same line constitute the feature space (line features).

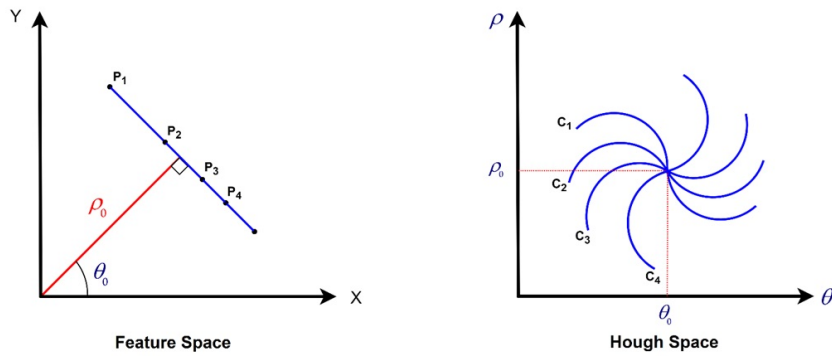


FIGURE 5 – Transformation of points into curves from feature space to Hough space.

HT uses the Hesse normal form of a line, transforming each point in the (x, y) space to a sinusoid in the parameter space indicated in Figure 5. An accumulator array is formed based on the discretization of (ρ, θ) in which θ can vary in $[0; \pi]$ and the values of ρ can be positive or negative depending on θ , with each cell incremented for each sinusoid passing through it. Intersections of sinusoids correspond to local peaks in the Hough space. This is applied to LIDAR range data to extract features like lines and curves representing walls and corners. The range data was acquired by two LIDARS mounted at the front and back of the robot with range of 10 meters and a measurement resolution of 0.25° . A range image (X, Y) was formed mapping the coordinates of the scan range points given by (x_f, y_f) for the front and (x_r, y_r) for the rear LIDAR respectively. HT was applied to the scans to extract line features. Figure 6 displays on the left the initial point distribution, and on the right the lines of interest resulting of the HT process superposed on the initial points.

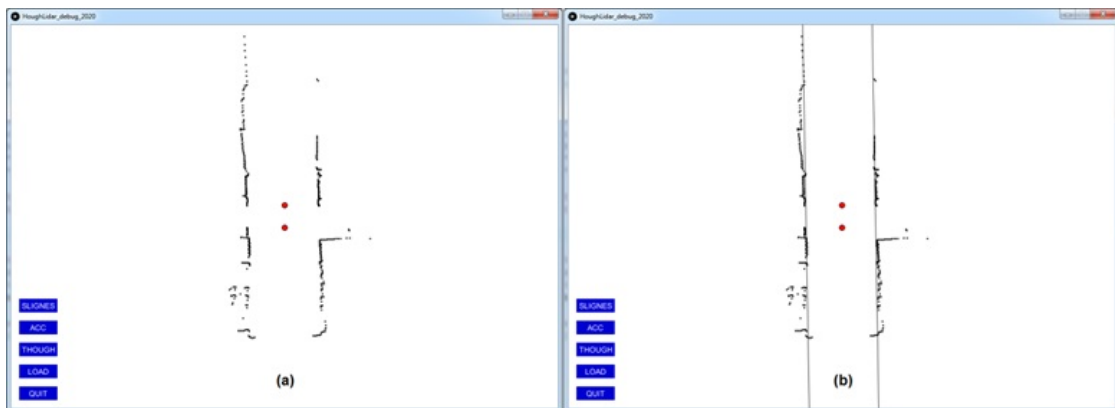


FIGURE 6 – Application of Hough Transform to the range data.

3.3.4 Control

The extracted Hough lines serve as control references for the robot to align with the rack and perform precise movements. The robot needs to achieve a specific separation distance and orientation relative to the rack. This was accomplished using a non-linear controller that considered the orientation offset and the linear distance to the rack. The robot aligned itself with the rack in real-time, adjusted its posture, and moved smoothly along the rack. The same controller also enabled the robot to align and make necessary adjustments while retrieving a package from a pallet. The details of the controller can be viewed in detail in the thesis Khan (2020) and in the video .

3.4 Manipulation

The 6 DOF UR5 arm installed on the robot is manipulated for picking tasks. The arm moves the gripper to the package location and grasps it. A custom motion planner produced the required trajectories for the 6 DOF arm in the Cartesian space, considering joint limits and collision avoidance. The target pick location was acquired by a camera and used by the motion planner. An inverse kinematic solver then produced joint trajectories to accurately position the gripper. By utilizing the Inverse Kinematics (IK) solver the custom motion planner produced linear and circular trajectories in the Cartesian space. Figure 7 left displays 4 poses of the arm while Figure 7 right displays the gripper center's trajectories.

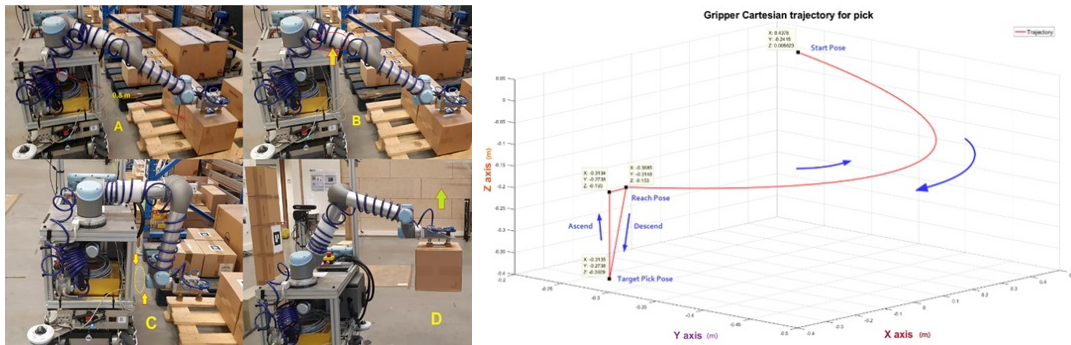


FIGURE 7 – UR5 arm motion

3.5 Behavior control

Before the picking operation, the Warehouse Management System (WMS) has assigned the box reference. The robot localizes itself, navigates to the pick location, aligns with the pallet, detects the package, grasps it, and transports it to another location. These actions form the robot's global behavior. In essence the dynamics of the picking task is hybrid because it is continuous between two events modifying the policy. Then it is necessary to use two different modelling : a continuous real-time controller (ROS) and a Discrete Event System (DES) controller based on Petri Nets (PN), Costelha and Lima (2007).

For each functional block, a state machine was developed using PN modelling, in which each node represents a specific state of the system governed by the firing of a transition for its state evolution as illustrated in Figure 8. Only three primitive behaviors were used : localization, navigation, and manipulation. Each behavior had a state machine, with sub-behaviors encoded in

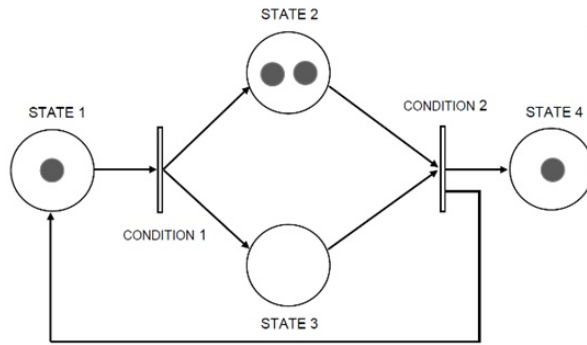


FIGURE 8 – A simple state machine using PN

each state. The PN graph was mathematically represented and simulated in Simulink (MATLAB) using equations of incidence matrix to evaluate the states and identify any deadlocks. This led to the implementation of the PN framework on the real robot. Three interconnected global PNs were developed to form a larger global supervisory net to manage the robot’s behavior, and implemented on the C++ platform in ROS. A simulation of each state machine was carried out in Matlab to verify the firing of transitions and evolution of the states as can be seen in the Figure 9. The simulation is an example of the localization state machine which shows the activation of different behaviors for a specific duration of the time.

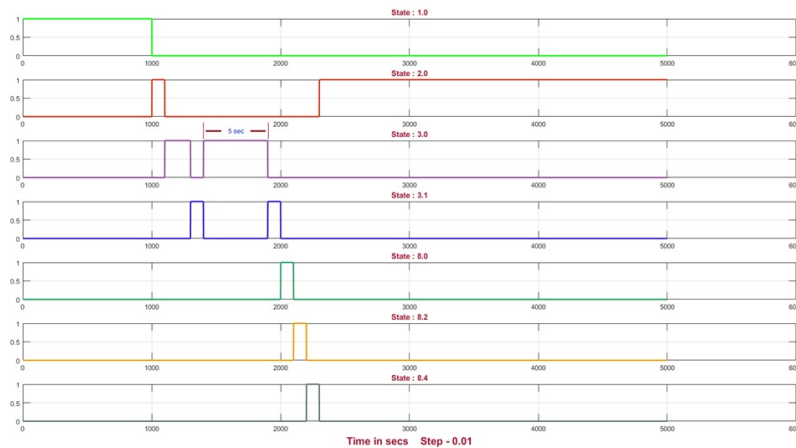


FIGURE 9 – Evolution of states in the state machine

3.6 Experimental results

To validate our approach and the frameworks of execution, we performed various tests and experiments. In the first step we developed the executional blocks in simulation leveraging ROS physics simulator Gazebo. After validating the results, we proceeded to the experiments in the real warehouse environment. In the first phase we tested the navigation and control, and in the second phase we performed experiments to grasp a box with the manipulator. Then combining the execution elements of the first and second phase, the third phase was the demonstration of a picking task in the warehouse employing the behavioral framework utilizing the state machines. The consecutive steps consisted of simultaneous localization and mapping, navigation, and manipulation of the box. For the localization we put the AR markers in the environment to specify the navigation references,

pallet locations and package placement.

To arrive at the pick location, the robot used the global planner to move through an aisle 3 meters wide, formed by the parallel placement of pallets. The robot navigates in the aisle keeping a fixed offset from one side which was possible due to our custom HT navigation controller, using the control parameter as the reference distance from the edge of the aisle. The robot after localizing itself for pallet location, aligns itself to the pallet position within 10 cm. After localizing the package, the manipulator is used to grasp the package with the vacuum gripper. The robot then de-aligns itself from the pallet and moves forward to detect the next pallet location and place the box. The successful execution of the pick and place task demonstrated the robustness of our behavioral control approach utilizing PNs as state machines, enabling to execute concurrent process and avoid deadlocks at the same time.

4. Conclusions

The objective of the order picking robotisation was to introduce mobile manipulators to pick up boxes below the racks and build up a pallet in a warehouse. We designed and implemented a customized mobile manipulator consisting of a XL Summit mobile base fitted with a UR5 arm. To reduce the complexity of picking up the large number of products stored, with large variations in shape, size and surface material, we chose to focus on the picking of parallelepiped boxes within 5 kg weight limit, by using a vacuum gripper able to pick up the box from the top. The autonomous picking task implies to develop the ability of the robot to localize and navigate. Hence again, we chose to reduce the complexity of the task by sticking AR markers on the racks. The navigation was successful thanks a global and local path planners using the HT method. Concerning the grasping task, the UR5 arm installed on the platform was used to execute linear trajectories divided in different steps. The high-level control of the robot via ROS was relying on a hybrid controller.

Experiments done in real industrial conditions gave impressive results close to simulation results. The mobile manipulator was able to pick up boxes in total autonomy, demonstrating the relevance of our design and control.

Such mobile manipulators will be more and more used in logistics warehouses where they will participate to mitigate high payload manipulation by humans resulting in musculoskeleton disorders and health problems. However, care will need to be taken to implement training and change management strategies to facilitate the societal adoption of such technologies.

Références

- Calzavara, M., Glock, C., Grosse, E., Persona, A., and Sgarbossa, F. (2016). Analysis of economic and ergonomic performance measures of different rack layouts in an order picking warehouse. *Computers Industrial Engineering*, 111.
- Costelha, H. and Lima, P. U. (2007). Modelling, analysis and execution of robotic tasks using petri nets. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1449–1454.
- de Koster, R., Roodbergen, K. J., and van Voorden, R. (1999). Reduction of walking time in the distribution center of de bijenkorf. In Speranza, M. G. and Stähly, P., editors, *New Trends in Distribution Logistics*, pages 215–234, Berlin, Heidelberg. Springer Berlin Heidelberg.

- Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15 :11–15.
- Ecommerce-Foundation (2015). <https://www.ecommerce-europe.eu/wp-content/uploads/2016/07/global-b2c-e-commerce-report-2015-light.pdf>. consulté le 1 novembre 2017.
- Hirata, Y., Kume, Y., Wang, Z., and Kosuge, K. (2005). Handling of a single object by multiple mobile manipulators in cooperation with human based on virtual 3-d caster dynamics. *JSME International Journal Series C-mechanical Systems Machine Elements and Manufacturing - JSME INT J C*, 48 :613–619.
- Khan, A. (2020). *Design and control of a robotic system based on mobile robots and manipulator arms for picking in logistics warehouses*. PhD thesis, Le Havre Normandie University.
- Khatib, O. (1999). Mobile manipulation : The robotic assistant. *Robotics and Autonomous Systems*, 26(2) :175–183. Field and Service Robotics.
- Peshkin, M., Colgate, J., Wannasuphprasit, W., Moore, C., Gillespie, B., and Akella, P. (2001). Cobot architecture. *Robotics and Automation, IEEE Transactions on*, 17 :377 – 390.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. (2009). Ros : an open-source robot operating system. volume 3.
- Ronzoni, D., Olmi, R., Secchi, C., and Fantuzzi, C. (2011). Agv global localization using indistinguishable artificial landmarks. pages 287–292.
- Wolf, D. and Sukhatme, G. (2004). “online simultaneous localization and mapping in dynamic environments”. pages 1301 – 1307 Vol.2.
- Xu, G., Meng, Z., Li, S., and Sun, Y. (2022). Collision-free trajectory planning for multi-robot simultaneous motion in preforms weaving. *Robotica*, 40 :1–20.

Biographie des auteurs



Aqib Khan est docteur en robotique. Sa thèse de doctorat, soutenue en décembre 2020, intitulée "Conception et commande d'un système robotisé à base de robots mobiles et bras manipulateurs pour le picking en entrepôt logistique", s'est inscrite dans le dispositif des thèses CIFRE financées par l'ANRT et a été réalisée au sein du laboratoire GREAH (Groupe de Recherche en Electrotechnique et Automatique du Havre), Université Le Havre Normandie et de l'entreprise FM Logistic.



Jean-François Brethé est Professeur des Universités en robotique à l'Université Le Havre Normandie, Institut Supérieur d'Etudes Logistiques (ISEL) et membre du GREAH. Ses travaux de recherche portent initialement sur la modélisation, la conception et l'étude des performances en précision des robots manipulateurs industriels, puis se focalisent aujourd'hui sur les verrous scientifiques posés par l'introduction de cobots dans les entrepôts logistiques (préparation de commandes robotisée par robot manipulateur mobile, saisie de colis de taille variable par torse humanoïde).



François Guérin est Agrégé de Génie Electrique-Electronique (1995) et Docteur de l'Université Le Havre Normandie en Automatique et Robotique (2004). Il a obtenu son Habilitation à Diriger des Recherches (2015) et travaille comme Maître de Conférences à l'Université Le Havre Normandie, au sein de l'ISEL et du GREAH. Ses enseignements portent sur l'Électronique (IoT), l'Automatique et la Robotique et ses travaux de recherche s'intéressent à la Commande Avancée des Systèmes, appliquée à la Robotique Mobile Terrestre et Aérienne. En 2021, il a créé au Havre (76), le Centre d'Innovation Drones de Normandie (CIDN) et a intégré en 2022, la Réserve Citoyenne de l'Armée de Terre auprès du 61eme Régiment d'Artillerie basé à Chaumont (52).

CONCEPTION ET MISE EN ŒUVRE D'UNE ALIMENTATION ROBOTISÉE AUTONOME D'UNE MACHINE-OUTIL PAR UN MANIPULATEUR MOBILE DANS LE CADRE DU PROJET COROT

Wafae SEBBATA¹, Jean-François BRETHÉ¹

¹Laboratoire GREAH, Université Le Havre Normandie

wafaesebbata@yahoo.fr

jean-francois.brethe@univ-lehavre.fr

RESUME

Le projet Interreg Trans-Manche CoRoT (2017-2022) avait pour objectif principal d'accompagner l'entrée des PME de l'Arc TransManche dans l'Industrie 4.0 en leur fournissant un certain nombre d'outils et de formations. Sur le plan des nouvelles technologies proposées, l'accent était mis sur la conception et la mise en œuvre de manipulateurs mobiles. Dans cet article est présentée la solution de manipulation mobile robotisée étudiée et réalisée au sein du laboratoire GREAH afin de pouvoir saisir des pièces brutes dans un magasin, les transporter en toute sécurité en traversant différents ateliers, puis les déposer avec précision dans les mors d'un étau de machine-outil. Pour fonctionner de manière autonome, chaque étape de ce processus nécessite l'utilisation d'algorithmes robustes et une modélisation fine des contraintes rencontrées. Les méthodes proposées permettent alors de lever les verrous scientifiques en utilisant des algorithmes issus de l'intelligence artificielle et ouvrent de nouvelles perspectives pour les entreprises utilisatrices.

MOTS CLÉS: Intralogistique, Manutention Robotisée, Manipulateur Mobile, Intelligence Artificielle, Industrie 4.0.

ABSTRACT

The Interreg Trans-Manche CoRoT project (2017-2022) aimed to support the transition of Arc Trans-Manche SMEs into Industry 4.0 by providing them with specific tools and training courses. The new technologies offer focused on the design and implementation of mobile manipulators. This article presents the mobile robotic handling solution designed and implemented by the GREAH laboratory to pick up raw parts from a storage area, transport them safely through various rooms, and place them precisely in the jaws of a machine-tool vice. In order to operate autonomously, each stage of this process requires the use of robust algorithms and detailed modelling of the constraints encountered. The proposed methods manage to overcome the scientific obstacles by using algorithms derived from artificial intelligence, and open up new opportunities for SMEs.

KEYWORDS: Intralogistics, Robotic Handling, Mobile Manipulator, Artificial Intelligence, Industry 4.0.

1. Introduction

Le projet Interreg Trans-Manche CoRoT a réuni de 2017 à 2022 des instituts de recherche (LINEACT CESI, IRSEEM-ESIGELEC, GREAH-Université Le Havre Normandie, Université d'Exeter, Université de Greenwich) et des entreprises fournisseurs de technologies (Autofina, Alstef Group, CERL..) afin d'améliorer la compétitivité des PME manufacturières dans le cadre de leur feuille de route vers l'Industrie 4.0. Les travaux du projet sont structurés en sept modules :

1. Gestion globale du projet (structurelle, opérationnelle, financière)
2. Communication (workshops, site internet, lettres d'information)
3. Conception et mise en œuvre en environnement industriel (CERI) d'un manipulateur mobile (Autofina, Alstef group, ESIGELEC, GREAH)
4. Utilisation de manipulateurs mobiles dans des cas d'usage spécifiques.
5. Supervision des systèmes manufacturiers intégrés flexibles
6. Organisation de formations (PMEs, écoles d'ingénieurs,...) et diffusion scientifique.
7. Études de marché

La flexibilité de l'outil de production a été identifiée comme l'un des défis majeurs à relever pour faire face à l'évolution observée des marchés (variabilité des volumes, petites séries, délais réduits). Pour répondre à ce challenge, le projet COROT a mené des actions spécifiques sur la conception et réalisation de manipulateurs mobiles : il s'agit de robots mobiles autonomes (AMR : Autonomous Mobile Robot) sur lesquels est fixé un bras manipulateur.

Les manipulateurs mobiles autonomes sont particulièrement adaptés à la préparation de commande où ils peuvent remplacer l'homme dans les tâches de saisie et transport dans les entrepôts logistiques. Un deuxième secteur d'activités prometteur est l'alimentation robotisée de machines, particulièrement les machines de test. En effet, si les centres d'usinage sont aujourd'hui conçus pour intégrer un dispositif de chargement/déchargement automatique de pièces à partir d'un magasin intégré, de nombreuses machines nécessitent encore d'être chargées par un humain. Par exemple les machines de tests (vibrations, température) pour les cartes électroniques sensibles doivent être alimentées à des intervalles réguliers (3 à 8h) et imposent une organisation de travail de nuit afin d'être utilisées en continu. Dans ce cadre, l'alimentation robotisée permet d'améliorer les conditions de travail et la productivité. C'est ainsi que le GREAH a proposé et travaillé sur un cas d'usage spécifique : l'alimentation robotisée des machines par des manipulateurs mobiles.

2. Présentation générale d'une étude de cas (GREAH)

2.1 Le scénario

Les opérations développées pour le cas d'études GREAH consistent à mettre en œuvre un manipulateur mobile afin de saisir des pièces brutes dans le magasin de stockage, transporter ces pièces en navigant à travers plusieurs ateliers pour arriver à la machine d'usinage, enfin placer la pièce dans la machine. L'ensemble des différentes étapes du processus doit fonctionner de manière autonome en temps réel et ceci nécessite de percevoir l'environnement et d'utiliser les informations collectées pour prendre les meilleures décisions. Les techniques utilisées et proposées

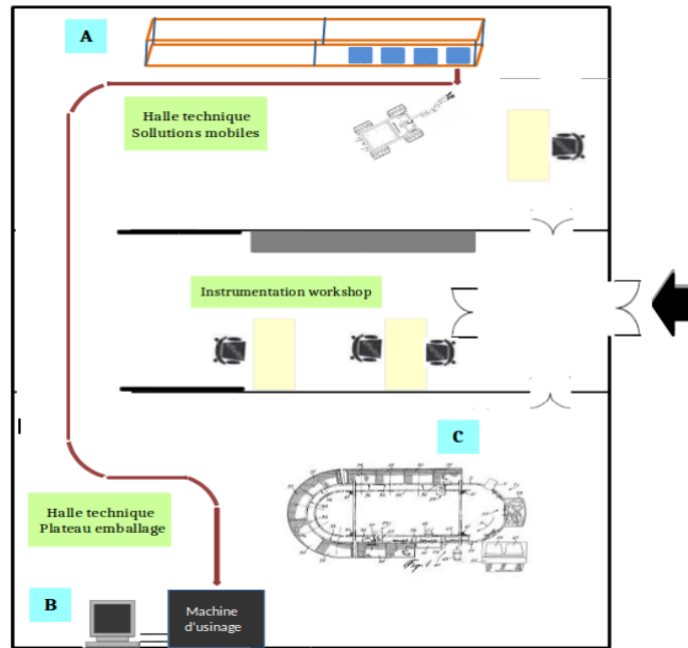


FIGURE 1 – La carte de la halle robotique du PIL

sont détaillées dans la thèse de Mme Sebbata (2021) et nous nous attachons dans cet article à restituer succinctement les verrous scientifiques à lever et les solutions mises en œuvre.

Le cas d’usage se déroule au sein de la halle robotique du PIL modélisée sur la Figure 1. Le manipulateur mobile doit assurer trois tâches distinctes qui comportent chacune des verrous scientifiques spécifiques :

- La manipulation d’objet : les pièces à usiner sont stockées dans la zone A. Le robot mobile Robotnik se positionne proche de la zone de stockage des pièces pour les saisir par le biais d’un bras manipulateur UR5 équipé d’une pince à deux doigts. Le déplacement du bras robotique dépend de l’estimation de la pose de la pièce obtenue par le traitement visuel. La pièce est alors déposée sur la plate-forme mobile et le bras replié en position de sécurité avant de passer à l’étape suivante. Les problématiques étudiées sont relatives au positionnement du robot mobile par rapport à la zone de saisie, l’identification et la localisation des pièces dans l’espace de stockage, le choix du mode de préhension (ouverture ou fermeture de pince) en fonction de la face présentée par la pièce et la planification de mouvement. Les travaux relatifs à la saisie des objets dans l’espace de stockage sont présentés dans la section 3.
- Le transport de l’objet : le robot mobile se déplace de manière autonome dans l’atelier pour aller de la zone A vers la zone C et s’approcher de la machine d’usinage. Les problématiques étudiées sont relatives à la navigation autonome, l’évitement d’obstacles et l’asservissement visuel pour le positionnement précis ; elles sont décrites dans la section 4.
- La dépose de la pièce dans la machine : cette tâche est contrainte par le jeu fonctionnel à respecter au niveau des mors de l’étau avant serrage automatisé. La répétabilité et la précision du robot manipulateur sont modélisées afin d’évaluer la fiabilité du processus et les travaux relatifs sont présentés dans la section 5.

2.2 Le manipulateur mobile

La Figure 2 montre le manipulateur mobile utilisé, constitué d'un bras manipulateur cobotique UR5 à 6 degrés de liberté embarqué sur un robot mobile holonome Summit XL. Une pince à deux doigts Robotiq est montée à l'extrémité terminale du bras robotique. Pour ce qui concerne la perception de son environnement, le manipulateur mobile est équipé des capteurs suivants : deux LiDARs permettant de repérer les obstacles dans un plan horizontal situé à 7 cm du sol et utilisés pour la navigation autonome ; deux caméras PTZ (Pan Tilt Zoom) permettant de scruter le demi-espace situé au-delà de 30 cm du sol. Deux caméras RGB-D (Red Green Blue - Depth) sont placées à des hauteurs différentes sur deux mâts fixés sur le robot mobile. Elles sont exploitées pour avoir une vue 2D et 3D d'une zone spécifique de l'espace de saisie. Une troisième caméra RGB-D est ajoutée à la plate-forme précisément au poignet de l'UR5. Cette caméra est utilisée pour la saisie adaptative des pièces brutes.



FIGURE 2 – Le manipulateur mobile « GREAH »

3. La saisie des pièces

L'objectif est de saisir de manière autonome une pièce brute dans le magasin de stockage. Nous avons conçu et imprimé en 3D une pièce industrielle en plastique, de couleur unie, sans texture et présentant des faces de caractéristiques différentes comme l'illustre la Figure 3. Le processus de saisie de cette pièce respecte les étapes suivantes : la détection d'objet, la classification des faces de l'objet pour un mode de préhension adaptatif, l'estimation de pose et le déploiement du bras robotique. La Figure 4 illustre les différentes étapes du processus.

Détection de l'objet par apprentissage profond : La pièce brute est stockée sur une table et est observée à l'aide d'une caméra 1 positionnée au dessus de la scène. La détection de la pièce dans l'image est réalisée en temps réel en utilisant un algorithme d'apprentissage profond basé sur un réseau neuronal convolutif dénommé YOLO (You Only Look Once). YOLO permet d'identifier la présence de l'objet recherché dans l'image, de construire les contours de la zone rectangulaire englobant l'objet et de calculer une probabilité d'appartenance à la classe identifiée. Redmon et al. (2016) ont présenté la première version de YOLO. Au cours des dernières années, les chercheurs ont publié plusieurs versions ultérieures de YOLO : YOLO-V2 par Redmon

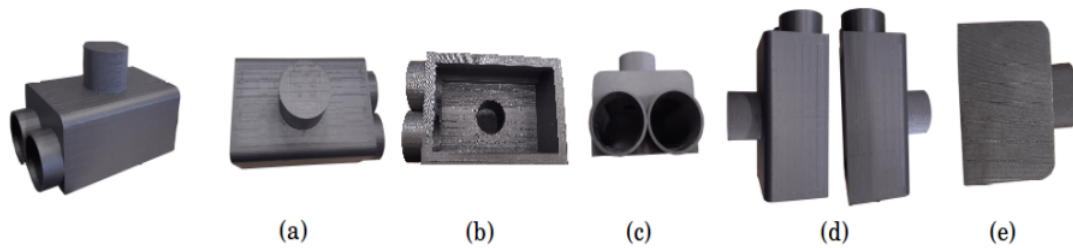


FIGURE 3 – La pièce industrielle conçue ; (a) face supérieure, (b) face inférieure, (c) face droite, (d) face avant / arrière (e) face gauche.

and Farhadi (2017), YOLO-V3 par Redmon and Farhadi (2018), YOLO-V4 par Bochkovski et al. (2020), YOLO-V5 par Zhan et al. (2022), YOLO-V6 par Li et al. (2022), YOLO-V7 par Wang et al. (2023) et récemment YOLO-V8. Il existe quelques versions révisées et limitées, telles que YOLO-LITE par Huang et al. (2018), PP-YOLOE par Xu et al. (2022) et plusieurs d'autres.

