

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd - Tlemcen -
Faculté de TECHNOLOGIE



MEMOIRE

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

En : Automatique

Spécialité : Automatique et Informatique Industrielle

Par : REZOUG Inas
SEKKAL Mohammed Hebri

Sujet

Outdoor Robot localization

Soutenu publiquement, le / 09 / 2023, devant le jury composé de :

Mme.BENMANSOUR Souhila
Mme.BENREZZAK Nawel
M. BENSALAH Choukri

MCB
MCB
MCB

Université de Tlemcen
Université de Tlemcen
Université de Tlemcen

Président
Examinatrice
Encadrant

Année universitaire : 2022 /2023

Dedicaces

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

حمداً لله الذي وفقني لإنهاء مساري الدراسي رغم كل المشاق والصعوبات التي واجهتها. فشكراً لله أولاً، ثانياً أودّ أن أشكر والديّ، أمي التي كانت سنداً ورفيقاً دائماً لي طوال حياتي، أبي الذي ساندني و دعمني ، وإلى إخوتي الذين قدموا لي دعماً بطريقتهم الفريدة من نوعها. وأخيراً ليس آخرًا، أودّ أن أشكر نفسي التي نجحتُ من أجلها رغم العقبات الصعبة.
ربما وُجد الظلام لإِراحةِ العين من نور ساطع

رزوق إيناس

d'abord on veut remercier nos chère parents, qui m'ont soutenu et encouragé tout au long de mon parcours académique. Vos sacrifices, votre amour et votre confiance ont été nos sources d'inspiration et de motivation les plus importantes.

À nos frères et sœurs, pour leur soutien indéfectible et leurs encouragements constants. Vos conseils avisés et votre présence à notre côtés ont été d'une grande aide pour moi

À nos amis, pour leur soutien, leurs encouragements et leurs mots d'encouragement. Votre amitié a été un pilier important dans ma vie et a contribué à mon bien-être mental et émotionnel.

À nos professeurs et notre encadreur, pour leur enseignement, leur expertise et leur patience tout au long de notre parcours académique. Vos conseils et votre soutien ont été essentiels dans la réalisation de ce travail.

Enfin, à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, je leur exprime ma gratitude sincère et mes remerciements les plus chaleureux.

sekhal habri

Remerciements

Tout d'abord, nous souhaitons exprimer notre profonde gratitude envers le Divin, le Tout-Puissant et le Miséricordieux, qui nous a donné la force, la patience et la détermination nécessaires pour mener à bien ce modeste projet.

Nous aimerions également remercier notre encadreur Monsieur Bensalah Choukri, pour ses conseils inestimables et sa disponibilité infailible tout au long de l'élaboration de ce travail. Sa précieuse orientation, sa confiance indéfectible, son soutien désintéressé ainsi que le temps généreusement accordé méritent une mention spéciale. Nous exprimons ainsi notre profonde reconnaissance envers lui.

Nos chaleureux remerciements sont également destinés aux membres du comité d'évaluation pour l'intérêt qu'ils ont manifesté en acceptant d'évaluer notre travail et de le bonifier par leurs suggestions constructives. Nous souhaitons exprimer nos sincères remerciements à l'ensemble des enseignants du département de technologie de la faculté des sciences à Tlemcen.

Enfin, nous tenons également à remercier toutes les personnes qui ont apporté leur contribution quelle que soit, qui ont participé de près ou de loin à la réalisation de ce travail.

Résumé

La localisation précise des robots dans des environnements extérieurs représente un défi majeur en raison de la nature dynamique et imprévisible de ces paramètres. Les robots doivent faire face à des complexités telles que les changements de conditions d'éclairage, les terrains irréguliers et la présence d'obstacles dynamiques, des éléments pour lesquels les méthodes de localisation traditionnelles peinent à fournir une position précise dans de tels scénarios.

À travers notre projet, nous avons proposé l'utilisation de techniques de localisation basées sur le flux optique, qui exploitent les informations visuelles pour estimer le mouvement du robot par rapport à l'environnement. L'adaptabilité du flux optique en temps réel et sa résistance aux variations d'éclairage en font une stratégie prometteuse pour la localisation en extérieur.

Cependant, il est essentiel de reconnaître que la localisation basée sur le flux optique présente des limites, telles que sa sensibilité aux occlusions et ses difficultés avec les scènes complexes. En réponse, notre projet intègre une unité de mesure inertielle (IMU) pour renforcer le système. Cette intégration fournit les données d'orientation du robot, améliorant ainsi la précision et la robustesse de notre approche proposée.

Mots clés

Robot mobile, Localisation, Milieu , Vision par ordinateur, Filtrage de Kalman.

Abstract

The accurate localization of robots in outdoor environments presents a significant challenge due to the dynamic and unpredictable nature of such settings. Robots face complexities like changing lighting conditions , uneven terrains, and the presence of dynamic obstacles, which traditional localization methods struggle to provide precise positioning in these scenarios.

Through our project, we proposed to use an optical flow-based localization techniques, which utilize visual information to estimate robot motion relative to the environment. The adaptability of optical flow in real-time and its resistance to lighting variations position as a promising strategy for outdoor localization.

However, it's imperative to acknowledge that optical flow-based localization has its limitations, such as susceptibility to occlusions and struggles with intricate scenes. In response, our project integrates an Inertial Measurement Unit (IMU) to augment the system. This integration provides the robot's orientation data, bolstering the accuracy and robustness of our proposed approach.

key words

Mobile robot, Localization, Environment, Computer vision, Kalman filtering.

ملخص

تمثل تحديات تحديد موقع الروبوتات بدقة في البيئات الخارجية تحدياً كبيراً بسبب طبيعة هذه الإعدادات الديناميكية وغير المتوقعة. تواجه الروبوتات تعقيدات مثل تغير ظروف الإضاءة، والتضاريس غير المستوية، ووجود العوائق الديناميكية، والتي تجعل من الصعب على الأساليب التقليدية لتحديد الموقع توفير تحديد دقيق في مثل هذه السيناريوهات.

من خلال مشروعنا، اقترحنا استخدام تقنيات تحديد الموقع بناءً على تدفق البصريات، والتي تستخدم المعلومات البصرية لتقدير حركة الروبوت بالنسبة للبيئة. إن قدرة تدفق البصريات على التكيف في الوقت الحقيقي ومقاومته لتغيرات الإضاءة تجعلها استراتيجية واعدة لتحديد الموقع في الهواء الطلق.

مع ذلك، من الضروري أن ندرك أن تقنية تحديد الموقع بناءً على تدفق البصر تحمل قيودها، مثل العرصة للتغطيات والصعوبات في المشاهد المعقدة. كإجراء استجابة، يُدمج مشروعنا وحدة قياس الحركة الداخلية (ايپو) لتعزيز النظام. يقدم هذا التكامل بيانات الاتجاه للروبوت، مما يعزز دقة وصلابة النهج المقترح لدينا. روبوت متنقل، تحديد الموقع، بيئة، رؤية الكمبيوتر، تصفية كالمان

Définitions et termes

Dans ce mémoire, un certain nombre de termes et de définitions importants sont utilisés. Pour permettre au lecteur de mieux comprendre le sujet ces termes sont listés ci-dessous :

Position Une position est un ensemble de coordonnées et une orientation liée à un référentiel de coordonnées.

Positionnement Le processus de détermination d'une position est appelé positionnement.

Localisation Une position en termes de relation topologique est appelée emplacement.

IMU Inertial Measurement Unit

Fritzing C'est un logiciel open source spécialement conçu pour créer des projets électroniques.

MATLAB est un environnement de programmation et de calcul numérique largement utilisé dans les domaines de l'ingénierie

Table des matières

Table des matières	6
Liste des figures	9
I Etat de l'art	12
I.1 Historique	12
I.2 Applications des robots mobiles	13
I.3 Caractéristiques des environnements naturels	13
I.4 Types d'opérations effectuées par les robots mobiles	15
I.4.1 Agriculture de précision	15
I.4.2 Lutte contre les mauvaises herbes	15
I.4.3 Plantation et récolte assistées par des robots mobiles	16
I.4.4 Récolte des cultures et des fruits	17
I.5 Objective	17
I.6 Motivations	17
I.7 Etat de l'art	18
I.7.1 Localisation odométrique	18
I.7.2 Localisation inertielle	20
I.7.3 Localisation à partir de la vision	20
I.7.4 Localisation topologique	21
I.7.5 Localisation à partir d'images de profondeur	22
I.8 Conclusion	22
II Théorie des Robots mobiles	23
II.1 Introduction	23
II.2 Classification du Robot mobile	23
II.3 Architecture des robots mobiles	24
II.3.1 Structure mécanique et la motricité	25
II.3.2 Système de localisation	27
II.3.3 Traitement des informations et gestion des tâches	28
II.3.4 Les organes de sécurité	29
II.4 Holonomie et non-holonomie	30
II.4.1 Robots mobiles non-holonomes	31
II.4.2 Robots mobiles holonomes	31
II.5 Classification de robot mobiles a roues	32
II.5.1 Robot unicycle	32
II.5.2 Robot tricycle	32
II.5.3 Robot voiture	33
II.5.4 Robot omnidirectionnel	33
II.5.5 Avantages et Inconvénients	34
II.6 Méthodes de navigation	35
II.6.1 Navigation Métrique	35
II.6.2 Navigation par action associé à un lieu	35

II.6.3	Navigation topologique	36
II.6.4	Navigation géométrique	36
II.7	Modélisation du robot mobile à 4 roues	37
II.7.1	Définitions	37
II.7.2	Roulement sans glissement	38
II.7.3	Différentes forces qui agissent sur le système	39
II.7.4	Modèle cinématique	41
II.7.5	Modèle dynamique	42
II.8	Conclusion	44
III	Généralités sur la localisation des robots mobiles	45
III.1	Introduction	45
III.2	Localisation d'un robot mobile	45
III.3	Traitement d'image	46
III.4	Image Numérique	47
III.4.1	Image Binaire	47
III.4.2	Image à Niveaux de Gris	47
III.4.3	Image en Couleur	48
III.5	Vision assisté par ordinateur (VAO)	49
III.5.1	Acquisition	49
III.5.2	Prétraitement	50
III.5.3	Analyse d'une image	50
III.5.4	Apprentissage et classification	51
III.5.5	Interprétation	51
III.6	Filtrage de l'image	51
III.6.1	Filtrage linéaire	52
III.6.2	Filtrage non linéaire	52
III.7	Calibrage de caméra	52
III.7.1	Principe général du calibrage de caméra	52
III.7.2	OpenCV	52
III.7.3	Calibrage avec OpenCV	53
III.7.4	Paramètres de camera	54
III.7.5	Protocole de calibration	55
III.8	Flux optique	56
III.8.1	Calcul de flux optique	56
III.8.2	Méthodes de flux optique	56
III.8.3	Méthode de Lucas-Kanade	57
III.8.4	Calcul de Lucas Kanade	57
III.8.5	Estimation de position 3D à partir d'images 2D	58
III.8.6	Amélioration de l'estimation	59
III.9	Théorie du filtre de Kalman	60
III.9.1	Initialisation de l'état et de la covariance de l'erreur	60
III.9.2	Prédiction de l'état	61
III.9.3	Mise à Jour de la Covariance de l'État	61
III.10	Conclusion	61
IV	Partie expérimental	62
IV.1	Introduction	62
IV.2	Matériel utilisé	62
IV.2.1	Châssis de robot mobile 6WD	62
IV.2.2	Raspberry Pi 3 Model B	63
IV.2.3	Caméra Module Rev 1.3	64
IV.2.4	Support caméra	64
IV.2.5	Batterie	65

IV.2.6	Régulateur de tension	66
IV.2.7	Conducteur de moteur L298N	66
IV.2.8	Capteur Ultrason HC-SR04	67
IV.2.9	MPU6050	68
IV.3	Montage électrique	69
IV.4	Configuration de Raspberry Pi	70
IV.4.1	Accès à distance au Raspberry Pi	71
IV.4.2	Langage de programmation (python)	72
IV.5	Calibration de MPU6050	73
IV.6	Localisation du robot par flux optique	74
IV.7	Résultats obtenus par le MPU6050	76
IV.8	Filtre de Kalman	78
IV.9	Conclusion	80
.1	Annexes	82
.1.1	Raspberry Pi 3 model B / Annexes	82
.1.2	La carte mère Raspberry Pi 3	82
.1.3	L298N	83
.1.4	Description du L298N	83
.1.5	Ultrason	84
.1.6	Les caractéristiques du 6WD	85
Bibliographie		87

Liste des figures

I.1	Le chien électrique de Hammond et Miessner	12
I.2	Exemples des scènes dans un environnement naturel	14
I.3	Exemple d'un robot d'agriculture de précision	15
I.4	Le robot Ecorobotix cible les mauvaises herbes	16
I.5	Robots planteurs de graines	16
I.6	Exemple d'un robot de récolte	17
I.7	Dérive de localisation d'un robot en mouvement	18
I.8	Schéma de fusion de données pour la localisation multi-capteurs	19
I.9	Principe de localisation inertielle.	20
I.10	Principe de localisation à partir de la vision	21
I.11	Structure d'une carte topologique..	21
II.1	Boucle de commande d'un robot mobile autonome.	25
II.2	Exemple d'un Robot mobile à roues	25
II.3	Robot mobile a chenilles	26
II.4	Robot mobile rampant	27
II.5	Navigation de robot mobile en environnement encombré	28
II.6	Synoptique de la sécurité	30
II.7	Exemple de plate-forme non holonome de type Ackerman.	31
II.8	Exemple d'un robot mobile holonome	32
II.9	Robot de type unicycle.	32
II.10	Robot de type tricycle	33
II.11	Robot de type voiture	33
II.12	Robot de type omnidirectionnel	34
II.13	Schéma simplifié de tâches à l'intérieur de la navigation autonome	35
II.14	Opérations réalisées pour définir une tâche de navigation	36
II.15	Repérage d'un robot mobile	37
II.16	Repérage d'un robot mobile	39
II.17	Représentation des vitesses de chaque roue et celle du robot	40
II.18	Représentation des différentes forces qui agissent sur une des roues du robot mobile.	40
II.19	La cinématique de la plateforme du robot mobile à 4 roues	41
II.20	La dynamique de la plateforme du robot mobile à 4 roues	42
III.1	Approches de Localisation pour les Robots Mobiles	46
III.2	Les pixels de l'image	47
III.3	Exemple d'un image binaire	47
III.4	Exemple d'une image gris	48
III.5	Le codage RVB	48
III.6	Le processus de la vision assistée par ordinateur (VAO)	49
III.7	les étapes de l'acquisition de l'image	50
III.8	Logo d'opencv	53
III.9	Prise de vue différentes d'un checkerboard	53

III.10	Modèle de caméra complet	54
III.11	Exemples de flux optique	56
III.12	Le modèle de projection de caméra	59
III.13	Implémentation pyramidale d'une méthode de calcul du flux optique	60
IV.1	Robot mobile 6WD	63
IV.2	Raspberry Pi 3 Model B	63
IV.3	5mp raspberry pi camera module rev 1.3	64
IV.4	Support de caméra	65
IV.5	Lipo batterie 2500 mAh	65
IV.6	Régulateur de tension XL6009E1	66
IV.7	Double pont en H (L298N)	67
IV.8	Capteur ultrason HC-SR04	68
IV.9	Capteur MPU6050	68
IV.10	Montage de circuit électrique du notre robot mobile (Fritzing)	70
IV.11	Montage réelle du robot mobile	70
IV.12	L'application Raspberry Pi Imager	71
IV.13	Adresse IP de la carte Raspberry Pi	72
IV.14	Connexion à distance de raspberry avec VNC	72
IV.15	Logo de python	73
IV.16	Résultats d'interface de la calibration de MPU6050	73
IV.17	Visualisation des points d'Intérêt	74
IV.18	Diagramme de flux optique	75
IV.19	Résultats des mouvements de bruit et de filtrage selon X par rapport au temps	78
IV.20	Résultats des mouvements de bruit et de filtrage selon Y par rapport au temps	79
IV.21	Résultats des mouvements de bruit et de filtrage selon Z par rapport au temps	79
IV.22	Résultats des mouvements de bruit et de filtrage pour l'orientation du robot selon Z par rapport au temps	80
23	Les composantes du Raspberry Pi 3	82
24	Broches de connexion du HC-SR04	85

Introduction générale

La localisation d'un robot mobile autonome en milieu extérieur vise à permettre son déplacement autonome dans un environnement inconnu en déterminant précisément sa position et en orientant sa trajectoire. Cette tâche revêt une importance capitale pour la navigation autonome, la planification des mouvements du robot, ainsi que pour diverses applications telles que la surveillance environnementale, l'inspection d'installations industrielles, la cartographie de zones à risques, et d'autres applications pratiques.

Cependant, la localisation en milieu naturel représente un défi complexe en raison de la diversité et de la variabilité des environnements, ainsi que des limites inhérentes aux capteurs et aux algorithmes de localisation. Les techniques de localisation peuvent être basées sur des approches de vision par ordinateur, de filtrage de Kalman, d'apprentissage en profondeur, parmi d'autres.

Dans ce contexte, des recherches sont en cours afin de développer des méthodes de localisation robustes et précises spécifiquement adaptées aux robots mobiles autonomes évoluant dans des environnements naturels. L'objectif principal de ces travaux de recherche est de permettre une localisation précise d'une plateforme mobile à six roues en utilisant des capteurs permettant à la fois une localisation précise et une génération optimale de trajectoire. Ces capteurs sont conçus pour collecter des informations environnementales pertinentes, les traiter de manière à permettre au robot de déterminer avec précision sa position et de planifier efficacement sa trajectoire dans le milieu naturel.

Cette étude se concentre sur l'analyse et l'évaluation de différentes techniques de localisation adaptées aux robots mobiles autonomes évoluant en milieu naturel, dans le but de fournir des connaissances pratiques et théoriques sur les techniques de localisation les plus avancées, ainsi que des recommandations pour les applications futures.

Ce mémoire s'articule autour de quatre chapitres qui explorent différents aspects :

Dans le premier chapitre, nous effectuons une revue de l'état de l'art sur la localisation des robots mobiles, en mettant particulièrement l'accent sur son application dans le domaine agricole. Nous examinons les définitions, les différents domaines d'application et le fonctionnement des robots mobiles dans ce contexte spécifique.

Le deuxième chapitre se concentre sur la modélisation cinématique et dynamique des robots mobiles à quatre roues. Nous analysons en détail les différentes composantes de ces robots, en tenant compte de leurs caractéristiques spécifiques liées à la localisation en extérieur dans le domaine agricole.

Dans le troisième chapitre, nous abordons les méthodes de localisation en extérieur pour les robots agricoles. Nous explorons les différentes techniques utilisées, telles que la localisation odométrique, inertielle, basée sur la vision, les images de profondeur et la localisation topologique. Nous évaluons également leurs avantages et leurs limitations dans le contexte agricole.

Enfin, le quatrième chapitre présente les composants utilisés dans notre approche expérimentale pour la conception et la validation des méthodes de localisation en extérieur du robot. Nous détaillons les expériences menées et analysons les résultats obtenus, mettant en évidence l'efficacité et la fiabilité des solutions proposées.

Chapitre I

Etat de l'art

Les robots mobiles sont des dispositifs automatisés dotés de capacités de mouvement et conçus pour interagir avec leur environnement. Ils intègrent des éléments mécaniques, électroniques et informatiques, ainsi que des capteurs et des effecteurs, leur permettant de percevoir leur environnement, de prendre des décisions et d'effectuer des actions en conséquence.

Dans ce chapitre, nous allons explorer l'historique et l'évolution des robots mobiles au fil du temps. Nous examinerons également les différents domaines d'application des robots mobiles, en mettant un accent particulier sur leur utilisation dans le domaine de l'agriculture. De plus, nous présenterons brièvement les méthodes couramment utilisées pour la localisation des robots mobiles

I.1 Historique

L'histoire de l'évolution des robots mobiles est marquée par des avancées significatives tout au long des décennies. Des premières machines rudimentaires aux robots sophistiqués d'aujourd'hui, ces engins ont considérablement amélioré l'efficacité, la rentabilité et la durabilité environnementale, notamment dans le secteur agricole.

Les robots ont fait leur apparition au début du XXe siècle, suscitant l'intérêt des ingénieurs qui cherchaient à tester les hypothèses avancées par les biologistes et les psychologues. Par exemple, le chien électrique conçu par Hammond et Missner en 1915 [1], inspiré des travaux de Loeb en 1918 sur le phototropisme des animaux, qui les pousse à être attirés par la lumière (figure I.1). D'autres machines, telles que celles développées par Russell (1913) et Stephens (1929) [2], ainsi que la tortue cybernétique de Gray Walter (1950) [3], ont également contribué à l'avancement de la robotique. Parallèlement, les recherches menées par des psychologues tels que Thorndike (1911), Hull (1943) et le physiologiste Pavlov (1903) [4], ont également enrichi notre compréhension du comportement humain et animal.

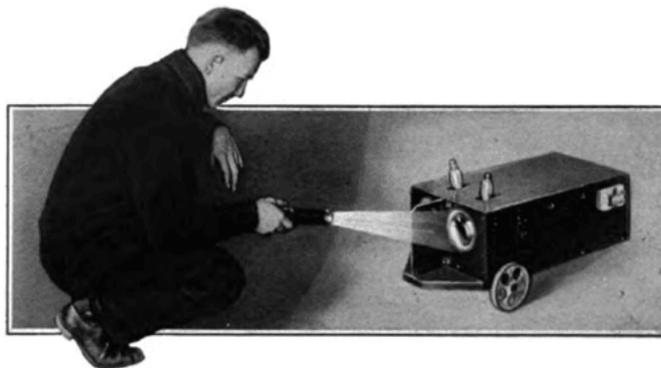


FIGURE I.1 – Le chien électrique de Hammond et Miessner

I.2 Applications des robots mobiles

Les robots mobiles offrent des solutions novatrices pour l'automatisation de tâches diverses et variées, couvrant plusieurs domaines d'activité.

Le tableau ci-après résume les diverses applications des robots mobiles [5] :

Le types de robot mobile :	L'applications :
Agricole	- cueillette de fruits - traite, moisson, traitement des vignes...
Médecine	- assistance d'urgence - aide aux handicapés physiques, aux aveugles
Industriel	convoyage et surveillance
Militaire et sécurité civile	- neutralisation d'activité terroriste - surveillance, patrouille - pose d'explosif - déminage
Chimique	- surveillance de site - manipulation de matériaux toxiques
Industrie nucléaire	- surveillance de sites - manipulation de matériaux radio-actifs - démantèlement de centrales
Lutte contre l'incendie	- localisation d'une source d'incendie - détection de fumée - suppression de flammes
Nettoyage	- coque de navire - nettoyage industriel

TABLE I.1 – Applications des robots mobiles dans différents domaines

I.3 Caractéristiques des environnements naturels

Le développement des robots mobiles dans le domaine agricole est largement motivé par les caractéristiques spécifiques de ce secteur. Les environnements naturels peuvent être très variés et complexes, ce qui rend la tâche des robots mobiles autonomes difficile.