Dans nos travaux, la détection de la pièce est effectuée en employant YOLO-V3 et retestée sur YOLO-V4 après son lancement. Une base de données de l'objet a été préparée, contenant 996 images dans différentes conditions. Après avoir annoté toutes les images, l'entraînement du réseau de neurones YOLO-V3 a été réalisé par le processus dit "transfert learning" à partir de la classe "plastic part". Cet entraînement a utilisé 60% des images comme des données d'apprentissage et le reste des images pour le test et la validation. Au final, un rectangle 2D couvrant la pièce détectée dans la scène apparaît dans l'image. Il est sous-titré d'une étiquette (labellisation) comme illustré sur la Figure 4 (a).

Estimation de pose : La méthode utilisée pour l'estimation de la position 3D de l'objet par rapport au robot consiste à utiliser deux informations : d'une part, les données du nuage des points de la scène obtenues par la caméra de profondeur 1 à l'aide de la bibliothèque PCL (Point Cloud Library) ; d'autre part, les coordonnées du rectangle 2D englobant l'objet détecté dans l'image 2D à l'étape précédente. La combinaison de ces deux informations permet de générer un pavé 3D englobant la pièce.

Classification de la face de l'objet par apprentissage profond : Cette étape consiste à identifier la face présentée par la pièce sur l'image produite par la caméra 2 placée au poignet du bras robotique comme l'illustre la Figure 4 (b). La méthode de reconnaissance utilise YOLO mais le réseau de neurones a été re-entraîné avec une base de données de 4430 images annotées correspondant à cinq classes qui représentent les faces de la pièce industrielle. A l'issue de cette étape nous identifions la face de la pièce vue par la caméra 2 parmi les différentes faces possibles présentées sur la Figure 3.

Préhension adaptative de l'objet : La préhension adaptative est réalisée par une pince à 2 doigts parallèles permettant de saisir l'objet suivant la face présentée, soit par l'ouverture, soit par la fermeture des doigts. La décision du type de prise (par ouverture ou fermeture des doigts) dépend de la face de la pièce identifiée à l'issue de l'étape de classification. Par exemple, sur la Figure 4 (c), la face observée est la face inférieure et la prise s'effectuera par ouverture des doigts.

Planification des trajectoires du bras robotique : Le déplacement de notre bras robotique UR5 à 6 degrés de liberté est réalisé par Moveit! développé par Chitta (2016). Moveit! est une plate-forme de planification de trajectoire compatible avec ROS (Robot Operating System) qui fournit un système assez complet d'outils de génération et d'exécution de mouvement tenant

compte de l'environnement perçu pour éviter les collisions avec les obstacles. Cette plate-forme intègre un ensemble d'algorithmes de planification de mouvement, d'évitement d'obstacles et de collisions sous différentes bibliothèques telles que : OMPLSukan et al. (2012), SBPL Likhachev (2024), CHOMP Ratliff et al. (2009).

Pour résumer, après la détection et l'estimation de la pose de la pièce dans la scène, le bras robotique se déplace au-dessus de la table pour observer la face de l'objet vue par la caméra 2 et la classer afin de déterminer le mode de préhension adapté. Le générateur de trajectoire est alors utilisé pour amener le robot au niveau de la pose identifiée pour saisir l'objet. Enfin, le bras robotique se déplace pour déposer la pièce sur la plate-forme du robot mobile. Plus de détails pourront être trouvés dans Sebbata et al. (2020)

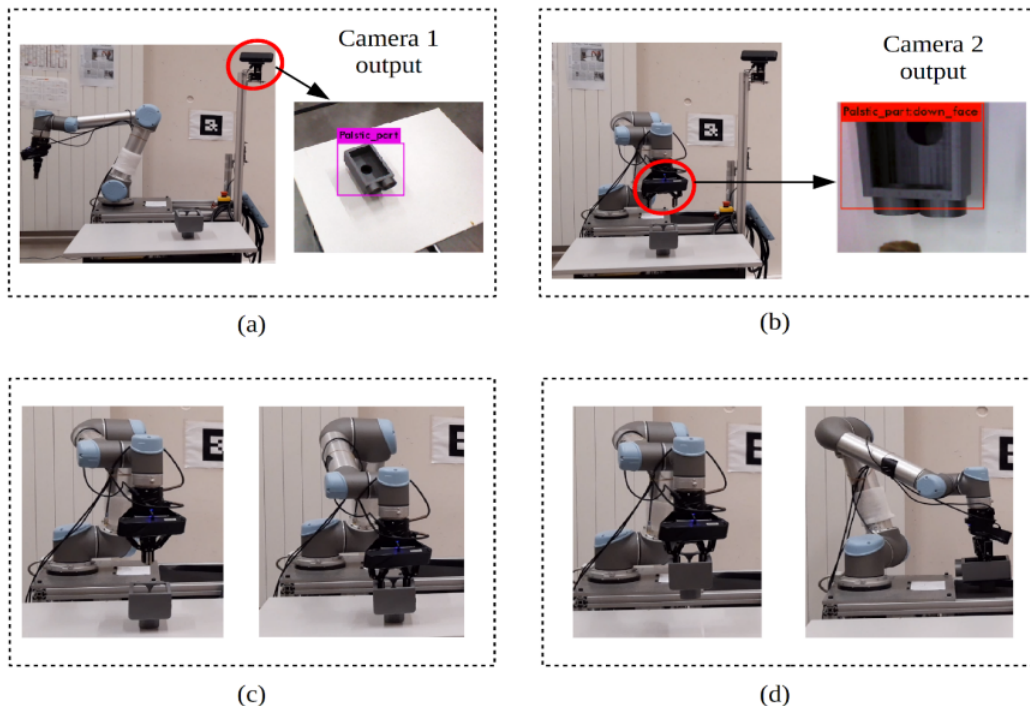


FIGURE 4 – Les étapes de picking de l'objet multifaces : (a) détection de l'objet et estimation de la pose, (b) classification de la face de l'objet et décision de mode de préhension, (c) saisie de l'objet, (d) planification et exécution du mouvement.

4. La navigation autonome

Le robot mobile peut évoluer dans un terrain ouvert (« outdoor navigation») ou fermé («indoor navigation»). Dans les deux cas, la navigation autonome nécessite d'une part de disposer de capteurs pour percevoir l'environnement (LiDARs, caméras, ...), d'autre part de construire une modélisation de l'environnement (cartes 2D ou 3D) pour pouvoir localiser le robot et planifier sa trajectoire. Du fait de l'évolution dynamique de certains obstacles, la trajectoire dite globale planifiée initialement est souvent modifiée lors de la navigation par l'utilisation d'un planificateur local.

Le middleware ROS propose un module de navigation autonome "Stack" détaillé par Zheng (2021) qui gère les trois fonctions de perception, localisation et planification comme illustré

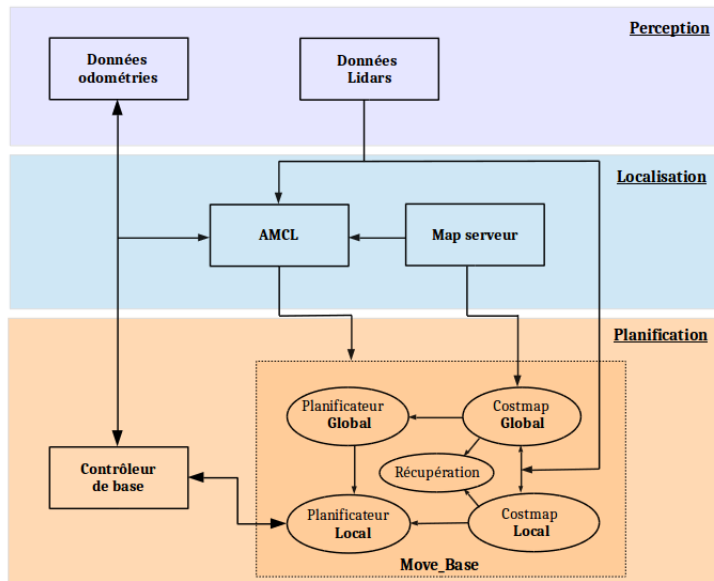


FIGURE 5 – L'architecture de navigation stack sous ROS.

sur la Figure 5. À partir des données de l'environnement perçues par le capteur LiDAR et les informations géographiques issues du serveur Map, le robot se localise dans l'espace de navigation en utilisant l'approche AMCL (Adaptive Monte Carlo Localization) initiée par Thrun et al. (2001) et complétée par Reis et al. (2021). La trajectoire se génère par Move_base pour atteindre la position désirée en tenant compte des données de localisation du robot. L'avantage de cette approche est qu'elle permet de localiser le robot mobile en temps réel.

Comme la carte de l'environnement n'est pas statique mais évolue en fonction des obstacles, l'approche SLAM (Simultaneous Localization And Mapping) décrite dans Taheri and Xia. (2021) s'avère particulièrement pertinente pour une utilisation robuste en milieu industriel. L'étape de cartographie est effectuée en continu ; les capteurs du robot balayent l'environnement pour mettre à jour la carte en temps réel. Pour la localisation, le robot utilise cette carte. Les algorithmes de SLAM présentent différentes variantes selon les types des capteurs (caméras RGB ou RGB-D ; capteurs LiDAR).

Pour la planification, le robot détermine le chemin optimal depuis son emplacement actuel jusqu'à la position souhaitée. La navigation Stack est basée sur Move_base présenté par Conner and Willis. (2017). Cette plate-forme contient plusieurs algorithmes de planification et prend en considération l'ensemble des entrées suivantes : les données de localisation, les informations de la carte de l'environnement et les signaux des LiDARs. Nous obtenons à la sortie deux trajectoires illustrées par la Figure 6) : une globale (en bleu) et une locale (en rose). La trajectoire globale est planifiée initialement du point de départ du robot jusqu'à la position finale et est construite à partir d'une carte initiale statique (global costmap) contenant toutes les informations statiques sur l'environnement de navigation. La trajectoire locale représente le chemin suivi par le robot qui s'ajuste en fonction des obstacles dynamiques détectés en temps réel autour du robot (local costmap). La carte locale de forme rectangulaire entourant le robot représentée par un rectangle blanc dans la Figure 6 ne prend en compte qu'un espace restreint autour du robot.

En résumé, la mise en œuvre de ces différents outils logiciels utilisant les informations des capteurs LiDAR installés sur la base mobile permet une navigation autonome et robuste dans

les allées des différents ateliers.

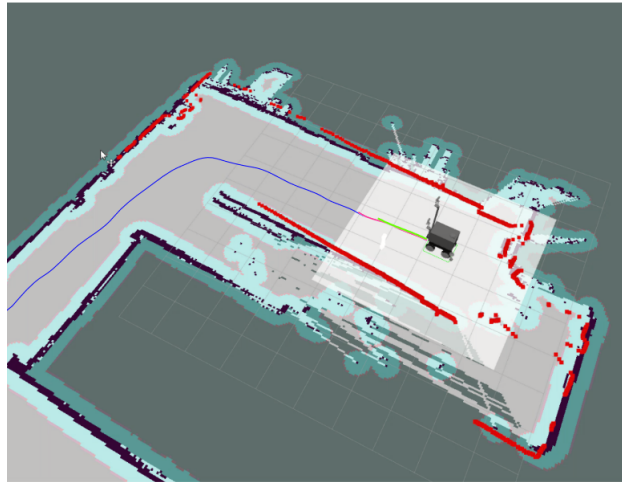


FIGURE 6 – Le robot localisé dans la carte de l'entrepôt réel PIL du laboratoire sur RVIZ avec les trajectoires, globale en bleu, locale en rose et la trajectoire suivie en temps réel en vert.

5. La dépose des pièces dans la machine-outil

La dépose d'une pièce dans les mors de l'étau d'une machine-outil est effectuée aujourd'hui de manière autonome par des bras manipulateurs fixés sur le bâti de la machine. Par exemple, les centres d'usinage japonais KITAMURA (2021) utilisent des bras robotiques pour le chargement et déchargement autonome. Ces opérations sont réalisées avec une grande fiabilité du fait que les performances en précision des robots industriels sont excellentes. Comme l'explique la norme ISO9283 (1998) illustrée par la Figure 7, il convient de définir deux indicateurs de performance liés à la précision : lorsque le robot est commandé pour atteindre une cible P_0 , et que la position réelle du robot par rapport à la cible est estimée par un dispositif métrologique, si ce mouvement est répété 30 fois, les différentes positions atteintes forment un nuage de points. La distance entre la moyenne des points atteints et la cible correspond à l'erreur d'exactitude, alors que le rayon de la sphère englobant le nuage est définie comme l'erreur de répétabilité. Or l'erreur de répétabilité de la plupart des bras manipulateurs industriels (Yaskawa, Fanuc, Kuka, ABB, Staubli,...) est de l'ordre de 0.05 mm donc permet de respecter le jeu fonctionnel entre la position nominale de dépose de la pièce et les mors de l'étau dont l'ordre de grandeur est du centimètre.

Par contre, lorsque le bras est embarqué sur un robot mobile, la précision du système robotique complet n'a pas été encore étudiée de manière approfondie et constitue encore un verrou scientifique freinant l'utilisation plus large des manipulateurs mobiles dans l'industrie 4.0. Aussi le travail de recherche concerne à modéliser cette précision en adaptant des outils statistiques et probabilistes développés lors de travaux menés par Brethé et al. (2006), Nzue et al. (2010), Brethé (2010).

Afin d'étudier les caractéristiques du nuage de points, nous avons mis en œuvre deux dispositifs métrologiques qui nous permettent d'étudier la précision de positionnement du robot mobile par rapport à la machine-outil et la précision de la dépose de la pièce par rapport à l'étau de la machine-outil.

Le dispositif métrologique qui permet d'étudier la précision de positionnement du robot

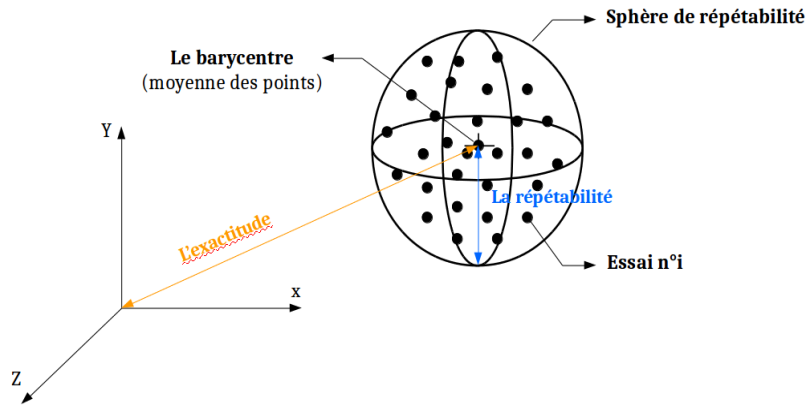


FIGURE 7 – La répétabilité de positionnement et l’exactitude suivant la norme ISO 9283

mobile (rectangle jaune) par rapport à la machine-outil (rectangle bleu) est illustré par la Figure 8. Il est constitué de deux équerres métrologiques de centre O_1 et O_2 . Les faisceaux des LiDARs placés en O_{RL} et O_{FL} viennent frapper les équerres et la position et l’orientation du robot mobile sont estimées à partir de l’exploitation statistique de ces nuages de points,

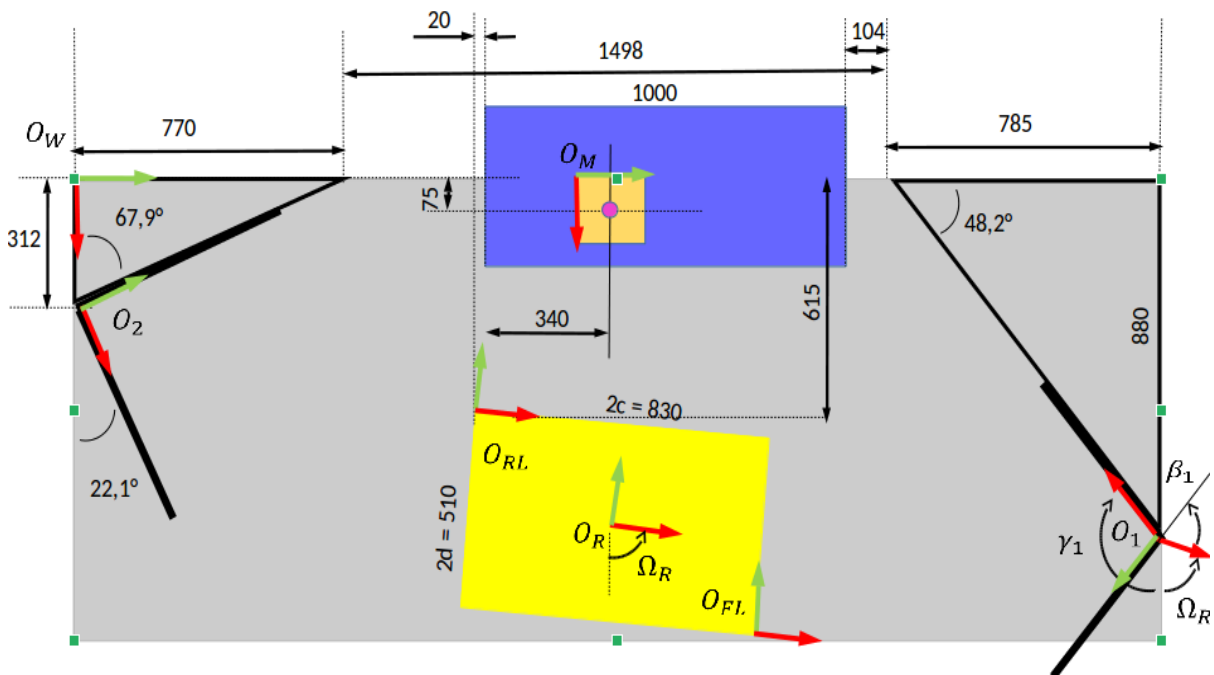


FIGURE 8 – Vue globale du dispositif métrologique.

Le dispositif métrologique qui permet d’estimer la précision de dépose de la pièce dans l’étai de la machine-outil utilise deux trièdres : un trièdre dit fixe tenu par les mors de l’étai de la machine-outil et équipé par trois micromètres numériques Mitutoyo d’une résolution de $1 \mu\text{m}$; un autre trièdre dit mobile constitué d’un cube en acier cémenté, est fixé au poignet du bras robotique comme illustré dans la Figure 9.

Les résultats expérimentaux obtenus sont particulièrement significatifs et nous donnons ici les principaux résultats. En premier lieu nous rappelons que le positionnement du robot mobile devant la machine-outil se fait en deux temps : tout d’abord, la phase finale de la navigation se base

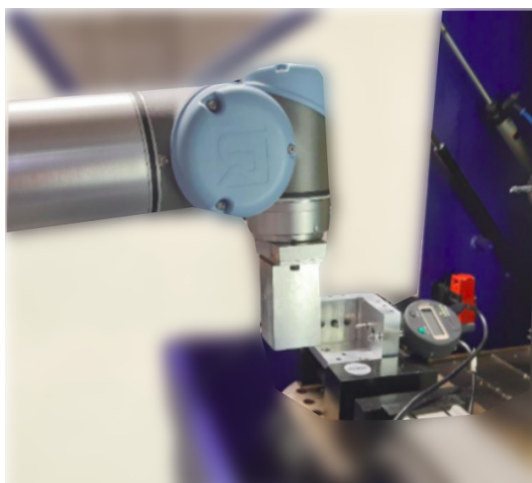


FIGURE 9 – Le dispositif de mesure pour les tests de précision du bras robotique

sur le module Stack de ROS et les signaux LiDAR ; ensuite nous avons utilisé un asservissement visuel sur un AR Tag (Augmented Reality Tag) fixé sur la machine-outil. Dans ces conditions, la répétabilité de pose du robot mobile est estimée à $\pm 4.15cm$ en position et à $\pm 3.49^\circ$ en orientation. Ceci signifie que le positionnement du robot mobile n'est pas assez répétable pour que le bras manipulateur puisse déposer la pièce dans l'état étant donné que le jeu fonctionnel est de l'ordre de grandeur du centimètre. On en déduit qu'il ne faut pas utiliser l'asservissement visuel pour obtenir une bonne répétabilité de pose de la base mobile et qu'il est préférable d'utiliser un asservissement basé sur les signaux LiDAR.

Par la suite, nous avons estimé la répétabilité du bras lorsque la base mobile est à l'arrêt devant la machine-outil. Dans ces conditions, la répétabilité de position est alors estimée à $\pm 7.8mm$ et la répétabilité d'orientation à $\pm 2.1^\circ$. Or sur la fiche technique du constructeur, la répétabilité du bras est estimée à $\pm 0.1mm$. Cette différence de performance s'explique logiquement par la compliance de la structure mécanique de la base mobile supportant le bras, qui entraîne une variabilité de la pose de la base du bras lors de la répétition des mouvements et détériore la répétabilité du bras robotique. Plus le support du bras robotique est compliant, plus le bras perd ses performances en précision. Avec une répétabilité en position et orientation ainsi estimée, nous avons alors appliqué la méthode des ellipsoïdes stochastiques pour obtenir les zones de confiance à 0.01 % et conclu qu'il était nécessaire d'exploiter d'autres stratégies détaillées dans la thèse de Sebbata (2021) pour pouvoir réussir la dépose avec la fiabilité nécessaire à une application industrielle.

6. Conclusions

Dans cet article, nous avons détaillé le cas d'usage développé au sein de la halle robotique du PIL (Pôle Ingénieur Logistique) de l'Université Le Havre Normandie dans le cadre du projet Interreg Trans-Manche CoRot. L'objectif était de concevoir et mettre en œuvre un robot manipulateur mobile pour alimenter de manière autonome une machine d'usinage en pièces brutes. A cet effet, nous avons utilisé un robot mobile Summit XL sur lequel a été embarqué un bras manipulateur cobotique UR5 muni d'une pince à deux doigts.

Les travaux de recherche ont mis en évidence plusieurs verrous scientifiques et se sont

attachés à les résoudre. La saisie autonome de la pièce nécessite une phase d'identification et de localisation qui a été mise en œuvre en utilisant un réseau neuronal convolutif YOLO pour de l'apprentissage profond. Cette technique issue de l'intelligence artificielle a montré ici tout son potentiel pour réussir une saisie adaptative. Pour la partie navigation, les contraintes concernaient la prise en compte de l'aspect dynamique dans la modélisation de l'environnement en temps réel. Nous avons pu mettre à profit un ensemble de méthodes innovantes et utiliser les modules adhoc associés au middleware ROS pour une navigation sûre parmi les pièces de la halle robotique. Enfin, pour la partie dépose de la pièce, nous avons proposé une modélisation inédite de la précision du système, donnant ainsi les éléments méthodologiques qui permettent de choisir les capteurs et le processus permettant de réussir cette tâche de dépose en respectant les contraintes fonctionnelles.

Le scénario complet de saisie de l'objet, de navigation et de dépose par le robot manipulateur mobile a été mis en œuvre en toute autonomie et sécurité, ce qui constitue en soi une avancée importante dans le domaine. Pour les perspectives à venir, nous prévoyons de disséminer ce savoir-faire et de travailler à la mise en œuvre expérimentale de l'ensemble des solutions proposées pour une dépose précise. Des algorithmes, pour ajuster la pose du robot par rapport à la machine et la pose du bras manipulateur, sont en cours d'études et développement afin d'assurer la dépose de la pièce dans la machine avec précision.

Références

- Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4 : Optimal speed and accuracy of object detection. *arXiv preprint arXiv :2004.10934*.
- Brethé, J.-F. (2010). Intrinsic repeatability : a new index for repeatability characterisation. In *IEEE International Conference on Robotics and Automation*, pages 3849–3854. IEEE. Anchorage, AK, USA.
- Brethé, J.-F., Vasselín, E., Lefebvre, D., and Dakyo, B. (2006). Modelling of repeatability phenomena using the stochastic ellipsoid approach. *Robotica*, page 477.
- Chitta, S. (2016). Moveit! : an introduction. *Robot Operating System (ROS) The Complete Reference*, 1 :3–27.
- Conner, D. C. and Willis., J. (2017). Flexible navigation : Finite state machine-based integrated navigation and control for ros enabled robots. *SoutheastCon*, 34(2) :1–8.
- Huang, R., Pedoeem, J., and Chen, C. (2018). Yolo-lite : A real-time object detection algorithm optimized for non-gpu computers. In *2018 IEEE International Conference on Big Data*, pages 2503–2510.
- ISO9283 (1998). Robots manipulateurs industriels.
- KITAMURA (2021). <https://www.kitamura-machinery.com/>. consulté en août 2021.
- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., et al. (2022). Yolov6 : A single-stage object detection framework for industrial applications. *arXiv preprint arXiv :2209.02976*.
- Likhachev, M. (2024). <http://sbpl.net/>. consulté en avril 2024.
- Nzue, R.-M. A., Brethé, J. F., Vasselín, E., and Lefebvre, D. (2010). Comparative analysis of the repeatability performance of a serial and parallel robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 63–68. IEEE. Taipei, Taiwan.
- Ratliff, N., Zucker, M., Bagnell, J. A., and Srinivasa, S. (2009). Chomp : Gradient optimization techniques for efficient motion planning. *IEEE international conference on robotics and automation*, pages 489–494.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once : Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Redmon, J. and Farhadi, A. (2017). Yolo9000 : better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.
- Redmon, J. and Farhadi, A. (2018). Yolov3 : An incremental improvement. *arXiv preprint arXiv :1804.02767*.

- Reis, W. P. N. d., Silva, G. J. d., Junior, O. M., and Vivaldini, K. C. T. (2021). An extended analysis on tuning the parameters of adaptive monte carlo localization ros package in an automated guided vehicle. *The International Journal of Advanced Manufacturing Technology*, 117 :1975–1995.
- Sebbata, W. (2021). *Etude et développement de tâches de manipulation mobile dans le contexte de l'industrie 4.0*. Thèse, Normandie Université.
- Sebbata, W., Brethé, J.-F., et al. (2020). An adaptive robotic grasping with a 2-finger gripper based on deep learning network. *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1 :620–627. Vienna, Austria.
- Sucan, I. A., Moll, M., and Kavraki, L. E. (2012). The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4) :72–82.
- Taheri, H. and Xia., Z. C. (2021). Slam; definition and evolution. *Engineering Applications of Artificial Intelligence*, 97 :104032.
- Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2) :99–141.
- Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). Yolov7 : Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475.
- Xu, S., Wang, X., Lv, W., Chang, Q., Cui, C., Deng, K., Wang, G., Dang, Q., Wei, S., Du, Y., et al. (2022). Pp-yoloe : An evolved version of yolo. *arXiv preprint arXiv :2203.16250*.
- Zhan, W., Sun, C., Wang, M., She, J., Zhang, Y., Zhang, Z., and Sun, Y. (2022). An improved yolov5 real-time detection method for small objects captured by uav. *Soft Computing*, 26 :361–373.
- Zheng, K. (2021). Ros navigation tuning guide. *Robot Operating System (ROS) The Complete Reference*, 6 :197–226.

Biographie des auteurs



Wafae SEBBATA est docteur en robotique. Sa thèse de doctorat, soutenue en octobre 2021, intitulée "Étude et développement de tâches de manipulation mobile dans le contexte de l'industrie 4.0", a été réalisée au sein du laboratoire GREAH (Groupe de Recherche en Electrotechnique et Automatique du Havre), Université Le Havre Normandie, sous la direction du professeur Jean-François Brethé. Elle a travaillé comme ingénieure d'études en robotique dans le cadre du projet CoRoT de 2017 à 2022. Aujourd'hui, elle exerce comme enseignante à l'école d'ingénieurs ECE-Paris "École Centrale d'Électronique".



Jean-François Brethé est Professeur des Universités en robotique à l'Université Le Havre Normandie, Institut Supérieur d'Etudes Logistiques (ISEL) et membre du GREAH (Groupe de Recherche en Electrotechnique et Automatique du Havre). Ses travaux de recherche portent initialement sur la modélisation, la conception et l'étude des performances en précision des robots manipulateurs industriels, puis se focalisent aujourd'hui sur les verrous scientifiques posés par l'introduction de cobots dans les entrepôts logistiques (préparation de commandes robotisée par robot manipulateur mobile, saisie de colis de taille variable par torse humanoïde).