Voici quelques spécificités importantes à considérer pour un robot mobile autonome :

Les surfaces Les environnements naturels peuvent inclure des surfaces variées, telles que des rochers, de la terre, du sable, de la neige, de l'eau, etc. Il est important que le robot soit capable de se déplacer sur ces surfaces de manière stable et sûre.

Les obstacles Les environnements naturels peuvent contenir de nombreux obstacles, tels que des arbres, des rochers, des branches, etc. Le robot doit être capable de les détecter et de les éviter pour éviter les collisions.

Les pentes et les gradients Les environnements naturels peuvent avoir des pentes et des gradients importants, ce qui peut affecter la traction et la stabilité du robot. Le robot doit être capable de détecter ces changements de terrain et de les gérer en conséquence.

Les conditions météorologiques Les environnements naturels peuvent être soumis à des conditions météorologiques extrêmes, telles que la pluie, la neige, le vent, etc. Le robot doit être capable de s'adapter à ces conditions et de continuer à fonctionner de manière fiable.

La visibilité Les environnements naturels peuvent avoir une visibilité limitée en raison de la végétation dense, du brouillard, etc. Le robot doit être capable de naviguer dans ces conditions en utilisant d'autres capteurs tels que les lasers, les radars, etc.

L'alimentation Les environnements naturels peuvent ne pas avoir d'alimentation électrique disponible. Le robot doit être capable de fonctionner pendant de longues périodes sans avoir besoin d'une source d'alimentation externe.

La communication Les environnements naturels peuvent avoir des obstacles physiques et électromagnétiques qui peuvent entraver la communication avec le robot. Le robot doit être capable de communiquer avec d'autres robots et avec la station de contrôle de manière fiable.



FIGURE I.2 – Exemples des scènes dans un environnement naturel

I.4 Types d'opérations effectuées par les robots mobiles

Les robots agricoles sont des machines spécialement développées pour effectuer diverses tâches liées à l'agriculture et à l'horticulture. Leur conception vise à répondre aux besoins spécifiques du secteur agricole et à contribuer à l'automatisation des activités agricoles.

Ces robots sont utilisés pour accomplir un large éventail d'opérations, notamment :

I.4.1 Agriculture de précision

Ces types de robots de terrain sont utilisés dans les petites exploitations pour les vignobles qui permettent d'appliquer des techniques d'agriculture de précision. Ils sont souvent utilisés pour surveiller de manière autonome le sol et les cultures, collecter des données et appliquer des mesures précises de protection des cultures [6].



FIGURE I.3 – Exemple d'un robot d'agriculture de précision

I.4.2 Lutte contre les mauvaises herbes

Les robots autonomes sont utilisés pour des tâches d'éradication des mauvaises herbes, privilégiant des méthodes non chimiques telles que le désherbage mécanique, l'utilisation d'eau chaude ou de chalumeaux, ainsi que la pulvérisation ciblée d'herbicides dans les pépinières. Ces robots contribuent à automatiser les activités de culture dans les pépinières, en particulier le déplacement des plantes dans de grandes serres, ce qui améliore l'efficacité et compense le manque de main-d'œuvre.[7].



FIGURE I.4 – Le robot Ecorobotix cible les mauvaises herbes

I.4.3 Plantation et récolte assistées par des robots mobiles

Une application émergente, les robots de terrain équipés de systèmes de vision 3D peuvent désormais planter et semer les cultures avec précision pour une croissance optimale [8].



FIGURE I.5 – Robots planteurs de graines

I.4.4 Récolte des cultures et des fruits

La récolte des légumes et des fruits peut être effectuée 24 heures sur 24, 7 jours sur 7 [9].



FIGURE I.6 – Exemple d'un robot de récolte

I.5 Objective

L'objectif principal de notre projet est de concevoir un système de localisation précis pour un robot agricole, lui permettant d'atteindre avec exactitude une position spécifique dans les champs.

Pour atteindre cet objectif, nous prévoyons de mettre en œuvre des techniques de localisation avancées, notamment l'utilisation du flux optique et l'intégration d'un capteur inertielle. Cette combinaison d'approches permettra au robot de déterminer avec précision sa position dans son environnement.

En plus de la localisation précise, nous prévoyons d'intégrer un filtre de Kalman dans notre système. Cette intégration permettra d'atténuer les perturbations, de réduire les bruits et d'améliorer la précision globale du système. En combinant ces différentes techniques, notre démarche préliminaire vise à accroître de manière significative l'efficacité et la productivité de l'agriculture en autorisant les robots agricoles à se déplacer de manière autonome et précise au sein des champs.

I.6 Motivations

Notre choix de sujet de recherche découle d'un fort intérêt pour le domaine de la robotique et ses applications pratiques. Nous sommes fascinés par la capacité des robots à accomplir de manière autonome des tâches complexes et à résoudre des problèmes dans divers domaines.

Nous sommes motivés par la nécessité de développer des solutions novatrices pour améliorer l'efficacité et la productivité dans le secteur de l'agriculture. Nous considérons qu'une meilleure gestion de la fertilité des sols est essentielle. En déployant des robots dans les zones agricoles, nous pouvons collecter des données précises et détaillées sur la composition du sol, les niveaux de nutriments et d'autres paramètres pertinents. Cette approche permet aux agriculteurs de prendre des décisions éclairées concernant la fertilisation, ce qui conduit à une utilisation plus efficace des ressources et à une amélioration de la productivité agricole.

De plus, l'automatisation de ce processus réduit considérablement la charge de travail manuel et les coûts associés. Cela permet de libérer du temps et des ressources pour se concentrer sur d'autres aspects essentiels de la gestion agricole.

En adoptant cette approche technologique avancée, nous contribuons à la transformation numérique de l'agriculture, favorisant ainsi la durabilité environnementale en réduisant les gaspillages et les impacts négatifs sur les écosystèmes.

I.7 Etat de l'art

La localisation en extérieur des robots mobiles est un domaine de recherche dynamique et en constante évolution, bénéficiant de nombreuses avancées récentes.

Diverses approches et technologies sont disponibles pour effectuer cette localisation, adaptées aux besoins spécifiques de chaque application. Parmi les méthodes les plus fréquemment utilisées pour localiser ces robots, on peut citer :

I.7.1 Localisation odométrique

La localisation odométrique est une méthode couramment utilisée pour déterminer la position des robots mobiles lors de leurs déplacements dans des environnements naturels. Elle repose sur l'intégration des rotations élémentaires des roues du robot afin de calculer sa position. Dans la plupart des applications de robotique mobile, les mesures de rotation des roues sont généralement effectuées à l'aide de divers types de codeurs tels que les codeurs optiques, magnétiques, inductifs, capacitifs, etc. Ces codeurs permettent de mesurer précisément les rotations des roues et d'obtenir des informations cruciales pour la localisation du robot.

Ainsi, en utilisant ces mesures de rotation, les systèmes odométriques peuvent estimer la position du robot en suivant et en intégrant les mouvements des roues au fil du temps. Le mouvement global du robot est reconstitué en intégrant les mesures individuelles des déplacements des roues à chaque instant. En partant d'une position initiale, la position actuelle du robot est calculée en intégrant les déplacements effectués. Toutefois, en raison de l'intégration de la vitesse pour déduire la position, cela implique également l'intégration de l'erreur, qui s'accumule progressivement au fil du temps. Cette accumulation d'erreur est illustrée par la figure (I.7), qui représente la dérive de l'odométrie d'un robot en déplacement. À mesure que le robot progresse dans le temps, les erreurs associées à l'odométrie augmentent progressivement[10].

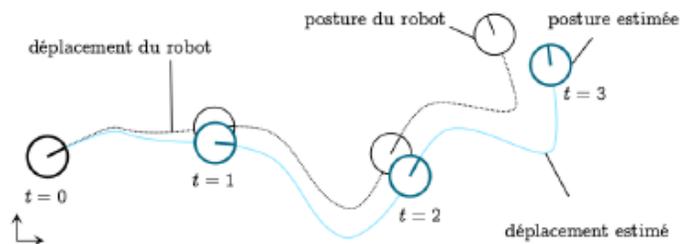


FIGURE I.7 – Dérive de localisation d'un robot en mouvement

Néanmoins, la localisation odométrique peut être sujette à des erreurs cumulatives qui se produisent progressivement au fil du temps. Des facteurs tels que les variations de la surface du sol, la friction des roues, la charge appliquée sur le robot, et d'autres sources d'incertitude peuvent également influencer la précision de la localisation odométrique. Par conséquent, afin d'améliorer la précision de la localisation dans des environnements naturels, il est souvent nécessaire de combiner la localisation odométrique avec d'autres méthodes complémentaires.

Une étude réalisée par Borenstein a démontré qu'il est possible de réduire l'erreur de localisation odométrique en utilisant des techniques de correction d'erreur, telles que le filtrage de Kalman. Le filtrage de Kalman est une approche statistique qui permet de combiner les mesures odométriques avec d'autres sources d'informations afin d'estimer plus précisément la position réelle du robot. Cette méthode de correction d'erreur contribue à réduire l'impact des erreurs cumulatives et à améliorer la précision globale de la localisation[11]. Pour approfondir la localisation odométrique, une variante intéressante est la localisation odométrique optique. Cette méthode combine les principes de l'odométrie (utilisant les capteurs de vitesse) avec le flux optique. Le flux optique exploite les informations de mouvement apparent dans les images capturées par une caméra pour estimer le déplacement du robot.

L'odométrie optique visuelle utilise des caméras pour capturer les images de l'environnement et analyse le flux optique pour estimer le mouvement du robot. En analysant les changements apparents dans les

images, cette méthode est capable de calculer le déplacement du robot. Elle peut être utilisée pour estimer la position et l'orientation du robot dans l'environnement. Cependant, elle peut être sensible aux conditions d'éclairage et aux changements de texture de l'environnement.

D'autre part, l'odométrie optique lidar utilise des capteurs lidar pour mesurer la distance entre le robot et les objets environnants. En analysant les scans lidar successifs, cette méthode est capable d'estimer le mouvement du robot. En utilisant les informations de mouvement lidar en combinaison avec les encodeurs de vitesse, la position et l'orientation du robot peuvent être estimées. Cette méthode peut être plus robuste dans des environnements avec des conditions d'éclairage difficiles, mais elle nécessite l'utilisation de capteurs lidar spécifiques [10].

Une approche alternative consiste à fusionner les données provenant de différentes sources de localisation, telles que l'odométrie optique, le lidar et le GPS, dans le but de bénéficier des avantages spécifiques de chaque méthode tout en compensant leurs limitations respectives. À cette fin, des algorithmes d'estimation de l'état, tels que les filtres de Kalman étendus ou les méthodes de fusion de capteurs, peuvent être employés. Cette fusion de données permet d'obtenir des estimations de position et d'orientation du robot plus précises et robustes. De plus, la redondance des capteurs peut également garantir une certaine résilience en présence de perturbations, telles que la surexposition des caméras ou les mesures fantômes. Une illustration de cette approche est présentée dans la Figure (I.8) [12].

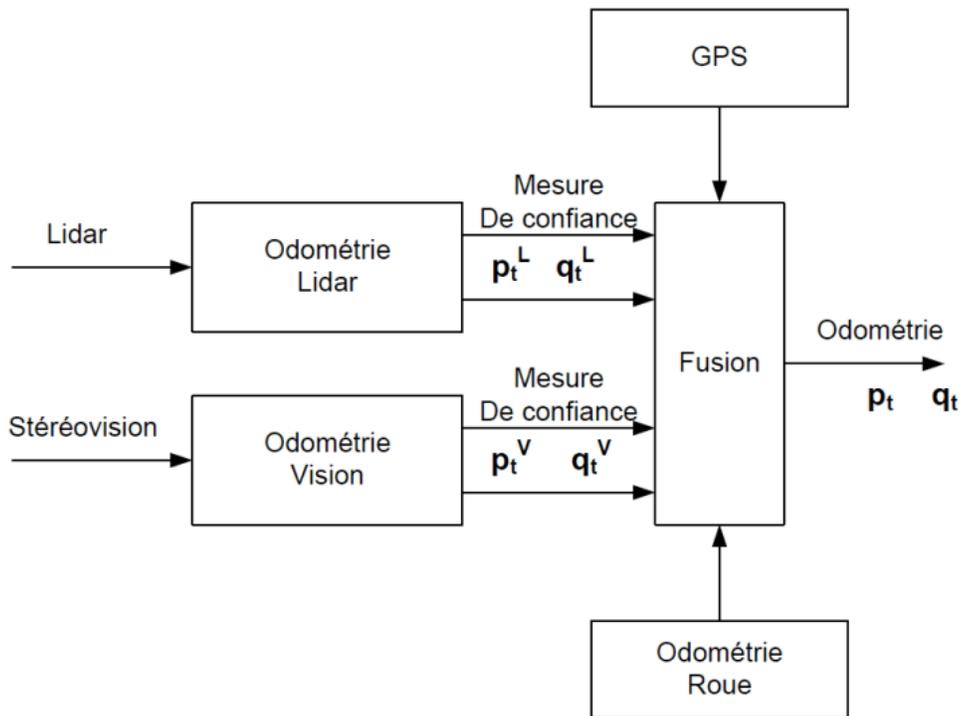


FIGURE I.8 – Schéma de fusion de données pour la localisation multi-capteurs

I.7.2 Localisation inertielle

La localisation inertielle est une méthode de localisation qui utilise des capteurs inertiels tels que des accéléromètres et des gyroscopes pour estimer la position, la vitesse et l'orientation d'un robot mobile. Cette méthode est souvent utilisée en complément d'autres techniques de localisation telles que la localisation par GPS ou la localisation par vision. Les capteurs inertiels peuvent être utilisés pour fournir des informations sur les mouvements du robot, telles que la vitesse linéaire et angulaire, ainsi que l'accélération. Ces informations peuvent être intégrées dans un filtre de Kalman étendu pour estimer la position et l'orientation du robot en temps réel.

La fusion des données provenant des capteurs inertiels et de la vision à partir d'une caméra ou d'autres capteurs de vision peut être utilisée pour améliorer la localisation et la navigation d'un objet. Les capteurs inertiels fournissent des mesures de l'accélération et de la rotation, permettant de suivre les mouvements de l'objet en temps réel, et la vision permet de détecter et de reconnaître les caractéristiques de l'environnement, telles que les objets, les murs et les marques distinctives. La fusion de ces informations permet de corriger les erreurs accumulées dans les mesures inertielle et d'améliorer la précision de la localisation.

Cependant, la localisation inertielle a quelques limitations, notamment l'accumulation d'erreurs au fil du temps, qui peut conduire à une dérive de la position estimée. De plus, les capteurs inertiels peuvent être sensibles aux vibrations et aux chocs, ce qui peut affecter la qualité des mesures. Par conséquent, la localisation inertielle est souvent utilisée en combinaison avec d'autres méthodes de localisation pour améliorer la précision et la fiabilité de l'estimation de la position du robot mobile autonome en environnements naturels [13].

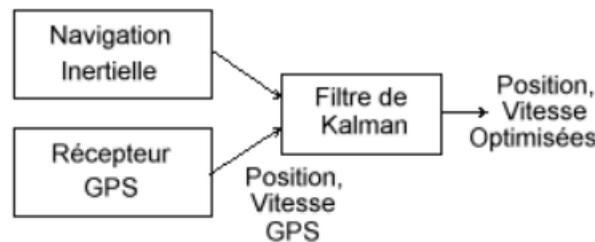


FIGURE I.9 – Principe de localisation inertielle.

I.7.3 Localisation à partir de la vision

La localisation à partir de la vision est une technique de localisation qui utilise les images capturées par une caméra pour estimer la position du robot. Cette méthode peut être basée sur différents éléments visuels tels que les marqueurs visuels, les bords, les coins ou les caractéristiques des objets environnants. Les marqueurs visuels, tels que les QR codes ou les AR tags, peuvent être utilisés pour aider à localiser le robot en environnement naturel. Ces marqueurs sont placés à des endroits stratégiques de l'environnement et sont facilement détectables par la caméra du robot.

Les bords et les coins peuvent également être utilisés pour la localisation. Les techniques basées sur les bords et les coins sont généralement utilisées en combinaison avec d'autres techniques telles que l'odométrie ou la localisation inertielle pour améliorer la précision de la localisation. Les caractéristiques des objets environnants, telles que les textures ou les motifs, peuvent également être utilisées pour la localisation. Cette méthode consiste à extraire les caractéristiques des objets de l'environnement à partir des images capturées par la caméra, puis à utiliser ces caractéristiques pour estimer la position du robot.

La localisation à partir de la vision est une technique très prometteuse pour la localisation en environnement naturel car elle utilise des capteurs peu coûteux et facilement disponibles, et peut être utilisée dans des environnements où les autres techniques de localisation sont limitées. Cependant, elle peut être affectée par des conditions d'éclairage et des changements dans l'environnement, ce qui peut entraîner une baisse de la précision de la localisation [14].

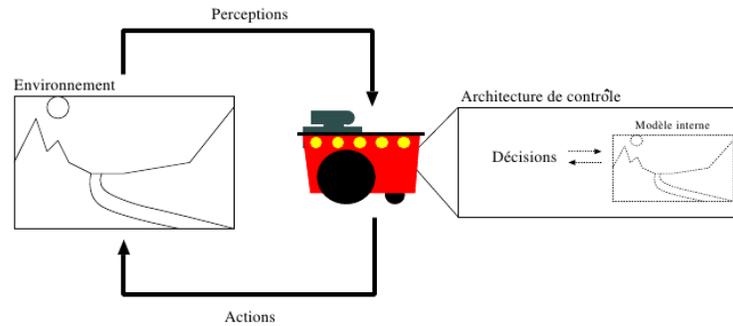


FIGURE I.10 – Principe de localisation à partir de la vision

I.7.4 Localisation topologique

La localisation topologique est une méthode de localisation qui utilise des points de repère dans l'environnement plutôt que des coordonnées géométriques précises. Elle est souvent utilisée dans les environnements intérieurs, tels que les bâtiments, où la précision de la localisation GPS est limitée. Dans le contexte des robots mobiles autonomes, la localisation topologique implique la création d'une carte topologique de l'environnement, dans laquelle chaque point est associé à une étiquette descriptive ou à un identifiant unique. Le robot utilise ensuite cette carte pour déterminer sa position actuelle en fonction des repères qu'il peut détecter à l'aide de capteurs tels que des caméras ou des scanners laser.

Les méthodes de localisation topologique comprennent souvent des techniques d'apprentissage automatique, telles que les réseaux de neurones ou les algorithmes de clustering, pour la reconnaissance de motifs et la classification des repères. La localisation topologique est également souvent utilisée en combinaison avec d'autres méthodes de localisation, telles que la localisation odométrique ou la localisation basée sur la vision, pour améliorer la précision de la localisation. Dans le contexte spécifique de la localisation d'un robot mobile autonome dans une serre agricole, la localisation topologique peut être utilisée pour cartographier les différents points de repère tels que les rangées de plantes, les portes ou les zones de stockage. En utilisant cette carte topologique, le robot peut naviguer de manière autonome dans la serre en évitant les obstacles et en atteignant les destinations désirées [15].

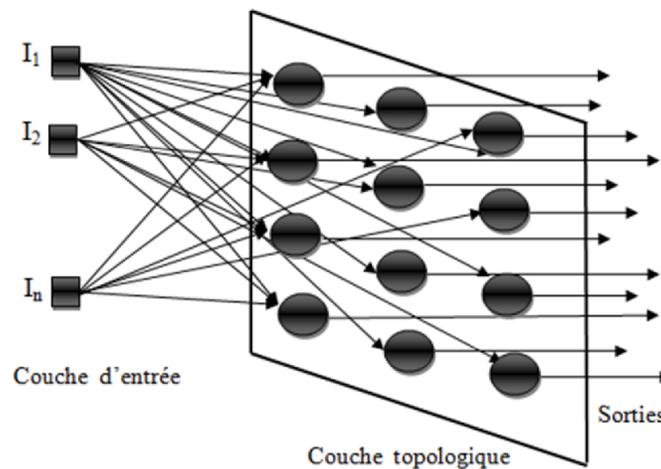


FIGURE I.11 – Structure d'une carte topologique..

I.7.5 Localisation à partir d'images de profondeur

La localisation à partir d'images de profondeur est une méthode de localisation qui utilise des capteurs tels que des caméras RGB-D (qui fournissent à la fois des images en couleur et des informations de profondeur) pour estimer la position et l'orientation d'un robot mobile autonome dans un environnement naturel. Cette méthode utilise la triangulation pour estimer la position de l'objet ou de la surface à partir de laquelle la profondeur est mesurée.

Les images de profondeur peuvent être utilisées pour créer des cartes 3D de l'environnement, qui peuvent être utilisées pour la localisation et la navigation. Ces cartes peuvent être créées en utilisant des techniques telles que la stéréovision ou la structure à mouvement (structure from motion), qui permettent de reconstruire la géométrie 3D de l'environnement à partir de plusieurs images.

Cette méthode de localisation présente quelques avantages, tels que la précision élevée, la robustesse aux variations d'éclairage et la capacité à fonctionner dans des environnements sans GPS. Cependant, elle peut être sensible aux changements de l'environnement, tels que les variations de la luminosité ou les changements dans la géométrie de l'environnement. En outre, la création de cartes 3D à partir d'images de profondeur peut être coûteuse en termes de ressources computationnelles [16].

I.8 Conclusion

Ce chapitre nous a permis d'acquérir une vision générale des robots mobiles en explorant leur histoire, leurs applications et les caractéristiques des environnements naturels dans lesquels ils évoluent. De plus, nous avons présenté un état de l'art sur les méthodes de localisation des robots mobiles, en mettant en évidence les approches couramment utilisées. Ces méthodes jouent un rôle crucial dans l'amélioration de l'autonomie et de l'efficacité des robots mobiles, notamment dans le domaine agricole, ouvrant ainsi la voie à des avancées significatives dans les opérations agricoles modernes.

Dans le prochain chapitre, nous aborderons la théorie des robots mobiles, en examinant leur classification, leur architecture et leur modélisation cinématique et dynamique. Cette étude approfondie nous permettra de mieux comprendre le fonctionnement interne des robots mobiles, ce qui sera essentiel pour la suite de notre projet.

Chapitre II

Théorie des Robots mobiles

II.1 Introduction

La modélisation cinématique et dynamique des robots mobiles est d'une importance cruciale pour comprendre leur comportement spatial et leurs interactions avec leur environnement. La modélisation cinématique permet de décrire leurs mouvements et positions, tandis que la modélisation dynamique étudie les forces et les moments qui influencent leur comportement. Combinées, ces deux approches permettent d'analyser, concevoir et contrôler efficacement les robots mobiles, en optimisant leur navigation, leur stabilité et leurs performances dans différentes applications.