COOPERATION UAV-UGV POUR L'INVENTAIRE D'ENTREPOTS LOGISTIQUES

François Guérin¹, Frédéric Guinand^{1,2}

¹Le Havre Normandy University GREAH - LITIS, 76600 Le Havre, France
{francois.guerin, frederic.guinand}@univ-lehavre.fr

²Faculty of Mathematics and Natural Sciences, Cardinal Stefan Wyszyński University,
01-815 Warsaw, Poland

RESUME

Cet article présente, à travers un projet d'innovation, l'utilisation de robots mobiles pour les opérations d'inventaire dans les entrepôts logistiques. Ces robots mobiles peuvent se déplacer de manière autonome à travers les zones de stockage pour recueillir des données (codes-barres, codes SKU, etc.) sur les produits entreposés et les transmettre directement au système de gestion d'entrepôt (WMS) via un réseau de communication. L'objectif de leur mise en œuvre est l'amélioration de la gestion logistique en apportant plus de fiabilité et en réduisant les coûts globaux des opérations d'inventaire. En effet, en utilisant des robots, ces opérations peuvent être réalisées à des heures qui n'interfèrent pas avec le fonctionnement habituel des installations logistiques. Cependant, la navigation autonome des drones peut poser des défis, car les entrepôts logistiques sont souvent des environnements dynamiques et encombrés, difficiles à équiper en instruments de guidage. Cet article propose une solution innovante pour la gestion des inventaires dans les entrepôts logistiques en utilisant des drones aériens (UAV) et terrestres (UGV). Le projet décrit a été mis en œuvre, en conditions réelles, dans un véritable entrepôt de la région havraise avec un drone aérien et un drone terrestre travaillant de manière coordonnée et coopérative. Les résultats obtenus montrent, d'une part que la solution fonctionne, et d'autre part qu'elle peut être déployée facilement sans qu'il ne soit nécessaire de pré-équiper l'entrepôt.

MOTS CLÉS: Robotique Mobile, Robot mobile aérien, Robot mobile terrestre, Navigation Autonome

ABSTRACT

This article describes, through an innovation project, the use of mobile robots for inventory operations in logistics warehouses. These mobile robots (drones) can move in an autonomous way through the storage facilities to acquire data (barcodes, SKU codes, ...) about the stored products and transmit them directly to the Warehouse Management System (WMS) using a communication network. Generally, they improve logistics management by providing reliability and by reducing the global cost of inventory operations since these can be carried out at hours that do not affect the normal operation of the logistics facilities. However, autonomous navigation of drones can be challenging because logistics facilities are known to be dynamic and cluttered environments potentially difficult to instrument. This article describes an innovating solution for inventory management in logistics warehouses using UAVs and UGVs.

KEYWORDS: Mobile Robotics, Unmanned Aerial Vehicle (UAV), Unmanned Ground Vehicle (UGV), Autonomous Navigation

1. Introduction

L'une des tâches les plus laborieuses dans le secteur de la logistique est la gestion des inventaires. Un inventaire consiste à compter les articles en stock en scannant les produits situés à des emplacements physiques spécifiques. Cette tâche n'apporte pas de valeur ajoutée mais doit être réalisée régulièrement (au moins une fois par an) pour vérifier que la valeur des stocks correspond bien à celle enregistrée dans le bilan comptable, ce qui est une obligation légale. L'inventaire est une tâche chronophage qui bloque les activités de l'entrepôt, mobilise d'importantes ressources humaines et matérielles, et qui peut s'avérer dangereuse lorsque les produits se trouvent à de grandes hauteurs (parfois plus de 10 mètres). Le coût élevé des opérations d'inventaire a conduit au développement de diverses solutions alternatives ((Airborne Data Collection, 2016), (Eyeseer, 2016), (The flying inventory assistant, 2016)) visant plusieurs objectifs : réduire les coûts, limiter les risques pour les personnels et limiter l'impact de l'inventaire sur les opérations courantes de l'entrepôt logistique. Parmi les solutions envisagées, certaines utilisent des véhicules aériens sans pilote (UAV) et/ou des véhicules terrestres sans pilote (UGV). Ces robots mobiles (drones) peuvent se déplacer dans les installations de stockage pour recueillir des données (codes-barres, codes SKU, etc.) sur les produits en entrepôt et les transmettre directement au système de gestion d'entrepôt (WMS) via le réseau de communication de l'entrepôt. L'utilisation des drones permet de répondre aux trois objectifs précédents. En premier lieu, ils ne nécessitent pas d'installation spécifique ou la location d'engins permettant d'accéder aux parties hautes des entrepôts. En second lieu, le nombre d'opérateurs nécessaires pour effectuer les tâches d'inventaire est nettement réduit et ces personnes ne sont pas exposées aux divers dangers inhérents aux opérations d'inventaire, celles-ci étant effectuées par les drones. En troisième lieu, en fonction de leur autonomie et des capteurs qu'ils embarquent, les drones sont en capacité d'effectuer ces opérations à des horaires qui n'interfèrent pas ou très peu avec le fonctionnement normal des installations logistiques. Enfin, ils permettent de générer des informations complémentaires à l'inventaire. Ils peuvent, par exemple, cartographier les étagères pour en déterminer les taux d'occupation. Une solution utilisant des robots a donc de sérieux atouts à faire valoir pour les tâches d'inventaire. Cependant la mise en œuvre d'une telle solution se heurte à de multiples défis. Par exemple, déterminer la position des UAV/UGV dans les installations de stockage peut poser de réels problèmes. En effet, l'environnement est souvent dynamique et encombré. Ainsi, les solutions actuelles impliquent généralement un opérateur humain pour contrôler le drone sur place ou pour définir un plan de vol via une interface homme-machine (HMI). Il est important de noter que la navigation autonome nécessite un système de positionnement en intérieur, tel que des balises (Ultra-Wide Band ou système de positionnement Wi-Fi) pour obtenir une estimation précise de la position du drone. Cette exigence technologique entraîne des coûts supplémentaires qui peuvent être significatifs. Néanmoins, bien que ces systèmes soient souvent coûteux à l'achat, ils peuvent être facilement accessibles en location sous forme de "drone-as-a-service". Cet article présente un projet innovant, le projet "DRIVELOG", développé à l'Institut Supérieur d'Études Logistiques (ISEL) de l'Université Le Havre Normandie, en partenariat avec la Région Normandie, Bolloré Logistics, Drone XTR et CIRCOE. Ce projet concerne les inventaires dans les entrepôts logistiques à l'aide de deux robots mobiles collaboratifs, un UAV et un UGV.

2. Inventaire dans les entrepôts logistiques

2.1 Description du problème

Effectuer une prise de vue aérienne d'une palette sur une étagère et détecter les informations associées (codes-barres, codes SKU) nécessite une estimation précise de la position du drone (UAV) par rapport à l'étagère à inventorier. Compte tenu du coût et des difficultés d'installation d'un système de positionnement en intérieur (balises Ultra-Wide Band (UWB), Positionnement Wi-Fi (WPS), etc.) dans un entrepôt logistique, la solution développée dans le cadre du projet DRIVELOG (Figure 1) consiste à combiner un véhicule aérien sans pilote (UAV) avec un véhicule terrestre sans pilote (UGV) (Harik, 2016).

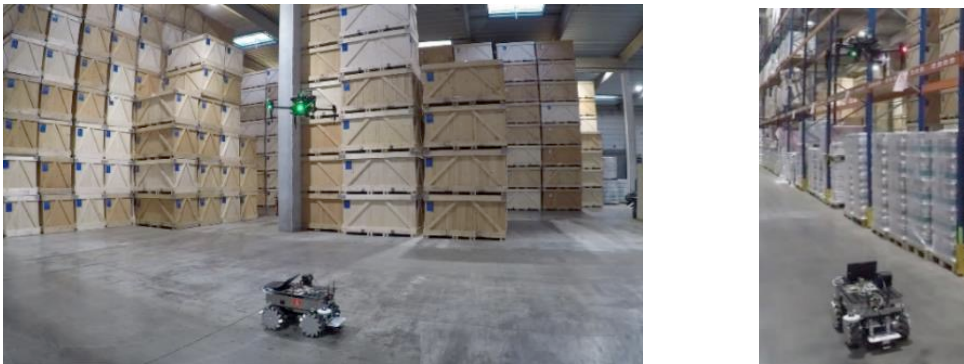


Figure 1 – Collaboration UAV/UGV pour les inventaires en entrepôts logistiques

Après s'être positionné correctement devant une colonne de palettes, l'UGV transmet au drone l'ordre de scanner les informations placées sur chaque palette (en montant ou descendant selon la colonne). Lors de ce mouvement vertical, l'objectif principal est de déterminer si chaque emplacement est vide ou (partiellement) rempli et d'identifier les codes-barres présents sur les palettes (Harik et al., 2016).

2.2 Navigation autonome du robot terrestre

La navigation autonome de l'UGV repose principalement sur ses capteurs extéroceptifs (odométrie, lidars, etc.). Ceux-ci permettent à l'UGV de se déplacer au milieu de l'allée tout en restant parallèle aux étagères (Figure 1). Pour détecter la position des colonnes de palettes à inventorier, un capteur de couleur industriel (Sensor Instruments SPECTRO-3-1000-COF) a été installé sur l'UGV. Ce capteur de couleur détecte (après une phase d'apprentissage) la couleur spécifique d'une étiquette placée sur la structure (poteaux) de l'étagère (Figure 2). Le capteur de couleur délivre une information binaire (codée sur cinq bits) par couleur apprise.

La reconnaissance se fait en quelques millisecondes, et l'information fournie par le capteur de couleur peut être utilisée directement par l'UGV sans traitement informatique complexe, ce qui permet de respecter le fonctionnement en temps réel du système. Une interface a été conçue pour transmettre les informations binaires à un module Arduino Mega 2560, sur lequel la bibliothèque "Rosserial" (<http://wiki.ros.org/rosserial>) a été installée. Les messages du capteur de couleur sont publiés (sous forme de "publisher") dans ROS (Robot Operating System (ROS, 2007)) et utilisés par l'UGV (en tant que "subscriber") durant la phase de navigation. Cette information permet à l'UGV de repérer le début (en rouge) et la fin (en vert) de l'allée, ainsi que

chaque poteau (en jaune) devant lesquels il marque une pause (Figures 2 et 3). Sur les étagères logistiques, une alvéole (délimitée par deux poteaux) contient quatre colonnes de palettes. Après avoir détecté un poteau rouge ou jaune et s'être arrêté devant, l'UGV avance par étapes régulières (environ 80 cm) pour atteindre les colonnes de palettes à inventorier. Pour ce faire, il utilise les données d'odométrie publiées ("publisher") dans ROS par l'unité de mesure inertielle (IMU) et les encodeurs incrémentaux installés sur chaque roue. Afin de rester parallèle aux étagères pendant son déplacement, l'UGV utilise ses deux lidars "Hokuyo UST-10LX" (Figure 4) installés à l'avant et à l'arrière. Ces lidars ont une résolution de 1080 incréments sur 270° et une portée maximale de 30 mètres.

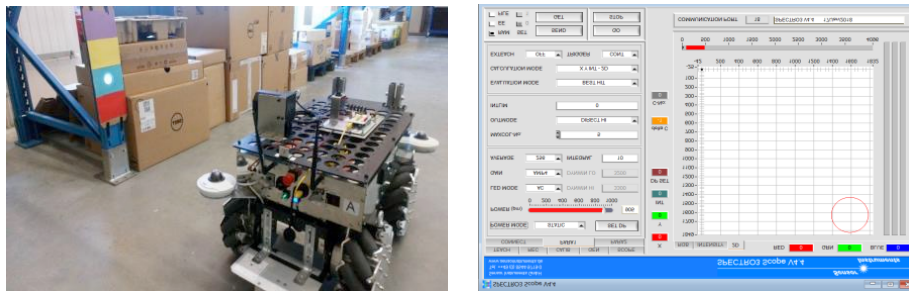


Figure 2 – Le capteur de couleur SPECTRO-3-1000-COF et son logiciel d'apprentissage

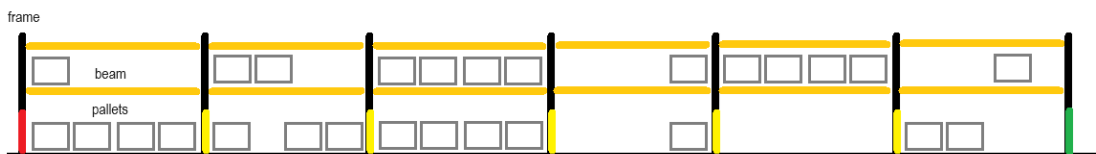


Figure 3 – Identification avec des repères de couleur

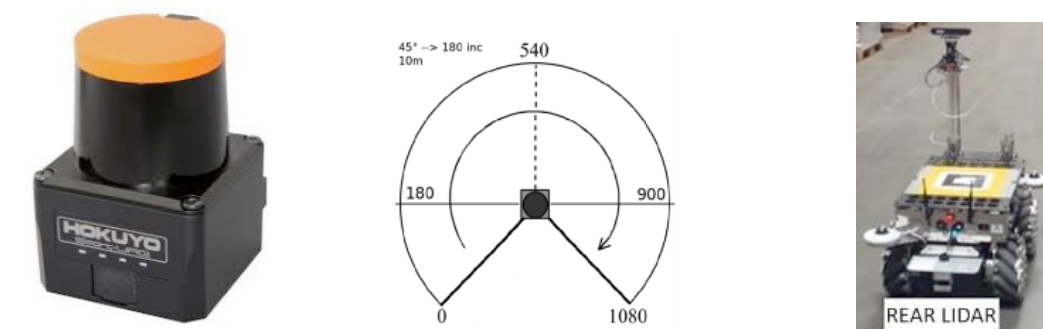


Figure 4 - Lidars « Hokuyo UST-10LX »

Un algorithme a été conçu pour reconstruire des murs virtuels à partir des impacts des lidars causés par les palettes placées au sol (Figure 5), en tenant compte des contraintes de stockage. Cet algorithme est basé sur la transformée de Hough (Hough, 1959), habituellement utilisée en traitement d'images pour extraire des lignes (Duda et al., 1972). Après avoir déterminé la matrice d'accumulation à partir des impacts du lidar, nous avons retenu les valeurs de cette matrice dépassant un certain seuil (Figure 6) et avons calculé pour chacune d'elles les paramètres de la ligne correspondante (Figure 7). Parmi toutes les lignes droites obtenues, seules sont retenues celles qui passent à droite et à gauche de

l'UGV et dont l'espacement correspond à la largeur de l'allée, avec une tolérance de 10 % (ces vérifications sont effectuées par rapport à l'axe latéral (y) de l'UGV). Pour chaque paire de lignes, nous calculons ensuite le déterminant (1) des vecteurs associés et le nombre d'impacts lidar (N_{IMP}) situés entre elles, puis nous calculons le critère J (2). La paire de lignes retenue pour la navigation est celle pour laquelle le critère J est minimal (Figure 8).

$$\det(\vec{V}_1, \vec{V}_2) = xd_1 \cdot yd_2 - xd_2 \cdot yd_1 \quad (1)$$

$$J = \frac{(1+N_{imp}) * (1 + |\det(\vec{V}_1, \vec{V}_2)|)}{Acc1 + Acc2} \quad (2)$$

Les résultats pour le cas sélectionné (Figure 8) sont présentés ci-dessous. Ils sont générés dès que les messages du lidar sont publiés ("publisher") dans ROS (toutes les 70 ms).

Nombre total de lignes : 166 - Impacts entre lignes : $N_{IMP} = 101$

Déterminant : $\det = 0.021362305$ – Critère : $J = 1.2113832$

Lignes sélectionnées : ligne 59 → $Acc1 = 42$ - ligne 108 → $Acc2 = 44$

Les erreurs de positionnement (3) de l'UGV (axe y) peuvent être estimées par rapport au rack situé à gauche, au rack situé à droite ou au centre de l'allée :

$$e_y = dy_l - d_{ref} \quad / \quad e_y = d_{ref} - dy_r \quad / \quad e_y = dy_r - dy_l \quad (3)$$

L'erreur d'orientation (4) de l'UGV (axe z) par rapport aux racks est donnée par :

$$e_\theta = \theta_z - \theta_{ref} \quad \text{with} \quad \theta_z = \tan^{-1} \frac{xd_1}{yd_1} \quad (4)$$

La distance et l'orientation de référence sont respectivement $d_{ref}(1,5 \text{ m})$ et $\theta_{ref}(0^\circ)$.

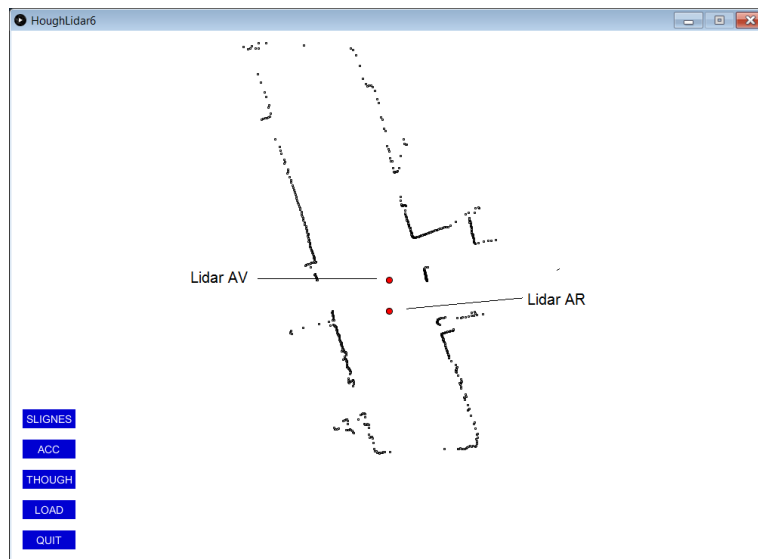


Figure 5 – Impacts des Lidars

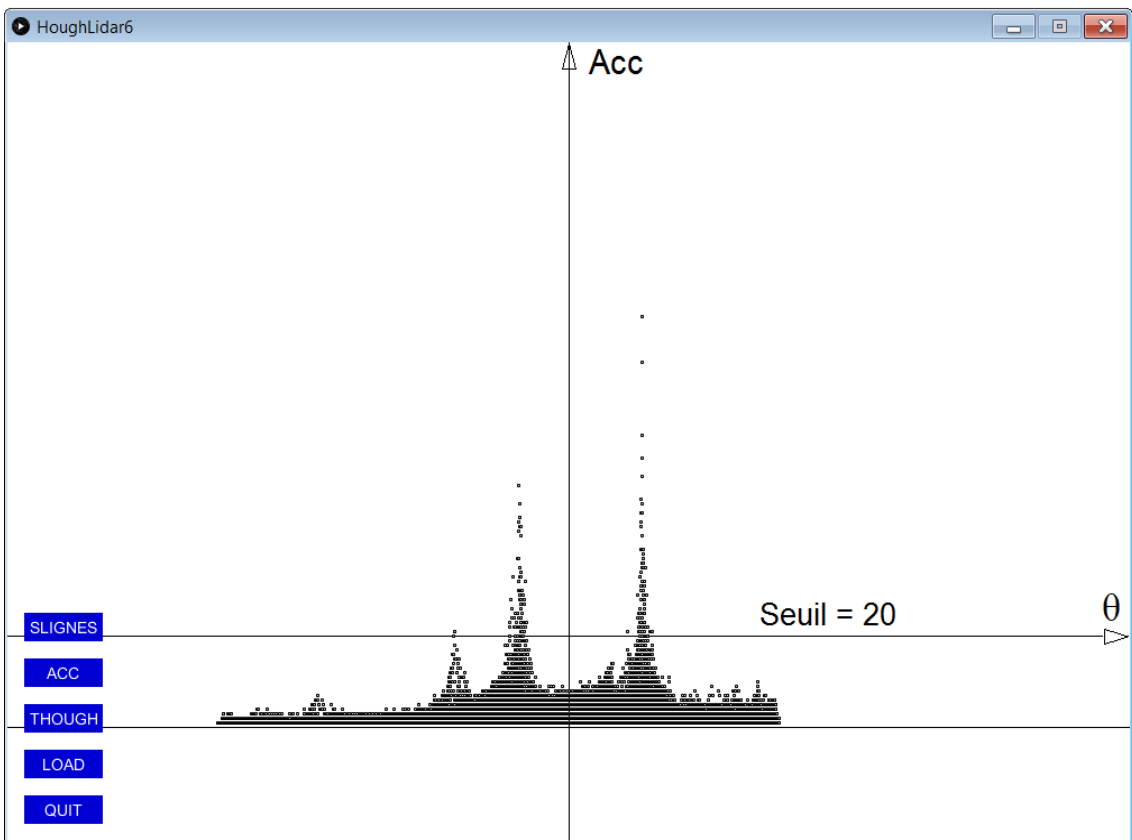
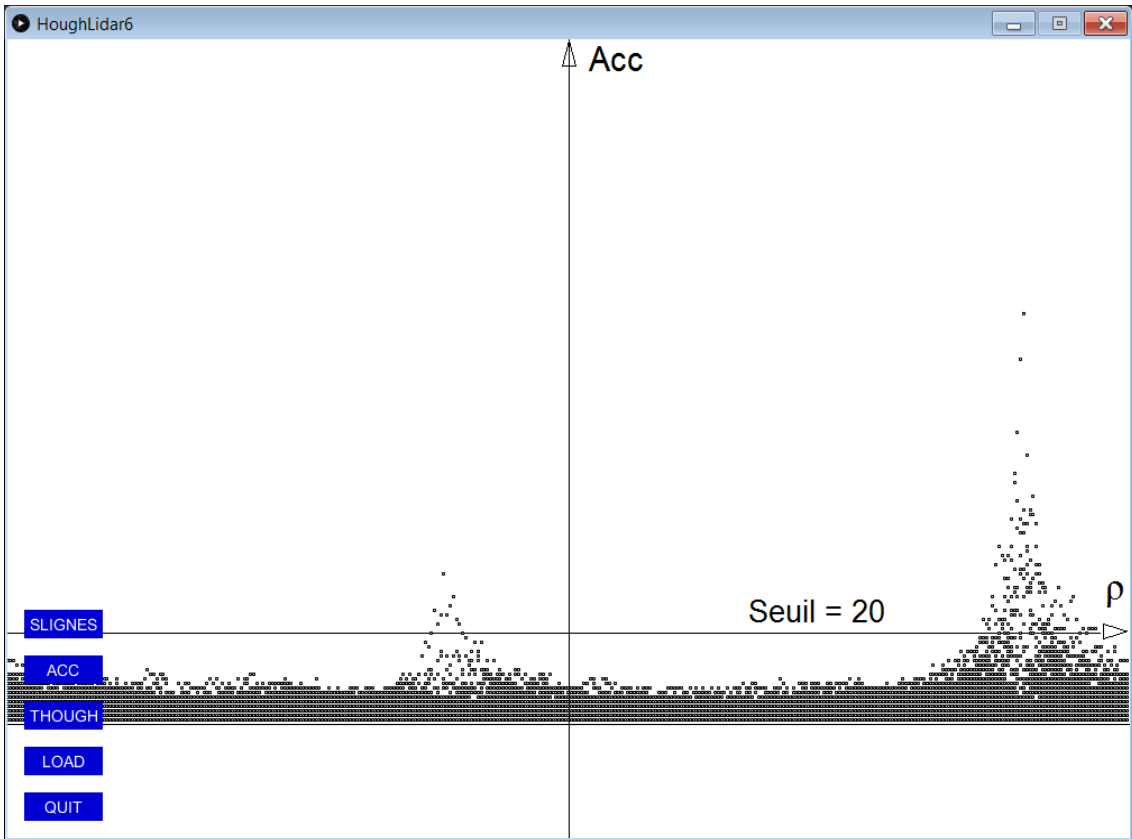


Figure 6 – Matrice d'accumulation / $\text{Acc}=f(\rho)$ - $\text{Acc}=f(\theta)$

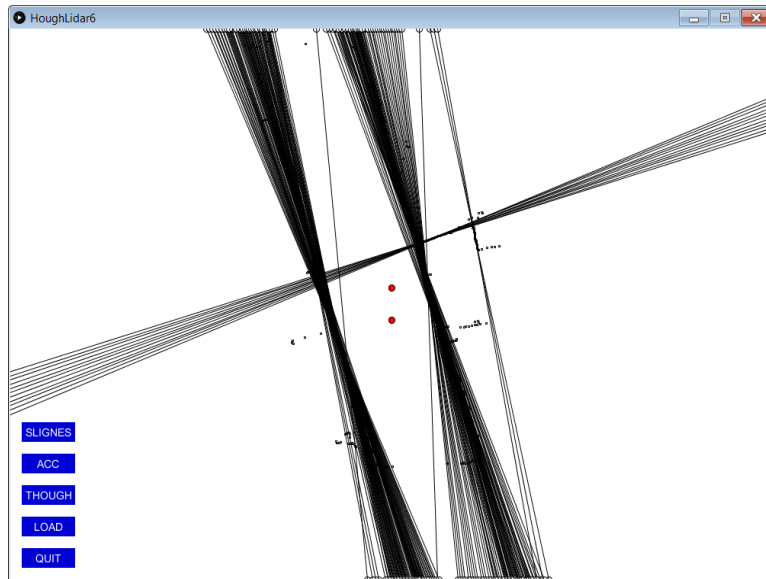


Figure 7 - Lignes obtenues avec la Transformée de Hough

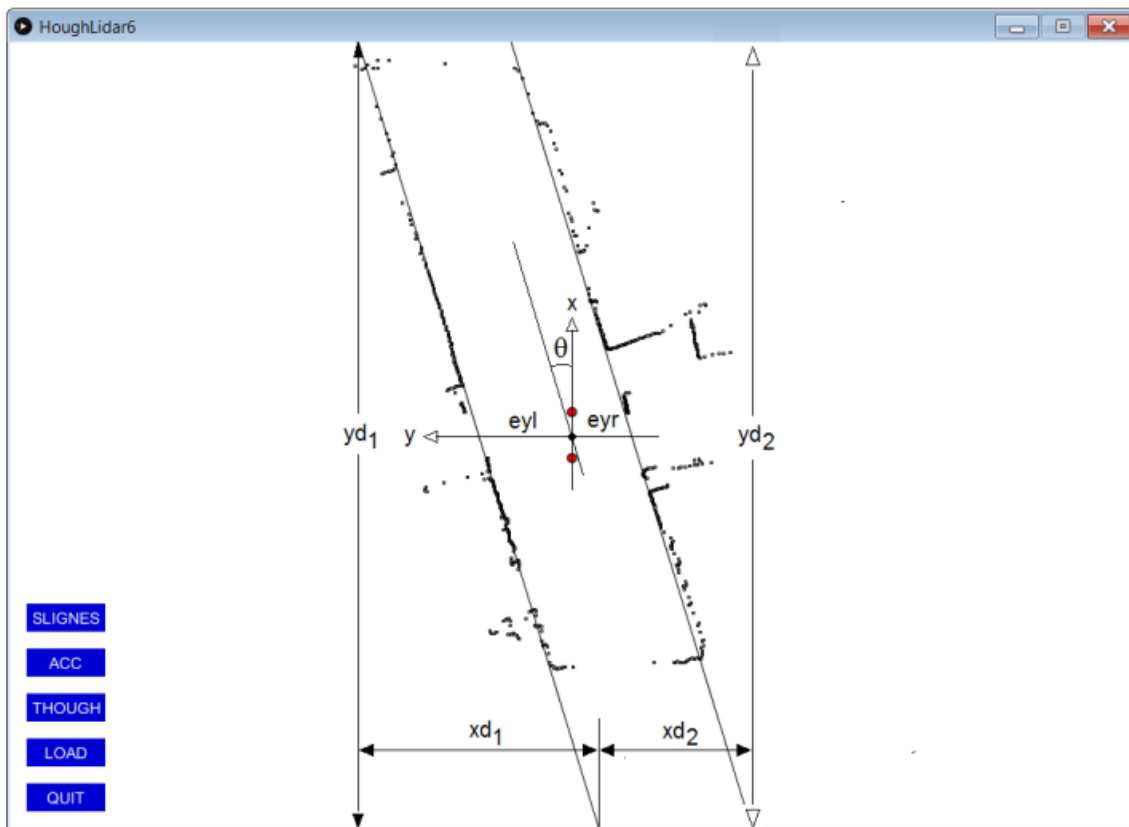


Figure 8 – Lignes sélectionnées et erreurs de positionnement

La loi de commande non linéaire (5) de l'UGV est donnée par :

$$\dot{e}_y = f(e_y, \cos(e_\theta), V_Y, D_Y) = -(\tau_v \cdot \dot{v} + v) \quad \dot{e}_\theta = f(e_\theta, R_z, \theta_z) = -(\tau_r \cdot \dot{r} + r) \quad (5)$$

V_Y et R_Z sont respectivement les vitesses latérales et angulaires maximales souhaitées. D_Y et θ_Z sont respectivement la distance et l'orientation à partir desquelles l'UGV atteint la vitesse maximale. Les constantes de temps τ_r et τ_v permettent des accélérations et décélérations progressives. La fonction f introduit une saturation et garantit un mouvement à vitesse constante et contrôlée lorsque l'erreur est importante. Elle est définie comme suit (6) :

$$f(eps, Vm, Dm) = \begin{cases} -Vm & \text{if } eps < -Dm \\ \frac{Vm}{Dm} \cdot eps & \text{if } -Dm \leq eps \leq +Dm \\ +Vm & \text{if } eps > +Dm \end{cases} \quad (6)$$

Remarque : une fonction « sigmoïde » peut également être utilisée.

2.3 Navigation autonome du robot aérien

Pendant les phases de montée et de descente (Figure 1), le drone (UAV) reste en position de "Hover" (vol stationnaire) par rapport à l'UGV en utilisant un asservissement visuel intégrant deux émetteurs infrarouges codés (mode pulsé) installés sur l'UAV, et une caméra infrarouge placée sur l'UGV. Cette technique permet de déterminer la position et l'orientation de l'UAV dans le repère de l'UGV, quelle que soit la distance (portée de 15 m, supérieure à la hauteur des étagères en entrepôt) et les conditions d'éclairage (souvent variables dans les entrepôts). L'asservissement visuel repose sur une "Sensor Bar" (comparable à celle de la console de jeu Wii) installée sous le drone (Figure 9) et une caméra infrarouge placée sur l'UGV. Cette dernière comprend deux groupes de 14 LEDs infrarouges (Vishay TSAL6100) émettant en mode pulsé (5 kHz). Une entrée "Enable" peut être contrôlée par une sortie logique de l'ordinateur embarqué (DJI Manifold) sur l'UAV (DJI Matrice 100) pour éteindre la "Sensor Bar" lorsqu'elle n'est pas nécessaire (économie de la batterie).

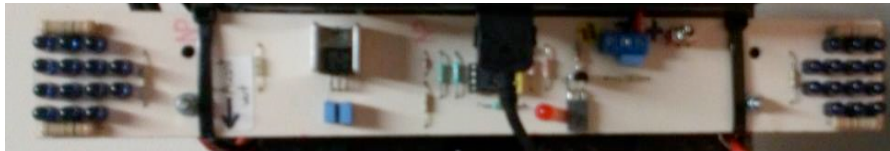


Figure 9 - « Sensor Bar » (PCB) installée sur l'UAV (DJI Matrice 100)

Les dimensions de la carte PCB correspondent à celles du drone (Matrice 100) pour permettre une fixation facile. Les LED infrarouges choisies fonctionnent en mode pulsé (5 kHz) pour améliorer la puissance et la portée. La caméra infrarouge est commercialisée par la société DFrobot (RB-Dfr-553) et fournit, via un bus de communication I2C, les coordonnées (x, y) de quatre cibles infrarouges dans le plan de l'image. Une phase de calibration a été réalisée (Figure 10) pour établir la correspondance entre les déplacements (en mètres) et les projections (en pixels) dans le plan de l'image (Guérin, 2004). Pour cela, la "Sensor Bar" a été placée à différentes distances de la caméra infrarouge (2m, 4,5m et 7m) puis déplacée horizontalement le long des axes x et y.

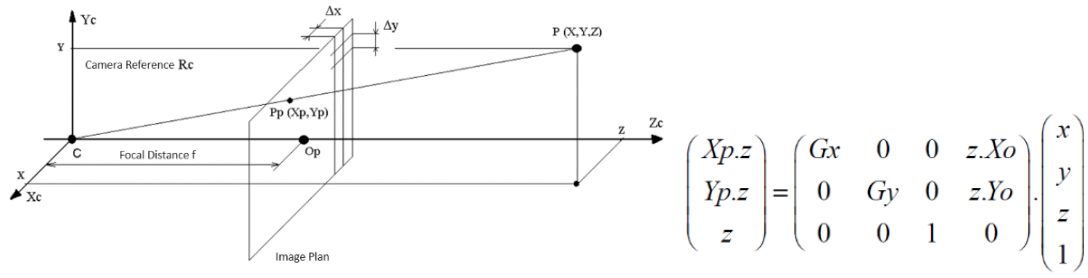


Figure 10 – Phase de calibration

Le maintien du drone en position de vol stationnaire est réalisé par asservissement visuel 2D ((Hutchinson et al., 1996), (Chaumette et al., 2016), (Guérin et al., 2013)). L'objectif est de garder le centre de la "Sensor Bar" (on suppose que son point de référence coïncide avec celui de l'UAV) au centre du plan de l'image de la caméra (on suppose que le point de référence de la caméra coïncide avec celui de l'UGV). Les erreurs de positionnement e_x , e_y et e_θ sont présentées ci-dessous (Figure 11).

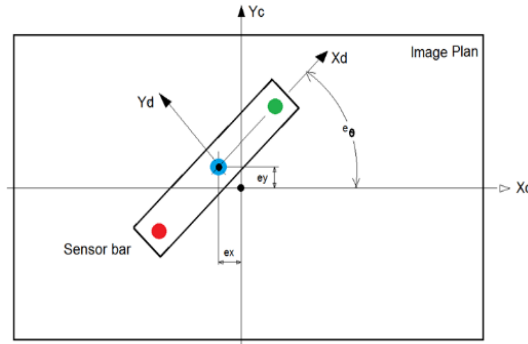


Figure 11 - Erreurs de positionnement dans le plan image

La loi de commande non linéaire (7) de l'UAV est la suivante :

$$\begin{aligned} \dot{e}_x &= f(e_x, V_X, D_X) = -(\tau_u \cdot \dot{u} + u) & \dot{e}_y &= f(e_y, V_Y, D_Y) = -(\tau_v \cdot \dot{v} + v) \\ \dot{e}_z &= f(e_z, V_Z, D_Z) = -(\tau_v \cdot \dot{v} + v) & \dot{e}_\theta &= f(e_\theta, R_Z, \theta_Z) = -(\tau_r \cdot \dot{r} + r) \end{aligned} \quad (7)$$

V_x , V_y , V_z et R_z sont respectivement les vitesses maximales linéaires, latérales, verticales et angulaires souhaitées. D_x , D_y , D_z et θ_z sont respectivement les distances et l'orientation à partir desquelles le drone atteint sa vitesse maximale. Les constantes de temps τ_r , τ_y et τ_x permettent des accélérations et décélérations progressives. La fonction f introduit une saturation et garantit un mouvement à vitesse constante et contrôlée lorsque l'erreur est importante. Elle est définie comme suit (8) :

$$f(eps, Vm, Dm) = \begin{cases} -Vm & \text{if } eps < -Dm \\ \frac{Vm}{Dm} \cdot eps & \text{if } -Dm \leq eps \leq +Dm \\ +Vm & \text{if } eps > +Dm \end{cases} \quad (8)$$

Remarque : une fonction « sigmoïde » peut également être utilisée.

Il est intéressant de noter qu'une estimation de l'altitude peut être obtenue en mesurant la distance entre les deux cibles (en rouge et vert dans la Figure 10). Cette estimation peut compléter les informations fournies par l'altimètre (peu fiables en intérieur) ou par le système "guidance" de DJI (DJI, 2006). Cette approximation est acceptable car le drone se déplace principalement de haut en bas pour ses opérations, en restant "à plat", c'est-à-dire sans inclinaison (tangage ou roulis). De plus, la valeur d'altitude est générée en continu, ce qui permet d'atténuer d'éventuelles perturbations. Par exemple, si l'altitude passe de 2,50 m à 4,30 m puis revient à 2,54 m, un traitement logiciel simple peut corriger cette incohérence. Lors de déplacements horizontaux, il peut y avoir un léger tangage et roulis, mais dans ces conditions, l'altitude n'est pas utilisée pour l'inventaire. En résumé, l'influence du roulis et du tangage est faible ($<10^\circ$) et compatible avec le diagramme de rayonnement des LED infrarouges (directivité).

2.4 Automatisation de l'opération d'inventaire

Pour réaliser l'opération d'inventaire, il est nécessaire d'établir un dialogue entre l'UGV et l'UAV afin de coordonner les mouvements, d'effectuer les ajustements finaux (par rapport aux étagères) et d'organiser les actions (décollage, atterrissage, position en vol stationnaire, etc.). Cette automatisation repose sur deux machines d'état (une pour chaque robot mobile) qui communiquent entre elles (Figures 12 et 13) via des messages ROS (ROS, 2007) comme indiqué dans la Figure 14.

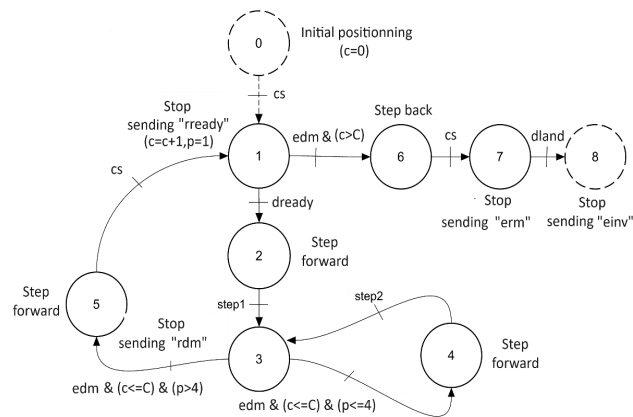


Figure 12 - Machine d'état de l'UGV

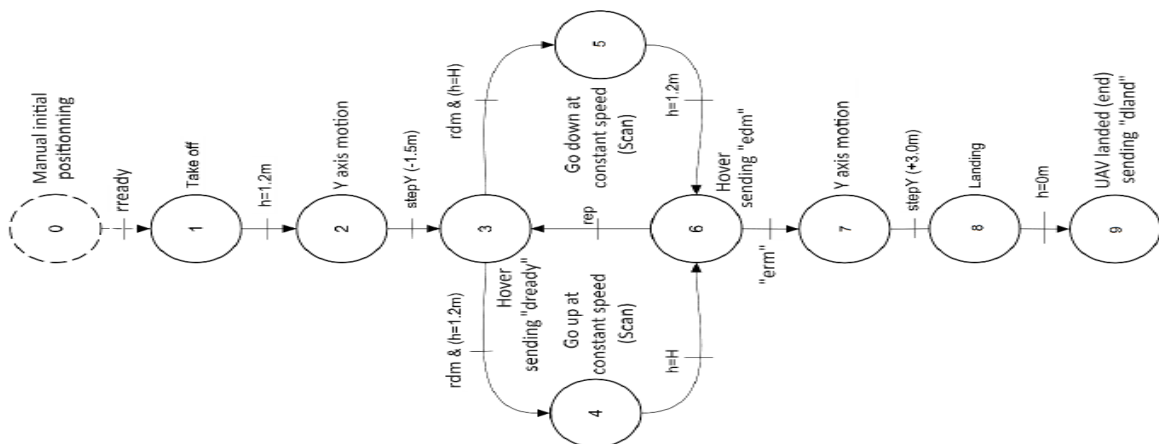


Figure 13 - Machine d'état de l'UAV

<p>Liste des messages ROS échangés :</p> <p>UGV→UAV:</p> <p>« rready »: UGV prêt</p> <p>« rdm »: Demande de mission UAV</p> <p>« erm »: Fin de mission UGV</p> <p>« einv »: Fin de l'inventaire</p> <p>UAV→UGV:</p> <p>« dready »: UAV prêt</p> <p>« dland »: UAV atterrissage</p> <p>« edm »: Fin de mission UAV</p>	<p>Transitions:</p> <p>UGV:</p> <p>« cs »: Capteur de couleur</p> <p>« step1 »: Poteau palette 1</p> <p>« step2 »: Entre palettes</p> <p>« p »: palettes – « c »: alvéole</p> <p>« C »: Nombre maximum d'alvéole</p> <p>UAV:</p> <p>« stepY »: Mouvement axe Y</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figure 14 – Liste des messages ROS échangés et transitions

L'approche « Système à Evénements Discrets » (SED) offre une grande flexibilité et permet de nombreux scénarios. La solution adoptée consiste à réaliser l'inventaire par colonnes en lisant les codes-barres lorsque le drone monte et/ou descend. Il est également possible de modifier les machines d'état pour réaliser l'inventaire en temps réel sans modifications logicielles majeures. Les deux options peuvent être programmées afin de laisser à l'opérateur le choix selon la configuration et les besoins de l'entrepôt. Le dialogue entre les deux robots mobiles peut facilement se faire via un réseau WIFI ou Bluetooth. La présence de l'UGV est également utile pour le rechargement éventuel des batteries de l'UAV (système de charge filaire (Elistair, 2014)) et pour le stockage des informations d'inventaire. De plus, contrairement à l'UAV, l'UGV est doté d'un système informatique embarqué puissant capable d'héberger un serveur web pour une supervision à distance via un navigateur classique (Firefox, par exemple). Les deux systèmes à événements discrets ont été modélisés par des réseaux de Petri et l'ensemble de l'opération d'inventaire a pu être simulé avec le logiciel Matlab/Simulink (Figures 15a et 15b).

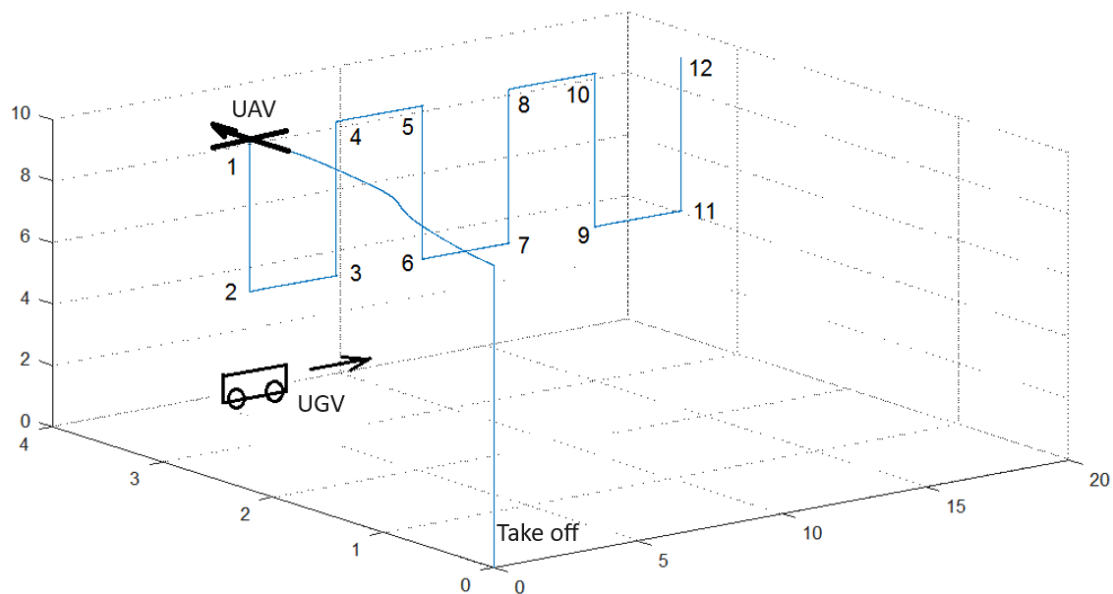


Figure 15a – Résultats de simulation

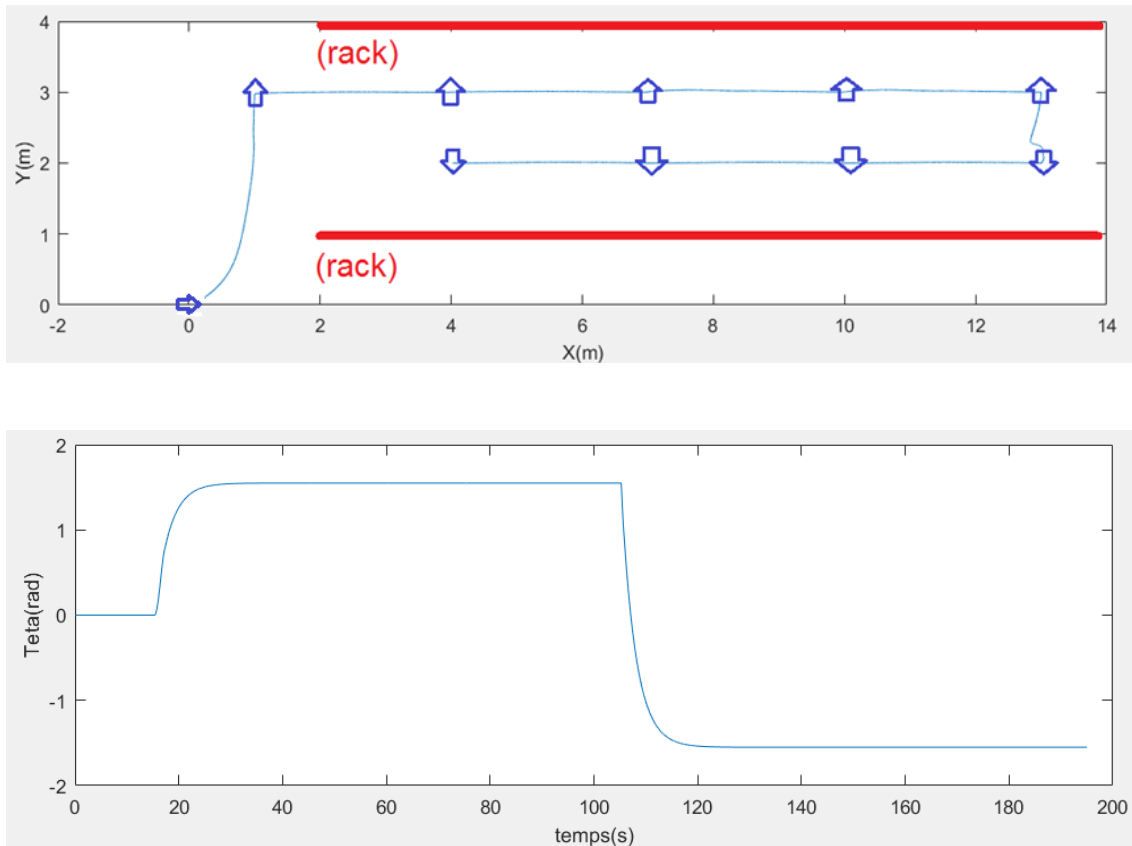


Figure 15b – Résultats de simulation

2.5 Acquisition des données d'inventaire

2.5.1 Fonctionnement général du système d'acquisition

Le système conçu pour l'acquisition des données se compose de deux smartphones équipés du système d'exploitation Android et d'une carte SIM (Figure 16). L'un des appareils joue le rôle de superviseur et l'autre celui de lecteur. L'opération d'acquisition des données peut être décrite par le diagramme suivant :

- a) Le superviseur reçoit un signal indiquant que l'opération d'inventaire doit commencer (ce signal est optionnel, l'opération peut aussi être lancée manuellement depuis le superviseur).
- b) Le superviseur se connecte au réseau WIFI.
- c) Le superviseur communique les informations de connexion au lecteur.
- d) Le lecteur se connecte au réseau WIFI.
- e) Le lecteur initie la communication directe avec le superviseur.
- f) Le lecteur écoute les robots mobiles pour obtenir des informations de positionnement.
- g) Le lecteur indique au superviseur qu'il est prêt.
- h) Le superviseur demande le démarrage de l'opération d'inventaire sur le lecteur.
- i) Le lecteur effectue le traitement des images et extrait les codes-barres (Figure 17).
- j) Le lecteur associe le(s) code(s)-barres extrait(s) avec un horodatage et des informations de positionnement (si disponibles).

- k) Le superviseur reçoit les données du lecteur et les traite.
- l) Les trois dernières opérations sont effectuées jusqu'à ce que le superviseur reçoive un signal de fin d'inventaire.
- m) Le superviseur informe le lecteur de la fin de l'inventaire.

2.5.2 Description des fonctionnalités du superviseur

- a) Piloter à distance et contrôler le lecteur.
- b) Connexion au réseau de l'entreprise ou, à défaut, à un autre point d'accès WIFI.
- c) Communication des informations de connexion au lecteur (via SMS) et des informations de connexion du superviseur (adresse IP, port de communication pour la lecture).
- d) Envoi de l'ordre de démarrer la lecture des codes-barres.
- e) Réception des informations du lecteur.
- f) Connexion au WMS (Système de Gestion d'Entrepôt) de l'entreprise.
- g) Reconstruction virtuelle de l'entrepôt (Figure 18) à partir :
 - des informations des codes-barres augmentées d'un horodatage et d'informations de positionnement.
 - des informations du WMS.
 - de la gestion des communications par SMS avec un ensemble de téléphones autorisés ou via l'utilisation d'un code.

2.5.3 Description des fonctionnalités du lecteur

- a) Gestion de la communication par SMS (uniquement avec le superviseur).
- b) Acquisition et analyse des images pour extraire les codes-barres.
- c) Obtention des informations de positionnement de l'UGV.
- d) Communication des informations (codes-barres couplés avec un horodatage et des informations de positionnement) au superviseur via le réseau WIFI.

2.5.4 Contraintes logicielles

L'application Android se présente sous la forme d'un fichier APK. Sa version actuelle est fonctionnelle à partir de l'API 23 (Android Marshmallow) et pour les versions suivantes, représentant en mai 2020 environ 85 % des téléphones actuellement utilisés. L'application sera donc compatible avec toutes les nouvelles versions du système Android à venir. En termes de matériel, les téléphones utilisés pour les tests sont des Samsung Galaxy S8 commerciaux. Les principales caractéristiques importantes pour le bon fonctionnement de l'application sont :

- a) Qualité et performance du processeur. Nos appareils sont équipés d'un processeur Exynos 8895 à 8 cœurs. Cette caractéristique permet l'exécution simultanée de plusieurs méthodes indépendantes et augmente donc considérablement la performance de l'application.
- b) Quantité de mémoire vive (RAM). Nos appareils disposent de 4 Go de RAM. Cette mémoire permet à l'application de résider entièrement en RAM et de garder dans cette même mémoire les données nécessaires au traitement des images.
- c) Qualité et performance du capteur photo. Le capteur utilisé a une définition de 12 Mpx. Un bon capteur photo permet d'augmenter le ratio distance de prise de vue / taille du

code-barres et offre plus de flexibilité dans l'utilisation et la portée des codes-barres pouvant être extraits.

- d) Capacité de la batterie. Nos appareils sont équipés d'une batterie de 3000 mAh, permettant une journée complète (12 heures) d'utilisation pour l'inventaire sans rechargement.

Ces appareils peuvent être remplacés par d'autres ayant des caractéristiques équivalentes ou supérieures en termes de performance. L'application est entièrement indépendante de la marque des appareils.

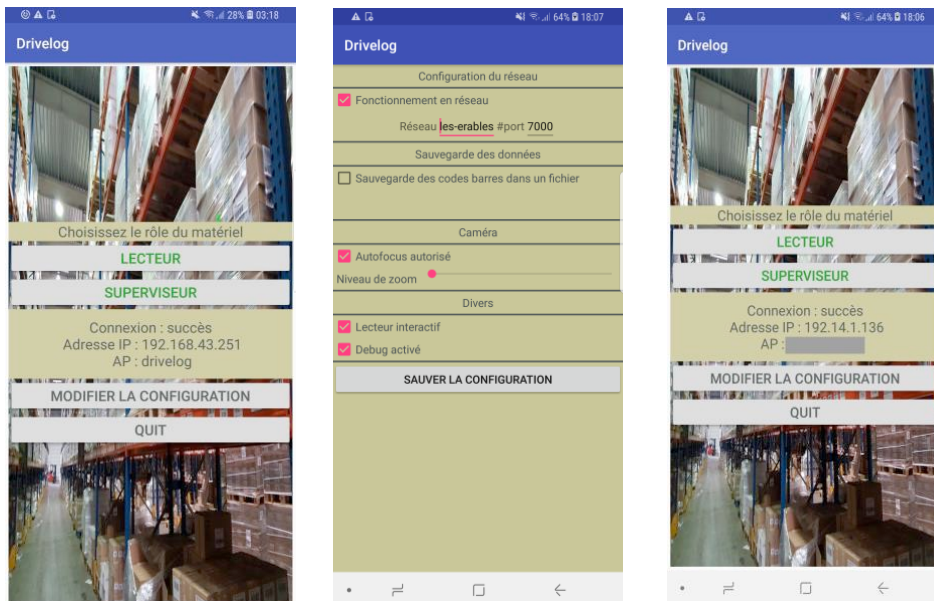


Figure 16 – Application Android pour l'acquisition de données

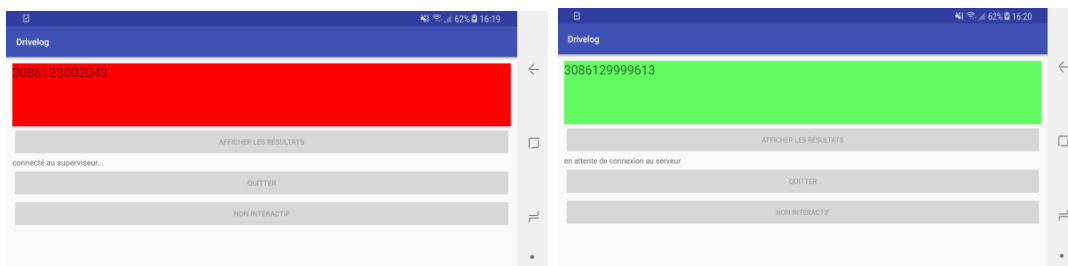


Figure 17 – Acquisition de données



Figure 18 – Reconstruction virtuelle de l'entrepôt

3. Conclusion

Cet article présente un projet innovant pour la gestion des inventaires dans les entrepôts logistiques. L'opération d'inventaire soulève de nombreux problèmes, à la fois en termes de coût, de sécurité des personnes et de perturbation du fonctionnement de l'entrepôt. La solution présentée répond à l'ensemble de ces questions en mettant en œuvre des robots mobiles aériens (UAV) et terrestres (UGV) sans pilote. Des expérimentations ont été menées en conditions réelles sur un site de la région havraise et ont montré la faisabilité du prototype que nous avons développé. Nous avons montré en particulier que le système proposé résout le problème de la géolocalisation intérieure en combinant la navigation autonome et les Systèmes à Événements Discrets (DES) pour coordonner les actions entre l'UAV et l'UGV. Le système, constitué des deux drones, peut ainsi naviguer de manière autonome à travers l'entrepôt. Par ailleurs, couplé à un dispositif composé de deux smartphones connectés, il permet également de collecter des données sur les produits stockés (telles que des codes-barres et des codes SKU), et de transmettre ces informations directement au système de gestion d'entrepôt (WMS) via un réseau de communication. Un aspect important de cette solution est la possibilité de l'étendre en utilisant plusieurs UAV et UGV, tous gérés par un superviseur global qui attribue les tâches. Cette configuration, qui repose sur l'utilisation de balises Ultra Wide Band (UWB), vise à optimiser les opérations d'inventaire dans les entrepôts. Elle a déjà été étudiée par les auteurs (Guinand et al., 2021) dans des recherches précédentes et consiste à mettre en œuvre un superviseur global pour l'allocation de tâches.

Références

- Airborne Data Collection (2016). <http://dronescan.co/>. Consulté 2018.
- Chaumette, F., Hutchinson, S., and Corke, P. (2016). Visual Servoing. In Handbook of Robotics, 2nd edition, O. Khatib B. Siciliano (ed.), pp. 841-866, Springer, 2016.
- DJI (2006). <https://www.dji.com/fr/guidance>. Consulté 2016.
- Duda, R. O., and Hart, P. E. (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Comm. ACM*, vol. 15, p. 11-15.
- Elistair (2014). <https://elistair.com/>. Consulté 2017.
- Eyeseer (2016). The drone allowing to automate inventory in warehouses. <http://www.hardis-group.com>. Consulté 2018.
- Guérin, F. (2004). Commande conjuguée d'un robot mobile, modélisation dynamique et vision artificielle. Thèse de doctorat de l'université du Havre. EUE – ISBN 978-613-1-54896-3, 2004.
- Guérin, F., Fabri, S. G., and Bugeja, M. K. (2013). Double exponential smoothing for predictive vision based target tracking of a wheeled mobile robot. In *52nd IEEE Conference on Decision and Control*, Firenze, Italy, 2013, pp. 3535-3540.
- Guinand, F., Guérin, F., and Petitprez, E. (2021). UAV-UGV Multi-Robot System for Warehouse Inventory. In *13th International Conference on Computational Collective Intelligence*, Rhodes, Greece, 29 sept - 1 oct 2021.