Dans ce chapitre, nous étudions en détail la théorie des robots mobiles en abordant leur classification, leur architecture et leurs méthodes de navigation. Nous mettons l'accent sur la modélisation d'un robot mobile à 4 roues en prenant en compte le phénomène du roulement sans glissement. Cette analyse approfondie nous permettra de mieux comprendre le fonctionnement et les applications des robots mobiles dans divers domaines.

II.2 Classification du Robot mobile

Dans le domaine de la robotique, il existe une classification établie dans la littérature pour définir le degré d'autonomie des robots mobiles. Cette classification les catégorise en fonction de leur mode de contrôle, on peut classer comme suivant :

Télécommandé par un opérateur

Dans cette catégorie, le robot mobile est entièrement contrôlé par un opérateur externe. L'opérateur lui transmet des instructions détaillées pour chaque tâche à accomplir. Le robot n'a pas d'autonomie décisionnelle et agit uniquement selon les directives fournies par l'opérateur.

Cette classification reflète le niveau le plus bas d'autonomie, où le robot dépend entièrement de l'opérateur pour toutes ses actions.

Télécommandé en termes de tâches

Le robot mobile est télécommandé pour effectuer des tâches spécifiques. Contrairement au mode "Télécommandé par un opérateur", il dispose ici d'une certaine automatisation pour exécuter la tâche donnée. Une fois que l'opérateur a spécifié la tâche à accomplir, le véhicule peut contrôler automatiquement ses actions pour la mener à bien, sans avoir besoin d'une supervision constante pour chaque mouvement effectué. Cependant, l'opérateur garde toujours le contrôle sur les tâches globales assignées au robot.

Semi-autonome

Le robot mobile est capable d'effectuer des tâches prédéfinies sans l'aide constante de l'opérateur. Il peut prendre des décisions simples de manière autonome pour accomplir les tâches assignées, mais reste limité aux tâches préprogrammées et ne possède pas la flexibilité pour s'adapter à des situations imprévues ou inconnues.

Autonome

Au sommet de l'échelle de l'autonomie robotique, nous trouvons les robots autonomes. Ces machines sont capables d'accomplir des tâches partiellement définies de manière indépendante, sans nécessiter une intervention constante de l'opérateur. Leur capacité de décision et leur flexibilité leur permettent de faire face à des défis complexes et de s'adapter à des situations changeantes.

La conception de ces robots représente un véritable défi d'ingénierie, où la complexité atteint son apogée. Des domaines tels que la représentation des connaissances, la prise de décisions et la génération de plans sont résolus au sein même du robot, lui conférant une intelligence embarquée.

Ces robots dotés d'une autonomie avancée posent des problèmes spécifiques, parmi lesquels on peut citer :

- La conception mécanique soigneusement pensée pour assurer une mobilité fluide et une adaptation parfaite à l'environnement dans lequel le robot évolue.
- La précision cruciale de la détermination de la position et de l'orientation du robot, garantissant l'exécution précise et fiable des tâches spécifiques qui lui sont assignées.
- La recherche minutieuse du chemin optimal pour atteindre la destination requise, prenant en compte les contraintes et les obstacles présents sur le parcours. Cela implique la planification de trajectoires à la fois efficaces et sûres pour accomplir brillamment les tâches qui lui sont confiées.

Après avoir présenté les différentes catégories de robots mobiles en fonction de leur degré d'autonomie, il est évident que les choix architecturaux jouent un rôle crucial dans la mise en œuvre de leurs capacités d'autonomie. Ces décisions déterminantes impactent directement la flexibilité, l'efficacité et la robustesse des robots dans divers environnements.

II.3 Architecture des robots mobiles

L'architecture d'un robot mobile est conçue pour assurer un fonctionnement cohérent et efficace du robot, lui permettant d'interagir intelligemment avec son environnement. Cette conception globale permet au robot de percevoir les informations environnementales, de les traiter de manière appropriée, de prendre des décisions éclairées en fonction de ces données, et de mettre en œuvre des actions adéquates en utilisant ses composants mécaniques et instrumentaux.

En général, l'architecture d'un robot mobile se compose de quatre structures fondamentales qui interagissent harmonieusement pour assurer son bon fonctionnement[17] :

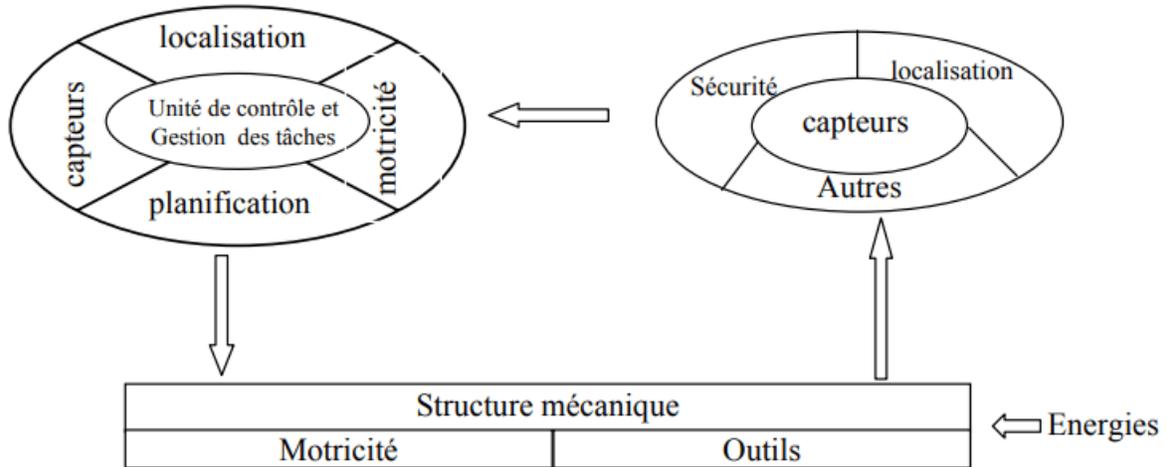


FIGURE II.1 – Boucle de commande d'un robot mobile autonome.

II.3.1 Structure mécanique et la motricité

La structure mécanique et la motricité jouent un rôle primordial dans l'architecture d'un robot mobile. Elle englobe la conception et la construction de sa structure physique, y compris les systèmes de propulsion tels que les roues, les chenilles ou les jambes, ainsi que les mécanismes de direction, les articulations et les systèmes de suspension. Grâce à ces éléments, le robot peut se déplacer avec agilité et s'adapter à divers types de terrains, ce qui assure une mobilité optimale et une performance efficace dans son environnement.

Les quatre principales catégories de structures mécaniques assurant la motricité sont les suivantes :

Les mobiles à roues

La mobilité à roues est l'une des structures mécaniques les plus couramment utilisées dans les robots mobiles. Elle permet au robot de se déplacer dans différentes directions avec une accélération et une vitesse significatives, en fonction de la disposition et des dimensions des roues. Cette configuration offre une certaine capacité de franchissement d'obstacles et d'escalade de marches d'escalier. Diverses configurations de roues, en termes de nombre, d'agencement et de fonction, peuvent être appliquées pour répondre aux besoins spécifiques du robot.



FIGURE II.2 – Exemple d'un Robot mobile à roues

Les mobiles à chenilles

Les robots mobiles à chenilles sont un autre type de robot qui utilise des chenilles comme moyen de locomotion pour se déplacer dans leur environnement. Les chenilles offrent une meilleure adhérence au sol par rapport aux roues, ce qui les rend particulièrement adaptées aux terrains perturbés et aux environnements extérieurs. Les robots mobiles à chenilles : Les robots mobiles à chenilles sont souvent utilisés dans des situations où les conditions du sol sont difficiles, comme sur des surfaces rugueuses, glissantes, accidentées ou non stabilisées. Les chenilles permettent au robot de mieux surmonter les obstacles et de maintenir une stabilité accrue lors de son déplacement.

La commande des robots mobiles à chenilles est réalisée en imposant une différence de vitesse entre les chenilles droite et gauche. Cette différence de vitesse permet de contrôler la direction du robot, en lui permettant de tourner sur place ou de suivre des trajectoires spécifiques.

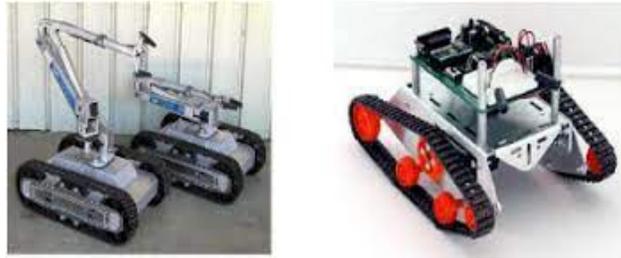


FIGURE II.3 – Robot mobile a chenilles

Les Robots rampants

La reptation est une méthode de locomotion adaptée aux environnements de type tunnel, permettant ainsi la création de structures filiformes. Cette approche repose sur l'utilisation d'un ensemble de modules, chacun possédant plusieurs degrés de mobilité. Les techniques utilisées s'inspirent des méthodes de déplacement observées chez les animaux.

Les robots rampants se déclinent en trois types distincts :

- Le type scolopendre, qui est composé d'une structure articulée selon deux axes orthogonaux, rendant l'ensemble inextensible.
- Le type péristaltique, qui implique un déplacement relatif d'un module par rapport à ses voisins, similaire aux mouvements de contraction et d'expansion observés dans la reptation de certains animaux.
- Le type lombric, caractérisé par trois articulations permettant des rotations orthogonales ainsi qu'une translation dans le sens du mouvement principal.

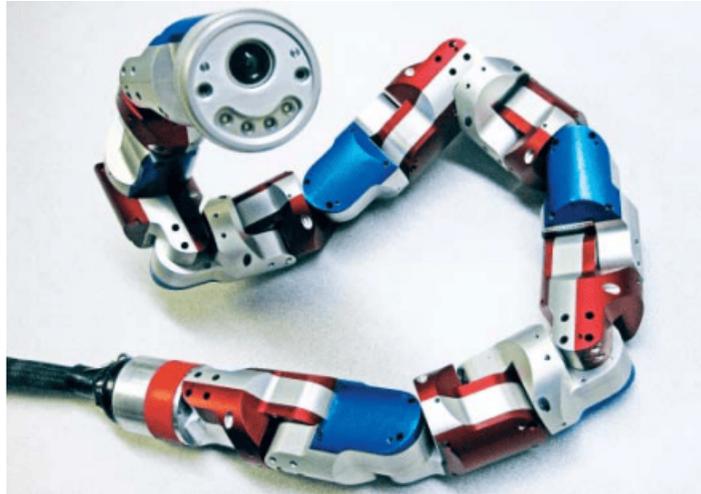


FIGURE II.4 – Robot mobile rampant

Les Robots marchants

Les robots mobiles marchants sont conçus pour accomplir diverses tâches dans des environnements difficiles, dangereux ou inaccessibles pour les humains. Leur anatomie complexe, qui comprend de nombreux degrés de liberté, présente des similitudes avec celle des robots manipulateurs. Ces robots utilisent la commande des articulations en termes de coordonnées pour contrôler leur locomotion, définissant ainsi l'allure du robot et assurant un déplacement stable et fluide.

Inspirés de la marche des animaux, en particulier celle des insectes, ces robots développent des techniques de déplacement efficaces. L'un des défis majeurs pour ces robots marchants est de s'adapter au support sur lequel ils évoluent, en choisissant les points de contact les plus appropriés pour assurer à la fois leur avancée et leur stabilité. Pour résoudre cette problématique, des capteurs de proximité, de contact ou de vision sont utilisés pour fournir des informations en temps réel au robot, l'aidant ainsi à prendre des décisions adaptées à son environnement changeant. Ces capteurs permettent aux robots marchants de s'adapter de manière efficace à leur environnement et de mener à bien leurs missions avec succès.

La motricité et l'énergie

Pour assurer leur déplacement, les robots utilisent des moteurs électriques, thermiques ou hydrauliques. L'énergie électrique est couramment employée en raison de sa facilité de commande. Cependant, son transport et sa génération peuvent poser des défis.

Différentes méthodes sont utilisées pour alimenter les robots en énergie :

- L'utilisation de batteries qui peuvent être rechargées périodiquement de manière automatique ou manuelle, ou bien échangées avec d'autres batteries lorsque celles-ci sont déchargées.
- L'emploi de groupes électrogènes embarqués, mais cela peut entraîner une masse élevée pour le robot, car l'énergie de base est alors de nature thermique.
- L'utilisation d'un cordon ombilical, bien que cela réduise l'autonomie du robot, car il dépend alors d'une source d'alimentation externe en continu.

II.3.2 Système de localisation

Le système de localisation est un élément essentiel de l'architecture des robots mobiles, car il permet au robot de connaître sa position et son orientation dans l'environnement. Cette information est cruciale pour la planification de trajectoire, l'évitement d'obstacles et la prise de décisions éclairées lors de l'interaction

avec l'environnement. Dans le premier chapitre, nous avons exploré différentes méthodes de localisation, telles que la triangulation, l'utilisation de balises, la détection d'obstacles à distance, et bien d'autres. Ces méthodes fournissent au robot les informations nécessaires pour naviguer de manière autonome dans son environnement, même dans des territoires inconnus.

II.3.3 Traitement des informations et gestion des tâches

Dans l'architecture des robots mobiles, le module central de traitement des informations et de gestion des tâches occupe une position critique. Il joue un rôle fondamental en permettant au robot de prendre des décisions éclairées et d'accomplir ses missions de manière efficiente. Notre attention est spécifiquement axée sur les robots mobiles.

Au cœur de cette problématique se trouve la planification du déplacement du robot, qui vise à déterminer comment ce dernier se déplace en fonction des informations disponibles. Ces informations peuvent être statiques, c'est-à-dire préalablement connues, ou dynamiques, c'est-à-dire acquises en temps réel lors de l'évolution du robot dans son environnement.

La génération de plans repose sur trois concepts fondamentaux [18] :

- La stratégie de navigation .
- La modélisation de l'espace.
- La planification .

Stratégie de navigation

La navigation revêt une importance capitale dans la robotique mobile, car elle confère une autonomie étendue au robot. Le système de navigation se compose de plusieurs modules, parmi lesquels la localisation et l'évitement d'obstacles jouent un rôle central.

La localisation est réalisée au moyen de capteurs à ultrasons qui permettent de détecter la présence d'obstacles sur le trajet du robot. Une fois les obstacles repérés, le robot peut entreprendre diverses actions, telles que la cartographie de son environnement, la vérification de la distance par rapport à l'obstacle en comparaison d'une distance limite, et, le cas échéant, l'adoption d'une stratégie d'évitement appropriée[18].

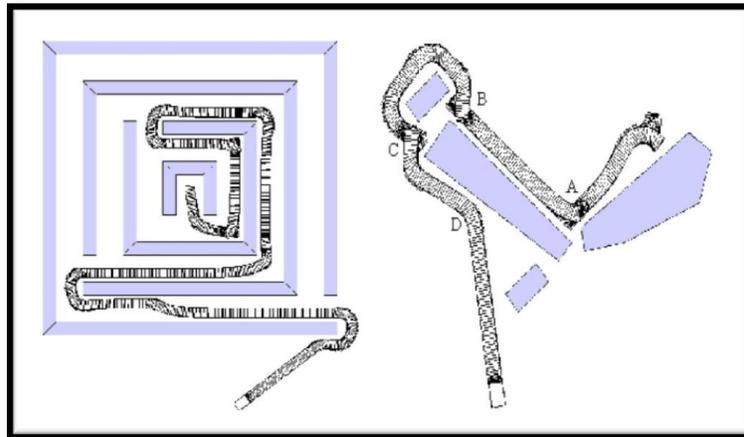


FIGURE II.5 – Navigation de robot mobile en environnement encombré

Modélisation de l'environnement

Pour qu'un robot mobile puisse évoluer efficacement dans son environnement, il est essentiel de comprendre les éléments qui le composent. Traditionnellement, cette compréhension est obtenue grâce à une campagne de mesure exhaustive des différents éléments de l'environnement. Cependant, cette approche peut être laborieuse et nécessite des efforts importants.

Une alternative consiste à permettre au robot de construire dynamiquement son propre modèle d'environnement. Cependant, pour que la planification de trajectoire soit efficace, le robot doit disposer d'un modèle de l'espace d'évolution, ce qui limite souvent son utilisation.

En utilisant un modèle préalable de l'environnement et une loi d'évaluation des erreurs de représentation, le planificateur peut générer des sous-trajectoires réalisables dans certaines parties de l'environnement et ajuster les sous-trajectoires dans d'autres parties en fonction des informations locales obtenues par les capteurs environnementaux. Lors de l'exécution d'une trajectoire, le robot acquiert des informations qui lui permettent de reconstruire de manière récursive et aussi fidèle que possible le modèle de l'environnement en utilisant un algorithme approprié[18].

Planification de trajectoire

La planification de trajectoire constitue un défi crucial en robotique mobile, et elle nécessite des approches ingénieuses pour assurer la navigation efficace du robot dans différents environnements.

Lorsque le territoire est préalablement cartographié, la tâche est facilitée, permettant ainsi l'utilisation d'algorithmes de routage pour guider le robot vers sa destination de manière optimale.

Cependant, dans des environnements inconnus, la planification de trajectoire devient plus complexe. Le robot doit être équipé de capteurs performants pour analyser son environnement en temps réel, déterminer sa position relative à son objectif, et prendre des décisions tactiques pour choisir la meilleure trajectoire. Cette phase de localisation peut être réalisée grâce à diverses méthodes sophistiquées, telles que la triangulation des signaux des balises, la détection d'obstacles à distance, ou encore la construction progressive d'une carte du site en temps réel.

L'implémentation d'algorithmes avancés est alors primordiale pour guider le robot sur des trajectoires optimales. Toutefois, la convergence des algorithmes reste parfois un défi, pouvant nécessiter des solutions de secours ou même une intervention humaine.

La planification de trajectoire représente un domaine de recherche actif et passionnant en robotique mobile, où les ingénieurs s'efforcent de concevoir des solutions toujours plus sophistiquées et performantes. Ces avancées permettent aux robots mobiles de devenir de véritables compagnons d'exploration, capables de s'adapter de manière autonome à des environnements variés et d'affronter des situations complexes avec succès [19].

II.3.4 Les organes de sécurité

Lorsque des robots interagissent à proximité du personnel, la sécurité devient une préoccupation primordiale. Afin d'assurer un environnement sans danger, ces robots sont équipés de dispositifs de sécurité sophistiqués. Ces dispositifs jouent un rôle critique dans la protection des individus et des équipements.

Pour détecter les obstacles et éviter les collisions potentielles, des capteurs de haute précision sont habilement répartis tout autour du robot, garantissant une couverture étendue. Deux types de capteurs sont déployés à cette fin. Les capteurs proximétriques permettent de repérer les obstacles avant qu'une collision ne survienne. Dotés de technologies avancées telles que les ultrasons, les hyperfréquences et l'infrarouge, ces capteurs opèrent avec une fiabilité inégale.

Complétant cette gamme de dispositifs de sécurité, les capteurs de contact sont conçus pour détecter les collisions ou les impacts avec l'environnement. Ces capteurs multifonctionnels, sous différentes formes telles que contacts électriques intégrés aux pare-chocs du robot, résistances variables ou fibres optiques, assurent une réaction rapide face aux situations potentiellement critiques.

L'architecture intelligente de sécurité du robot intègre une redondance d'informations provenant des capteurs, renforçant ainsi la fiabilité et la réactivité du système global. Cette conception réfléchie garantit que le robot peut travailler aux côtés du personnel en toute confiance, minimisant efficacement les risques d'accidents et de dommages matériels.

Le schéma de la figure (II.6) illustre clairement l'agencement et le fonctionnement coordonné de ces dispositifs de sécurité essentiels, permettant une interaction harmonieuse et sécurisée entre les robots et les êtres humains.

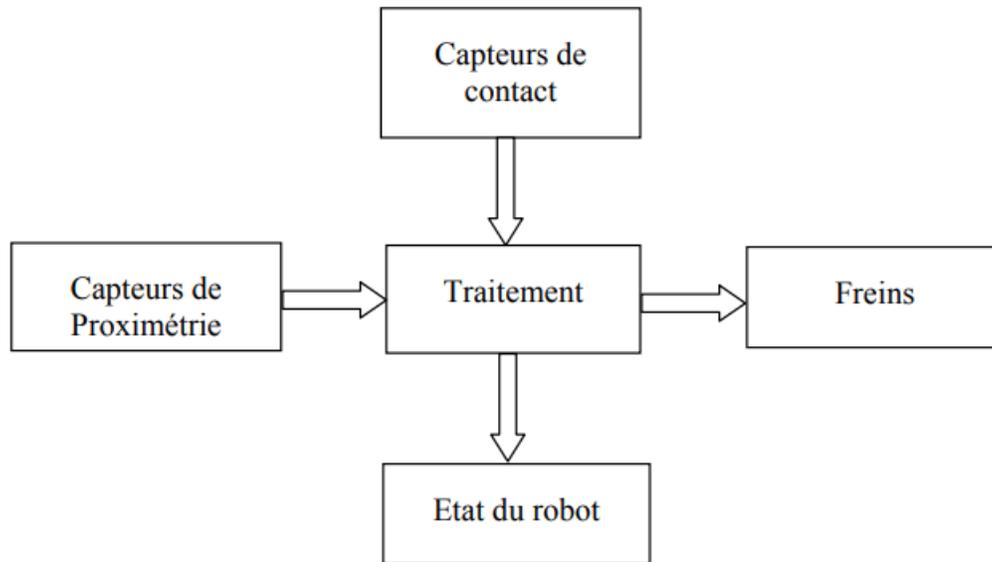


FIGURE II.6 – Synoptique de la sécurité

Le système de sécurité du robot comporte une surveillance continue de l'état de fonctionnement de tous ses organes. Le traitement des données issues des capteurs de sécurité suit plusieurs scénarios bien définis. Lorsque le capteur de contact est activé, le robot peut s'arrêter de manière permanente, s'immobiliser tant que le contact persiste, ou effectuer un mouvement opposé à la direction du contact.

En revanche, si un capteur de proximité détecte la présence d'une personne, plusieurs stratégies peuvent être mises en œuvre. Le robot peut choisir de s'immobiliser en attendant que la personne s'éloigne, ralentir son mouvement si la personne n'est pas trop proche, ou opter pour un itinéraire alternatif qui l'éloigne de la personne détectée.

L'architecture des robots mobiles est un élément essentiel qui détermine leur holonomie et leur classification en tant que robots mobiles à roues. Les composants clés tels que la structure mécanique, la motricité et le système de localisation, que nous avons abordés dans les sections précédentes, jouent un rôle déterminant dans leurs capacités. La manière astucieuse dont ces éléments sont conçus et intégrés a un impact direct sur la manœuvrabilité et l'autonomie du robot, lui permettant ainsi d'interagir intelligemment avec son environnement. Dans les prochaines sections, nous examinerons en détail l'influence de ces choix architecturaux sur les performances et les capacités spécifiques des robots mobiles à roues, ouvrant ainsi la voie à de passionnantes avancées technologiques.

II.4 Holonomie et non-holonomie

Dans le monde des robots La mobilité des robots mobiles constitue un aspect essentiel de leur architecture et de leur conception. Elle se divise en deux catégories principales en fonction de leur degré de liberté de mouvement : les robots holonomes et les robots non-holonomes[20].

II.4.1 Robots mobiles non-holonomes

Les robots mobiles non holonomes adoptent une configuration mécanique similaire à celle des véhicules conventionnels tels que les voitures, les bus et les camions. Ils sont équipés de roues motrices, de roues directrices et de roues porteuses, parfois combinant deux ou trois fonctions en une seule roue.

Cependant, contrairement aux robots holonomes, ces robots mobiles présentent des limitations pour effectuer des translations latérales, ce qui signifie qu'ils ont seulement deux degrés de liberté sur un plan : la translation (avancer ou reculer) et la rotation (tourner à droite ou à gauche). Leur configuration mécanique restreint leur capacité à se déplacer latéralement, ce qui rend leur retour à leur position initiale plus complexe sans des manœuvres spécifiques. Ces contraintes de mouvement nécessitent une planification plus précise pour garantir leur bon déplacement dans leur environnement.

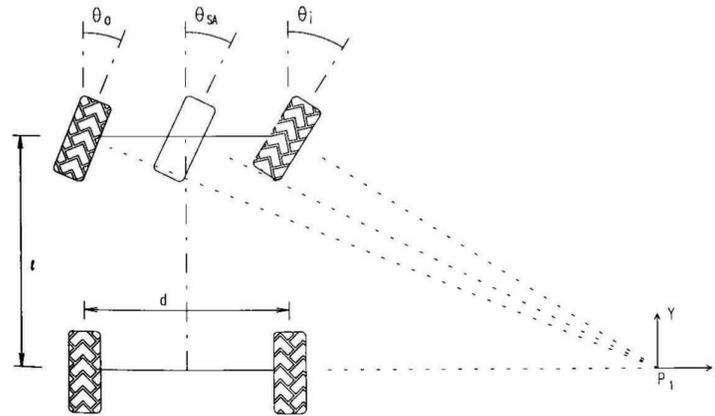


FIGURE II.7 – Exemple de plate-forme non holonome de type Ackerman.

II.4.2 Robots mobiles holonomes

En robotique, une plateforme est considérée comme holonome lorsque le nombre de degrés de liberté contrôlables est égal au nombre total de degrés de liberté. Dans le cas des robots mobiles se déplaçant sur un plan, cela signifie qu'ils peuvent se déplacer dans n'importe quelle direction indépendamment de leur orientation. Ces robots disposent de 3 degrés de liberté : deux pour les translations (avancer, reculer ou se déplacer latéralement) et un pour la rotation (tourner sur place).

Grâce à cette capacité holonome, le contrôle de ces robots est simplifié, car ils peuvent effectuer n'importe quel déplacement imaginable dans leur environnement. Cette propriété facilite grandement la planification de trajectoire, permettant aux robots holonomes de se déplacer avec agilité et d'effectuer des manœuvres précises pour atteindre leur destination.

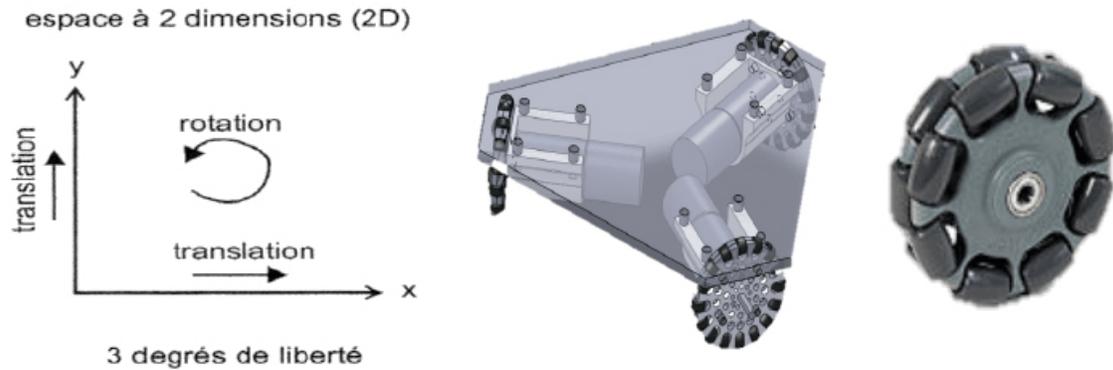


FIGURE II.8 – Exemple d'un robot mobile holonome

II.5 Classification de robot mobiles a roues

Les robots mobiles à roues peuvent être classés en quatre catégories principales, basées sur le nombre et la disposition des roues utilisées [21].

II.5.1 Robot unicycle

Un robot mobile est qualifié d'unicycle lorsqu'il est équipé de deux roues indépendantes pour la propulsion et qu'il possède également plusieurs roues passives pour assurer sa stabilité. Dans ce type de configuration, les roues passives ne sont pas directement impliquées dans la cinématique du robot. La figure (II.9) illustre un schéma représentatif d'un robot unicycle, où les roues passives ont été omises pour des raisons de clarté, mais elles sont judicieusement placées pour maintenir la stabilité du robot. Les robots unicycles sont largement utilisés en raison de leur construction simple et de leurs propriétés cinématiques intéressantes.

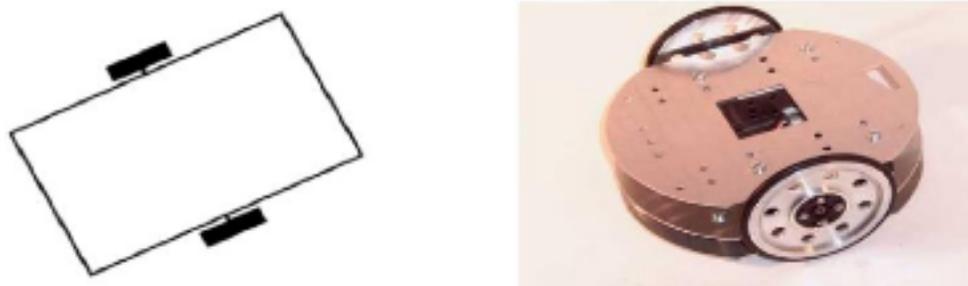


FIGURE II.9 – Robot de type unicycle.

II.5.2 Robot tricycle

Le robot tricycle est équipé de deux roues passives à l'arrière et d'une roue motrice à l'avant, ou inversement. Son mouvement est contrôlé par deux actions principales : la vitesse longitudinale et l'orientation de la roue orientable. Pour assurer sa localisation, ce type de robot est équipé d'un capteur d'orientation associé à la roue orientable, tandis que deux codeurs sont utilisés pour les roues motrices. Ces dispositifs permettent au robot de déterminer avec précision sa position et son orientation dans l'environnement, facilitant ainsi ses déplacements et ses interactions.

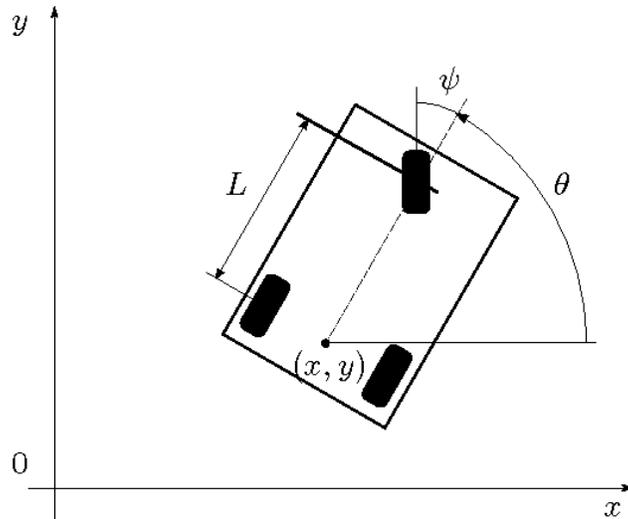


FIGURE II.10 – Robot de type tricycle

II.5.3 Robot voiture

Le robot de type "voiture" présente une configuration similaire à celle du tricycle, avec deux roues fixes placées sur un même axe et deux roues orientables également alignées sur un même axe. Cependant, ce type de robot se distingue par sa stabilité accrue grâce à un point d'appui supplémentaire.

Toutes les autres caractéristiques de ce type de robot sont semblables à celles du robot tricycle. En effet, le deuxième type de robot peut être transformé en robot tricycle en remplaçant simplement les deux roues avant par une seule roue centrale, tout en maintenant le centre de rotation inchangé.

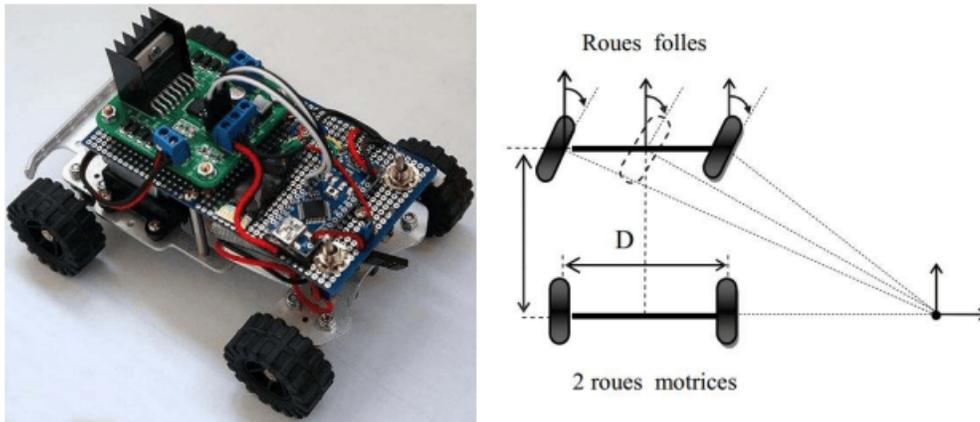


FIGURE II.11 – Robot de type voiture

II.5.4 Robot omnidirectionnel

Le robot omnidirectionnel se distingue par sa capacité à se déplacer librement dans toutes les directions. Sa configuration comprend généralement trois roues orientables décentrées, disposées de manière à former un triangle équilatéral.

L'un des principaux avantages de ce type de robot réside dans sa holonomie, qui lui permet d'effectuer des déplacements dans n'importe quelle direction avec aisance. Cependant, cette holonomie accrue entraîne une complexité mécanique plus élevée que d'autres types de robots.

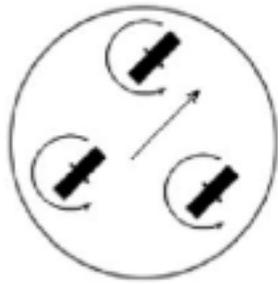


FIGURE II.12 – Robot de type omnidirectionnel

II.5.5 Avantages et Inconvénients

Dans cette section, nous examinerons attentivement les caractéristiques distinctives de chaque type de robot mobile, en mettant en évidence les avantages et les inconvénients associés. Cette évaluation approfondie permettra aux ingénieurs et concepteurs de mieux comprendre les spécificités de chaque configuration, facilitant ainsi la sélection d'un robot adapté à des tâches spécifiques et répondant aux exigences environnementales particulières.

Le tableau suivant contient les avantages et des inconvénient des différents types de robots mobile.

Type de robot	Avantages	Inconvénient
Robot unicycle	-stable -Rotation sur soi-même -Complexité mécanique faible	non holonome
Robot tricycle	-Complexité mécanique modérée	-non holonome -peu stable -pas de rotation sur soimême
Robot voiture	-complexité mécanique modérée -stable	-non holonome
Omnidirectionnel	Holonome Stable Rotation sur soi-même	Complexité mécanique importante

TABLE II.1 – Avantages et Inconvénients des Différents Types de Robots Mobiles

II.6 Méthodes de navigation

La navigation des robots mobiles comprend des stratégies qui leur permettent de se localiser dans leur environnement et de planifier efficacement des itinéraires du point A au point B. Pour atteindre cet objectif, diverses approches sont utilisées, et l'utilisation de cartes préétablies fait partie des techniques couramment adoptées. Le choix de la méthode de navigation dépend généralement de la précision requise pour la localisation du robot dans son environnement spécifique. Parmi les différentes approches disponibles, les suivantes sont couramment utilisées [22] :

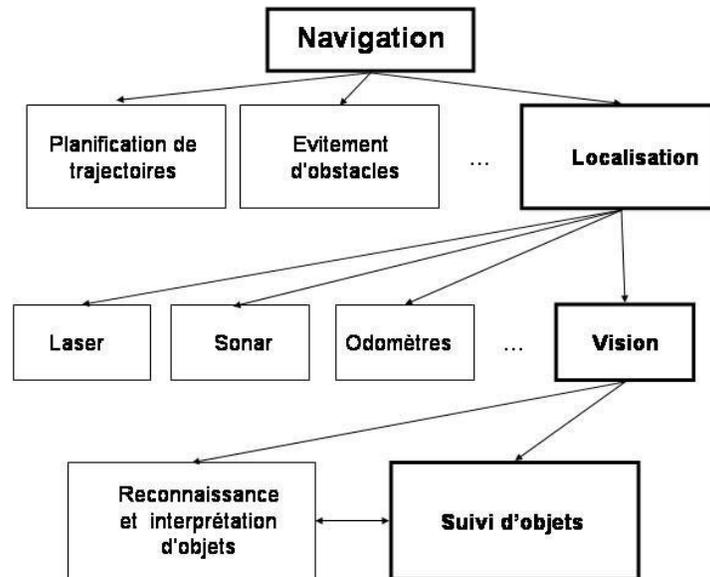


FIGURE II.13 – Schéma simplifié de tâches à l'intérieur de la navigation autonome

II.6.1 Navigation Métrique

La navigation métrique est une méthode de navigation couramment utilisée pour les robots mobiles, qui repose sur l'utilisation de capteurs pour mesurer les mouvements du robot. Cette méthode permet d'estimer la position et l'orientation du robot en se basant sur des données géométriques. Parmi les capteurs fréquemment utilisés pour la navigation métrique, on trouve les encodeurs de roues, les centrales inertiels (IMU) et les capteurs de vision.

Les algorithmes de navigation métrique s'appuient généralement sur des modèles cinématiques du robot et utilisent des méthodes d'estimation de l'état telles que les filtres de Kalman, les filtres de particules ou les filtres d'observation étendus pour estimer en temps réel la position et l'orientation du robot.

La navigation métrique trouve de nombreuses applications dans le domaine industriel, notamment pour la navigation de robots d'entrepôt, de véhicules autonomes et de drones. Cette approche permet d'assurer une navigation précise et fiable, favorisant ainsi l'efficacité et la sécurité des opérations robotiques dans des environnements variés.

II.6.2 Navigation par action associé à un lieu

La navigation par action associée à un lieu est une méthode de navigation utilisée pour les robots mobiles. Elle consiste à associer une action spécifique à chaque lieu visité par le robot, permettant ainsi de suivre une séquence prédéfinie d'actions pour atteindre un objectif donné. Cette approche de navigation

est souvent combinée avec des techniques de localisation afin de permettre au robot de déterminer sa position par rapport à la séquence d'actions à effectuer.

En utilisant la navigation par action associée à un lieu, le robot peut se déplacer vers sa destination en suivant un plan d'action clairement défini. À chaque étape du parcours, il sait quelle action doit être réalisée en fonction de sa position dans l'environnement. Cette méthode de navigation est particulièrement utile pour les robots qui doivent suivre des trajectoires spécifiques ou effectuer des tâches précises dans des environnements structurés.

En combinant des actions spécifiques à des lieux particuliers, le robot peut atteindre efficacement son objectif, tout en s'adaptant à d'éventuelles modifications de la séquence d'actions en fonction de sa localisation précise. Cette approche offre une solution pratique pour des applications telles que la patrouille de zones prédéfinies, la livraison de colis ou l'exploration de sites spécifiques.

II.6.3 Navigation topologique

La navigation topologique est une méthode de navigation qui se fonde sur la représentation topologique de l'environnement. Cette approche consiste à créer un graphe reliant les lieux importants de l'environnement, appelés nœuds, ainsi que les connexions entre ces nœuds, appelées arêtes. Grâce à cette représentation simplifiée, la planification de trajectoire pour le robot mobile est grandement facilitée.

Les robots utilisant la navigation topologique évoluent généralement dans des environnements structurés et prévisibles tels que les entrepôts, les usines ou les bâtiments. La représentation topologique permet de créer une carte abstraite de l'environnement, ignorant certains détails géométriques spécifiques. Cela offre l'avantage de réduire la complexité de la carte, ce qui facilite la planification des mouvements du robot.

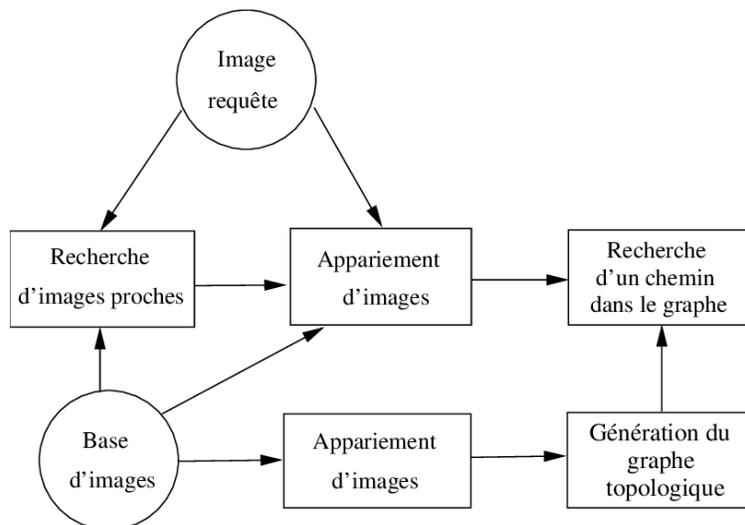


FIGURE II.14 – Opérations réalisées pour définir une tâche de navigation

II.6.4 Navigation géométrique

La navigation géométrique est une méthode de déplacement utilisée par les robots mobiles pour se déplacer dans leur environnement. Elle repose sur le calcul précis des déplacements et des rotations nécessaires pour atteindre une position ou une orientation désirée en utilisant des données géométriques de l'environnement, telles que des distances et des angles.

Les méthodes de navigation géométrique peuvent être divisées en deux catégories principales : la navigation basée sur la trajectoire et la navigation basée sur le point de référence. La navigation basée sur la trajectoire consiste à suivre une trajectoire planifiée à l'avance, tandis que la navigation basée sur le point de référence consiste à se déplacer vers un point de référence spécifique dans l'environnement.

Les algorithmes de navigation géométrique les plus couramment utilisés sont les algorithmes de contrôle de trajectoire, tels que les algorithmes de commande proportionnelle-intégrale-dérivée (PID) et les algorithmes de commande de mouvement. Ces algorithmes utilisent des capteurs tels que des encodeurs de roue et des capteurs de distance pour suivre précisément une trajectoire planifiée à l'avance.

D'autre part, la navigation basée sur le point de référence utilise des capteurs tels que des caméras, des capteurs LiDAR et des capteurs de distance pour détecter des points de référence spécifiques dans l'environnement. En utilisant ces points de référence, les algorithmes de navigation basée sur le point de référence calculent les déplacements et les rotations nécessaires pour atteindre une position ou une orientation désirée avec une grande précision. Cette approche est particulièrement utile dans des environnements où des caractéristiques spécifiques doivent être atteintes, telles que des points d'intérêt ou des cibles spécifiques.

II.7 Modélisation du robot mobile à 4 roues

Dans notre projet pratique, nous avons choisi de modéliser le robot mobile en utilisant une configuration à quatre roues, bien que le robot physique possède six roues. Cette décision repose sur la similitude cinématique et dynamique entre les deux configurations. Les principes de mouvement et de comportement sont fondamentalement les mêmes, permettant ainsi une modélisation simplifiée sans compromettre les performances globales du robot.

La configuration à quatre roues nous offre également une approche bien documentée et largement étudiée dans la littérature scientifique et l'industrie de la robotique. Cela facilite notre travail de conception et nous permet de nous concentrer sur des aspects clés tels que la planification de trajectoire, la localisation et la navigation autonome.

II.7.1 Définitions

Dans notre étude, nous considérons deux repères : un repère fixe $R = (O, \vec{x}, \vec{y}, \vec{z})$ et un repère mobile $R' = (O', \vec{x}', \vec{y}', \vec{z}')$ attaché au robot. Le repère mobile est généralement choisi avec O' situé à un point remarquable de la plate-forme du robot, tel que le centre de l'axe des roues motrices si existant, comme illustré dans la Figure (II.15) [23].

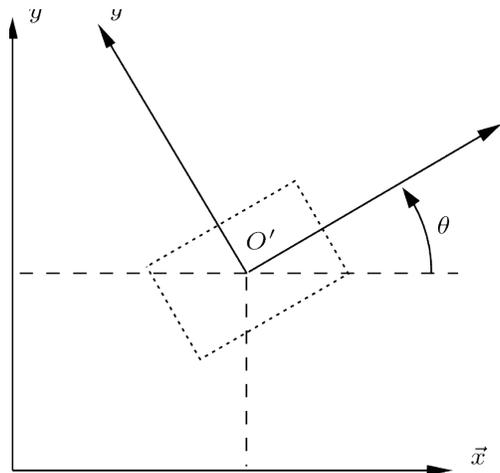


FIGURE II.15 – Repérage d'un robot mobile

Pour décrire la situation du robot, nous utilisons un vecteur ϵ représentant sa posture, qui se compose de trois éléments :

$$\epsilon = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (\text{II.1})$$

Ici, x et y représentent respectivement l'abscisse et la coordonnée de l'origine locale O' dans le repère fixe R , tandis que θ représente l'angle (\vec{x}, \vec{x}') . La situation du robot est donc définie dans un espace M de dimension $m = 3$, qui peut être comparé à l'espace opérationnel d'un manipulateur plan.

La configuration d'un système mécanique [24] est établie lorsque la position de tous ses points est connue par rapport à un repère spécifié. Pour un bras manipulateur, cette notion est clairement définie par les positions angulaires de ses articulations. Cependant, dans le cas d'un robot mobile, la configuration peut être décrite de manière plus ou moins précise, comme nous le verrons par la suite. Dans tous les cas, la configuration d'un robot mobile est représentée par un vecteur q .

$$q = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} \quad (\text{II.2})$$

Ce vecteur q comprend n coordonnées, appelées coordonnées généralisées, qui définissent entièrement la configuration du robot. La configuration du robot mobile est donc définie dans un espace N de dimension n , appelé l'espace des configurations.