Harik, E. H. C. (2016). Commande et coordination d'une flotte de robots mobiles terrestres et aériens pour la surveillance et l'alerte. Thèse de doctorat, Université du Havre.

Harik, E. H. C., Guérin, F., Guinand, F., Brethé, J. F., and Pelvillain, H. (2016). Towards An Autonomous Warehouse Inventory Scheme. In IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2016), Athens, Greece, December 6-9, 2016.

Hough, P.V.C. (1959). Machine Analysis of Bubble Chamber Pictures. In Proc. Int. Conf. High Energy Accelerators and Instrumentation, p. 554-558.

Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. In IEEE. Trans. on Robotics and Automation, 12(5): p. 651-670, October 1996.

ROS (2007). <https://www.ros.org/> . Consulté 2015.

The flying inventory assistant (2016). <http://www.fraunhofer.de>. Consulté 2018.

Biographie des auteurs



Dr François Guérin a obtenu son Master (2000), son Doctorat (2004) et son Habilitation à Diriger des Recherches (2015) en Automatique et Robotique à l'Université Le Havre Normandie. Il exerce en tant que Maître de Conférences HDR à l'Université Le Havre Normandie, au sein de l'Institut Supérieur d'Etudes Logistiques (ISEL) et du Groupe de Recherche en Electrotechnique et Automatique (GREAH). Ses enseignements concernent l'Electronique (IoT), l'Automatique et la Robotique et ses recherches portent sur la commande avancée, appliquée à la robotique mobile terrestre et aérienne. En 2021, il a créé au Havre (76) le Centre d'Innovation Drones de Normandie (CIDN) et a rejoint en 2022, le 61ème Régiment d'Artillerie de l'Armée de Terre basé à Chaumont (52) en tant qu'officier réserviste.



Pr Frédéric Guinand a obtenu son master et son doctorat en informatique à l'Institut National Polytechnique de Grenoble (INPG, aujourd'hui Grenoble Institut de Technologie) en France en 1991 et 1995, respectivement. Après ses études universitaires, le Pr Guinand s'est dirigé vers l'industrie numérique, acquérant une précieuse expérience en tant qu'ingénieur dans une start-up en 1996. Par la suite, il a obtenu une bourse postdoctorale INRIA, menant des recherches à l'École Polytechnique Fédérale de Lausanne (EPFL - Suisse). Fin 1997, il a rejoint l'Université Le Havre Normandie en tant que professeur assistant et a été nommé Professeur des Universités en 2005. Ses intérêts de recherche couvrent un large éventail de sujets, notamment les graphes dynamiques, les réseaux complexes, l'informatique distribuée, mobile et inspirée de la nature, et la robotique collective. Il est également Professeur invité à la Faculté de mathématiques et de sciences naturelles de l'Université Cardinal Stefan Wyszyński de Varsovie en Pologne.

MODÉLISATION ET COMMANDE D'UN ROBOT À DEUX BRAS POUR LA SAISIE DE BOÎTES EN CARTON DANS LES ENTREPÔTS LOGISTIQUES

Khairidine BENALI¹, Jean-François BRETHER², François GUERIN²

¹École d'Informatique, Université de Nottingham
Wollaton Rd, Lenton, Nottingham NG8 1BB

² GREAH, Université Le Havre Normandie
75, Rue Bellot, 76058 Le Havre Cedex (France)

RESUME

Dans les entrepôts logistiques, la manipulation robotisée des boîtes en cartons, composants basiques du conditionnement en palette, est effectuée presque exclusivement par un bras robotique anthropomorphe muni d'un préhenseur de type pince ou ventouse. Or les opérateurs humains qui accomplissent ces tâches utilisent généralement leurs deux bras et sont beaucoup plus polyvalents pour saisir la large variété de colis présents. Dans ce cadre, une méthode innovante de saisie robotisée de colis est proposée à partir d'un torse humanoïde (CROM) équipé de deux bras cobotiques (UR10). La solution vise à reproduire la dextérité de manipulation d'un opérateur humain. Les phases de conception et modélisation sont détaillées ainsi que la description d'une loi de commande innovante optimisée de type position-Force/Position.

MOTS CLÉS: Logistique, Manutention, Torse humanoïde, Cobotique, Commande en Force/Position.

ABSTRACT

In logistics warehouses, the robotic handling of cardboard boxes, the basic components of pallet packaging, is carried out almost exclusively by one anthropomorphic robotic arm fitted with gripper or suction cups. However, the human operators who perform these tasks generally use their two arms and are much more versatile when it comes to gripping the wide variety of packages present. In this context, an innovative method of robotised parcel gripping is proposed based on a humanoid torso (CROM) equipped with two cobotic arms (UR10). The solution aims to reproduce the handling dexterity of a human operator. The design and modelling phases are detailed, as is the description of an innovative optimised position-Force/Position control law.

KEYWORDS: Logistics, Handling, Humanoid torso, Cobotics, Force/Position control.

1. Introduction

Aujourd'hui les opérations de manutention dans les entrepôts logistiques sont encore majoritairement effectuées par des opérateurs humains. Or la pénurie de ressources humaines et la pénibilité de ces tâches TMS (Troubles Musculo-Squelettiques) amènent à envisager la robotisation de ces opérations dans le cadre plus général de l'évolution de la logistique vers l'industrie 4.0. Cependant du fait de la grande variété des textures, des géométries et des masses des objets stockés en entrepôt, leur manutention par des robots est beaucoup plus complexe que sur une chaîne d'assemblage traditionnelle où les préhenseurs sont conçus spécifiquement pour un objet. Si on restreint l'analyse au cas de la dépalettisation de cartons de taille variable, la solution classique du bras anthropomorphe est limitée par les contraintes du préhenseur (largeur d'ouverture de la pince, surface couverte par les ventouses, etc). Il s'avère en pratique que les opérateurs humains n'ont aucune difficulté à saisir des cartons de taille variable du fait qu'ils utilisent leur deux bras. De ce fait, l'utilisation d'un robot humanoïde semble une solution prometteuse. Il faut cependant au préalable résoudre un certain nombre de verrous scientifiques : entre autres, la reconnaissance des géométries et la localisation des cartons grâce à des techniques d'analyse d'images ou de deep learning ou la mise en œuvre d'une commande avancée permettant au robot de fonctionner dans des environnements dynamiques et imprévisibles.

Concernant la commande avancée, des recherches antérieures, telles que Park and Lee (2016), Erhart et al. (2013) et Erhart and Hirche (2015), ont exploré la manipulation à deux bras, en se concentrant sur la commande en espace de tâches en mode coopératif étendu et en traitant des erreurs cinématiques dans les chaînes cinématiques fermées. Dans Pliego-Jimenez and Arteaga-Perez (2017) sont décrites des lois de commande hybrides adaptatives Force/Position, et dans Lin (2018), des lois de commande contraintes imposent le contact et le frottement avec un comportement d'impédance cartésienne. Des stratégies de planification de mouvement coordonnées pour des tâches complexes qui ne peuvent pas être exécutées par un seul bras manipulateur ont été détaillées dans Salehian et al. (2016). Nurkan Yagiz and Arslan (2010), Yim et al. (1999), Liu (2000) ont étudié une commande par mode glissant pour les systèmes à deux bras, nécessitant la prise en compte à la fois des paramètres dynamiques des robots et des objets qu'ils manipulent. Les contributions d'Almeida and Lopes (1999) sur la commande Force-Impédance ont aussi montré tout leur intérêt.

Nos travaux se focalisent essentiellement sur la synthèse d'une commande optimale pour saisir un carton par les faces latérales en maintenant une force minimale pour éviter le glissement et la chute de l'objet et sans dépasser une force trop forte risquant de détériorer l'objet. La contribution majeure de cet article est la proposition d'un schéma de commande innovant qui combine des méthodes maître/esclave et hybrides force-position. Tout d'abord une présentation et une modélisation synthétique du dispositif utilisé est effectuée. Elle est suivie d'une description détaillée de la commande innovante et d'une comparaison aux approches classiques.

2. Présentation et modélisation du manipulateur à deux bras

2.1 Description du système

Notre manipulateur à double bras est constitué par l'assemblage de deux bras cobotiques Universal Robots UR10 (6 degrés de liberté et 10 kg de charge utile) sur un torse CROM 1-DOF

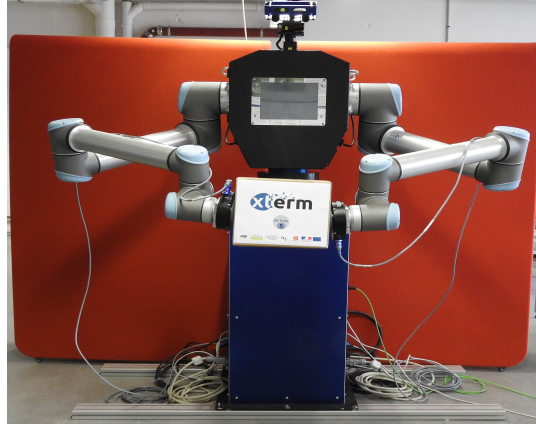


FIGURE 1 – Le robot manipulateur à double bras CROM

(Figure 1). Le CROM est équipé d'une tête munie d'une caméra 3D XTION de type Pan-Tilt (moteurs Dynamixel Pro H42). Les bras UR10 sont commandés à une fréquence de 125 Hz par le biais d'une interface utilisateur, ou par des commandes de script et par le niveau C-API. Les commandes sont en vitesses articulaires ou portent sur une combinaison de position articulaire, de vitesse articulaire et d'accélération articulaire. L'extrémité de chaque bras est équipée d'un capteur 6-axes de force et de couple FT 150 Robotiq. Les composants de notre système sont coordonnés par ROS (Robot Operating System), qui fournit des services de transmission de messages inter-processus (IPC) au sein d'un réseau.

2.2 Modélisation cinématique du système coopératif

La modélisation cinématique du système coopératif composé de deux bras manipulateurs UR10 et du torse CROM est décrite en détail dans Benali et al. (2018). Nous rappelons les points essentiels dans ce paragraphe. La commande prend en compte la contrainte unilatérale selon laquelle les bras peuvent pousser mais ne peuvent pas tirer l'objet saisi. Nous supposons en outre que le torse reste fixe et par conséquent le schéma de commande n'intègre pas à ce stade la rotation du torse.

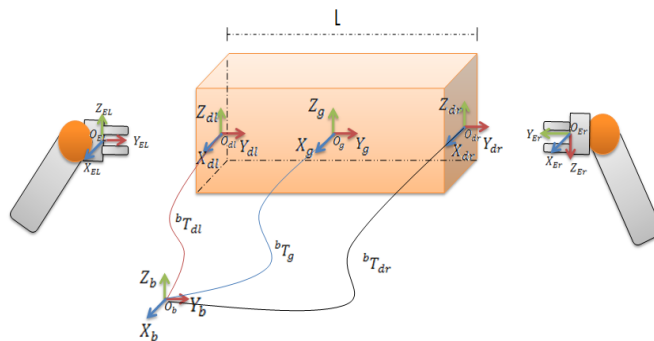


FIGURE 2 – Schéma des repères pour le système robotique à double bras.

La matrice de préhension Gr d'un système manipulateur à deux bras représente la relation entre les vitesses des objets et celles des effecteurs finaux, décrivant comment les mouvements des bras influencent les mouvements de l'objet manipulé.

Le système robotique comporte deux bras manipulateurs, ce qui signifie que la matrice de préhension est formée des composants de la matrice de préhension de chaque bras.

L'équation $\dot{x}_i = Gr_i \dot{x}_o$ lie la vitesse \dot{x}_o de l'objet à la vitesse \dot{x}_i du bras i .

3. Modélisation de la commande du système coopératif

Nous rappelons maintenant les concepts de commande en position et de commande en force qui sont au fondement des commandes hybrides Force/Position. Le détail des équations figure dans Benali et al. (2018).

3.1 Commande en position ou vitesse

Les erreurs de position et d'orientation (e_p, e_r) sont les variables d'entrée de notre boucle d'asservissement :

$$e_p = p_d - p \quad \text{et} \quad e_r = \xi(R_d R^T) \quad (1)$$

La loi de commande proportionnelle calcule les vitesses de chaque manipulateur en fonction des erreurs de position et d'orientation, en utilisant les gains de réaction K_i et les constantes spécifiques au manipulateur Gr_i :

$$v_i = Gr_i K_i \begin{bmatrix} e_p \\ e_r \end{bmatrix} \quad (2)$$

Les vitesses dans l'espace articulaire s'obtiennent à partir des vitesses cartésiennes v_i grâce au pseudo-inverse de la matrice jacobienne J_a :

$$\dot{q}_i = J_a^T (J_a J_a^T)^{-1} v_i \quad (3)$$

3.2 Commande en Force

Nous avons intégré deux capteurs d'efforts 6 axes à l'extrémité des deux bras. La loi de commande en effort vise à assurer une force donnée et se traduit par l'ajout d'une vitesse supplémentaire v_f résultant de l'écart e_f entre la force de consigne et la force mesurée et d'un gain de réaction K_f :

$$v_f = K_f e_f \quad (4)$$

3.3 Commande en Force/Position

La loi de commande hybride Force/Position assure un compromis entre les consignes à respecter au niveau de la trajectoire à suivre et de la force à exercer. Elle comprend donc un premier terme correctif pour la position et un deuxième terme pour la force. L'agrégation se fait au final en terme de vitesses cartésiennes et est traduite ensuite en vitesses articulaires suivant l'équation :

$$\dot{q}_i = \mathbf{N} (J_a^T (J_a J_a^T)^{-1} Gr_i K_i e_i) + (\mathbf{I} - \mathbf{N}) (J_F^T (J_F J_F^T)^{-1} K_f e_f) \quad (5)$$

4. Comparaison de deux commandes

Dans cette section, nous étudions deux architectures hybrides de commande. La première est l'architecture Force/Position-Force/Position, qui est la plus couramment rencontrée dans la littérature. La seconde est l'architecture Force/Position-Position que nous proposons d'introduire. Dans cette optique, nous allons optimiser les paramètres des commandes ce qui nécessite un travail de modélisation dynamique. Dans ce but, nous introduisons la notion de compliance sur les faces latérales du carton avec les raideurs k_1 et k_2 comme indiqué sur la Figure 3.

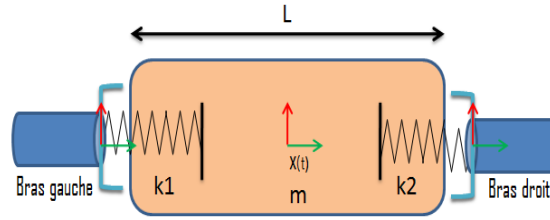


FIGURE 3 – Modélisation de la compliance de l'objet et des deux bras manipulateurs

Nous considérons que le bras gauche (indice L=left) est contrôlé en utilisant une commande en Position, tandis que le bras droit (indice R=right) est contrôlé en utilisant une commande en Force/Position et nous obtenons les équations suivantes :

$$\dot{x}_L = K_L(x_L^{th}(t) - x_L(t)) \quad (6)$$

$$\dot{x}_R = K_R(x_R^{th}(t) - x_R(t)) + K_f(F(t) - F_0) \quad (7)$$

Nous fixons les notations suivantes :

- \dot{x}_R, \dot{x}_L représentent les vecteurs de vitesse du bras droit et gauche respectivement.
- $x_R^{th}(t), x_L^{th}(t), x^{th}(t)$ sont les positions désirées du bras droit et gauche, et du centre du carton.
- $x_R(t), x_L(t)$ représentent les positions mesurées sur les bras.
- $x(t)$ est la position du centre de gravité du carton.
- $F_R(t), F_L(t)$ représentent les forces exercées par le bras droit et le bras gauche sur le carton.
- F_0 est la force minimale à exercer pour éviter tout glissement.
- K_f, K_L, K_R sont les gains des lois de commande à ajuster.

Les positions des extrémités des bras droit et gauche sont définies comme suit :

$$x_L^{th}(t) = x^{th}(t) - \frac{L}{2} \text{ et } x_R^{th}(t) = x^{th}(t) + \frac{L}{2} \quad (8)$$

Pour simplifier, les forces sont supposées agir uniquement dans la direction (x) du repère de l'objet, d'où :

$$F_R(t) = -K_1(x(t) - \frac{L}{2} - x_L(t)) + F_0 \text{ et } F_L(t) = K_2(x(t) + \frac{L}{2} - x_R(t)) + F_0 \quad (9)$$

La force relative $F_r(t)$ imposée sur l'objet peut être décrite par la relation (10) :

$$F_r(t) = F_R(t) - F_L(t) \quad (10)$$

Ces expressions sont valides tant que les manipulateurs sont en contact avec l'objet ($F_L(t) \geq 0$ et $F_R(t) \leq 0$).

En appliquant le principe fondamental de la dynamique sur le carton, avec M la masse du carton, le modèle dynamique du système est le suivant :

$$M\ddot{x}(t) = (-K_1(x(t) - \frac{L}{2} - x_L(t)) + F_0) + (-K_2(x(t) + \frac{L}{2} - x_R(t)) - F_0) \quad (11)$$

$$M\ddot{x}(t) + (K_1 + K_2)x(t) + \frac{L}{2}(K_2 - K_1) - K_1x_L(t) - K_2x_R(t) = 0 \quad (12)$$

$$\dot{x}_L(t) = K_L(x^{th}(t) - \frac{L}{2} - x_L(t)) \quad (13)$$

$$\dot{x}_R(t) = K_R(x^{th}(t) + \frac{L}{2} - x_R(t)) + K_f(F_R(t) - F_0) \quad (14)$$

$$F_R(t) = K_2(x(t) + \frac{L}{2} - x_R(t)) + F_0 \quad (15)$$

Par conséquent, l'équation (16) peut être déduite à partir des équations (14) et (15) :

$$\dot{x}_R(t) = K_R(x^{th}(t) + \frac{L}{2} - x_R(t)) + K_f(K_2(x(t) + \frac{L}{2} - x_R(t))) \quad (16)$$

Les équations précédentes correspondent à un système différentiel linéaire à 4 équations, où le vecteur d'état est : $X = [X_1 \ X_2 \ X_3 \ X_4]^t = [x_L(t) \ \dot{x}_R(t) \ x_L(t) \ x_R(t)]^t$; le vecteur d'entrée est : $U = [1 \ x^{th}(t)]^t$ et le vecteur de sortie est : $Y = [Y_1 \ Y_2 \ Y_3 \ Y_4]^t$

La représentation dans l'espace d'état devient : $\dot{X} = AX + BU$ et $Y = CX$ où :

$$A = \begin{bmatrix} 0 & -\frac{K_1+K_2}{M} & \frac{K_1}{M} & \frac{K_2}{M} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -K_L & 0 \\ 0 & K_2K_f & 0 & K_R + K_fK_2 \end{bmatrix} \text{ et } B = \begin{bmatrix} (K_1 - K_2)\frac{L}{M} & 0 \\ 0 & 0 \\ -K_L\frac{L}{2} & K_L \\ K_R\frac{L}{2} + K_fK_2\frac{L}{2} & K_R \end{bmatrix} \text{ et } C = I_4 \quad (17)$$

Pour une trajectoire imposée $x^{th}(t)$ et un F_0 fixé, on peut résoudre le système et en déduire $x(t)$, $x_L(t)$, $x_R(t)$ et $F_R(t)$.

En suivant la même procédure, nous pouvons déduire la représentation dans l'espace d'état pour l'architecture Force/Position-Force/Position. Seules les matrices A et B changent et deviennent :

$$A = \begin{bmatrix} 0 & -\frac{K_1+K_2}{M} & \frac{K_1}{M} & \frac{K_2}{M} \\ 1 & 0 & 0 & 0 \\ 0 & K_1K_{fl} & -K_{fl}K_1 - K_L & 0 \\ 0 & K_2K_{fr} & 0 & -(K_R + K_{fr}K_2) \end{bmatrix} \text{ et } B = \begin{bmatrix} (K_1 - K_2)\frac{L}{M} & 0 \\ 0 & 0 \\ -K_g\frac{L}{2} - K_{fg}K_1\frac{L}{2} & K_g \\ K_d\frac{L}{2} + K_{fd}K_2\frac{L}{2} & K_d \end{bmatrix} \quad (18)$$

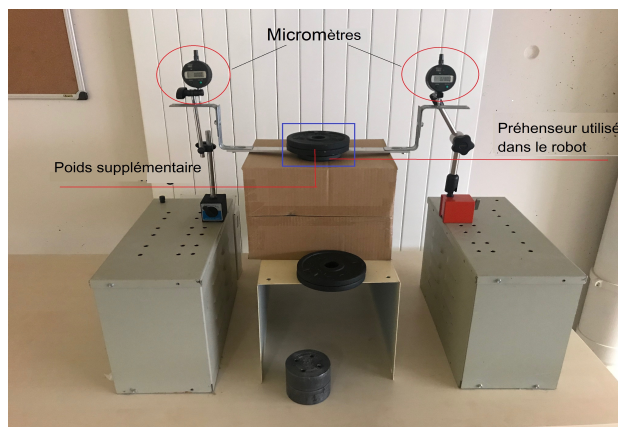
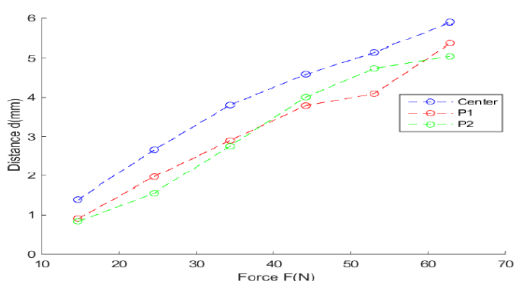


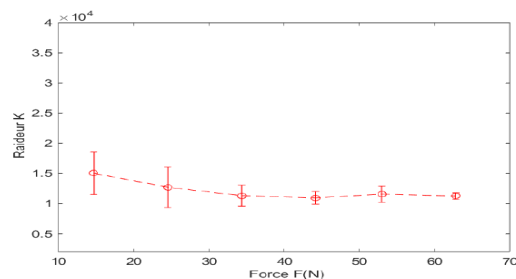
FIGURE 4 – Dispositif expérimental pour l’estimation de la raideur k

4.1 Estimation expérimentale de la compliance du carton

Pour optimiser la commande, il est nécessaire d’estimer expérimentalement la compliance $\frac{1}{k} = \frac{x}{F}$ du carton où F est la force appliquée au centre P de la face latérale du carton (en N) et x est la déflexion au point de contact (en m). Nous utilisons deux micromètres situés de part et d’autre d’une barre indéformable en contact ponctuel avec le centre du carton et nous ajoutons des masses additionnelles pour augmenter la force verticale et mesurer la déflexion correspondante (Figure 4). Une campagne d’essais permet de tracer les courbes affichées sur la Figure 5 où P1 (resp. P2) est un point de contact décalé de 2 cm vers l’arrière du centre de la face (resp. vers l’avant).



(a) Détermination expérimentale de la relation force/déflexion.



(b) Détermination expérimentale de la raideur au centre de la face latérale

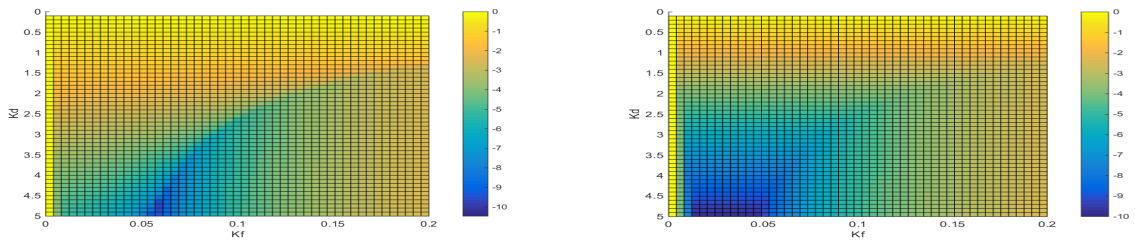
FIGURE 5 – Détermination expérimentale de la raideur du carton

La raideur obtenue expérimentalement a une valeur moyenne de $12000\text{N} \cdot \text{m}^{-1}$ avec une variabilité de $3000\text{N} \cdot \text{m}^{-1}$ due aux différences des matériaux et des contenus.

4.2 Optimisation des commandes

Pour optimiser les commandes, nous utilisons les valeurs moyennes expérimentales précédemment estimées pour les deux paramètres K_1 et K_2 et nous nous intéressons maintenant à déterminer les valeurs optimales des coefficients K_d et K_f . Dans cette optique, pour les deux types d’architectures de commande (Force/Position-Position et Force/Position-Force/Position), nous calculons les parties réelles des valeurs propres de la matrice d’état du système et les représentons sur les Figures 6a et 6b. Les valeurs propres ont toutes une partie réelle négative et

cela indique que la stabilité du système est bien assurée. Nous pouvons alors améliorer la robustesse de la commande en choisissant les parties des valeurs propres les plus négatives possible (zones bleues), ce qui permet d'optimiser les paramètres de nos commandes avant de pouvoir les comparer.



(a) Les valeurs propres de la matrice d'état (avec partie réelle négative) pour une architecture Force/Position-Force/Position

(b) Les valeurs propres de la matrice d'état (avec partie réelle négative) pour une architecture Force/Position-Position

FIGURE 6 – Les valeurs propres de la matrice d'état pour les deux architectures

Les paramètres K_d et K_f ayant été fixés, la commande Force/Position est appliquée sur le bras droit et la commande Position sur le bras gauche suivant le schéma général de la Figure 7.

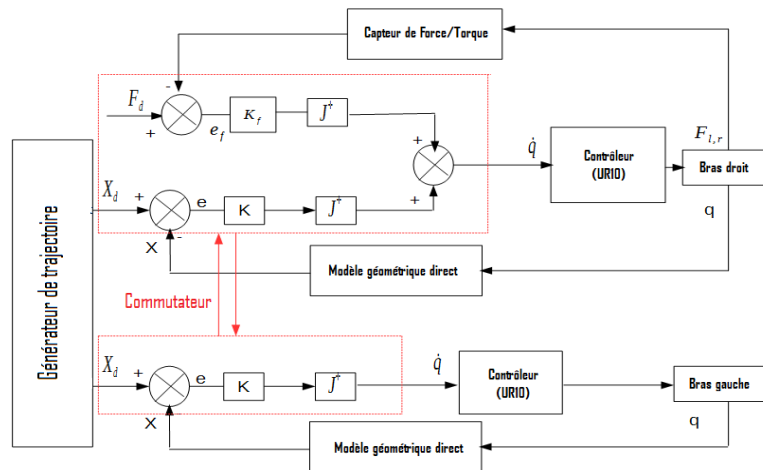
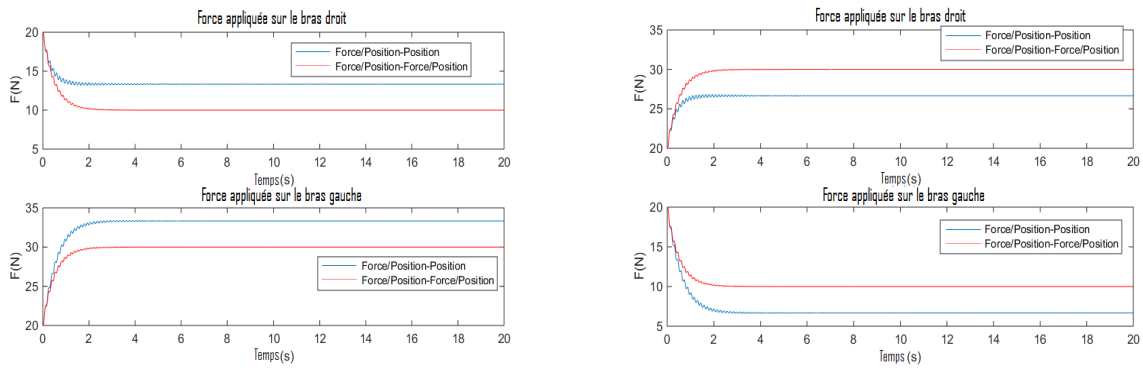


FIGURE 7 – Diagramme de commande du robot à double bras

5. Expérimentations et discussions

Nous comparons expérimentalement les performances des deux méthodes de commande : commande hybride Force/Position sur les deux bras ; commande en position sur un bras et commande en Force/Position sur l'autre bras. La trajectoire planifiée est un mouvement rectiligne avec une accélération constante donnée ($\pm 10m/s^2$) menant à une trajectoire $x^{th}(t) = \frac{1}{2}at^2$ et une force initiale appliquée de $F_0 = 20N$. Les performances sont évaluées par rapport à la force minimale $F(t)$ qui doit rester au-dessus de la valeur minimale F_0 qui évite le glissement du carton. Les forces sont mesurées et affichées sur la Figure 8.