II.7.2 Roulement sans glissement

La locomotion à l'aide de roues exploite la friction au contact entre roue et sol. Dans des conditions optimales, le roulement sans glissement (R.S.G.) se produit lorsque la roue se déplace sur le sol sans glisser, ce qui signifie que la vitesse relative de la roue par rapport au sol au point de contact est nulle. Pour vérifier théoriquement cette condition, il est nécessaire de considérer les hypothèses suivantes :

- Le contact entre la roue et le sol est considéré comme ponctuel, c'est-à-dire qu'il se produit en un seul point à la surface de la roue.
- Les roues sont considérées comme indéformables, ayant un rayon constant noté " r ". Cette hypothèse suppose que les roues ne subissent aucune déformation pendant le mouvement.

En pratique, lorsqu'une roue est en contact avec le sol, de légères variations de vitesse peuvent entraîner des glissements. De plus, il est important de noter que l'hypothèse d'indéformabilité des roues n'est généralement pas valable, surtout lorsque les roues sont équipées de pneus. Cependant, malgré ces réalités pratiques, pour des raisons de simplicité et d'analyse, nous considérerons toujours le roulement sans glissement (R.S.G.) et supposerons que le sol est parfaitement plat.

Sur le plan mathématique, la condition de roulement sans glissement (R.S.G.) peut être exprimée pour une roue donnée. Soit P le centre de la roue, Q le point de contact entre la roue et le sol, ϕ l'angle de rotation propre de la roue, et θ l'angle entre le plan de la roue et le plan (O, \vec{x}, \vec{z}) , comme illustré dans la figure (II.16). La condition de vitesse relative nulle au point de contact \vec{v}_Q entre la roue et le sol conduit à une relation vectorielle entre la vitesse \vec{v}_P du centre P de la roue et le vecteur vitesse de rotation ω de la roue [23].

$$\vec{v}_Q = \vec{v}_P + \omega \wedge \overrightarrow{PQ} = \vec{0} \quad (\text{II.3})$$

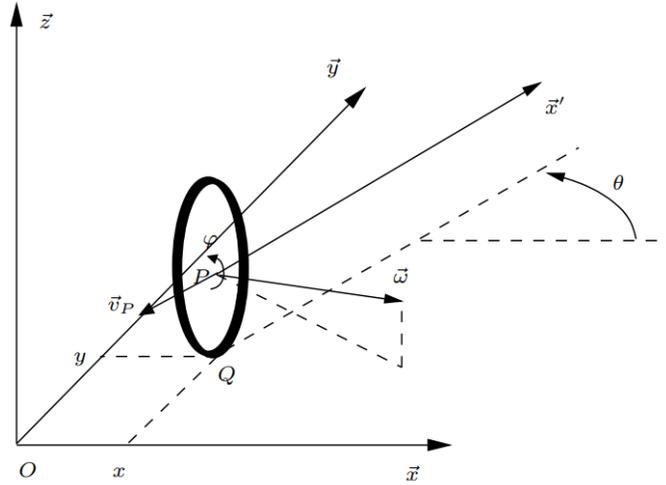


FIGURE II.16 – Repérage d'un robot mobile

Les points P et Q ont pour coordonnées respectives $(x \ y \ r)^T$ et $(x \ y \ 0)^T$. Il vient alors :

$$\dot{x} \vec{x} + \dot{y} \vec{y} + (\dot{\theta} \vec{z}) + \dot{\phi}(\sin \theta \vec{x} - \cos \theta \vec{y}) \wedge (-r \vec{z}) = \vec{0} \quad (\text{II.4})$$

$$(\dot{x} + r\dot{\phi} \cos \theta) \vec{x} + (\dot{y} + r\dot{\phi} \sin \theta) \vec{y} = \vec{0} \quad (\text{II.5})$$

Ceci nous donne le système de contraintes scalaires :

$$\dot{x} + r\dot{\phi} \cos \theta = 0 \quad (\text{II.6})$$

$$\dot{y} + r\dot{\phi} \sin \theta = 0 \quad (\text{II.7})$$

que l'on peut transformer pour faire apparaître les composantes de vitesse dans le plan de la roue d'une part et perpendiculairement à la roue d'autre part :

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \quad (\text{II.8})$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta = -r\dot{\phi} \quad (\text{II.9})$$

Ces contraintes traduisent le fait que le vecteur \vec{v}_p soit dans le plan de la roue et ait pour module $r\dot{\phi}$

II.7.3 Différentes forces qui agissent sur le système

Les différentes forces qui agissent sur le robot mobile à quatre roues type différentiels et ces principales forces et leurs effets sur le système sont représentés sur la figure (II.17) [25].

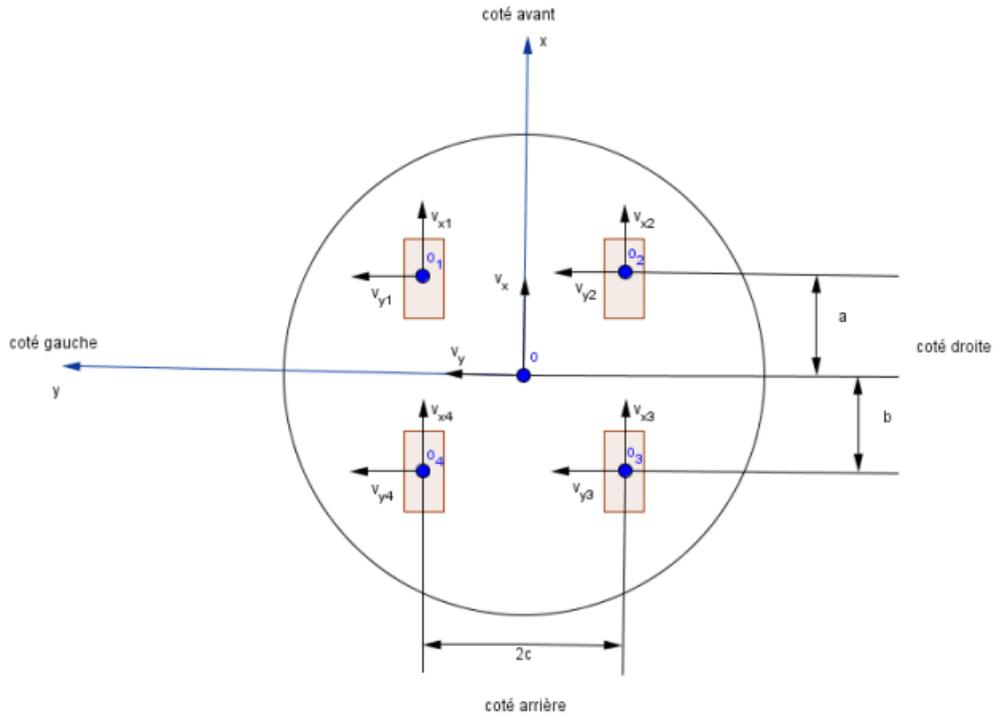


FIGURE II.17 – Représentation des vitesses de chaque roue et celle du robot

Nous donnerons par la suite les différentes forces qui régissent le comportement dynamique de notre système :

Tout d'abord nous définissons toutes les forces qui agissent sur chaque roue, ensuite au robot.

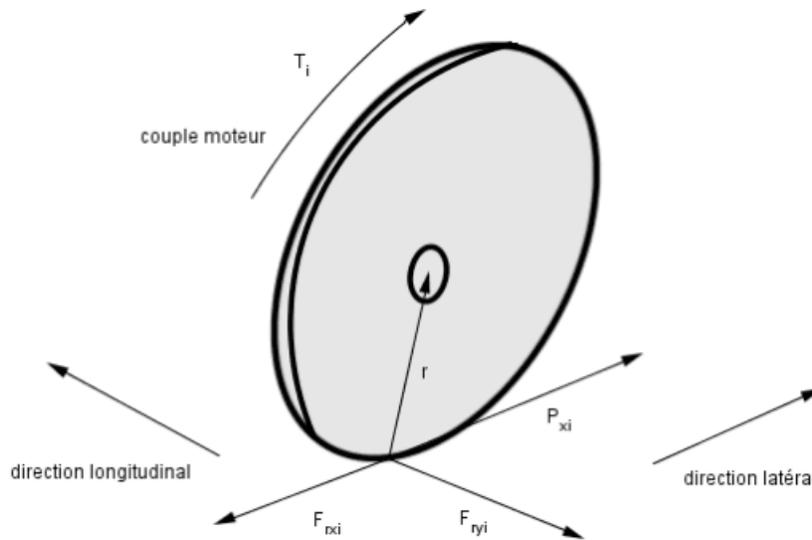


FIGURE II.18 – Représentation des différentes forces qui agissent sur une des roues du robot mobile.

— τ_i : le couple moteur de chaque roue.

- P_{xi} : la force générée par le couple moteur de chaque roue qui permet de pousser chaque roue et qui est générée par le couple moteur τ_i est défini par :

$$P_{xi} = \frac{\tau_i}{r} \quad (\text{II.10})$$

- F_{rxi} : la force de frottement latéral de la roue avec le sol.
- F_{ryi} : la force de frottement longitudinal de la roue avec le sol.
- r : le rayon de la roue.

II.7.4 Modèle cinématique

La cinématique de la plateforme du robot mobile est illustrée dans la Figure (II.19) [26].

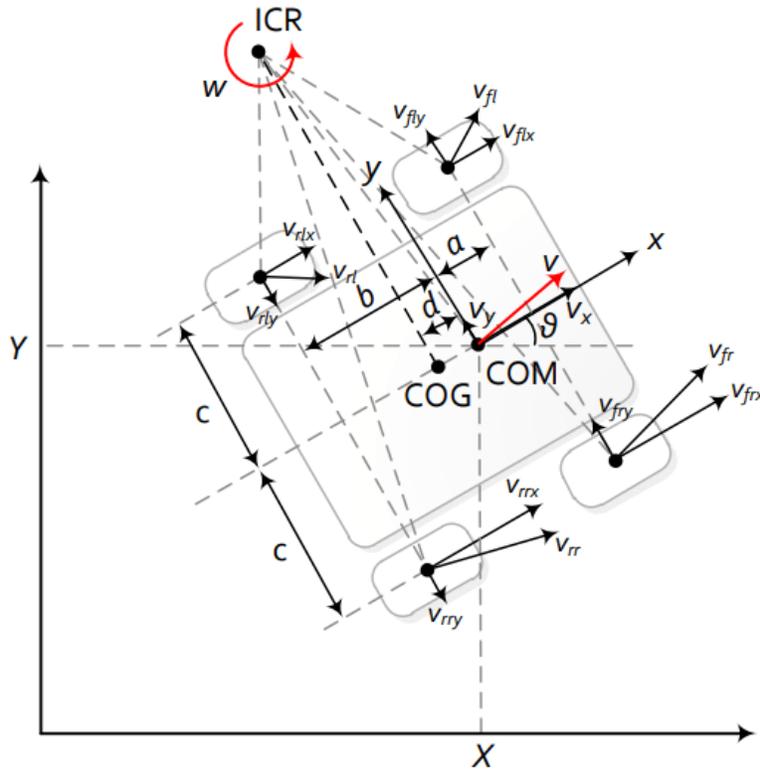


FIGURE II.19 – La cinématique de la plateforme du robot mobile à 4 roues

Le vecteur de configuration du véhicule dans le repère de coordonnées global est le suivant :

$$q = [X \quad Y \quad \theta]^T \in \mathbb{R}^3 \quad (\text{II.11})$$

où X, Y et θ représentent respectivement la position et l'orientation du véhicule. La transformation entre les vitesses locales définies dans les repères de coordonnées locales attachés au centre de masse (COM) du véhicule et les vitesses généralisées est la suivante :

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix} \quad (\text{II.12})$$

Le centre de masse (COM) se situe à une distance d dans l'intervalle $(0, b)$ à partir du centre de géométrie (COG). Le point de rotation instantané (ICR) se trouve sur l'axe qui intersecte le COG, comme indiqué dans

la figure (II.19). Dans ce cas, la contrainte opérationnelle non holonome qui limite le glissement latéral est définie comme suit :

$$v_y - dw = 0 \quad (\text{II.13})$$

$$\begin{bmatrix} -\sin \theta & \cos \theta & -d \end{bmatrix} \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T = A(q)\dot{q} = 0 \quad (\text{II.14})$$

Ensuite, nous pouvons réécrire l'équation (II.12) sous la forme ci-dessous, où $S(q)$ est une matrice appartenant à l'ensemble, et η est appelé le vecteur d'entrée de contrôle au niveau cinématique, défini comme suit :

$$\dot{q} = S(q)\eta \quad (\text{II.15})$$

$$S(q) = \begin{bmatrix} \cos \theta & -d \sin \theta \\ \sin \theta & d \cos \theta \\ 0 & 1 \end{bmatrix}, \eta = \begin{bmatrix} v_x & w \end{bmatrix}^T \quad (\text{II.16})$$

Et comme les colonnes de $S(q)$ se trouvent toujours dans l'espace nul de $A(q)$, l'expression suivante est satisfaite :

$$S^T A^T = 0 \quad (\text{II.17})$$

II.7.5 Modèle dynamique

La dynamique de la plateforme du robot mobile est illustrée dans la figure (II.20). Tout d'abord, nous présentons les forces latérales gauche et droite, puis nous décrivons la dynamique des roues [26].

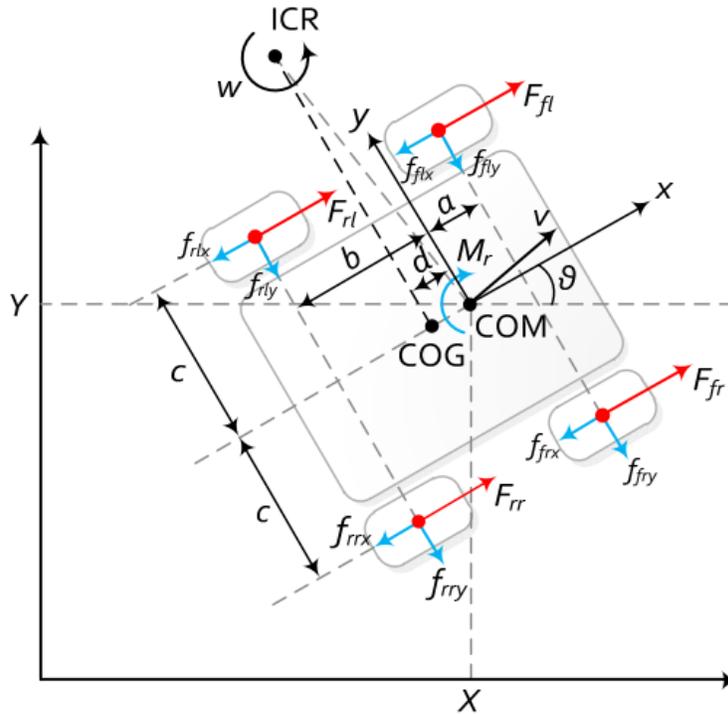


FIGURE II.20 – La dynamique de la plateforme du robot mobile à 4 roues

$$F_L/2 = F_{fl}, F_R/2 = F_{fr} = F_{rr} \quad (\text{II.18})$$

$$I_W \dot{W}_W = \tau - DF \quad (\text{II.19})$$

- I_W : l'inertie de la roue
- $W_W = [W_{fl} \ W_{rl} \ W_{fr} \ W_{rr}]$: le vecteur vitesse angulaire de la roue
- $\tau = [\tau_{fl} \ \tau_{rl} \ \tau_{fr} \ \tau_{rr}]$: le vecteur de couple de roue
- $F =$: le vecteur du force
- D : la matrice de conversion force-couple, définie comme :

$$D = \frac{r}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}^T \quad (\text{II.20})$$

où r représente le rayon de la roue. Ensuite, les équations du mouvement du robot dans les coordonnées globales sont présentées.

$$m\ddot{X} = (M_L + M_R) \cos \theta - (f_x \cos \theta - f_y \sin \theta) \quad (\text{II.21})$$

$$m\ddot{Y} = (M_L + M_R) \sin \theta - (f_x \sin \theta + f_y \cos \theta) \quad (\text{II.22})$$

$$I\ddot{\theta} = c(-M_L + M_R) - M_r \quad (\text{II.23})$$

- m : la masse de vehicule
- I : l'inertie du véhicule autour de l'axe z
- f_x et f_y : il sont les forces de frottement de roulement et de glissement respectivement
- M_r : le moment résistif autour de l'axe z

Les forces de frottement de roulement sont très faibles par rapport aux forces de frottement de glissement dans le cas idéal. Un modèle réaliste pour le frottement de roulement en cas de glissement peut être trouvé. Nous définissons les forces de frottement pour la seule roue avant gauche comme suit :

$$f_{flx} = \mu_x N_{fl} \text{sgn}(v_{flx}) \quad (\text{II.24})$$

$$f_{fly} = \mu_y N_{fl} \text{sgn}(v_{fly}) \quad (\text{II.25})$$

Les forces normales agissant sur les points de contact roue-sol dues à la gravité sont calculées comme suit :

$$N_{fl} = N_{fr} = \frac{b}{a+b} \frac{mg}{2} \quad (\text{II.26})$$

$$N_{rl} = N_{rr} = \frac{a}{a+b} \frac{mg}{2} \quad (\text{II.27})$$

et enfin, le moment de résistance est calculé de la manière suivante :

$$M_r = a[f_{fly} + f_{fry}] - b[f_{rly} + f_{rry}] + c[f_{frx} + f_{rrx} - f_{flx} - f_{rlx}] \quad (\text{II.28})$$

La forme générale de la dynamique du véhicule, y compris la contrainte non holonomique utilisant le principe d'Euler-Lagrange et l'introduction d'un vecteur supplémentaire pour représenter les perturbations est la suivante :

$$M(q)\ddot{q} + R(\dot{q}) + F_d = B(q)F + A^T \lambda \quad (\text{II.29})$$

- M : la matrice de masse et d'inertie
- R : le vecteur des forces et couples résistants
- F_d : le vecteur de perturbations
- B : la matrice d'entrée
- F : est appelée comme entrée de commande au niveau dynamique précédemment défini
- A : A est le vecteur de contrainte
- λ : est le vecteur des multiplicateurs de Lagrange.

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \quad (\text{II.30})$$

$$B(q) = \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ -c & c \end{bmatrix} \quad (\text{II.31})$$

II.8 Conclusion

La deuxième partie de ce mémoire nous a permis d'explorer en détail la modélisation des robots mobiles à roues. Nous avons mis l'accent sur les aspects essentiels de la représentation mathématique du mouvement et du contrôle de ces robots. Toutefois, nous avons également relevé le défi de traiter de vastes quantités d'informations, ce qui nécessite une puissance de calcul importante dans le domaine de la robotique mobile.

Passons maintenant à l'exploration approfondie de la localisation des robots mobiles, en abordant les mouvements, le traitement d'images, la technique de flux optique. Cela nous aidera à comprendre comment les robots déterminent leur position et à créer des systèmes de navigation avancés, en mettant en avant le rôle crucial du filtre Kalman.

Chapitre III

Généralités sur la localisation des robots mobiles

III.1 Introduction

Le problème de localisation en robotique mobile réside dans la détermination précise de la posture d'un robot mobile, comprenant sa position et son orientation, dans son environnement. Cependant, il n'existe pas des capteurs parfaits permettant une mesure directe et sans erreur de cette posture. Même si le GPS est un système capable de mesurer directement la position, sa précision est de l'ordre du mètre et il n'est pas adapté aux environnements clos tels que l'intérieur de bâtiments ou les zones forestières.

Pour résoudre cette problématique, la localisation en robotique mobile requiert l'exploitation exclusive d'une source de données spécifique. Dans ce contexte, l'utilisation du flux optique combiné avec les données d'orientation issues du capteur inertiel se révèle cruciale pour obtenir une estimation fiable de la posture du robot. Cette approche technique permet au robot de réaliser une localisation efficace en se basant sur une seule source d'information, ce qui évite la complexité associée à la fusion de données provenant de diverses sources, une situation fréquente lors de l'utilisation de multiples capteurs ou de balises.

Ce chapitre constitue une introduction au domaine de la localisation en robotique mobile. Il met en évidence l'importance de connaître la position et l'orientation du robot par rapport à son environnement, et souligne le rôle central de la localisation dans la navigation autonome des robots. Nous explorerons également les différentes approches de localisation ainsi que les types de capteurs associés à ce processus.

III.2 Localisation d'un robot mobile

Dans n'importe quel domaine d'application, la fonctionnalité essentielle d'un robot mobile est de posséder un système permettant une certaine autonomie dans la localisation et la navigation. Pour simplifier, le robot doit être capable de répondre à trois questions fondamentales : "Où suis-je?", "Où vais-je?" et "Comment y aller?". La première question concerne le problème de la localisation, tandis que les deux autres sont liées à la planification de trajectoire et à la navigation. La réussite des tâches de planification et de navigation dépend grandement de la qualité de la localisation.

Dans cette étude, nous nous concentrons sur les robots mobiles qui évoluent dans un plan en deux dimensions (2D), englobant ainsi une grande partie des systèmes existants. La localisation du robot consiste alors à déterminer trois paramètres : les coordonnées cartésiennes pour la position et un angle pour l'orientation. Formellement, cette tâche de localisation implique de calculer la transformation entre un repère lié au robot et un repère lié à l'environnement, comme illustré dans la Figure III.1 [27].

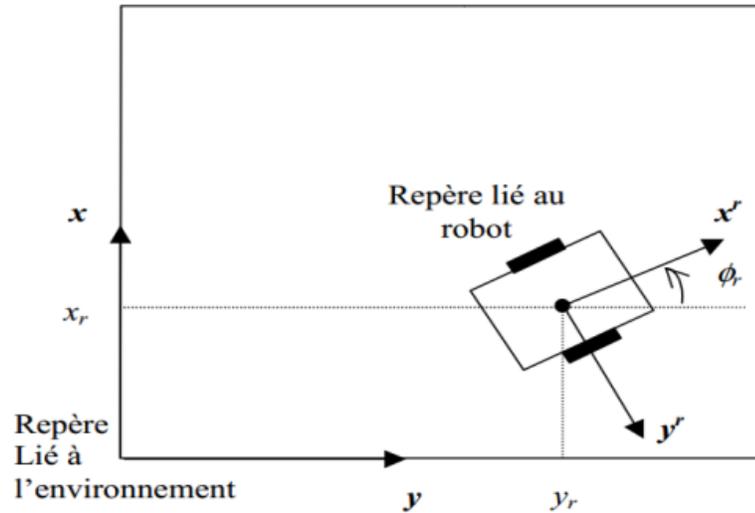


FIGURE III.1 – Approches de Localisation pour les Robots Mobiles

Plusieurs techniques et méthodes ont été développées pour permettre à un robot mobile de déterminer sa position dans son environnement. À ce jour, ces techniques peuvent être regroupées en deux catégories principales : les méthodes relatives et les méthodes de localisation absolue.

Les méthodes relatives se basent sur des informations internes au robot, telles que les mesures de déplacement des roues ou des capteurs inertiels, pour estimer sa position et son orientation par rapport à sa position initiale. Par exemple, la localisation odométrique utilise les rotations des roues pour estimer le déplacement du robot, tandis que la localisation inertielle utilise des capteurs inertiels tels que les accéléromètres et les gyroscopes pour suivre les mouvements du robot (voir section I.7).

D'un autre côté, les méthodes de localisation absolue font appel à des références externes pour déterminer la position du robot de manière absolue dans l'environnement. Par exemple, la localisation basée sur la vision utilise des caméras pour détecter des marqueurs visuels ou des caractéristiques de l'environnement, tandis que la localisation par GPS se sert des signaux satellites pour estimer la position géographique du robot (voir section I.7).

Parmi ces différentes méthodes de localisation, notre étude se concentre particulièrement sur la méthode du flux optique, qui est une approche de vision par ordinateur. Cette méthode utilise le traitement d'image pour estimer le mouvement apparent des objets dans une séquence d'images capturées par une caméra embarquée sur le robot. En analysant les variations d'intensité des pixels entre les images successives.

III.3 Traitement d'image

Le traitement d'image est un domaine passionnant de l'informatique et des mathématiques appliquées, dédié à la manipulation et à l'analyse d'images numériques. Il vise à améliorer, modifier ou extraire des informations utiles à partir de ces images. Cette discipline spécifique du traitement du signal opère dans le domaine numérique, ce qui lui permet d'utiliser des méthodes algorithmiques sophistiquées pour traiter les données visuelles. Le traitement d'image offre ainsi d'innombrables possibilités pour améliorer la qualité des images, effectuer des reconnaissances de motifs, détecter des objets et bien plus encore [28].

III.4 Image Numérique

L'image numérique est une matrice codée sous un certain nombre de bit, Les images numériques peuvent être acquises par plusieurs dispositifs qui s'adaptent aux domaines d'application.

L'image numérique est une matrice dont les éléments sont des pixels, chaque pixel est repère par sa position spatiale et sa luminance. Cette luminance indique le niveau de gris (voir la figure III.2) pour les images à niveau de gris et la couleur pour les images en couleurs [29].

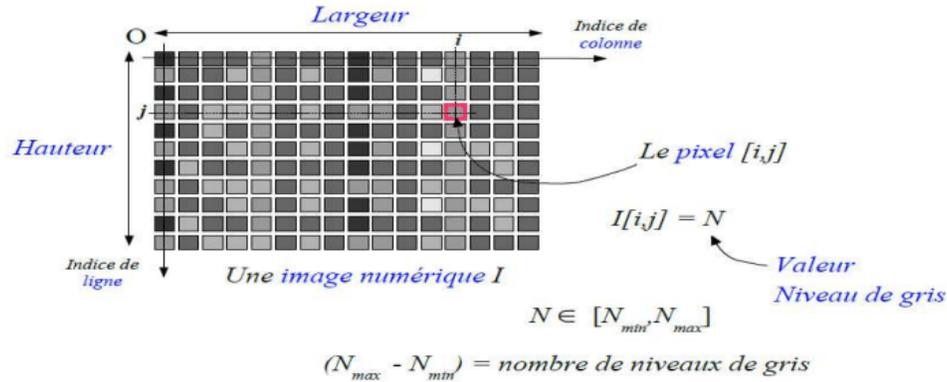


FIGURE III.2 – Les pixels de l'image

Les images numériques peuvent être classées en trois types principaux : les images binaires, les images en niveaux de gris et les images en couleur [27].

III.4.1 Image Binaire

L'image binaire est une forme d'image numérique où chaque pixel possède deux valeurs distinctes, généralement représentées par le noir (valeur 0) et le blanc (valeur 1). Elle est couramment utilisée pour représenter des éléments graphiques simples, tels que des dessins en noir et blanc, des logos, des symboles ou des masques de segmentation. En d'autres termes, chaque pixel de cette image est soit entièrement noir, soit entièrement blanc. Cette représentation binaire est particulièrement utile pour des applications nécessitant une distinction claire entre deux états distincts.

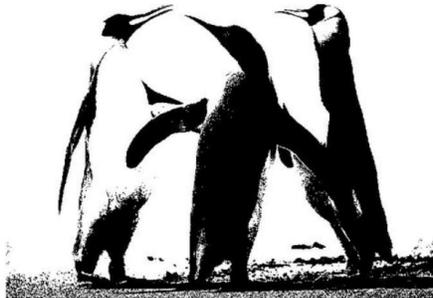


FIGURE III.3 – Exemple d'un image binaire

III.4.2 Image à Niveaux de Gris

L'image en niveaux de gris est un type d'image numérique qui permet à chaque pixel de prendre une valeur d'intensité allant du noir absolu (0) au blanc absolu (255), sans tenir compte des informations de

couleur. Dans une image en niveaux de gris, chaque valeur de pixel représente son niveau d'intensité lumineuse, et les valeurs intermédiaires correspondent à différentes nuances de gris entre le noir et le blanc. Par exemple, une valeur de pixel de 128 indique un gris moyen. Ce type d'image est couramment utilisé dans divers domaines tels que la photographie en noir et blanc, l'imagerie médicale, l'analyse d'images et la vision par ordinateur. Sa capacité à représenter des variations subtiles d'intensité lumineuse en fait un outil précieux pour de nombreuses applications.



FIGURE III.4 – Exemple d'une image gris

III.4.3 Image en Couleur

Une image en couleur est un type d'image numérique dans lequel chaque pixel est caractérisé par trois valeurs : une pour le canal rouge (r), une pour le canal vert (v), et une pour le canal bleu (b). Chacune de ces valeurs est un entier compris entre 0 et 255, déterminant l'intensité des couleurs primaires (rouge, vert, bleu) qui composent le pixel. Par exemple, un pixel avec les valeurs $(r, v, b) = (255, 0, 0)$ représente une couleur purement rouge, tandis qu'un pixel avec les valeurs $(0, 255, 0)$ représente une couleur purement verte, et ainsi de suite.

L'affichage d'une image en couleur sur un écran utilise les règles de la synthèse additive des couleurs, où les intensités des trois canaux sont combinées pour former une large gamme de couleurs. Par exemple, un pixel avec les valeurs $(r, v, b) = (255, 0, 255)$ est un mélange de rouge et de bleu, ce qui donne une couleur jaune à l'écran.

Les images en couleur sont largement utilisées dans divers domaines tels que la photographie, la vidéo, le cinéma, les applications graphiques et la vision par ordinateur

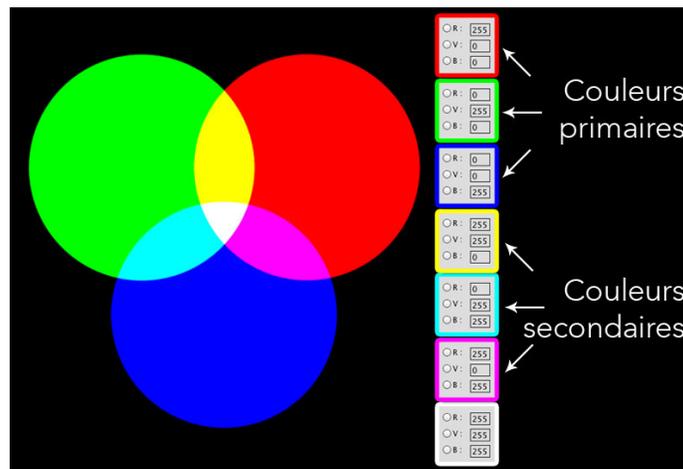


FIGURE III.5 – Le codage RVB

III.5 Vision assisté par ordinateur (VAO)

La vision assistée par ordinateur (VAO) est une branche de l'informatique qui vise à doter les ordinateurs de capacités de perception visuelle similaires à celles de l'être humain. Elle permet à l'ordinateur d'analyser et d'interpréter des images brutes afin d'extraire des informations utiles pour la prise de décisions. Cependant, la VAO présente des défis importants en raison de la nature des images 2D fournies par les capteurs, qui ne sont que des projections du monde 3D. Cette transformation 2D/3D entraîne une perte d'informations significative, et les images capturées peuvent être affectées par des imperfections telles que la numérisation des capteurs, les déformations dues aux objectifs et les bruits. Malgré ces difficultés, la vision assistée par ordinateur offre un potentiel considérable pour de nombreuses applications, allant de la reconnaissance d'objets à la navigation autonome des robots.

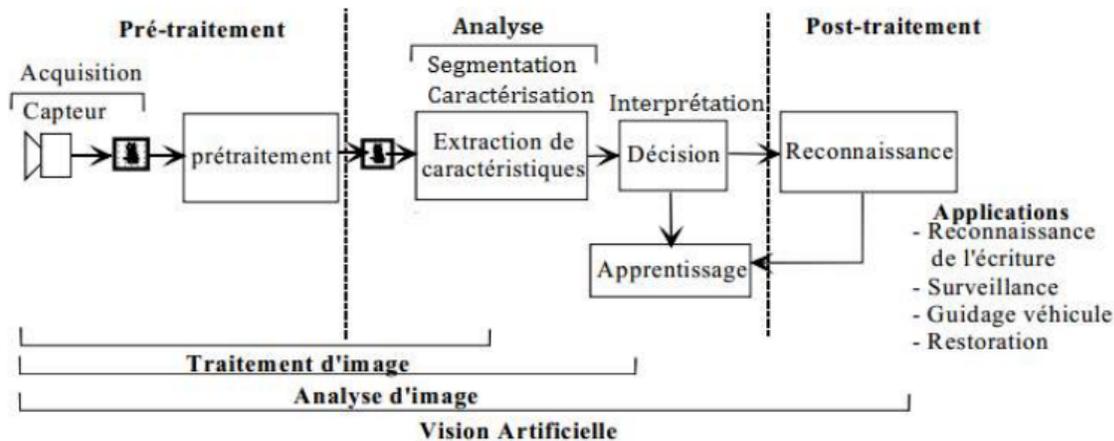


FIGURE III.6 – Le processus de la vision assistée par ordinateur (VAO)

III.5.1 Acquisition

Lorsqu'on procède à l'acquisition d'une image, il est nécessaire de suivre deux étapes essentielles pour convertir une image analogique en une version numérique : l'échantillonnage et la quantification [27].

L'échantillonnage

L'échantillonnage représente le processus de discrétisation spatiale d'une image, où chaque pixel se voit attribuer une valeur unique. Cette étape consiste à convertir l'image continue en une représentation discrète, où les pixels sont placés à des positions spécifiques sur une grille. Si l'image est déjà discrétisée, on peut également appliquer le sous-échantillonnage pour réduire le nombre de pixels tout en préservant les informations essentielles. Le sous-échantillonnage permet ainsi de représenter une version réduite de l'image originale, ce qui peut être utile dans certaines situations où la résolution peut être réduite sans affecter considérablement la qualité de l'analyse.

La quantification

La quantification concerne la discrétisation des niveaux de couleur ou de niveaux de gris de l'image. Chaque pixel se voit attribuer une valeur spécifique dans une plage prédéfinie de valeurs. Par exemple, dans une image en niveaux de gris, la quantification limitera le nombre de niveaux de gris disponibles pour chaque pixel. Cette étape vise à réduire le nombre de valeurs différentes que chaque pixel peut prendre.

Bien que cela entraîne une perte d'informations subtiles, la quantification est essentielle pour représenter une image analogique sous forme numérique avec un nombre fini de niveaux de couleur ou de niveaux de gris.

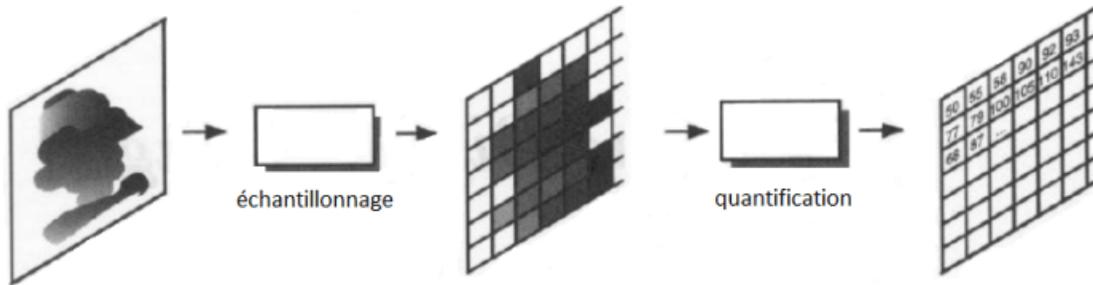


FIGURE III.7 – les étapes de l'acquisition de l'image

Ces étapes d'échantillonnage et de quantification sont fondamentales dans la numérisation d'images, car elles permettent de représenter des images analogiques en format numérique, rendant ainsi possible leur traitement et leur analyse à l'aide d'algorithmes et de techniques de traitement d'image. Cela joue un rôle crucial dans divers domaines tels que la vision par ordinateur, l'imagerie médicale, la photographie numérique, etc.

III.5.2 Prétraitement

Le prétraitement, également appelé traitement d'image, est une étape essentielle dans le domaine du traitement d'images et de la vision par ordinateur. Il regroupe un ensemble de techniques visant à améliorer la qualité et la lisibilité des images afin de faciliter leur interprétation tant par les observateurs humains que par les systèmes informatiques.

Ce processus se déroule en plusieurs étapes successives. Tout d'abord, la restauration permet de corriger les éventuels défauts présents dans l'image, tels que le bruit ou les distorsions, afin de la ramener le plus fidèlement possible à son état d'origine. Ensuite, l'amélioration s'attache à adapter l'image aux capacités visuelles humaines en optimisant le contraste et en mettant en évidence les caractéristiques importantes. Différentes techniques, telles que la modification de l'histogramme ou l'utilisation de filtres linéaires et non linéaires, sont employées dans ce but [30].

III.5.3 Analyse d'une image

L'analyse d'une image est un processus composé d'opérations visant à extraire l'information essentielle qu'elle contient. Ce processus se déroule généralement en deux phases principales : la segmentation et la caractérisation.

Segmentation

La segmentation est une étape essentielle du traitement d'image qui vise à identifier et délimiter les objets présents dans une image en les divisant en zones homogènes. L'objectif est de séparer les différents composants visibles pour les identifier distinctement. Pour cela, deux approches complémentaires sont utilisées [31] :

Segmentation en régions La première approche consiste à regrouper les pixels partageant une caractéristique commune, généralement basée sur l'intensité des pixels, pour former des régions. Ces régions correspondent à des zones de l'image ayant des propriétés similaires.

Segmentation en contours La seconde approche consiste à définir les frontières entre les régions en plaçant ces frontières là où la variation d'un critère est localement maximale. Ces frontières, également appelées contours, marquent les limites entre les différentes régions identifiées.

Caractérisation

La caractérisation d'une image implique l'extraction d'attributs ou de caractéristiques spécifiques qui peuvent être utiles pour décrire et analyser cette image.

Voici quelques exemples d'attributs couramment utilisés pour chaque aspect :

- Caractéristiques de forme (élongation, périmètre, surface, capacité)
- Caractéristiques de couleur (l'analyse de l'histogramme, moyenne de couleur, teinte, saturation et luminosité)
- Caractéristiques de texture (écart type, moyenne, corrélation, entropie, contraste, énergie)

III.5.4 Apprentissage et classification

L'apprentissage et la classification d'images sont des domaines essentiels dans le traitement et l'analyse des données visuelles. Différentes approches sont utilisées pour réaliser la catégorisation ou la classification d'images. Voici quelques exemples d'approches couramment utilisées :

- Intelligence artificielle
- Séparateurs à vaste marge (SVM)
- Apprentissage non supervisé (clustering)

III.5.5 Interprétation

L'analyse d'images représente généralement la dernière étape avant de prendre une décision. Son rôle est d'attribuer une signification à l'information extraite de l'image, en la comparant, par exemple, avec une base de données préexistante. Ainsi, elle permet de déterminer des ensembles de pixels connectés dans l'image. Cette interprétation peut requérir des méthodes avancées telles que l'intelligence artificielle, les réseaux neuronaux et la logique floue pour être menée à bien [32].

III.6 Filtrage de l'image

Le filtrage d'image est une étape essentielle du traitement d'images qui vise à améliorer la qualité en appliquant des filtres spécifiques pour éliminer les artefacts indésirables tels que le bruit, les imperfections ou les dégradations. Cette opération est réalisée sur chaque pixel ou groupe de pixels, permettant ainsi d'obtenir une image plus claire et adaptée pour une meilleure interprétation par les humains et les systèmes informatiques. En facilitant les étapes d'analyse et de reconnaissance d'objets, le filtrage contribue à obtenir des résultats plus précis dans les applications de vision par ordinateur et de traitement d'images.

Parmi les processus altérant une image réelle lors de sa numérisation, le bruit électronique du capteur et la qualité de la numérisation jouent un rôle crucial. Pour atténuer les effets du bruit, des filtres passe-bas sont utilisés, prenant en compte le voisinage de chaque pixel. Les filtres passe-bas sont l'une des méthodes couramment utilisées pour réduire le bruit et améliorer la qualité de l'image.

Il existe différents types de filtrage d'image, chacun ayant des objectifs et des effets spécifiques. Parmi les filtres les plus couramment utilisés, on retrouve les filtres passe-haut, les filtres de flou, les filtres de détection de contours et les filtres de couleur. Chaque type de filtre peut être adapté en fonction des besoins spécifiques de l'application [33].

III.6.1 Filtrage linéaire

Le filtrage linéaire repose sur des opérations mathématiques linéaires appliquées à l'image en utilisant des masques de convolution représentés sous forme de matrices. Ces masques sont déplacés pixel par pixel sur l'image, et à chaque position, une opération de convolution est réalisée pour obtenir la valeur filtrée du pixel. Les filtres linéaires incluent les filtres passe-bas, passe-haut, de flou et de netteté, qui ont été mentionnés précédemment. L'une des principales caractéristiques des filtres linéaires est leur propriété de superposition, permettant d'appliquer plusieurs filtres successivement pour obtenir l'effet de filtrage souhaité. Cela offre une flexibilité supplémentaire pour adapter le filtrage aux besoins spécifiques de traitement d'image.

III.6.2 Filtrage non linéaire

Le filtrage non linéaire vise à altérer l'image en fonction de relations non linéaires entre les pixels. Contrairement aux filtres linéaires, ces filtres utilisent des opérations non linéaires telles que la valeur maximale, la valeur minimale ou la médiane des pixels dans un voisinage donné. Cette approche est plus adaptée aux cas où les relations entre les pixels sont complexes et ne peuvent pas être bien modélisées par des filtres linéaires. Les filtres non linéaires offrent une solution efficace pour des tâches spécifiques, comme la réduction du bruit impulsif avec le filtre médian, la restauration d'images dégradées ou la segmentation d'images. Grâce à leur flexibilité, ces filtres peuvent mieux s'adapter aux propriétés locales de l'image et produire des résultats de traitement plus adaptés à des situations variées.

III.7 Calibrage de caméra

Le calibrage géométrique d'une caméra est un processus essentiel visant à établir une relation mathématique entre les coordonnées des points 3D dans la scène observée et leurs projections en coordonnées 2D dans l'image capturée.

III.7.1 Principe général du calibrage de caméra

Lorsque l'on souhaite obtenir des informations métriques à partir des images acquises, notamment pour des applications de mesure dimensionnelle, le calibrage de la caméra devient particulièrement crucial. En effet, il permet de prendre en compte les distorsions géométriques induites par le système optique de la caméra, garantissant ainsi des mesures dimensionnelles précises.

Les différentes étapes du processus de calibrage de caméra offrent une vue d'ensemble de cette procédure. Plusieurs bibliothèques et outils logiciels, tels qu'OpenCV ou MATLAB, proposent des fonctionnalités de calibrage de caméra prêtes à l'emploi, facilitant ainsi la réalisation de cette tâche.

Le calibrage de caméra constitue le point de départ pour de nombreuses applications de vision artificielle, telles que la reconnaissance et la localisation d'objets, le contrôle dimensionnel de pièces, ou encore la reconstruction d'environnements pour la navigation des robots mobiles. En fournissant des informations précises sur la géométrie de la caméra, cette étape est fondamentale pour assurer la fiabilité et l'efficacité de ces diverses applications [34].

III.7.2 OpenCV

(Vision par ordinateur open-source) représente une bibliothèque de fonctions de programmation conçue principalement pour la vision en temps réel. Elle regroupe plus de 2500 algorithmes optimisés, y compris des techniques de pointe en matière de traitement d'images et de vision par ordinateur. Cette ressource polyvalente permet d'effectuer diverses tâches telles que le traitement d'images, la détection, la reconnaissance faciale, l'identification d'objets, la classification d'actions, le suivi, ainsi que d'autres fonctions.

En utilisant cette bibliothèque, il est possible de mettre en œuvre facilement ces fonctionnalités sur un ordinateur, offrant ainsi une vision par ordinateur simple. Bien que basée sur C++, elle propose également des

interfaces Python pour faciliter son utilisation. Dans notre projet spécifique, nous exploitons pour localiser le robot dans son environnement et de le guider en toute sécurité vers sa destination [35].

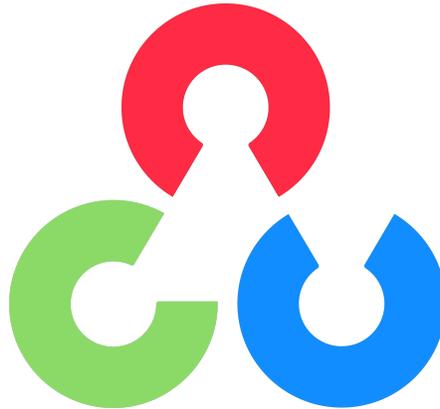


FIGURE III.8 – Logo d'opencv

III.7.3 Calibrage avec OpenCV

Le calibrage d'une caméra est un processus crucial réalisé en utilisant un objet bien connu, généralement une mire de calibrage. Selon la méthode de calibrage choisie, les types de mires peuvent varier, allant des objets plans aux cubes, par exemple. Dans le cas spécifique d'OpenCV, il est recommandé d'utiliser un quadrillage non symétrique de carreaux noirs et blancs dont les paramètres, tels que le nombre de carreaux et leurs dimensions, sont préalablement connus.

OpenCV propose une approche performante pour déterminer les paramètres intrinsèques et extrinsèques d'une caméra en utilisant plusieurs prises de vue d'un checkerboard créé à partir de la caméra à calibrer.