(a) Forces exercées sur le carton ($a = 10m/s^2$)

(b) Forces exercées sur le carton ($a = -10m/s^2$)

FIGURE 8 – Forces exercées sur le carton par les bras droit et gauche

5.1 Architecture de Commande Hybride Force-Position sur les deux bras

Lorsque les deux bras sont commandés en Force/Position, dans le cas où le bras gauche avance et le bras droit recule ($a = 10m/s^2$), la force exercée par le bras droit diminue d'environ 10 N pour se stabiliser à 10 N, tandis que la force exercée par le bras gauche augmente d'environ 10 N pour se stabiliser à 30 N. Ce faisant, il y a un risque de glissement du carton sur la face latérale droite. Du fait que le schéma de commande est symétrique, les mêmes remarques s'appliquent pour une accélération négative ($a = -10m/s^2$) correspondant à un mouvement vers la gauche.

5.2 Architecture de Commande mixte Force/Position et position

Le bras gauche est commandée en position tandis que le bras droit est commandé en Force/Position. Lors d'un déplacement vers la droite, la force exercée par le bras gauche sur le carton est de 33 N tandis que le bras droit exerce une force de 13 N. Le bras gauche commandé en position "pousse" en quelque sorte le carton et le bras droit s'adapte pour garder un différentiel de force de 20 N. Par-contre, lors d'un déplacement vers la gauche, la force exercée par le bras gauche sur le carton est de 7 N tandis que le bras droit exerce une force de 27 N. Le bras gauche "se rétracte" vers la gauche et le bras droit s'adapte pour garder un différentiel de force de 20 N. Le premier enseignement est que cette commande est fortement dissymétrique. La deuxième conclusion est que l'on peut exploiter cette dissymétrie en commandant en position le bras qui se déplace dans la direction de l'accélération du carton. En procédant de cette manière, la force minimale exercée par le bras commandé en Force/Position est plus élevée que celle obtenue lorsque les deux bras sont commandés en Force/Position. Cependant cette méthode nécessite d'intégrer un processus de commutation de commande entre les bras en fonction du signe de l'accélération désirée pour le carton.

6. Conclusions

La saisie de cartons par les faces latérales en exerçant une pression par les deux bras d'un torse humanoïde utilise généralement une commande Force/Position sur les deux bras. Dans cet article, nous avons présenté une architecture de commande innovante, utilisant une commande en position pour le bras qui se déplace dans la direction de l'accélération positive souhaitée pour le carton et une commande en Force/Position pour l'autre bras. Afin de pouvoir comparer les deux méthodes, nous avons d'abord cherché à modéliser le système mécanique complet en prenant en

compte la compliance du carton. Nous avons mené une campagne d'expérimentation pour estimer la raideur du carton au point de contact et utilisé cette valeur pour optimiser les performances des deux modes de commande choisis. Nous avons ensuite comparé expérimentalement les deux modes de commande sur une trajectoire rectiligne avec accélération constante et mesuré les valeurs des forces exercées par les bras sur les faces latérales du carton. Les résultats montrent que la nouvelle commande proposée (Force/Position-position) permet d'assurer une meilleure adhérence et d'éviter les glissements du fait que la force minimale exercée est supérieure à celle d'une commande symétrique Force/Position. Cette nouvelle commande peut donc être mise en œuvre avec succès mais nécessite de bien gérer la commutation de commande entre les deux bras car le bras qui se déplace dans la direction de l'accélération positive doit être commandé en position. Cette commutation de commande peut être réalisée par une fonction non-linéaire d'hystérésis.

7. Remerciements

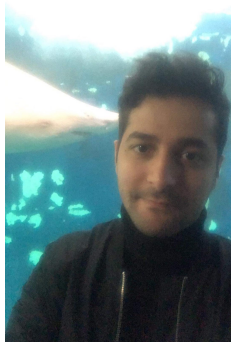
Les auteurs remercient l'Université le Havre Normandie, la Région Haute-Normandie et le FEDER (Fonds Européen de Développement Régional) qui ont financé les travaux dans le cadre du projet XTERM : Usine du Futur.

Références

- Almeida, P. A. F. and Lopes, A. (1999). Force-impedance control : a new control strategy of robotic manipulators.
- Benali, K., Brethé, J.-F., Guérin, F., and Gorka, M. (2018). Dual arm robot manipulator for grasping boxes of different dimensions in a logistics warehouse. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 147–152, Lyon, France.
- Erhart, S. and Hirche, S. (2015). Internal force analysis and load distribution for cooperative multi-robot manipulation. *IEEE Transactions on Robotics*, 31(5) :1238–1243.
- Erhart, S., Sieber, D., and Hirche, S. (2013). An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 315–322, Tokyo, Japan.
- Lin, H.-C. e. a. (2018). A projected inverse dynamics approach for multi-arm cartesian impedance control. In *IEEE/RSJ Int. Conf. on Robotics and Automation*, pages 5421–5428, Brisbane, Australia.
- Liu, A.-M. J. (2000). Robust control of planar dual-arm cooperative manipulators. *Robotics and Computer Integrated Manufacturing*.
- Nurkan Yagiz, Y. H. and Arslan, Y. Z. (2010). Load transportation by dual arm robot using sliding mode control. *Journal of Mechanical Science and Technology*, 24(5) :1177–1184.
- Park, H. A. and Lee, C. S. G. (2016). Dual-arm coordinated-motion task specification and performance evaluation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 929–936, Daejeon, South Korea.

- Pliego-Jimenez, J. and Arteaga-Perez, M. (2017). On the adaptive control of cooperative robots with time-variant holonomic constraints. *International Journal of Adaptive Control and Signal Processing*, 31(8) :1217–1231.
- Yim, W. R. W., Selvarajan, M., and Wells (1999). Sliding mode cooperative motion control of dual arm manipulators. *Artif Life Robotics*, 3 :166–169.

Biographie des auteurs



Khairidine Benali est un chercheur en robotique, dont les spécialités englobent la robotique médicale, la collaboration entre robots et l'intelligence artificielle appliquée à ce domaine. Il a obtenu son doctorat en sciences robotiques à l'Université de Normandie Le Havre en 2019, après avoir suivi ses études de Master à l'Université de Montpellier. Actuellement, il exerce en tant que chercheur en robotique et intelligence artificielle à l'Université de Nottingham.



Jean-François Brethé est Professeur des Universités en robotique à l'Université Le Havre Normandie, Institut Supérieur d'Etudes Logistiques (ISEL) et membre du GREAH (Groupe de Recherche en Electrotechnique et Automatique du Havre). Ses travaux de recherche portent initialement sur la modélisation, la conception et l'étude des performances en précision des robots manipulateurs industriels, puis se focalisent aujourd'hui sur les verrous scientifiques posés par l'introduction de cobots dans les entrepôts logistiques (préparation de commandes robotisée par robot manipulateur mobile, saisie de colis de taille variable par torse humanoïde).



Dr François Guérin est Agrégé de Génie Electrique-Electronique (1995) et Docteur de l'Université Le Havre Normandie en Automatique et Robotique (2004). Il a obtenu son Habilitation à Diriger des Recherches (2015) et travaille comme Maître de Conférences à l'Université Le Havre Normandie, au sein de l'Institut Supérieur d'Etudes Logistiques (ISEL) et du Groupe de Recherche en Electrotechnique et Automatique (GREAH). Ses enseignements portent sur l'Électronique (IoT), l'Automatique et la Robotique et ses travaux de recherche s'intéressent à la Commande Avancée des Systèmes, appliquée à la Robotique Mobile Terrestre et Aérienne. En 2021, il a créé au Havre (76), le Centre d'Innovation Drones de Normandie (CIDN) et a intégré en 2022, la Réserve Citoyenne de L'Armée de Terre auprès du 61eme Régiment d'Artillerie basé à Chaumont (52).

UTILISATION D'ALGORITHMES D'INTELLIGENCE ARTIFICIELLE POUR L'IDENTIFICATION ET LA LOCALISATION DE COLIS ET LEUR SAISIE PAR UN TORSE HUMANOÏDE DANS DES OPÉRATIONS DE DÉPALETTISATION

Santheep YESUDASU¹, Jean-François BRETHÉ¹, Wafae SEBBATA¹, Patrick BONNIN²

¹GREAH, Université Le Havre Normandie, France
santheep.yesudasu@univ-lehavre.fr
jean-francois.brethe@univ-lehavre.fr

²ISTY, University of Versailles Mantes-la-Ville, France
patrick.bonnin@gmail.com

RESUME

Dans le contexte Logistique 4.0, la robotisation des opérations de manutention se développe mais doit encore résoudre certains verrous scientifiques pour réussir à remplacer la dextérité des opérateurs humains. L'utilisation d'un torse humanoïde s'est révélé une solution pertinente pour la saisie de colis de taille variable. Cependant la robotisation complète des opérations de dépalettisation par un torse humanoïde reste un problème complexe lorsque les cartons sont disposés en couche et juxtaposés. Il s'agit dans un premier temps d'identifier et localiser les cartons unité par unité pour pouvoir les manipuler par la suite. Les travaux présentés dans cet article montrent que l'approche usuelle par analyse d'image atteint ses limites, puis introduit l'utilisation de l'intelligence artificielle et de l'apprentissage profond sur les données capteurs fournies par une caméra RGD-D. La conception et la mise en œuvre de cette méthode sont détaillées et les résultats expérimentaux sont présentés.

MOTS CLÉS: Dépalettisation robotisée, Torse humanoïde, Apprentissage profond, Vision par ordinateur, Intelligence artificielle.

ABSTRACT

In the context of Logistics 4.0, the robotisation of handling operations is developing but still needs to resolve specific scientific issues to succeed in replacing the dexterity of human operators. The use of a humanoid torso has proved to be a relevant solution for picking up parcels of varying sizes. However, the complete robotisation of depalletising operations by the humanoid torso remains a complex problem when the boxes are set up in layers and juxtaposed. First of all, the boxes have to be identified and located unit by unit so that they can then be handled. The work presented in this paper shows that the image analysis approach is reaching its limits and introduces the use of artificial intelligence and deep learning on the sensor data provided by an RGD-D camera. The design and implementation of this method are detailed and the experimental results are presented.

KEYWORDS: Depalletising Robot, Humanoid torso, Deep learning, Computer vision, Artificial intelligence.

1. Introduction

La logistique est en pleine mutation et intègre les concepts de l'Industrie 4.0 dans ses processus (Echelmeyer et al., 2008). Les nombreuses tâches de manutention nécessaires au fonctionnement des process de préparation de commandes sont aujourd'hui robotisées partiellement (taux estimé à 30 %). On estime que dans 10 ans, ce seront 70 % des tâches de manutention qui seront robotisées. Ceci permettra de diminuer le nombre d'accidents du travail et d'arrêts maladies, et soulagera les opérateurs victimes de Troubles Musculo-Squelettiques (TMS). La difficulté à recruter les ressources humaines dans le secteur est aussi un autre facteur qui pousse à une automatisation plus rapide des opérations de manutention.

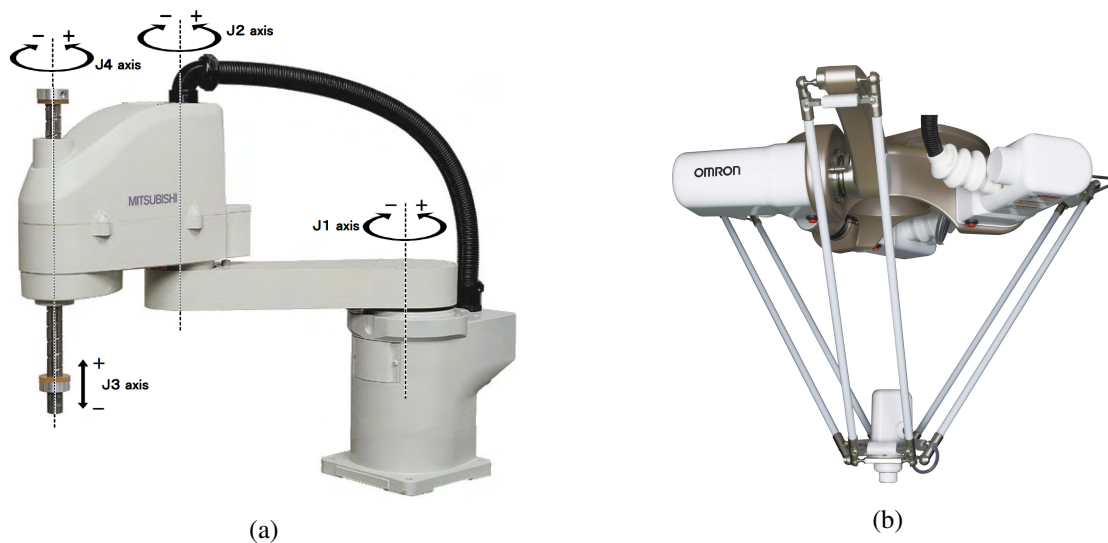


FIGURE 1 – (a) SCARA (Selective Compliant Assembly Robotic Arm) (b) bras robotiques parallèles (Delta)

Les opérations de saisie et de manutention robotisée de cartons sont majoritairement effectuées par des bras manipulateurs série de type anthropomorphes à 6 degrés de liberté, ou SCARA (Selective Compliant Assembly Robotic Arm) à 4 degrés de liberté, ou des bras robotiques parallèles (Delta, Quattro, etc) présentés sur la Figure 1b. Les préhenseurs utilisés sont généralement des systèmes de type ventouses (où la dépression est obtenue par utilisation d'une pompe à vide ou par effet Venturi à partir d'une source d'air comprimé), de type pinces à deux doigts ou des préhenseurs conçus spécifiquement pour un objet donné. Cependant dans le secteur de la préparation de commandes, ces bras sont confrontés à des objets de taille, de géométrie, de texture et de masse variable et ne peuvent pas toujours effectuer la manutention de cette grande variété d'objets. Or l'observation du comportement des opérateurs humains montre qu'ils sont souvent amenés à utiliser leurs deux bras simultanément pour saisir les objets. Ceci nous a amené à mettre en œuvre un torse humanoïde pour la saisie par les faces latérales de cartons de taille variable en mettant en œuvre un algorithme de commande spécifique (Benali et al., 2019) pour d'une part, exercer le juste effort empêchant le glissement sans écraser l'objet et d'autre part, assurer une coordination efficace des mouvements des deux bras.

Dans les travaux présentés ici, nous travaillons sur la dépalettisation automatisée effectuée par le torse humanoïde. Dans ce cas, les cartons présents sur la palette sont placés les uns à côté

des autres en couches superposées et ne sont plus isolés comme dans le cas de la thèse citée précédemment. Les géométries des cartons ne sont pas connues a priori. Nous sommes confrontés à deux verrous scientifiques : l'un concerne l'exploitation efficace des informations issues de la caméra RGB-D pour identifier les géométries et localiser unitairement les cartons ; l'autre la conception de préhenseurs adaptés et la mise en œuvre d'une série de mouvements de manipulation pour pouvoir saisir et transporter les cartons unitairement pour les déposer par exemple sur un convoyeur. Dans cet article, nous présentons les résultats obtenus relatifs à l'identification et la localisation. Après avoir dressé un état de l'art permettant de cerner les limites des techniques d'analyse d'images classique, nous nous intéressons aux potentialités offertes par les dernières méthodes d'intelligence artificielle de type apprentissage profond. Nous proposons par la suite une méthodologie de mise en œuvre de ces méthodes pour lever le verrou scientifique étudié. Enfin, nous présentons les résultats expérimentaux obtenus avec le torse humanoïde.

2. Etat de l'art

La détection d'objets basée sur la vision est un domaine largement étudié dans le domaine de la vision par ordinateur, et de nombreuses études existantes se sont concentrées sur la résolution des différents défis et limitations associés à ce type de détection. Ces défis peuvent inclure la sensibilité aux conditions d'éclairage et la difficulté à détecter les objets à faible texture. Les systèmes existants (Katsoulas et al., 2002), (Monica et al., 2020), (Prasse et al., 2011) ont ajouté un capteur supplémentaire tel que ToF (Time-of-Flight) et RFID (Radio-Frequency IDentification). Certains systèmes de vision 3D nécessitent toujours les dimensions de l'objet à détecter en entrée, bien que cette exigence diminue progressivement à mesure que la technologie évolue. En plus des systèmes de vision 3D, d'autres méthodologies (Caccavale et al., 2020) peuvent également être utilisées dans un système d'étiquetage et de dépalettisation. Une telle approche consiste à étiqueter les boîtes en carton cibles et à utiliser un système de dépalettisation hiérarchiquement structuré. Cette méthode implique d'appliquer des étiquettes aux boîtes contenant des informations sur le contenu, l'emplacement et la destination de la boîte. Le système d'étiquetage peut être intégré à un système de dépalettisation utilisant une approche hiérarchique pour retirer les boîtes de la palette et les trier dans leurs destinations finales.

Le système basé sur la vision mentionné dans (Nakamoto et al., 2016) peut utiliser diverses techniques de traitement d'image telles que l'extraction de contours et la détection de motifs à partir d'images lumineuses. Ces techniques sont utilisées pour extraire des informations pertinentes des images d'objets et les utiliser pour les identifier et les classer. L'extraction de contours 3D basée sur le nuage de points est une autre méthode bien connue qui peut être utilisée dans les systèmes basés sur la vision, comme décrit dans (Vohra et al., 2021). Cependant, cette approche ne fonctionne que lorsque les dimensions de l'objet sont connues. En d'autres termes, le système doit avoir une connaissance préalable de la taille et de la forme de l'objet pour extraire précisément ses contours 3D à partir des données de nuage de points. Cela peut limiter la polyvalence et la mise à l'échelle du système, car il peut ne pas être capable de manipuler une large gamme d'objets sans informations supplémentaires.

YOLO (You Only Look Once) (Redmon et al., 2016) est une méthode de détection d'objets en temps réel bien connue pour sa vitesse de traitement rapide et sa haute précision. En 2015, YOLOv1 a été introduit avec une approche matricielle performante pour la détection d'objets dans les images mais présentant des performances plus faibles lorsque les objets sont petits. YOLOv2 a amélioré les performances avec des boîtes d'ancrage, une normalisation par lots et une architecture de réseau neuronal plus puissante. En 2018, YOLOv3 a introduit une architecture de réseau plus grande et d'autres améliorations, atteignant une précision de pointe et une vitesse d'inférence plus rapide. YOLOv4 a ajouté une architecture de réseau neuronal plus puissante et un regroupement de boîtes basé sur des ancrages, atteignant une précision encore plus élevée et une vitesse d'inférence plus rapide. YOLOv5 est une réécriture complète avec une nouvelle architecture de réseau neuronal et une approche d'entraînement, atteignant une précision encore plus élevée et une vitesse d'inférence plus rapide que YOLOv4 (Jiang et al., 2022).

Dans la recherche de (Redmon et al., 2016) (Reyes et al., 2019) (Redmon and Farhadi, 2018) YOLOv3 utilise un seul réseau neuronal pour prédire plusieurs boîtes englobantes et les probabilités de classe associées pour chaque objet dans l'image. Le score d'objectivité pour chaque boîte englobante est déterminé par régression linéaire, et représente la probabilité que la boîte contienne un objet d'intérêt. Les scores de classification et les coordonnées de la boîte englobante sont combinés avec le score d'objectivité pour produire la prédiction finale pour chaque objet dans l'image. Cette approche permet à YOLOv3 d'effectuer la détection et la reconnaissance d'objets en temps réel avec une grande précision. Il utilise également une combinaison de convolutions 3x3 et 1x1 dans l'architecture Darknet-53 pour l'extraction de caractéristiques, qui est plus rapide que ResNet-101 et similaire en performance à ResNet-152. De plus, YOLOv3 surpasse les versions précédentes de YOLO et d'autres modèles de détection d'objets en termes de précision moyenne et de valeurs d'intersection sur l'union. Comme prouvé dans (Redmon et al., 2016) (Reyes et al., 2019) (Redmon and Farhadi, 2018) les résultats d'entraînement et de test basés sur le darknet montrent qu'ils ont une meilleure capacité dans YOLOv3, qui s'exécute de manière significativement plus rapide que d'autres méthodes de détection en temps réel avec des performances comparables. Dans la recherche (Sebbata et al., 2020) une application robotique utilisant YOLOv3 a été développée pour identifier des objets de même texture mais présentant différentes géométries. L'approche de (Schwarz et al., 2018) utilise l'apprentissage par transfert pour obtenir la puissance des réseaux neuronaux profonds dans la robotique avec des quantités minimales de données d'entraînement. En apprentissage profond, les couches de départ sont chargées d'identifier les caractéristiques significatives. Ensuite, seules les couches finales, qui effectuent des tâches avancées comme la détection ou la segmentation doivent être affinées.

Le travail présenté dans cet article est conçu pour surmonter les défis posés par les conditions d'éclairage pour la reconnaissance de boîtes polyvalentes sans texture (géométrie en forme de pavé droit) dans les opérations de dépalettisation en temps réel. En se concentrant sur la sensibilité des conditions d'éclairage standard et faible et des boîtes alignées étroitement, le système vise à fournir une détection rapide et précise des objets dans ces environnements difficiles. L'utilisation d'un torse humanoïde industriel avec des bras doubles UR10 est un choix intéressant car il offre un niveau élevé de flexibilité et de polyvalence dans la manipulation des objets. Il est important de noter que la détection d'objets en temps réel peut être une tâche complexe, car elle implique le traitement de grandes quantités de données en temps réel et la prise de décisions rapides basées sur l'apprentissage profond. Dans l'ensemble, la conception de ce système garantit

que le système peut répondre aux spécifications de performance requises.

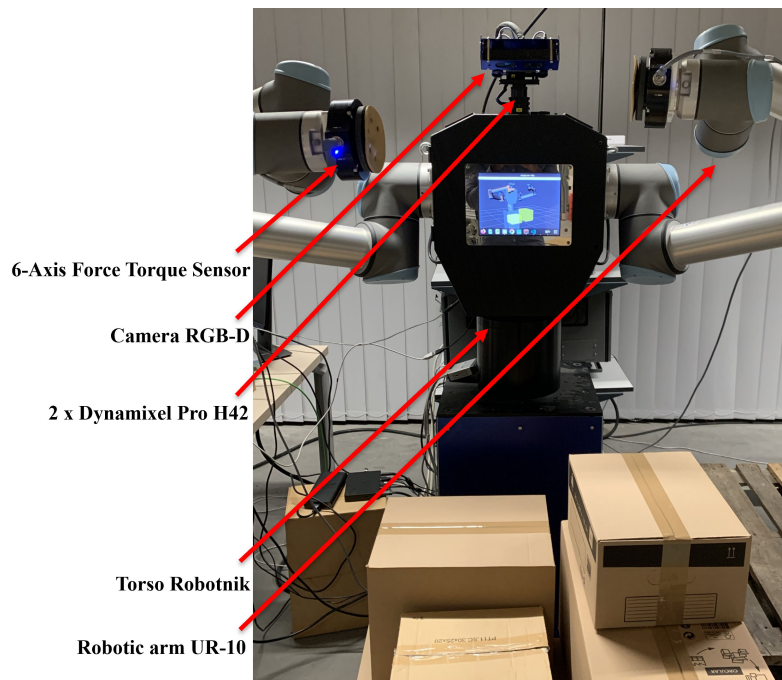


FIGURE 2 – Torse humanoïde pour la saisie et la manipulation.

3. Méthodologie

Cette recherche porte sur le développement d'un système automatisé de dépalettisation alimenté par l'IA pour des objets parallélépipédiques. Cet article est principalement axé sur le développement de la technologie de détection d'objets et d'apprentissage par transfert, en mettant particulièrement l'accent sur la résolution des défis précédemment abordés. De plus, l'article explore les avancées dans l'estimation de la pose et l'algorithme de dépalettisation pour les bras doubles.

La méthodologie décrite dans la Figure 3 utilise une seule caméra RGB-D et Convolutional Neural Network (CNN) basé sur YOLOv3 pour détecter les boîtes en carton sur une palette. Les boîtes englobantes détectées sont ensuite utilisées pour sélectionner une région d'intérêt dans les nuages de points, et Darknet 3D est utilisé pour estimer la pose de l'objet. Ce processus utilise les données de profondeur obtenues à partir de la boîte englobante 2D de YOLOv3. L'algorithme du système s'adapte au succès de la détection et continue de recueillir plusieurs positions de boîtes, ce qui permet une estimation précise de la pose. L'algorithme sélectionne la boîte avec le taux de précision de détection le plus élevé et la proximité la plus proche du bras double pour le processus de prélèvement, même lorsque les boîtes sont partiellement occultées par d'autres boîtes. Afin d'atteindre et de manipuler efficacement un objet, le manipulateur à bras double détermine d'abord la stratégie de préhension optimale pour l'objet, puis planifie le chemin le plus approprié pour l'atteindre. Après l'exécution du processus de dépalettisation, une base de données est générée, qui peut être utilisée pour gérer de manière plus efficace les poses complexes des boîtes et améliorer les processus de dépalettisation futurs.

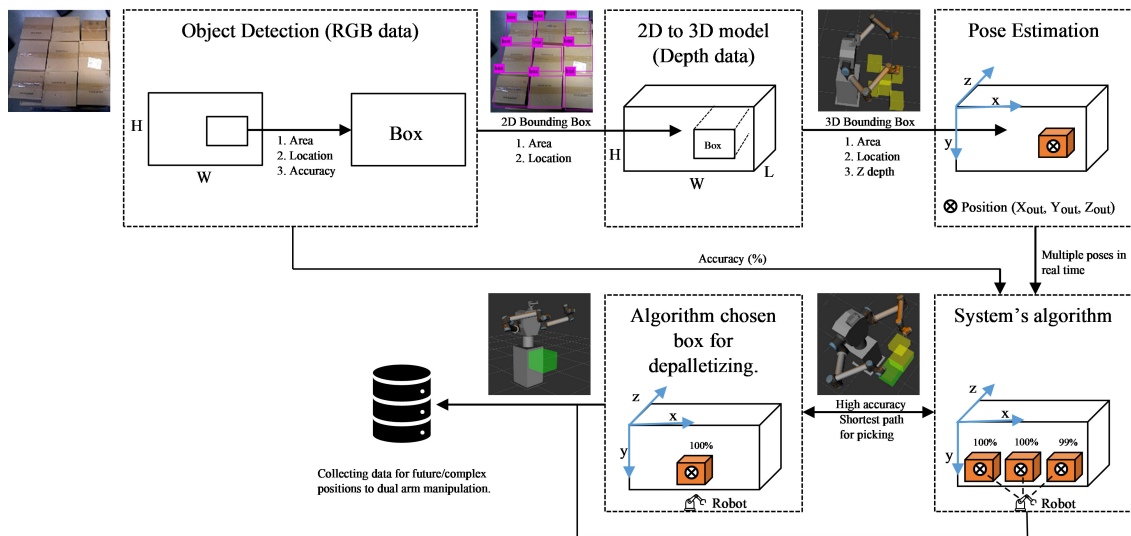


FIGURE 3 – Schéma de principe du processus global

3.1 Détection d'objets en 2D

Le modèle CNN YOLOv3, conçu avec une architecture d'entraînement à un seul stade, a été utilisé pour reconnaître avec précision les objets dans les images RGB. Au début, notre modèle a été entraîné en utilisant des images obtenues exclusivement dans notre environnement de dépalettisation. Cependant, afin d'améliorer son adaptabilité à des environnements divers, nous avons incorporé des images de boîtes en carton courantes dans l'ensemble de données. L'ensemble de données se compose de boîtes ("objets") étroitement annotées avec des étiquettes prioritaires pour des processus de prélèvement efficaces dans notre environnement. De plus, la surface supérieure des boîtes a été prise en compte pour détecter quand la couche supérieure de la palette était complètement remplie de boîtes en une couche. Pendant la tâche de dépalettisation, la caméra diffusait des images en temps réel, qui étaient traitées par le modèle CNN pour extraire la zone sous forme d'une boîte englobante en 2D.