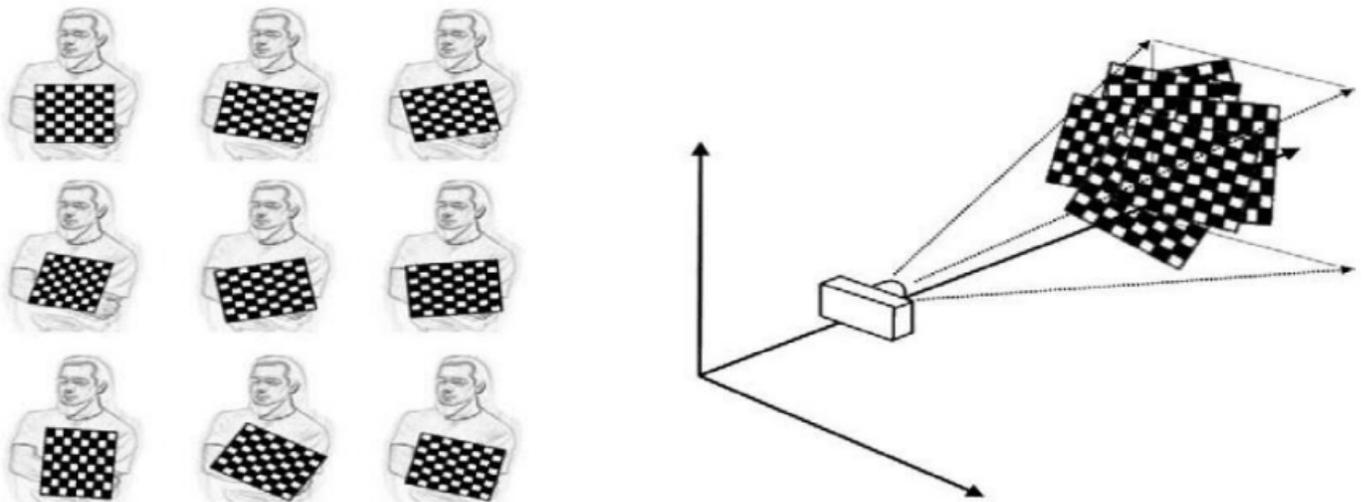


FIGURE III.9 – Prise de vue différentes d'un checkerboard

Pourquoi choisir un checkerboard? Cette sélection repose sur la facilité de détection de ses coins grâce aux algorithmes de vision par ordinateur, ainsi que sur la simplicité de sa forme géométrique. Pour déterminer la position de n'importe quel coin du checkerboard, il suffit de connaître le nombre de carrés horizontaux et verticaux présents dans la mire, ainsi que la taille d'un carré. Cette simplicité facilite grandement le processus de calibrage tout en assurant des résultats de haute précision.

III.7.4 Paramètres de camera

Les paramètres intrinsèques sont internes à l'appareil photo et restent constants pour un appareil donné. Ils comprennent la distance focale, ainsi que le centre de l'image sur l'axe x et y .

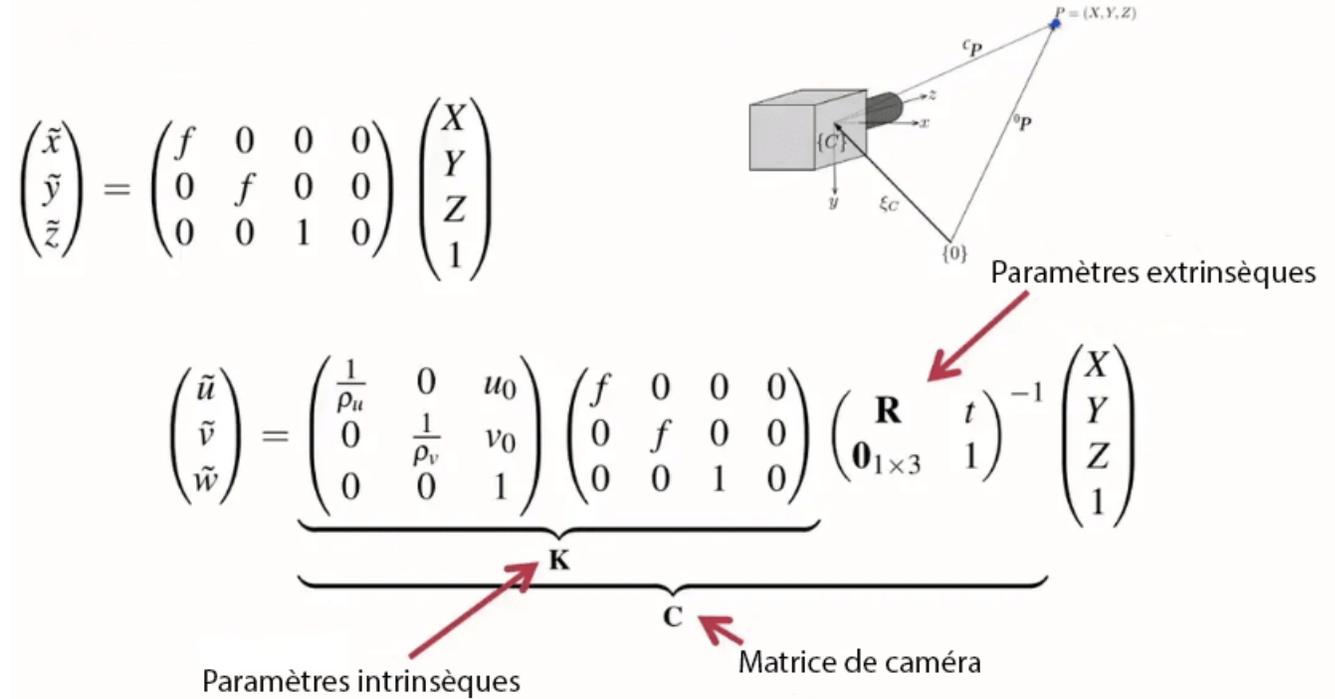


FIGURE III.10 – Modèle de caméra complet

Le modèle d'appareil photo mentionné ci-dessus est basé sur un modèle d'appareil photo à trou d'épingle (pinhole). Cependant, dans la réalité, nous avons une caractéristique importante d'un appareil photo qui est absente dans ce modèle : l'objectif (ou la lentille). La lumière a tendance à se courber légèrement trop aux bords d'une lentille courbée d'un appareil photo, ce qui crée un effet de distorsion sur les bords des images. Par conséquent, cette distorsion doit être corrigée. C'est pourquoi nous calculons ces paramètres.

La distorsion peut être exprimée soit comme une distorsion tangentielle, soit comme une distorsion radiale. Ces deux types de distorsion peuvent être exprimés par cinq paramètres [36].

$$\text{Distortion coefficients} = (k_1 k_2 p_1 p_2 k_3) \quad (\text{III.1})$$

Où :

- k_1, k_2, k_3 sont des coefficients de distorsion radiale. Ces paramètres contrôlent le niveau de distorsion radiale. k_1 corrige la distorsion radiale principale, k_2 corrige la distorsion radiale secondaire et k_3 corrige les distorsions d'ordre supérieur.
- p_1, p_2 sont des coefficients de distorsion tangentielle. Ces paramètres corrigent la distorsion tangentielle causée par le désalignement de l'objectif. Ils prennent en compte l'étirement ou l'inclinaison de l'image.

III.7.5 Protocole de calibration

Le processus de calibration de la caméra se déroule en quatre étapes, utilisant les fonctions fournies par la bibliothèque OpenCV :

Extraction des points du checkerboard Dans cette première étape, nous extrayons et suivons les points situés aux intersections des cases blanches et noires du checkerboard. Ces points serviront de repères pour la calibration.

Réorganisation des points détectés Une fois les points détectés, il est nécessaire de les réorganiser dans un ordre spécifique pour faciliter le processus de calibration.

Détermination des paramètres internes de la caméra Avec les coordonnées des points du checkerboard sur l'image obtenues, nous pouvons déterminer les paramètres intrinsèques de la caméra, tels que la matrice de caméra (camera matrix) qui contient les paramètres focaux et les coordonnées du centre de l'image.

Détermination des paramètres externes de la caméra Enfin, nous pouvons déterminer les paramètres extrinsèques de la caméra, qui décrivent la position et l'orientation de la caméra par rapport au monde réel.

Une fois le processus de calibration terminé, nous obtenons les résultats suivants :

Matrice de caméra (camera matrix) :

$$\begin{bmatrix} 565.20028664 & 0 & 320.4840189 \\ 0 & 585.31174699 & 251.86771572 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2})$$

Coefficients de distorsion (distortion coefficients) :

$$\left[0.139790776 \quad -1.00169522 \quad 0.00100196342 \quad 0.00398510255 \quad 1.61675094 \right] \quad (\text{III.3})$$

Ces résultats de calibration permettent d'établir une correspondance précise entre le monde réel et l'image capturée par la caméra, ce qui facilite des tâches telles que la mesure de distances, la reconstruction 3D et la localisation précise des objets dans l'environnement.

III.8 Flux optique

Le flux optique est une technique couramment utilisée en vision par ordinateur et en robotique pour estimer le mouvement des objets dans une séquence d'images. Il repose sur le principe que les pixels d'une image se déplacent dans le temps lorsque l'objet est en mouvement. En exploitant ces déplacements, il devient possible de calculer la vitesse et la direction du mouvement de chaque pixel [37].

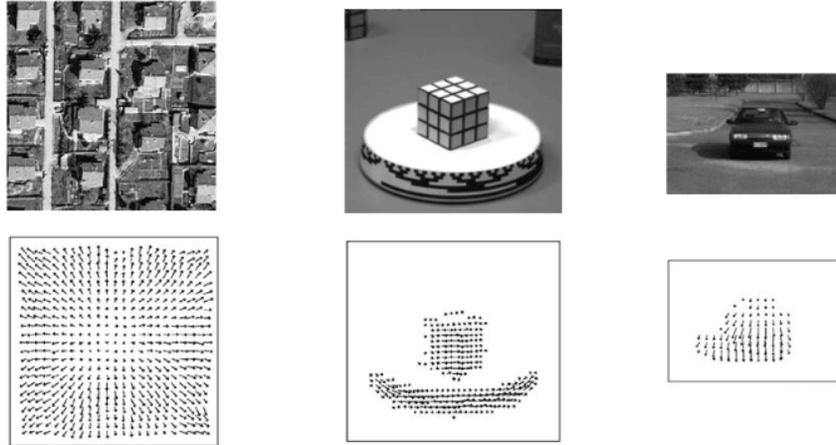


FIGURE III.11 – Exemples de flux optique

III.8.1 Calcul de flux optique

On considère une fonction d'image $I(u, v, t)$ qui représente la position du pixel (u, v) à l'instant t . Le flux optique entre deux images capturées aux instants t et $t + \Delta t$ peut alors être représenté par le déplacement $(\Delta u, \Delta v)$ et la différence de temps Δt .

La plupart des approches d'estimation du flux optique sont basées sur une contrainte de constance de luminosité (équation (III.4)). Cette contrainte suppose que les pixels en mouvement conservent la même luminosité entre les images consécutives :

$$I(u, v, t) = I(u + \Delta u, v + \Delta v, t + \Delta t) \quad (\text{III.4})$$

La constance de luminosité peut être linéarisée en utilisant l'approximation de Taylor, ce qui donne :

$$I_u V_u + I_v V_v + I_t = 0 \quad (\text{III.5})$$

où I_u , I_v et I_t sont les dérivées partielles de la fonction de l'image par rapport à u , v et t respectivement, et V_u et V_v sont les vitesses du flux optique.

Comme l'équation III.5 comporte deux variables, elle a un nombre infini de solutions. Cette ambiguïté signifie que d'autres contraintes doivent être appliquées, telles que la contrainte de régularité spatiale.

La contrainte de régularité spatiale suppose que les pixels voisins appartiennent aux mêmes objets et représentent donc le même mouvement.

III.8.2 Méthodes de flux optique

Il existe plusieurs méthodes pour calculer le flux optique, chacune avec ses propres avantages et limitations. Les méthodes les plus couramment utilisées sont les méthodes basées sur le suivi de points caractéristiques (sparse optical flow) et les méthodes basées sur l'estimation dense du flux optique (dense optical flow).

Les méthodes de suivi de points caractéristiques, telles que la méthode de Lucas-Kanade, sélectionnent un ensemble de points dans l'image et suivent leur déplacement entre les images consécutives. Ces méthodes sont efficaces pour estimer le flux optique dans des régions avec des variations de luminosité faibles à modérées, mais elles peuvent échouer dans des régions avec des variations de luminosité importantes ou en présence d'occlusions.

Malgré leurs différences, les méthodes de suivi de points caractéristiques peuvent être vues conceptuellement en termes de trois étapes de traitement :

- Préfiltrage ou lissage avec des filtres passe-bas et passe-bande pour avoir la structure du signal d'intérêt et améliorer le rapport signal/bruit.
- L'extraction des mesures de base, telles que les dérivées spatio-temporelles, pour mesurer les composantes normales de vitesse.
- L'intégration de ces mesures pour produire un flux optique 2D, ce qui implique souvent des hypothèses sur la régularité.

Les méthodes d'estimation dense du flux optique, telles que la méthode de Farneback, calculent le flux optique pour chaque pixel de l'image. Ces méthodes sont plus précises car elles prennent en compte l'ensemble de l'image, mais elles sont également plus coûteuses en termes de calcul.

Dans notre travail, nous nous intéressons particulièrement à la méthode de suivi de points caractéristiques (Lucas-Kanade), car elle offre un bon équilibre entre précision et efficacité pour notre application spécifique. Cette approche est reconnue pour sa capacité à estimer le flux optique avec une grande précision tout en optimisant les temps de calcul.

III.8.3 Méthode de Lucas–Kanade

La méthode de Lucas-Kanade est une technique largement utilisée en vision par ordinateur pour estimer le flux optique, c'est-à-dire le mouvement des objets dans une séquence d'images. Elle se base sur l'idée principale de trouver une approximation du flux optique en utilisant les gradients de l'image. Pour ce faire, la méthode utilise une fenêtre locale autour de chaque pixel pour calculer les gradients dans les directions horizontale (x) et verticale (y).

Cette approche repose sur l'hypothèse que le flux optique est relativement constant dans le voisinage local du pixel considéré, ce qui signifie que les objets ne se déplacent pas trop rapidement d'une image à l'autre dans cette région. En utilisant cette hypothèse, la méthode résout un système d'équations linéaires pour estimer le déplacement du pixel en minimisant la somme des différences entre les valeurs des pixels dans les deux images.

Développée par Bruce D. Lucas et Takeo Kanade, la méthode de Lucas-Kanade est un outil puissant pour le suivi de points caractéristiques dans les séquences d'images, car elle permet d'estimer avec précision le mouvement local des objets tout en étant efficace en termes de temps de calcul. Elle offre ainsi un bon équilibre entre précision et efficacité, ce qui en fait une approche populaire dans de nombreuses applications de vision par ordinateur et de robotique.

III.8.4 Calcul de Lucas Kanade

La méthode de Lucas-Kanade repose sur l'hypothèse selon laquelle le déplacement d'un point de l'image entre deux moments consécutifs est de petite amplitude et relativement constant dans un environnement proche du point p . On considère que l'équation du flot optique est applicable à l'ensemble des pixels situés dans une fenêtre centrée autour du point p . Le vecteur de vitesse local (V_x, V_y) doit être en accord avec cette hypothèse.

$$I_x(q_1) V_x + I_y(q_1) V_y = -I_t(q_1) \tag{III.6}$$

$$I_x(q_2) V_x + I_y(q_2) V_y = -I_t(q_2) \tag{III.7}$$

⋮

$$I_x(q_n) V_x + I_y(q_n) V_y = -I_t(q_n) \tag{III.8}$$

Où :

- q_1, q_2, \dots, q_n sont les pixels à l'intérieur de la fenêtre
- $I_x(q_i), I_y(q_i), I_z(q_i)$ sont les dérivées partielles de l'image I selon les variables d'espace x, y et de temps t , évaluée au point q_i et au temps courant.

Ces équations peuvent être écrites sous la forme matricielle suivante :

$$A_v = b \quad (\text{III.9})$$

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad \text{et} \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \quad (\text{III.10})$$

Le système contient un nombre d'équations supérieur à celui des inconnues, ce qui le classe comme sur-déterminé. La méthode de Lucas-Kanade propose une solution en utilisant l'approche des moindres carrés, résolvant ainsi un ensemble d'équations normales :

$$A^T A_v = A^T b \quad (\text{III.11})$$

$$v = (A^T A)^{-1} A^T b \quad (\text{III.12})$$

Où A^T est la matrice transposée de la matrice A . Alors, on trouve :

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i) I_y(q_i) \\ \sum_i I_x(q_i) I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i) I_t(q_i) \\ -\sum_i I_y(q_i) I_t(q_i) \end{bmatrix} \quad (\text{III.13})$$

Avec les sommes allant de $i = 1$ à n .

La matrice $A^T A$ est appelée le tenseur de structure de l'image au point p .

La solution précédemment exposée accorde une importance égale à tous les n pixels q_i de la fenêtre. En pratique, il est souvent souhaitable de privilégier les pixels situés à proximité du pixel p en leur attribuant un poids plus significatif. Pour ce faire, on recourt à la variante pondérée de l'équation des moindres carrés :

$$A^T W A_v = A^T W b \quad (\text{III.14})$$

$$v = (A^T W A)^{-1} A^T W b \quad (\text{III.15})$$

Où W est matrice diagonale nm contenant les poids $W_{ii} = w_i$ associé à l'équation du pixel q_i . Alors, le calcul est le suivant :

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i) I_y(q_i) \\ \sum_i w_i I_x(q_i) I_y(q_i) & \sum_i w_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i w_i I_x(q_i) I_t(q_i) \\ -\sum_i w_i I_y(q_i) I_t(q_i) \end{bmatrix} \quad (\text{III.16})$$

Le poids w_i est habituellement un ensemble de Gaussienne de la distance entre q_i et p [38].

III.8.5 Estimation de position 3D à partir d'images 2D

Un point dans l'espace $P = [X, Y, Z]^T$ est projeté par une caméra (figure III.12). Le point correspondant sur le plan de l'image $p = [x, y, f]^T$ peut être calculé comme suit :

$$p = -\frac{f}{Z} P \quad (\text{III.17})$$

où f , la distance entre le plan de l'image et l'origine de projection O , est la distance focale de la caméra. Étant donné que la caméra est montée perpendiculairement à la carrosserie d'un véhicule, la coordonnée Z est égale à la distance entre le point d'intérêt et l'origine de projection de la caméra.

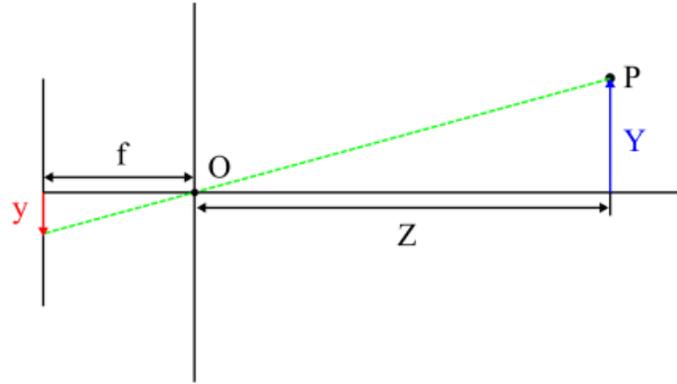


FIGURE III.12 – Le modèle de projection de caméra

La distance Z doit être obtenue à partir d'un capteur externe, tel qu'un capteur de distance ultrason. Étant donné que Z est approximativement constante entre deux images consécutives, un déplacement dans le plan de l'image $(\Delta x, \Delta y)$ peut être converti en un déplacement dans le monde réel $(\Delta X, \Delta Y)$:

$$\Delta X = -\frac{1}{f}\Delta x.Z \quad (\text{III.18})$$

$$\Delta Y = -\frac{1}{f}\Delta y.Z \quad (\text{III.19})$$

Le déplacement dans le plan de l'image peut être obtenu en utilisant l'algorithme de flux optique (Lucas-Kanade). Comme le déplacement calculé $(\Delta u, \Delta v)$ est généralement en pixels, il est nécessaire de le convertir en unités du monde réel (par exemple, en mètres).

$$\Delta X = -\frac{s}{f}\Delta u.Z \quad (\text{III.20})$$

$$\Delta Y = -\frac{s}{f}\Delta v.Z \quad (\text{III.21})$$

où s est la taille du pixel [39].

III.8.6 Amélioration de l'estimation

L'amélioration du calcul du flux optique peut être réalisée en appliquant le principe de l'implémentation pyramidale. Cette approche repose sur la création d'une pyramide d'images, où chaque niveau de la pyramide représente une version sous-échantillonnée de l'image originale. Plus précisément, on définit une hauteur L_m pour cette pyramide, et au niveau le plus bas (L_m), l'image est la plus grossière.

À chaque niveau de la pyramide, les deux images successives sont sous-échantillonnées d'un facteur 2. Le niveau zéro correspond à l'image initiale non modifiée, tandis que le niveau L_m représente l'image la plus réduite. Au niveau L_m , on procède au calcul du flux optique à l'aide de la méthode choisie.

Ensuite, le flux optique obtenu au niveau L_m est propagé vers le niveau inférieur en translatant l'image avec les informations calculées au niveau supérieur. Cette procédure est répétée de manière récursive jusqu'à atteindre le niveau 0, qui correspond à l'image initiale d'origine. Ainsi, à partir de ces calculs successifs, on obtient finalement le flux optique final [40].

Grâce à cette approche pyramidale, la méthode de Lucas-Kanade, par exemple, est capable de résoudre de manière plus robuste certaines incertitudes liées à l'estimation du flux optique, notamment le problème de l'ouverture. Cependant, il est important de souligner que cette méthode reste limitée aux informations locales et ne peut pas fournir d'informations détaillées sur le flux optique à l'intérieur de régions uniformes de l'image.

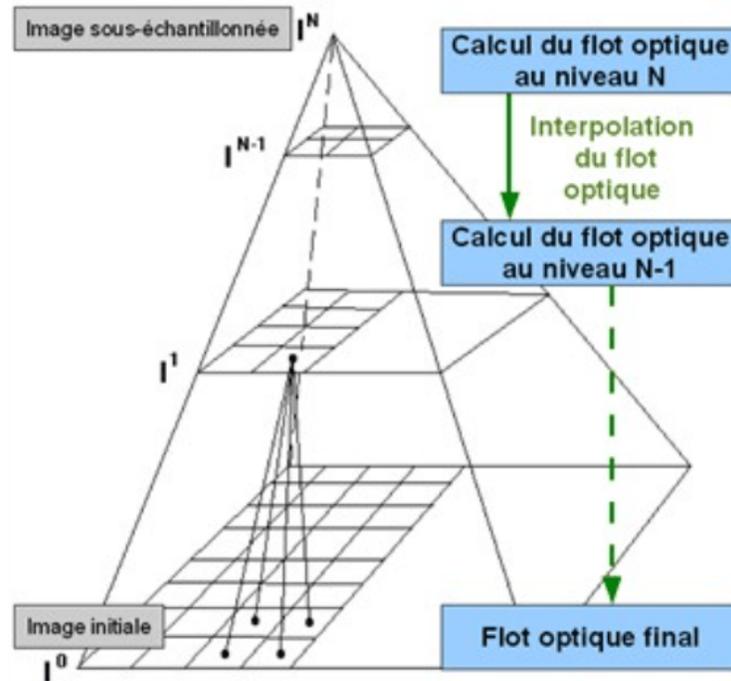


FIGURE III.13 – Implémentation pyramidale d'une méthode de calcul du flux optique

III.9 Théorie du filtre de Kalman

Le Filtre de Kalman est un algorithme de filtrage récursif largement utilisé dans le domaine de la robotique, du suivi d'objets et de l'estimation de l'état. Il a été développé par Rudolf E. Kalman qui est particulièrement efficace pour estimer l'état d'un système dynamique en présence de bruit dans les mesures.