3.2 Traitement des nuages de points 3D

La méthode proposée convertit les boîtes englobantes 2D en coordonnées 3D en cartographiant la relation géométrique entre l'image et le nuage de points. La région cartographiée est ensuite filtrée pour extraire la Region of Interest (ROI) optimale de l'objet. Cela est réalisé en calculant les valeurs minimales et maximales des coordonnées x , y et z de ces points (x_{min} , y_{min} , z_{min} , x_{max} , y_{max} , z_{max}), formant une boîte englobante 3D dans le nuage de points. Avec ces valeurs, la surface, l'emplacement et la profondeur de l'objet dans l'espace 3D peuvent être estimés. L'étape de détection d'objet est essentielle pour garantir l'exactitude et la précision du traitement des nuages de points 3D, tout en réduisant simultanément l'impact du désordre causé par d'autres boîtes en carton dans l'environnement.

3.3 Estimation de la pose

Après avoir extrait la partie 3D du nuage de points, elle est traitée pour estimer la pose de la boîte englobante spécifique en vue de la manipulation et de la préhension. Une boîte englobante 3D est utilisée pour localiser l'objet dans le cadre de la palette, et le point central des coordonnées dédiées de l'objet est utilisé pour l'estimation de la pose. Ce processus permet la détection et l'estimation en temps réel de plusieurs objets dans le cadre considéré.

3.4 Algorithme du système

Cet algorithme a été créé pour prédire la priorité de la dépalettisation des boîtes en fonction de l'exactitude de la prédiction du CNN et de la boîte prédite la plus proche dans le tableau. Les coordonnées cartésiennes de chaque détection sont stockées dans le tableau avec une grille unique basée sur son taux de précision. Simultanément, l'algorithme traite ces coordonnées pour localiser la boîte la plus proche de la position réelle du Tool Center Point (TCP) des deux bras. Pour ce faire, nous utilisons un algorithme de force brute (Sandström et al., 2020), qui accélère la recherche de paires proches pour la planification de mouvement et l'optimisation de trajectoire. Cet algorithme utilise les coordonnées X, Y et Z du centre de chaque boîte pour identifier le point le plus proche du TCP en utilisant la distance euclidienne. Une fois que nous avons calculé les distances entre toutes les paires de points, l'Algorithme 1 peut trouver la plus petite distance entre un point et un autre. Cet algorithme utilise des planificateurs basés sur l'échantillonnage pour éviter le cas d'utilisation où le bras double a des contraintes et des limitations d'accessibilité pour effectuer les mouvements souhaités.

3.5 Dépalettisation

La planification de trajectoire des bras est réalisée par MoveIt (Sachin et al., 2012), progiciel de planification de mouvement basé sur ROS, comprenant Kinematics and Dynamics Library (KDL) et Open Motion Planning Library (OMPL) gérant aussi l'évitement d'obstacles. L'algorithme de dépalettisation utilise les résultats de détection d'objet et d'estimation de pose pour planifier la trajectoire du bras robotique et exécuter la tâche de dépalettisation.

Algorithm 1 Optimize closest coordinate with Brute Force Algorithm

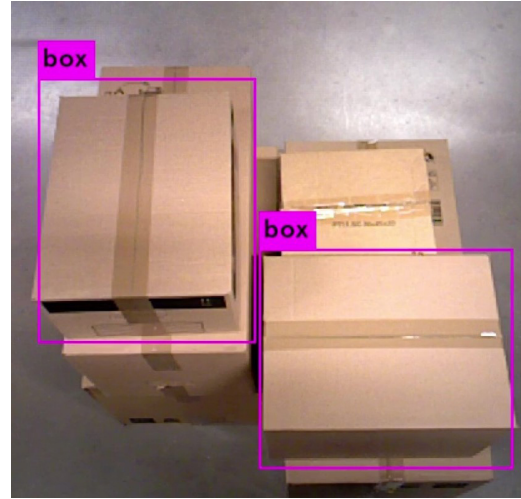
Input : P list of coordinates

Output : Indices index1 and index2 of the closest pair of coordinates

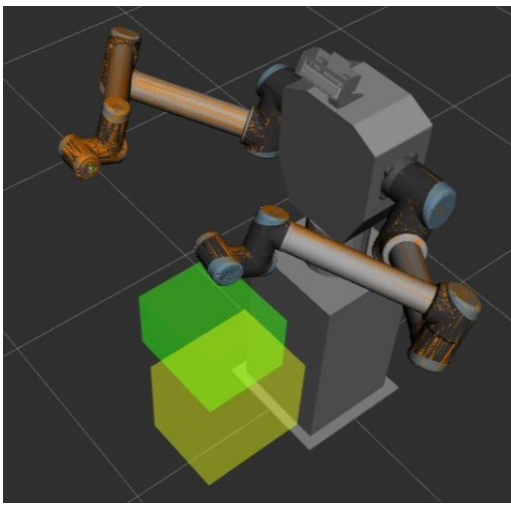
```
 $dmin \leftarrow \infty$ 
1: for  $i \leftarrow 1$  to  $n-1$  do
2:   for  $j \leftarrow i+1$  to  $n$  do
3:      $d \leftarrow \text{sqrt}((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2)$ 
4:     if ( $d < dmin$ ) then
5:        $dmin \leftarrow d; index1 \leftarrow i; index2 \leftarrow j$ 
6:     end if
7:   end for
8: end for
9: return  $index1, index2$ 
```



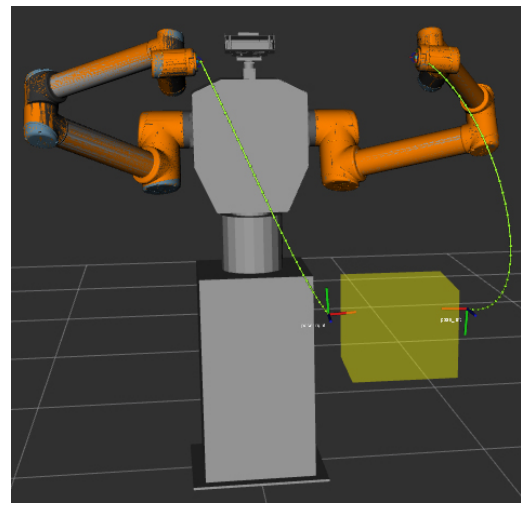
(a)



(b)



(c)



(d)

FIGURE 4 – (a) Boîtes à détecter (b) Boîtes englobantes 2D, (c) Nuage de points 3D, (d) Boîte à saisir en priorité.

4. Expériences

Le système robotique comporte deux bras UR10 montés sur un torse Robotnik, possédant des capteurs de force/couple à six axes et une caméra RGB-D à vision à 360 degrés (Asus XTION). Les mouvements du robot sont contrôlés par MoveIt, un package logiciel open source largement utilisé pour la planification de mouvement, la manipulation et le contrôle des systèmes robotiques. MoveIt fournit un contrôleur de haut niveau pour le robot, qui peut être programmé en utilisant C++ et Python via la distribution et les packages ROS Noetic. L'unité centrale est un processeur Intel Core i7-10875H haute performance, de 32 Go de RAM, d'un GPU NVIDIA Quadro RTX4000 avec 8 Go de mémoire, de Cuda 11.1, de cuDNN v8, et du système d'exploitation Ubuntu 20.04.

4.1 Création et entraînement de notre ensemble de données

Nous avons rassemblé une collection d'images de notre environnement en utilisant une caméra robotique, et avons également inclus des images de boîtes en carton courantes. Enfin, nous

avons annoté manuellement 807 images avec la classe 'boîte' en utilisant l'outil LabelImg. Plus précisément, nous nous sommes concentrés sur l'étiquetage des boîtes visibles en 3D et de la surface supérieure de ces boîtes. Notre priorité a été d'étiqueter les boîtes visibles en 3D tandis que l'étiquetage de la face supérieure minimale de la boîte a été de moindre priorité. Dans les cas où il existe deux possibilités pour la visibilité de la face supérieure et de la boîte en forme de 3D, nous avons constaté que la détection des boîtes de la face supérieure est moins précise que celle des boîtes en forme de 3D. Pour garantir une meilleure précision dans la détection et l'estimation de la pose, nous avons évité d'étiqueter à partir de l'autre possibilité de la portion visible de la boîte pendant l'annotation.

Notre ensemble de données comprend des images avec une résolution de 608×608 pixels. Pour l'entraînement, nous avons fixé la taille du lot à 64 et spécifié le nombre maximum d'itérations à 4000, avec des étapes intermédiaires à 3200 et 3600. L'outil annoté utilisé pour l'étiquetage extrait des fichiers texte au format $\langle classe - objet \rangle \langle x_centre \rangle \langle y_centre \rangle \langle largeur \rangle \langle hauteur \rangle$ à partir des images annotées. Nous avons utilisé un ratio de 80 :20 pour l'entraînement et le test. Notre formation a été réalisée en utilisant le GPU Colab avec les poids initiaux de darknet53.conv.74.

5. Résultats et discussions

Comme le montre la Figure 4, ce système de détection d'objets fonctionne sur deux boîtes inconnues situées en haut d'un ensemble, avec seulement deux faces visibles disponibles pour la détection. Un CNN en temps réel est utilisé pour prédire la classe des boîtes, ce qui donne une représentation visuelle avec une boîte englobante 2D. Ces boîtes englobantes 2D sont ensuite combinées avec des données de nuage de points et affichées dans le logiciel Rviz, fournissant une visualisation en 3D des objets prédits. L'algorithme du système choisit la boîte prioritaire, qui présente une précision plus élevée et est la plus proche de la boîte en fonction de l'Algorithme 1 : Optimisation de la coordonnée la plus proche avec l'algorithme de force brute. Comme le montre la Figure 4 (a), la boîte sélectionnée est au sommet de l'ensemble et la plus proche de la prise pour les bras doubles.

5.1 Test de détection et validation

Les performances vont être testées et validées sur la base des métriques de précision, de rappel et de score F (Goutte and Gaussier, 2005). Ces mesures sont utilisées pour évaluer l'efficacité de ce système dans la détection d'objets et comparer différentes méthodes.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

La précision mesure les prédictions vraies positives, tandis que le rappel mesure les vrais positifs trouvés. Une précision élevée est liée à un faible taux de faux positifs, tandis qu'un rappel

élevé est lié à un faible taux de faux négatifs. Un score de 1.0 pour la précision ou le rappel indique une classification ou une détection parfaite pour cette classe.

TABLE 1 – Score F1

Visible face of box to the detection(FDbox)	<i>F1-Score</i>
Single FDbox	0.81
Multi-FDbox	0.90
Single FDbox when Multi-FDbox	0.25
Multi-FDbox when Single FDbox	0.72
Mixed FDbox	0.82

Le Tableau 1 montre que lors de la phase de test en temps réel, à la fois les boîtes à une seule face et les boîtes avec plus d'une face ont obtenu des scores F1 de 0.81 et 0.90, respectivement. Ces scores sont restés constants quelles que soient les conditions d'éclairage normales et faibles. Le CNN a atteint une précision de détection maximale lors de la détection de boîtes avec au moins deux faces visibles (FDbox). Cependant, même lorsque seule une face de la boîte était visible dans l'ensemble complet de palettes, le CNN pouvait toujours détecter la boîte, bien que la précision soit plus faible que lorsque plusieurs faces étaient visibles. Cela s'explique par le fait que plusieurs faces visibles contiennent plus d'informations contextuelles. En revanche, lorsque seule une face de la boîte est visible, le CNN a plus de difficulté à la différencier du fond de la scène ou d'autres objets similaires. Sur la base des résultats globaux, il ressort que même lors de combinaison de différents scénarii (boîtes alignées étroitement, boîtes situées dans différentes couches et positions, conditions d'éclairage variables), le CNN donne des réponses précises, comme illustré dans la Figure 5.



FIGURE 5 – Résultats de la détection dans différentes situations

5.2 Validation des poses prédites

La précision de l'estimation de la pose est évaluée en réalisant une prédiction de boîtes englobantes 2D sur un ensemble de données annotées, ce qui implique de normaliser les valeurs de pixels et les coordonnées 3D. Les paramètres du modèle sont optimisés pour minimiser la différence entre les points clés prédits et les vérités terrain. L'évaluation en temps réel des boîtes détectées dans la couche supérieure de la palette montre que les coordonnées estimées ont une erreur de moins de 1cm sur tous les axes. Cette erreur est due au chevauchement des boîtes englobantes causé par l'orientation angulaire de la caméra. Cependant, cette erreur est attendue et peut être corrigée en utilisant cette distance pour initier la préhension. Comme l'objet (les boîtes) a une hauteur minimale de 20cm, l'erreur dans la direction z est négligeable. L'ensemble de la tâche est effectué dans un espace de travail non figé.

6. Conclusion

Cette recherche présente un système de dépalettisation automatisé à double bras humanoïde pour la saisie de boîtes en carton supervisé par l'IA. Le système surmonte les défis de la détection d'objets, de l'estimation de la pose et des algorithmes de dépalettisation pour les manipulateurs à double bras. Le système utilise une seule caméra RGB-D et YOLOv3 pour la détection d'objets et darknet 3D pour l'estimation de la pose. Le système peut détecter et estimer avec précision la pose des boîtes en carton, même dans des scénarios difficiles avec des boîtes partiellement visibles ou occultées et des conditions d'éclairage variables. Bien que la précision soit plus élevée pour les boîtes avec plusieurs faces visibles, le système est toujours capable de détecter les boîtes avec seulement une face visible. Le manipulateur à double bras peut saisir et manipuler efficacement les boîtes avec des stratégies de préhension optimales et des chemins appropriés. Le système génère également une base de données pour manipuler les poses complexes des boîtes de manière plus efficace dans les futurs processus de dépalettisation. Globalement, les performances du système sont fiables et précises, atteignant une précision de détection élevée même dans des conditions d'éclairage variables et avec des boîtes partiellement visibles. Cette recherche démontre le potentiel de l'automatisation alimentée par l'IA pour le processus de dépalettisation, offrant des avantages significatifs pour les industries qui traitent des marchandises palettisées.

Références

- Benali, K., Brethé, J.-F., Guerin, F., Gorka, M., and Sayah, A. (2019). A humanoid torso for palletization in logistics warehouses. In *12th International Doctoral Student Workshop on Logistics*.
- Caccavale, R., Arpentì, P., Paduano, G., Fontanelli, A., Lippiello, V., Villani, L., and Siciliano, B. (2020). A flexible robotic depalletizing system for supermarket logistics. *IEEE Robotics and Automation Letters*, 5(3) :4471–4476.
- Echelmeyer, W., Kirchheim, A., and Wellbrock, E. (2008). Robotics-logistics : Challenges for automation of logistic processes. In *2008 IEEE International Conference on Automation and Logistics*, pages 2099–2103. IEEE.

- Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European conference on information retrieval*, pages 345–359. Springer.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., and Ma, B. (2022). A review of yolo algorithm developments. *Procedia computer science*, 199 :1066–1073.
- Katsoulas, D., Bergen, L., and Tassakos, L. (2002). A versatile depalletizer of boxes based on range imagery. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 4313–4319. IEEE.
- Monica, R., Aleotti, J., and Rizzini, D. L. (2020). Detection of parcel boxes for pallet unloading using a 3d time-of-flight industrial sensor. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 314–318. IEEE.
- Nakamoto, H., Eto, H., Sonoura, T., Tanaka, J., and Ogawa, A. (2016). High-speed and compact depalletizing robot capable of handling packages stacked complicatedly. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 344–349. IEEE.
- Prasse, C., Skibinski, S., Weichert, F., Stenzel, J., Müller, H., and Ten Hompel, M. (2011). Concept of automated load detection for de-palletizing using depth images and rfid data. In *2011 IEEE International Conference on Control System, Computing and Engineering*, pages 249–254. IEEE.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Redmon, J. and Farhadi, A. (2018). Yolov3 : An incremental improvement. *arXiv preprint arXiv :1804.02767*.
- Reyes, E., Gómez, C., Norambuena, E., and Ruiz-del Solar, J. (2019). Near real-time object recognition for pepper based on deep neural networks running on a backpack. In *RoboCup 2018 : Robot World Cup XXII 22*, pages 287–298. Springer.
- Sachin, C., Ioan, S., and Steve, C. (2012). Moveit![ros topics]. *IEEE Robotics and Automation Magazine*, 19(1) :18–19.
- Sandström, R., Denny, J., and Amato, N. M. (2020). Asymptotically-optimal topological nearest-neighbor filtering. *IEEE Robotics and Automation Letters*, 5(4) :6916–6923.
- Schwarz, M., Milan, A., Periyasamy, A. S., and Behnke, S. (2018). Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter. *The International Journal of Robotics Research*, 37(4-5) :437–451.
- Sebbata, W., BRETHÉ, J.-F., et al. (2020). An adaptive robotic grasping with a 2-finger gripper based on deep learning network. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 620–627. IEEE.
- Vohra, M., Prakash, R., and Behera, L. (2021). Edge and corner detection in unorganized point clouds for robotic pick and place applications. *arXiv preprint arXiv :2104.09099*.

Biographie des auteurs



Santheep Yesudasu est doctorant en robotique à l'Université Le Havre Normandie au laboratoire GREAH (Groupe de Recherche en Électrotechnique et Automatique du Havre). Ses recherches se concentrent sur le développement de l'automatisation basée sur l'IA pour améliorer la détection et l'estimation de pose des robots humanoïdes dans les applications logistiques. Depuis 2023, il travaille également comme ingénieur de recherche au laboratoire CRISTAL UMR CNRS 9189, associé à l'Université de Lille.



Jean-François Brethé est Professeur des Universités en robotique à l'Université Le Havre Normandie, Institut Supérieur d'Études Logistiques (ISEL) et membre du GREAH (Groupe de Recherche en Électrotechnique et Automatique du Havre). Ses travaux de recherche portent initialement sur la modélisation, la conception et l'étude des performances en précision des robots manipulateurs industriels, puis se focalisent aujourd'hui sur les verrous scientifiques posés par l'introduction de cobots dans les entrepôts logistiques (préparation de commandes robotisée par robot manipulateur mobile, saisie de colis de taille variable par torse humanoïde).



Wafae SEBBATA est docteure en robotique. Sa thèse de doctorat, soutenue en octobre 2021, intitulée "Étude et développement de tâches de manipulation mobile dans le contexte de l'industrie 4.0", a été réalisée au sein du laboratoire GREAH (Groupe de Recherche en Électrotechnique et Automatique du Havre), Université Le Havre Normandie, sous la direction du Pr Jean-François Brethé. Elle a travaillé comme ingénieure d'études en robotique dans le cadre du projet CoRoT de 2017 à 2022. Aujourd'hui, elle exerce comme enseignante à l'école d'ingénieurs ECE-Paris "École Centrale d'Électronique".



Patrick Bonnin est Professeur des Universités à l'ISTY (Institut des Sciences et Techniques des Yvelines), Ecole d'Ingénieurs de l'Université de Versailles Saint Quentin, au Département Systèmes Electroniques Embarqués Il effectue sa recherche au LISV (Laboratoire d'Ingénierie des Systèmes de Versailles) dans l'équipe Robotique Interactive. Ses thématiques de recherche sont au cœur des problématiques de vision temps réel dans les systèmes embarqués.

TASK ALLOCATION ALGORITHMS FOR MOBILE ROBOTS, APPLICATION TO INDUSTRIAL LOGISTICS

Arnaud BELHOMME¹, François GUERIN²

¹GREAH, University Le Havre Normandie
Le Havre, France
arnaud.belhomme@univ-lehavre.fr

²GREAH, University Le Havre Normandie
Le Havre, France
francois.guerin@univ-lehavre.fr

RESUME

Dans cette publication, il est exploré les algorithmes d'allocation de tâches pour les robots mobiles appliqués à la logistique industrielle. Nous y examinons les systèmes de décision centralisés et décentralisés, selon leur flexibilité, robustesse et évolutivité. L'étude se concentre sur l'optimisation de l'allocation des tâches dans un système multi-robots en utilisant diverses stratégies telles que la programmation linéaire, le calcul itératif, la résolution de conflits par consensus, les modèles d'apprentissage automatique et l'apprentissage par renforcement. Les méthodes proposées sont évaluées par des simulations dans un environnement industriel, notamment au sein d'une entreprise d'assemblage de nacelles de moteurs d'avion. L'objectif est d'améliorer l'efficacité en permettant aux robots d'assigner des tâches de manière autonome en fonction de leurs capacités et contraintes. La recherche se conclut par une comparaison des performances des algorithmes, offrant des perspectives pour de futures applications dans l'automatisation industrielle.

MOTS CLÉS: Allocation de tâches, Robots mobiles, Logistique industrielle.

ABSTRACT

This publication explores task allocation algorithms for mobile robots in industrial logistics. It examines both centralised and decentralised control methods, according to their flexibility, robustness, and scalability. The study focuses on optimising task allocation in a multi-robot system using different strategies such as linear programming, iterative calculation, conflict resolution by consensus, machine learning models, and reinforcement learning. The proposed methods are evaluated through simulations in an industrial setting, specifically within an aircraft engine nacelle assembly company. The goal is to enhance efficiency by allowing robots to autonomously assign tasks based on their capabilities and constraints. The research concludes with a comparison of the algorithms' performance, providing insights for future applications in industrial automation.

KEYWORDS: Task Allocation, Mobile Robots, Industrial Logistics.

1. Introduction

In recent years, industrial enterprises have constantly sought the development of their products, processes and organisation, with the aim of maintaining a competitive advantage. In this context, industrial logistics provides essential support which makes it possible to optimise the supply of tools and raw materials to production lines, Shang (2022). Mobile robots form a large and important part of the logistics transportation systems in today's industry are widely used. Mobile robots offered by global manufacturers almost all operate under some form of centralised control where a single central controller coordinates the entire fleet of mobile robots. There is a trend towards decentralised control of these systems where mobile robots make individual decisions that promote the flexibility, robustness and scalability of transport, De Ryck et al. (2021).

Task allocation can be done in both a centralised or decentralised way. In many situations, due to the lack of infrastructure, higher possibility of single-point-of-failure, communication bottlenecks, loss of communication with the central node, etc., centralised task allocation becomes impractical. Especially the applications like war-field, disastrous scenario, underwater exploration where any infrastructure is not available, centralised allocation of tasks is not feasible, Mahato et al. (2023). Multi-agent systems are decentralised control systems where the autonomous agents in the system coordinate and cooperate with each other in order to achieve a global goal, Teck et al. (2023). In recent years, there has been a growing association between robotics and artificial intelligence, aiming to enable robots to make autonomous decisions, Sharma et al. (2023).

In a traditional decentralised task allocation strategy, first, a leader is elected, which oversees the overall task allocation. Such strategies will also have similar problems as a centralised solution in hostile scenarios. It may ultimately lead to the frequent election of the leaders, wasting time and energy. Decentralised task allocation without the intervention of a dedicated leader has also been considered in methods following various bidding-based schemes. Given a set of tasks, a robot can derive a bid value indicating its suitability in carrying out each of the tasks. Optimal distributed allocation of the tasks can be ensured using a simple two-step algorithm. In the first step, the robots would exchange the bid-values for each task with each of the robots. Next, each robot can run the task allocation algorithm locally as all of them possess the same input (bid value). This algorithm involves broadcast data sharing, which is a very costly operation in terms of the amount of communication needed in the underlying network, Mahato et al. (2023).

Multi-Robot Task Allocation (MRTA) can be solved by various optimisation approaches such as Integer Linear Program (ILP), heuristics, and meta-heuristics algorithms, Chakraa et al. (2023). Hungarian algorithm: A decentralised task allocation algorithm based on the Hungarian approach is proposed in Ismail and Sun (2017). The proposed algorithm guarantees an optimal solution as long as the agent network is connected, i.e., the second smallest eigenvalue of the Laplacian matrix of the agent graph is greater than zero. Patel et al. (2020) propose an approach for decentralised task allocation based on a decentralised Genetic Algorithm (GA). The approach parallelizes a genetic algorithm across the team of agents, making efficient use of their computational resources. Buckman (2018) presents a fully decentralised, dynamic task allocation algorithm that extends the Consensus-Based Bundle Algorithm (CBBA) to allow for allocating new tasks. Nayak et al. (2020) compare the performance of five state of the art decentralised task allocation algorithms under imperfect communication conditions. The decentralised algorithms we consider are CBAA (Consensus Based Bundle Algorithm), ACBBA (Asynchronous Consensus Based Bundle Algorithm), DHBA (Decentralised Hungarian Based Algorithm), HIPC (Hybrid Information and

Plan Consensus) and PI (Performance Impact). A decentralised allocation algorithm for distributed supply chains with critical tasks: Binetti et al. (2013) considers the problem of allocating tasks to a network of interconnected nodes in a supply chain, considering functional heterogeneity, resource constraints, and critical tasks whose assignment has to be considered mandatory. The proposed approach is an auction-based algorithm which uses a consensus algorithm to obtain a conflict-free solution fulfilling all the constraints. Numerical simulations and a comparison with a centralised optimisation algorithm are performed to evaluate the effectiveness of the proposed approach. With recent advances in mobile robotics, autonomous systems, and artificial intelligence, there is a growing expectation that robots are able to solve complex problems. Many of these problems require multiple robots working cooperatively in a multi-robot system. Complex tasks may also include the interconnection of task-level specifications with robot motion-level constraints. Many recent works in the literature use multiple mobile robots to solve these complex tasks by integrating task and motion planning, Antonyshyn et al. (2023).

In our work, we propose to compare different families of task allocation algorithms, with the aim of identifying the most relevant ones which will be applied to an example of an industrial case.

2. Problem description

We consider a fleet of R mobile robots, with N tasks to complete. Each robot is endowed with unique abilities that are relevant to the completion of these tasks. All robots can complete the tasks. The objective is to develop an embedded algorithm for each robot allowing it to assign itself to the task for which it is most competent, due to battery level (state of charge) and time (distance, speed). Then provide a consensus mechanism in the event of a task assignment conflict between several robots.

2.1 Industrial Application Example

The solution proposed in this paper will be implemented in an aircraft engine nacelle assembly company, as part of a line-side supply automation project.

Consider the example in figure 1 with 3 robots and 5 workstations. Each operator has a replenishment request push button for his workstation. The robot fleet is then responsible for managing the replenishment of component bins at various workstations. The supply of full bins (using a First In First Out table) is carried out by an operator, as well as the evacuation of empty bins.

For the purpose of a modelization (figure 2), let us consider the following notations :

- " N " operator workstations ($N \geq 1$), with 2 parts:
 - 1 : full bins => capacity = $C_{oF}(1)$
 - 2 : empty bins => capacity = $C_{oE}(1)$
- " R " mobile robots ($R \geq 1$)
- S_F : full bin station => capacity = $C_F(25)$
- S_E : empty bin station => capacity = $C_E(25)$

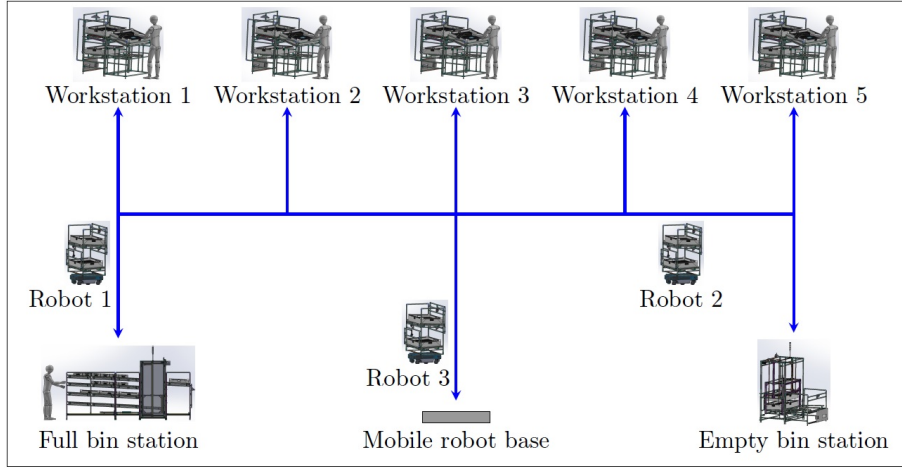


Figure 1: Industrial application (example for 3 robots and 5 workstations).

- O_{T_1}, \dots, O_{T_N} : workstation operators
- O_F : full bin station operator
- O_E : empty bin station operator
- Each workstation and station is equipped with a “level sensor” connected to an IoT network.
- Each mobile robot can transport “ B_{R_1}, \dots, B_{R_N} ” bins and perform actions:
 - GoS_F = go to full bin station + bin recovery
 - $GoO_{F_1}, \dots, GoO_{F_N}$ = go to the "full" operator workstation 1, ... N + bin deposit
 - $GoO_{E_1}, \dots, GoO_{E_N}$ = go to the "empty" operator workstation 1, ... N + bin recovery
 - GoS_E = go to empty bin station + bin deposit
 - $GoBase$ = return to base (battery charge)
- The navigation function of mobile robots is supposed to be perfect.

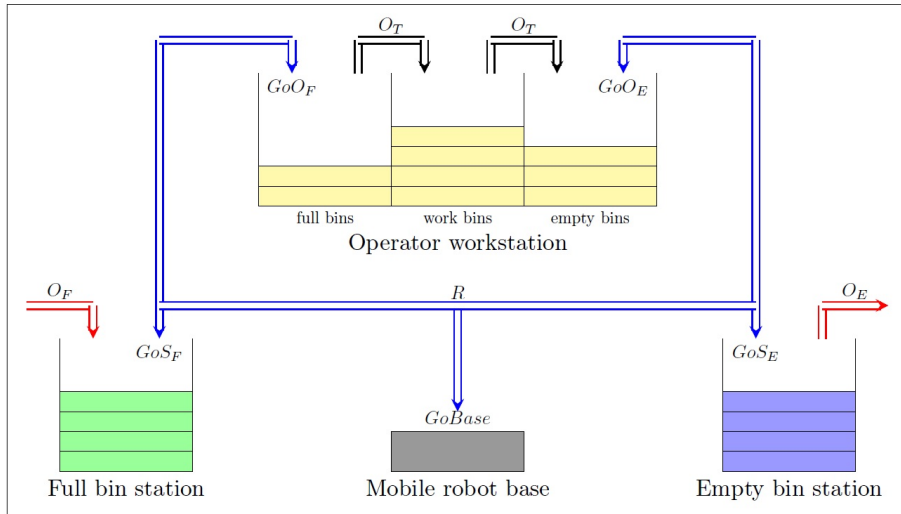


Figure 2: Workstation modelization (one workstation represented).

3. Contribution

3.1 Formulation of the Problem

Consider a set of robots $R = \{r_1, \dots, r_n\}$ and a set of tasks $T = \{t_1, \dots, t_m\}$. Each robot r_i has attributes including position p_i , battery level b_i , speed s_i , and payload capacity c_i .

3.2 Algorithms developed

We chose to implement different task allocation algorithms in order to compare them, both in terms of efficiency, but also in calculation time. Efficiency is measured by calculating the sum of the task capacities assigned to the different robots. The sum of the greatest capacity then corresponds to the best overall task allocation efficiency.

The algorithm 1 (Task Allocation with Mixed-Integer Linear Programming - MILP) uses linear programming to maximise task allocation capacities by assigning each robot exactly one task and each task to exactly one robot, subject to the constraints of unique allocation and binary decision variables.

The algorithm 2 (Task Allocation with Iterative Calculation) iteratively assigns tasks to robots using random permutations to maximize the sum of capacities, updating the best allocation found whenever a higher sum of capacities is achieved.

The algorithm 3 (Task Allocation with Conflict Resolution) initially assigns tasks to robots by selecting the task with the maximum capacity for each robot, checks for assignment conflicts, and if conflicts exist, iteratively solves them by randomly permuting task assignments to maximise the sum of capacities.

The algorithm 4 (Task Allocation with Conflict Resolution and Consensus) assigns tasks to robots by selecting the maximum capacity task for each robot, checks for assignment conflicts, and if conflicts exist, uses a local consensus method to resolve them by selecting the robot with the lowest capacity in conflicts or randomly if tied, and marking its task allocation for reassignment.

The algorithm 5 (Task Allocation Based on Ranking) ranks robot-task pairs based on a custom scoring criterion, sorts these pairs by score, and then allocates tasks to robots sequentially based on this ranking, ensuring each robot and task is assigned only once.

The algorithm 6 (Task Allocation based on a Linear Regression Machine Learning Model) imports a dataset from an Excel file, splits it into training and testing sets, trains a linear regression model on the training data, evaluates the model on the test data, and then uses the trained model to predict outputs based on a given input matrix.

The algorithm 7 (Task Allocation based on Machine Learning with Perceptron Model) iteratively trains a Perceptron model for each robot using a dataset imported from an Excel file, evaluates the model's performance, and uses it to predict task allocations based on given capacity data, storing the predictions in a task allocation matrix.

The algorithm 8 (Task Allocation based on a Reinforcement Learning Model with a Q-Learning Method) employs Q-learning with an epsilon-greedy strategy over multiple episodes to optimise task allocation for robots, updating a Q-matrix based on the total capacity achieved in each episode and ultimately determining the best task assignment for each robot.

Algorithm 1 Task Allocation with Mixed-Integer Linear Programming (MILP)

INITIALISATION:

Import linear programming solver
Create the objective coefficients (f) to maximise the sum of capacities

Initialise integer variables for task allocation ($intcon$)
Initialise equality constraints matrix (Aeq) with zeros
Initialise equality constraints bounds (beq) with ones for unique allocation constraint

CALCULATION:

```
for each robot  $i$  do
    Set sum constraints in  $Aeq$  for each robot
end for
for each task  $j$  do
    Set sum constraints in  $Aeq$  for each task
end for
Set bounds for each variable representing task allocation to a robot (0 or 1)
Solve the MILP problem with linprog
OUTPUTS:
Reshape the result to match the number of robots and tasks
for each robot  $i$  do
    Print the allocated task for the robot
end for
```

Algorithm 2 Task Allocation with Iterative Calculation

INITIALISATION:

Initialise number of iterations
Initialise TaskAllocation matrix

CALCULATION:

```
for each iteration do
    Randomly permute the tasks and reshape to (numTasks, 1)
    for TaskAllocation
    Set NewCapaSum to 0
    for each robot  $i$  do
        Add the capacity of robot  $i$  for its allocated task to NewCapaSum
    end for
    if NewCapaSum is greater than CapaSum then
        Copy TaskAllocation to NewTaskAllocation
        Set CapaSum to NewCapaSum
    end if
end for
OUTPUTS:
for each robot  $i$  do
    Print the allocated task for the robot ( $R\{i + 1\}$  ->  $T\{int(NewTaskAllocation[i]) + 1\}$ )
end for
```

Algorithm 3 Task Allocation with Conflict Resolution

INITIALISATION:

Initialise number of iterations
Initialise TaskAllocation matrix

CALCULATION:

```
for each robot  $i$  do
    Find the maximum value in the capacity of robot  $i$ 
    Find the positions of the maximum value
    if more than one maximum value (conflict) then
        Randomly choose one of the maximum positions
    else
        Directly assign the task
    end if
end for
if all tasks are uniquely allocated then
    Copy TaskAllocation to NewTaskAllocation
else
    for each iteration do
        Randomly permute the tasks and reshape to (numTasks, 1) for TaskAllocation
        Set NewCapaSum to 0
        for each robot  $i$  do
            Add the capacity of robot  $i$  for its allocated task to NewCapaSum
        end for
        if NewCapaSum is greater than CapaSum then
            Copy TaskAllocation to NewTaskAllocation
            Set CapaSum to NewCapaSum
        end if
    end for
end if
OUTPUTS:
for each robot  $i$  do
    Print the allocated task for the robot ( $R\{i + 1\}$  ->  $T\{int(NewTaskAllocation[i]) + 1\}$ )
end for
```

Algorithm 4 Task Allocation with Conflict Resolution and Consensus

INITIALISATION:

Initialise number of iterations
Initialise TaskAllocation matrix

CALCULATION:

for each robot i **do**

 Find the maximum value in the capacity of robot i
 Find the positions of the maximum value
 if more than one maximum value (conflict) **then**
 Randomly choose one of the maximum positions
 else

 Directly assign the task

end if

end for

if all tasks are uniquely allocated **then**

 Directly assign the task

else

for each robot i **do**

 communicate capability $K_i(t_j)$ for each task t_j to other robots for contested tasks

 the robot with the highest $K_i(t_j)$ is assigned the task.

 in case of equal capabilities, the robot that communicated first is assigned the task

end for

end if

OUTPUTS:

for each robot i **do**

 Set NewTaskAllocation[i] to consensus[i , 0]

if NewTaskAllocation[i] == -1 **then**

 Print the robot has no task (R{ $i + 1$ } ->)

else

 Print the allocated task for the robot (R{ $i + 1$ } -> T{int(NewTaskAllocation[i] + 1)})

end if

end for

Algorithm 5 Task Allocation Based on Ranking

INITIALISATION:

Initialise TaskAllocationFG matrix with zeros of size (numRobots, numTasks)

Initialise an empty list for ranking (classement)

CALCULATION:

for each robot i **do**

for each task j **do**

 Calculate score as sum of capacities for robot i and task j , adjusted by capacity of robot i for task j

 Append ($i, j, score$) to ranking list (classement)

end for

end for

Sort ranking list (classement) by score in ascending order

Initialise allocated_tasks with zeros of size (numTasks)

Initialise allocated_robots with zeros of size (numRobots)

for each ($robot, task, score$) in ranking list (classement) **do**

if allocated_robots[robot] == 0 and allocated_tasks[task] == 0 **then**

 Allocate task to robot in TaskAllocationFG

 Mark robot as allocated in allocated_robots

 Mark task as allocated in allocated_tasks

if sum of allocated tasks equals numTasks **then**

 Break

end if

end if

end for

OUTPUTS:

for each robot i **do**

 Print the allocated task for the robot (R{ $i + 1$ } -> T{task number allocated to robot i in TaskAllocationFG})

end for

Algorithm 6 Task Allocation based on a Linear Regression Machine Learning Model

INITIALISATION:

Import train_test_split and LinearRegression from sklearn

Set nbSet

CALCULATION:

Read the first columns from 'DataSet.xlsx' into X_data with nbSet rows

Convert X_data to a numpy array

Read from 'DataSet.xlsx' into y_data with nbSet rows

Convert y_data to a numpy array

Split X_data and y_data into X_train, X_test, y_train, and y_test with 80% training and 20% testing data

Initialise model as LinearRegression

Fit model with X_train and y_train

Calculate score of model on X_test and y_test

Reshape capacity to (1, 25) and store in capacities

Predict y_pred from capacities using model

Round y_pred to the nearest integers

Convert y_pred to integer type

OUTPUTS:

for each robot i from 0 to numRobots - 1 **do**

 Print "R{ $i + 1$ } -> T{y_pred[0][i]}"

end for

Algorithm 7 Task Allocation based on a Machine Learning Model with Perceptron

INITIALISATION:

```
Import numpy as np
Import train_test_split and Perceptron from sklearn
Initialise TaskAllocationML2 matrix with zeros of size (numRobots, 1)
```

```
Set nbSet to 1000
```

CALCULATION:

```
for each robot  $i$  from 1 to numRobots do
```

```
  Read the first columns from 'DataSet.xlsx' into X_data with nbSet rows
```

```
  Convert X_data to a numpy array
```

```
  Read columns from 'DataSet.xlsx' into y_data with nbSet rows
```

```
  Convert y_data to a numpy array and flatten it
```

```
  Split X_data and y_data into X_train, X_test, y_train, and y_test with 80% training and 20% testing data
```

```
  Initialise model as Perceptron with max_iter=40, eta0=0.1, random_state=0
```

```
  Fit model with X_train and y_train
```

```
  Calculate score of model on X_test and y_test
```

```
  Reshape capacity to and store in capacities
```

```
  Predict y_pred from capacities using model
```

```
  Round y_pred to the nearest integers
```

```
  Convert y_pred to integer type
```

```
  Store y_pred in TaskAllocationML2 at index  $i - 1$ 
```

```
end for
```

OUTPUTS:

```
for each robot  $i$  from 0 to numRobots - 1 do
```

```
  Print "R{ $i + 1$ } -> T{int(TaskAllocationML2[ $i$ ])}"
```

```
end for
```

Algorithm 8 Task Allocation based on a Reinforcement Learning Model with a Q-Learning Method

INITIALISATION:

```
Set num_episodes
```

```
Set learning_rate
```

```
Set discount_factor
```

```
Set exploration_prob
```

CALCULATION:

```
# Function to choose an action (task) with epsilon-greedy
```

choose_action(robot)

```
if random number < exploration_prob then
```

```
  Return a random task from numTasks
```

```
else
```

```
  Return the task with the highest Q value for the given robot
```

```
end if
```

```
# Simulation of episodes
```

```
for each episode from 0 to num_episodes - 1 do
```

```
  Initialise total_capacity to 0
```

```
  Assign tasks randomly to task_assignment
```

```
  for each robot from 0 to numRobots - 1 do
```

```
    Set action to the assigned task for the robot
```

```
    Get capacity2 from robot_capacities for the robot and action
```

```
    Add capacity2 to total_capacity
```

```
    Set reward to total_capacity
```

```
    Update Q value for the robot and action using the formula:
```

$$Q[robot, action] = Q[robot, action] + learning_rate * (reward - Q[robot, action])$$

```
  end for
```

```
end for
```

```
# Find the best task assignment for each robot
```

```
Set best_task_assignment to the task with the highest Q value for each robot
```

OUTPUTS:

```
for each robot  $i$  from 0 to numRobots - 1 do
```

```
  Print "R{ $i + 1$ } -> T{int(best_task_assignment[ $i$ ] + 1)}"
```

```
end for
```

3.3 Comparison and results

In order to compare the different algorithms, we tested them using a set of capacities determined randomly (for each of the tasks carried out by each robot). We then repeated the test on a large number of iterations to identify a trend and make a reliable comparison of the different algorithms.

The algorithm 9 shows how we conducted this comparison. Table 1 summarizes the average task capacities and processing times for five robots and tasks after 1000 iterations. Figure 3 shows the average capacities (y-axis) for each algorithm (x-axis), while Figure 4 depicts their average execution times (y-axis). Both highlight performance differences under the 1000-iteration stopping criterion.

We conclude that Algorithm 8 is the most efficient based on the criterion of the sum of capacities, while Algorithm 5 excels in processing time efficiency. However, for a decentralised approach, Algorithm 4 is the most efficient one, in both sum of capacities and running time performance.

Algorithm 9 Robot Task Allocation

INITIALISATION:

```
numRobots ← 5  
numTasks ← 5  
nbitmax ← 1000  
out put ← zeros(nbitmax, 8 × 7)
```

CALCULATION:

```
for nbit ← 0 to nbitmax - 1 do  
  capacity ← random between 0 and 1 for each robot-task pair  
  Display the capacity table  
  print "Robot Abilities:"  
  print T  
  start ← current time  
  for algo ← 1 to 8 do  
    Run algorithm  
    duration ← round(current time - start, 3)  
    Write sum of capacities duration to output matrix  
  end for  
end for
```

end for**OUTPUTS:**

```
print out put
```

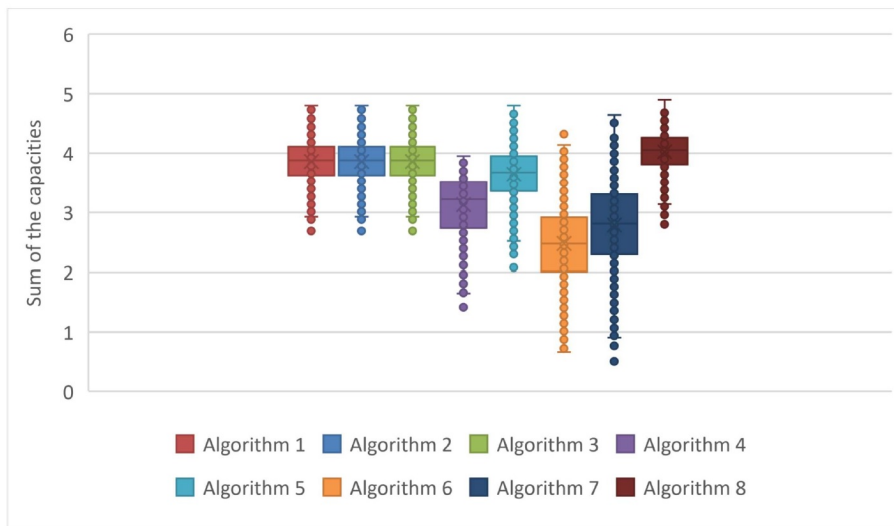


Figure 3: Results of the comparison of algorithms according to calculation of the sum of capacities.

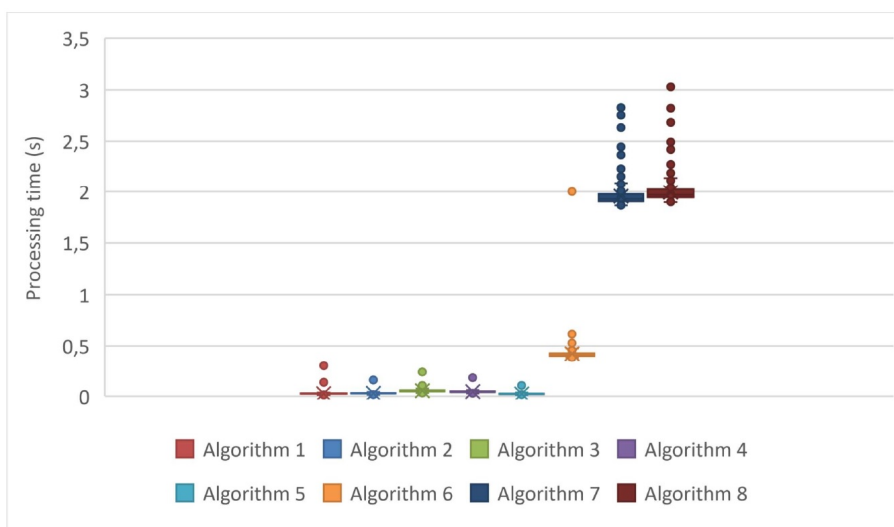


Figure 4: Results of the comparison of algorithms according to execution times.

	Control method	Average of the sums of the capacities	Average processing time (s)
Algorithm 1	centralised	3,86	0,026
Algorithm 2	centralised	3,86	0,029
Algorithm 3	decentralised & centralised	3,86	0,053
Algorithm 4	decentralised	3,13	0,045
Algorithm 5	decentralised & centralised	3,65	0,024
Algorithm 6	centralised	2,48	0,423
Algorithm 7	centralised	2,79	1,954
Algorithm 8	centralised	4,02	1,995

Table 1: Results of comparing algorithms.

3.4 Experimental results

To evaluate our chosen task allocation algorithm, we employed three ROS-based mobile robots. We replicated the industrial application scenario in a laboratory setting, constructing a model within a room equipped with replicas of various workstations. Each robot was loaded with the same program, enabling decentralised operation. All results can be found in Belhomme and Guerin (2024) publication.

4. Conclusions and perspectives

By evaluating various strategies—including linear programming, iterative calculation, conflict resolution, machine learning models, and reinforcement learning—the study identifies the advantages and challenges of each method. The proposed algorithms were tested in a simulated industrial environment, specifically within an aircraft engine nacelle assembly company, demonstrating their potential to enhance operational efficiency. By allowing robots to autonomously assign tasks based on their capabilities and constraints, the research provides valuable insights for improving task allocation processes in industrial automation. The comparative analysis of algorithm performance offers a foundation for future developments in this field, aiming to increase flexibility, robustness, and scalability in industrial logistics systems. The findings underscore the importance of adopting advanced control strategies to optimise multi-robot systems in dynamic and complex industrial settings.

Future research could focus on integrating advanced machine learning techniques, robust communication protocols, and hybrid control systems to enhance task allocation for mobile robots in diverse industrial settings, emphasising energy efficiency, human-robot collaboration, and real-world implementation.

Acknowledgement

We would like to thank the company Safran Nacelles, located in Gonfreville L’Orcher France, for trusting us with the implementation of this project.

References

- Antonyshyn, L., Silveira, J., Givigi, S., and Marshall, J. (2023). Multiple Mobile Robot Task and Motion Planning: A Survey. *ACM Computing Surveys*, 55(10):213:1–213:35.
- Belhomme, A. and Guerin, F. (2024). Decentralised Bio-inspired Task Allocation for Mobile Robots, Application to Industrial Logistics. In *10th International Conference (IEEE-IFAC) on Control, Decision and Information Technologies (CoDIT 2024) July 01-04, 2024, Valetta (Malta)*.
- Binetti, G., Naso, D., and Turchiano, B. (2013). A decentralized allocation algorithm for distributed supply chains with critical tasks. *IFAC Proceedings Volumes*, 46(9):192–197.
- Buckman, N. N. M. . (2018). *Decentralized task allocation for dynamic, time-sensitive tasks*. Thesis, Massachusetts Institute of Technology. Accepted: 2019-02-05T15:17:44Z.
- Chakraa, H., Guérin, F., Leclercq, E., and Lefebvre, D. (2023). Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art. *Robotics and Autonomous Systems*, 168:104492.
- De Ryck, M., Pissoort, D., Holvoet, T., and Demeester, E. (2021). Decentral task allocation for industrial AGV-systems with resource constraints. *Journal of Manufacturing Systems*, 59:310–319.
- Ismail, S. and Sun, L. (2017). Decentralized hungarian-based approach for fast and scalable task allocation. pages 23–28.
- Mahato, P., Saha, S., Sarkar, C., and Shaghil, M. (2023). Consensus-based fast and energy-efficient multi-robot task allocation. *Robotics and Autonomous Systems*, 159:104270.
- Nayak, S., Yeotikar, S., Carrillo, E., Rudnick-Cohen, E., Jaffar, M. K. M., Patel, R., Azarm, S., Herrmann, J. W., Xu, H., and Otte, M. (2020). Experimental Comparison of Decentralized Task Allocation Algorithms Under Imperfect Communication. *IEEE Robotics and Automation Letters*, 5(2):572–579.
- Patel, R., Rudnick-Cohen, E., Azarm, S., Otte, M., Xu, H., and Herrmann, J. W. (2020). Decentralized Task Allocation in Multi-Agent Systems Using a Decentralized Genetic Algorithm. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3770–3776. Conference Name: 2020 IEEE International Conference on Robotics and Automation (ICRA) ISBN: 9781728173955 Place: Paris, France Publisher: IEEE.
- Shang, Y. (2022). Design of an Integrated Approach to Industrial Logistics Information Based on Supply Chain Management. *Advances in Multimedia*, 2022:1–8.
- Sharma, N., Pandey, J. K., and Mondal, S. (2023). A Review of Mobile Robots: Applications and Future Prospect. *International Journal of Precision Engineering and Manufacturing*, 24(9):1695–1706.
- Teck, S., Vansteenwegen, P., and Dewil, R. (2023). An efficient multi-agent approach to order picking and robot scheduling in a robotic mobile fulfillment system. *Simulation Modelling Practice and Theory*, 127:102789.

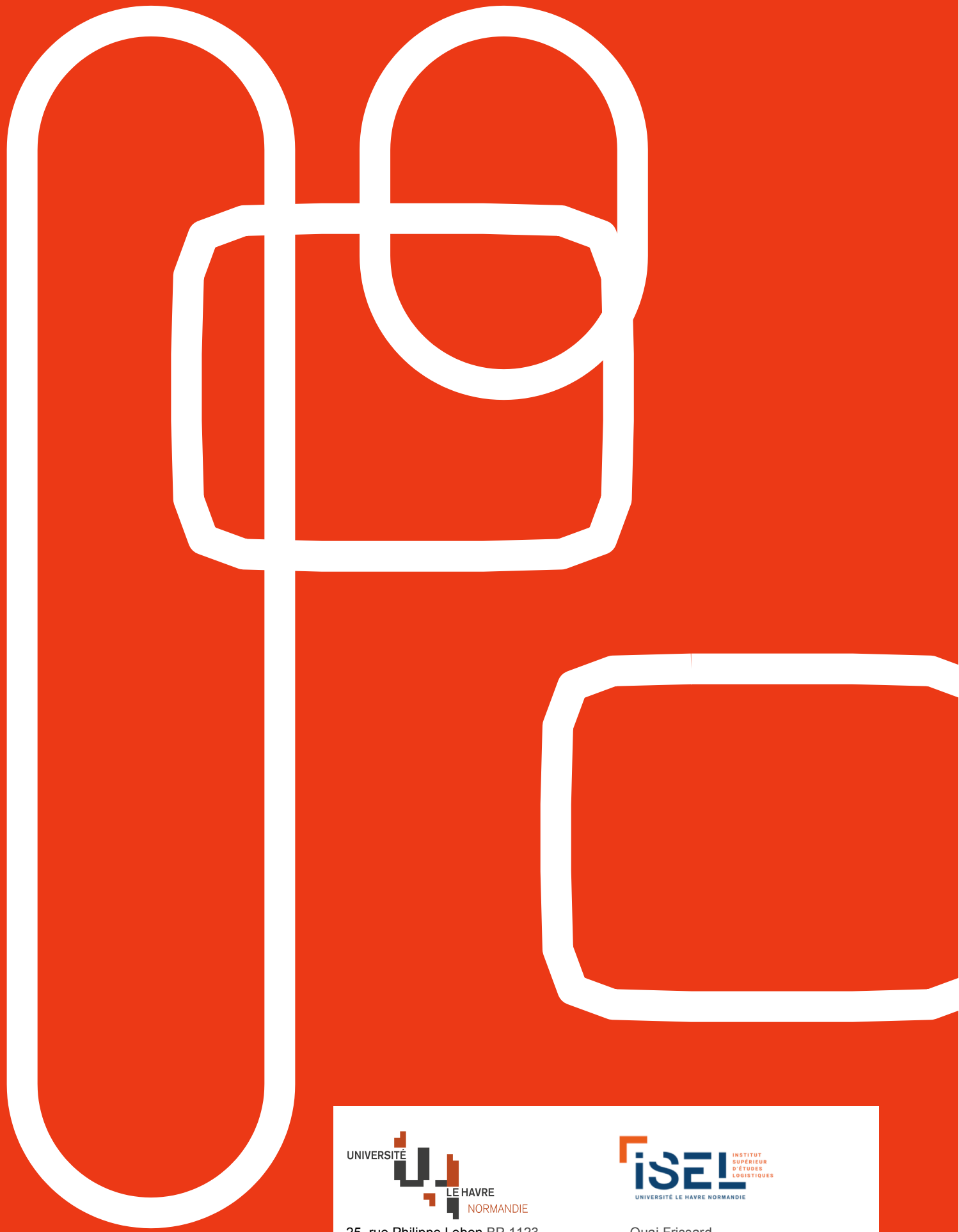
Author biography



Arnaud Belhomme began his career in 1996 as a mechanical engineering teacher. In 2014, he expanded his teaching scope to include robotics and operations management at the University of Le Havre, France, in the department of the higher institute of logistics studies (ISEL). In 2021, pursuing further academic growth, Mr Belhomme joined the Research Group on Electrical Engineering and Automatic Control (GREAH) as a Ph.D. student. His research focuses on the innovative hybrid control of a fleet of land and air mobile robots, specifically targeting industrial logistics applications. This work highlights his commitment to integrating advanced robotics into practical industry solutions.



Dr François Guérin got the "Agrégation" in Electrical Engineering (Electronics), an exam to access to civil service in the French public education system. Then, he obtained his Ph.D (2004) and his Habilitation Degree (2015) in Robotics and Automatic Control from Le Havre Normandy University. Currently, he works as Associate Professor within the Higher Institute of Logistics Studies (ISEL) and the Research Group in Electrotechnics and Automatic Control (GREAH) of Le Havre Normandy University. Dr François Guérin teaches Automatic Control, Robotics and Electronics (IoT) and focuses his researches on Advanced Automatic Control applied to Mobile Robotics. From 2021, he is also the President of the Normandy Drone Innovation Center (CIDN).



UNIVERSITÉ

LE HAVRE
NORMANDIE

25, rue Philippe Lebon BP 1123
76063 Le Havre Cedex France


INSTITUT
SUPÉRIEUR
D'ÉTUDES
LOGISTIQUES
UNIVERSITÉ LE HAVRE NORMANDIE

Quai Frissard
76600 Le Havre