Son utilisation dans cette application de suivi de position à partir du flux optique est justifiée par sa capacité à gérer de manière élégante les incertitudes et les fluctuations inhérentes aux capteurs optiques. Le Filtre de Kalman nous permet de suivre et de lisser les données du flux optique, offrant ainsi une estimation de position précise et robuste même en présence de perturbations et de bruit, ce qui en fait un outil indispensable pour de nombreuses applications de navigation et de suivi dans le domaine de la robotique et de la vision par ordinateur.

Ce filtre repose sur deux étapes principales : la prédiction de l'état futur du système et la mise à jour de cette prédiction en fonction des mesures réelles. Comprendre les éléments clés de ces étapes est essentiel pour saisir le fonctionnement du filtre de Kalman.

III.9.1 Initialisation de l'état et de la covariance de l'erreur

Le processus débute par l'initialisation de l'état initial du système, noté x_0 , qui représente la meilleure estimation de l'état du système à l'instant initial. De plus, la matrice de covariance de l'état initial, notée P_0 , est définie pour quantifier l'incertitude associée à cette estimation. Ces paramètres sont essentiels pour démarrer le filtre de Kalman.

$$x_0 = \begin{bmatrix} \text{position initiale} \\ \text{vitesse initiale} \end{bmatrix} \quad (\text{III.22})$$

$$P_0 = \begin{bmatrix} \text{covariance de la position initiale} & 0 \\ 0 & \text{covariance de la vitesse initiale} \end{bmatrix} \quad (\text{III.23})$$

III.9.2 Prédiction de l'état

Dans cette phase, le filtre de Kalman utilise le modèle dynamique du système pour anticiper l'état futur du système en se basant sur l'état actuel et, le cas échéant, sur les commandes appliquées. Cette prédiction est accompagnée d'une estimation de la covariance de l'erreur associée à cette prédiction, ce qui permet de quantifier l'incertitude de la prédiction.

L'équation de prédiction de l'état est formulée comme suit :

$$x_{k+1} = F \cdot x_k + B \cdot u_k \quad (\text{III.24})$$

Où :

- x_{k+1} : est le vecteur d'état prédit à l'instant $k + 1$.
- x_k : est le vecteur d'état actuel à l'instant k .
- F : est la matrice de transition du système.
- B : est la matrice de commandes (si applicable).
- u_k : est le vecteur de commandes à l'instant k .

III.9.3 Mise à jour de la Covariance de l'État

Dans cette étape, le filtre de Kalman utilise les mesures réelles du système pour corriger la prédiction précédente. Il compare la prédiction avec les mesures pour estimer l'erreur entre les deux. Ensuite, il calcule la mise à jour de l'état estimé en utilisant cette erreur et met à jour la covariance de l'erreur associée

L'équation de mise à jour de la covariance de l'état est formulée de la manière suivante :

$$P_{k+1} = F \cdot P_k \cdot F^T + Q \quad (\text{III.25})$$

Où :

- P_{k+1} : est la matrice de covariance de l'état prédite à l'instant $k + 1$.
- P_k : est la matrice de covariance de l'état à l'instant k .
- F : est la matrice de transition du système.
- Q : est la matrice de covariance du bruit du processus [41].

III.10 Conclusion

Ce chapitre traite de la localisation des robots mobiles, abordant des sujets tels que la vision par ordinateur, la calibration de caméras et l'estimation de position 3D. Les techniques abordées sont essentielles pour permettre aux robots de naviguer et d'interagir avec leur environnement de manière précise et intelligente, en utilisant notamment le filtre Kalman pour estimer les mouvements de bruit dans un système.

Nous allons maintenant passer à la phase pratique, où nous concrétiserons ces concepts en construisant notre robot de localisation et en mettant en œuvre ces techniques pour atteindre la localisation souhaitée. Tout cela sera expliqué dans le quatrième chapitre.

Chapitre IV

Partie expérimental

IV.1 Introduction

Ce chapitre présente les composants essentiels de notre robot mobile, avec une décision de construire un robot piloté par Raspberry Pi. Nous avons intégré plusieurs éléments clés, tels que les capteurs de distance (ultrasons HC-SR04), la caméra Pi et le capteur inertiel (MPU6050), pour lui permettre de déterminer sa position en temps réel.

IV.2 Matériel utilisé

IV.2.1 Châssis de robot mobile 6WD

Le châssis, construit en métal, assure une grande robustesse et résistance. Il est doté de six roues motrices avec une puissance de 350 mA * 6 et un moteur de vitesse nominale 17000 tr/min et vitesse de sortie 500 tr/min, offrant une excellente traction dans des environnements variés. Ses dimensions de 280 x 200 x 105 mm permettent une installation pratique des composants électroniques, avec un poids d'environ 1,7 kg et une capacité de charge maximale de 5 kg. La tension de fonctionnement nominale est de 12V DC, avec une plage recommandée de 12V à 24V DC.

La suspension indépendante améliore la capacité de franchissement, tandis que la facilité de personnalisation et l'assemblage simplifié rendent ce châssis adapté aux besoins individuels. Les trous pré-perçés pour les composants électroniques, tels que les moteurs, les capteurs et les circuits imprimés, simplifient l'installation. Cette polyvalence permet d'utiliser la plate-forme pour diverses applications robotiques, notamment la cartographie, la surveillance et l'exploration, en particulier dans des missions de recherche et de sauvetage en environnements difficiles, pour accéder à des zones dangereuses ou inaccessibles



FIGURE IV.1 – Robot mobile 6WD

IV.2.2 Raspberry Pi 3 Model B

Le choix d'utiliser un Raspberry Pi spécifiquement conçu pour fonctionner avec la caméra Pi, par opposition à l'Arduino, présente plusieurs avantages. Le Raspberry Pi offre une connectivité étendue grâce à ses ports USB et au Wi-Fi, simplifiant ainsi l'intégration de la caméra Pi avec d'autres périphériques et permettant l'envoi de données sur le réseau. Les caractéristiques du Raspberry Pi 3 Model B incluent :

- Démarrage plus rapide.
- Une meilleure compatibilité avec les derniers programmes et logiciels éducatifs.
- Un processeur intégré Quad-core ARM Cortex-A53 cadencé à 1.2 GHz (Broadcom BCM2837).
- 1024 Mo de RAM et un lecteur de cartes Micro SD, avec Wi-Fi b/g/n et Bluetooth 4.1.
- prise en charge de distributions GNU/LINUX ARM telles que Snappy Ubuntu, ainsi que de Microsoft Windows 10 .



FIGURE IV.2 – Raspberry Pi 3 Model B

Réalisé par la Raspberry Pi Foundation en 2011 en Angleterre, le Raspberry Pi est devenu l'un des ordinateurs à carte unique les plus populaires et les plus largement utilisés au monde. Cette plate-forme polyvalente est largement employée dans l'éducation, les projets de développement logiciel, le divertissement, le travail, l'expérimentation, la visualisation de vidéos, la navigation Web et la création de montages électroniques.

Le Raspberry Pi trouve une variété d'applications, notamment en tant que véritable ordinateur, support pour l'apprentissage de la programmation, plate-forme pour interagir avec le monde physique, lecteur multimédia et outil de programmation bas niveau. Cette combinaison de fonctionnalités en fait un outil puissant pour de multiples usages.

IV.2.3 Caméra Module Rev 1.3

Le Caméra Module Rev 1.3 est un composant essentiel de notre robot de localisation. Cette petite carte caméra se connecte de manière transparente à notre ordinateur embarqué Raspberry Pi via le port CSI (Camera Serial Interface). Grâce à son capteur de 5 mégapixels et à la technologie OmniBSI (rétro-éclairage), il est capable de capturer des images haute résolution d'une qualité exceptionnelle, avec une résolution allant jusqu'à 2592 x 1944 pixels. De plus, il offre des fonctionnalités vidéo étendues, avec des résolutions de 1080p30, 720p60 et 640x480p60/90.

Ce module est équipé d'une lentille de distance focale de 3,04 mm, ce qui lui permet de capturer des images nettes et détaillées. Chaque pixel de cette caméra a une taille de 1,12 μm x 1,12 μm , assurant une résolution fine et précise. Cette combinaison de caractéristiques optiques et de taille de pixel garantit des images de haute qualité, ce qui est crucial pour la précision de notre système de localisation et pour l'ensemble de nos applications liées à la vision.



FIGURE IV.3 – 5mp raspberry pi camera module rev 1.3

IV.2.4 Support caméra

Bien conscient des défis posés par les perturbations de vibrations sur la qualité de la lecture des caméras utilisant le flux optique, on a fait une approche pour résoudre ce problème. On a réussi à concevoir un support de caméra sur mesure par une imprimante 3D. Ce support a été spécialement conçu pour minimiser les effets néfastes des vibrations indésirables, on a pu réduire considérablement les vibrations nuisibles qui pouvaient altérer la qualité du flux optique capturé par la caméra.



FIGURE IV.4 – Support de caméra

IV.2.5 Batterie

Les batteries LiPo (Lithium-ion Polymère) sont grandement appréciées dans diverses applications électroniques grâce à leur rapport poids/puissance exceptionnel. Cette technologie offre des performances optimales. Un élément LiPo (1S) fournit une tension de 3,7V, on utilise en général des batteries à 3 ou 4 éléments (3S ou 4S).

Pour notre cas particulier, la batterie LiPo Turnigy 3S ($3 \times 3.7V = 11.2V$) possède une capacité de 2500 mAh, ce qui signifie qu'elle peut fournir un courant de 2500 mA pendant une heure. et elle a un taux de décharge de 20C.



FIGURE IV.5 – Lipo batterie 2500 mAh

IV.2.6 Régulateur de tension

Le régulateur de tension XL6009E1 est un composant électronique largement adopté pour assurer la stabilité de la tension d'alimentation. Il fonctionne en tant que convertisseur de tension à découpage (buck-boost), permettant de modifier la tension d'entrée pour obtenir une sortie régulière et stable.

Sa popularité réside dans sa grande polyvalence et ses performances supérieures. Avec une plage de tensions d'entrée étendue, généralement de 3V à 32V, et une capacité de courant pouvant atteindre 4 A, ce régulateur peut être utilisé avec diverses sources d'alimentation, comme les batteries ou les panneaux solaires.

Nous l'avons choisi pour alimenter le Raspberry Pi, intégré dans notre robot, et ainsi en faire un système embarqué. Cela permet de garantir une alimentation fiable et constante, essentielle pour le bon fonctionnement du robot. Il assure une tension de sortie de 5,2 V à partir de la tension de batterie de 11,2 V.



FIGURE IV.6 – Régulateur de tension XL6009E1

IV.2.7 Conducteur de moteur L298N

Le L298N est un circuit intégré, souvent utilisé comme un double pont en H, qui permet de contrôler la vitesse et la direction des moteurs électriques à courant continu. Ce dispositif offre une capacité de courant de 2A par canal et peut contrôler simultanément jusqu'à six moteurs à courant continu. C'est pourquoi nous avons opté pour l'utilisation du L298N au lieu du L293D, car les moteurs du robot ont un courant nominal de 350mA par moteur, et en en ayant six, cela nécessite une capacité de courant supérieure.

Le tableau ci-dessous présente une comparaison des caractéristiques entre les composants L293D et L298N :

	L298N	L293D
Tension logique	4.5V-7V	4.5V-5.5V
Tension de commande	5V-46V	4.5V-36V
Courant logique	0-36mA	1.2A
Courant de sortie continu maximal	2A	0.6A
Puissance maximale	25W	0.6W
Efficacité	30-70 %	50-90 %

TABLE IV.1 – Tableau de Comparaison entre les Composants L293D et L298N

Il peut être contrôlé à l'aide d'un microcontrôleur, tel qu'un Raspberry Pi, pour fournir un contrôle de vitesse et de direction aux moteurs. Le circuit intégré est également équipé de diodes de roue libre intégrées pour protéger les circuits contre les surtensions induites par le moteur lorsqu'il est en arrêt. Le L298N est souvent utilisé dans des projets de robotique, de domotique et de contrôle de véhicules.

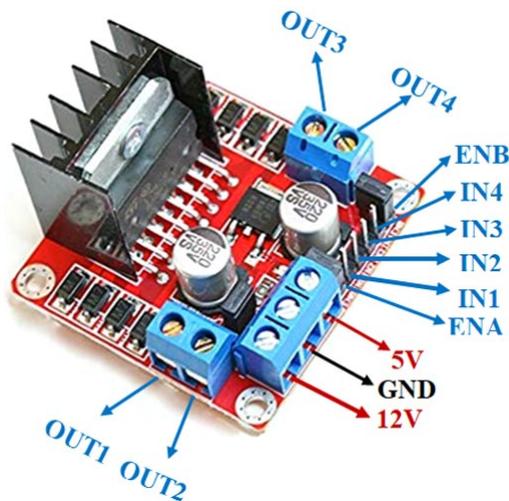


FIGURE IV.7 – Double pont en H (L298N)

IV.2.8 Capteur Ultrason HC-SR04

Le capteur ultrasonique HC-SR04 utilise des ondes sonores à haute fréquence pour mesurer la distance entre le capteur lui-même et un objet cible. Il émet des impulsions sonores ultrasoniques qui se propagent jusqu'à l'objet et sont ensuite renvoyées au capteur. En mesurant le temps écoulé entre l'émission du signal et la réception de l'écho, le capteur peut calculer la distance avec une précision raisonnable (une distance entre 3 cm et 3 m).

Dans notre projet, ce composant est essentiel pour la détermination précise de la position en 3D du robot à partir d'informations en 2D. Comme expliqué précédemment, la valeur de la coordonnée Z, représentant la distance entre le point d'intérêt et l'origine de projection de la caméra, est obtenue à l'aide de ce capteur.



FIGURE IV.8 – Capteur ultrason HC-SR04

IV.2.9 MPU6050

Le MPU6050 est un module de mesure inertielle qui possède 6 degrés de liberté pour déterminer sa position dans l'espace. Il intègre à la fois un gyroscope à 3 axes et un accéléromètre à 3 axes dans un seul boîtier. Cette configuration lui permet de mesurer les accélérations linéaires et angulaires. Il communique avec un microcontrôleur via l'interface I2C, fournissant des données brutes des capteurs. Ces données peuvent ensuite être utilisées pour calculer l'orientation et le mouvement. En plus de cela, il embarque un processeur de mouvement numérique qui peut effectuer des calculs avancés, facilitant la récupération de données.

Le MPU6050 est particulièrement utile pour des applications de navigation, de goniométrie, de stabilisation, ainsi que pour le contrôle gestuel, permettant de déterminer la position et les mouvements d'un objet dans l'espace.

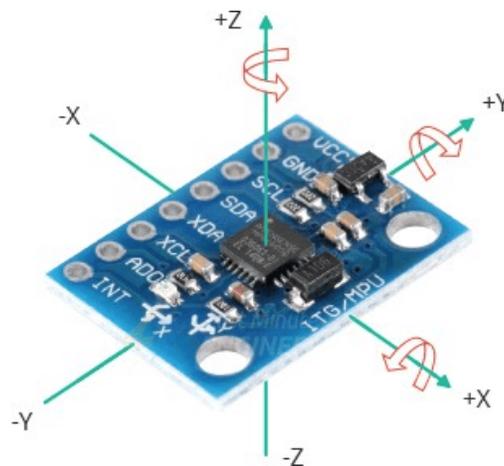


FIGURE IV.9 – Capteur MPU6050

Accéléromètre

L'accéléromètre est un capteur qui mesure l'accélération, c'est-à-dire le changement de vitesse par unité de temps ($a = dv / dt$). Cet appareil se base sur la deuxième loi de Newton, qui utilise les paramètres de force et de masse de l'objet ($a = F / m$), pour son fonctionnement. Pour mettre en œuvre cette mesure dans le domaine de l'électronique, des techniques MEMS (Micro Electro Mechanical Systems) sont utilisées.

Ces techniques diffèrent des méthodes de fabrication des puces électroniques traditionnelles, car elles permettent la création de composants mécaniques à l'intérieur d'un MEMS.

Gyroscope

Le gyroscope est un dispositif qui mesure la vitesse angulaire d'un objet, c'est-à-dire le changement d'orientation par unité de temps ou la vitesse de rotation d'un corps autour de son axe. Dans ce cas, des techniques MEMS sont également employées pour mesurer cette vitesse en utilisant un phénomène appelé effet Coriolis.

Pour notre projet, nous avons intégré l'IMU de modèle MPU6050. Les données fournies par le MPU6050 nous permettent de déterminer l'orientation du robot le long de l'axe Z, Étant donné la configuration de robot plane, ainsi que d'extraire des informations cruciales telles que l'accélération et la vitesse angulaire.

IV.3 Montage électrique

La figure ci-dessous illustre la configuration électrique réelle de tous les composants.

Pour garantir une connexion correcte des composants du robot, nous avons réalisé les connexions suivantes :

Branchement du MPU 6050 à la Raspberry Pi

Le capteur MPU6050 est équipé de 8 broches, mais nous utilisons seulement 4 :

- Vcc/GND : une source d'alimentation 3.3V et à la masse (GND) de la Raspberry Pi.
- SCL : Cette broche est utilisée pour les impulsions d'horloge pour le fonctionnement en I2C.
- SDA : Cette broche est utilisée pour transférer des données via une communication I2C.

Nous avons fixé le MPU de manière à ce que ses axes coïncident avec notre proposition : l'axe Z est perpendiculaire à la surface terrestre, l'axe X pointe vers la droite, et l'axe Y est dirigé vers l'avant.

Branchement de l'ultrason HC-SR04 à la Raspberry Pi

Alimentation : les broches VCC (alimentation) et GND (mise à la masse) de l'HC-SR04 ont été reliées respectivement à une source d'alimentation 5V et à la mise à la masse (GND) de la Raspberry Pi.

Trig : cette broche est utilisée pour déclencher l'émission des signaux ultrasoniques.

Echo : cette broche reçoit les signaux ultrasoniques réfléchis.

Nous avons monté l'ultrason à l'avant du châssis du robot pour mesurer la distance le long de l'axe Y du robot.

Branchement du motor driver L298N à la Raspberry Pi

Le branchement du motor driver L298N à la Raspberry Pi a été effectué en utilisant les connexions suivantes :

IN1, IN2, IN3, IN4 : ces broches contrôlent les moteurs du robot.

ENA et ENB : sont été dédiées à la régulation de la vitesse des moteurs.

Alimentation : la broche VCC 12V du L298N à une source d'alimentation adaptée, à savoir une batterie LiPo. De plus, le fil 5V du L298N a été connecté au 5V de la Raspberry Pi, tandis que la mise à la masse (GND) de la Raspberry Pi a été reliée au GND du L298N.

Toutes les connexions de masse entre les composants sont correctement reliées dans le raspberry pi pour éviter tout problème de référence de tension.

Le schéma global de notre projet est illustré dans la figure ci-dessous (figure IV.10).

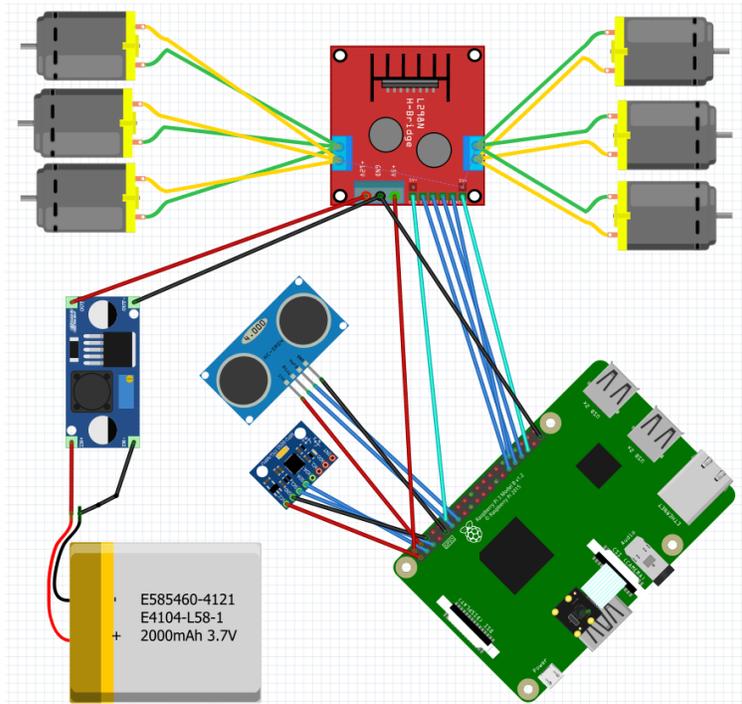


FIGURE IV.10 – Montage de circuit électrique du notre robot mobile (Fritzing)



FIGURE IV.11 – Montage réelle du robot mobile

IV.4 Configuration de Raspberry Pi

Pour démarrer le Raspberry Pi à partir de la carte microSD et installer le système d'exploitation, on a fait le travail suivant :

On a utilisé le Raspberry Pi Imager qui est un utilitaire graphique développé par la Fondation Raspberry Pi. Il permet aux utilisateurs d'installer facilement et rapidement des systèmes d'exploitation sur l'appareils Raspberry Pi, ainsi que sur d'autres cartes microSD compatibles.



FIGURE IV.12 – L'application Raspberry Pi Imager

Avec le Raspberry Pi Imager, nous avons pu télécharger le système d'exploitation sur une carte SD pour une utilisation avec le Raspberry Pi. Le processus est simple :

- On téléchargé et installé le Raspberry Pi Imager sur l'ordinateur à partir du site officiel.
- Connecté une carte SD vide à l'ordinateur par un lecteur de carte SD.
- Ouvrir le Raspberry Pi Imager et choisissé le système d'exploitation Raspberry Pi OS (32 bits) recommandé par l'application.
- Sélectionné la carte SD et après cliqué sur "Écrire" pour démarrer le processus de gravure.
- Une fois le processus de flashage terminé, on éjecte la carte microSD en toute sécurité de l'ordinateur.

après avoir inséré la carte SD sur sa port correctement dans le raspberry pi et branché un clavier, une souris et un écran (HDMI) au Raspberry Pi, et alimenté le Raspberry Pi par son chargeur original pour le démarrer. Le Raspberry Pi démarrera à partir de la carte microSD et lancera la configuration initiale, et mettre à jour le système on utilisant le "Terminal" pour avoir les derniers paquets logiciels et mises à jour de sécurité.

IV.4.1 Accès à distance au Raspberry Pi

En raison de la nature mobile de notre robot, il est nécessaire de le contrôler à distance. Pour ce faire, il est possible d'accéder à la carte Raspberry Pi via le protocole VNC, à condition que l'adresse IP de la carte soit connue.

L'adresse IP du Raspberry Pi

Pour trouver l'adresse IP de Raspberry Pi, nous avons utilisé un écran avec un clavier et une souris. La figure (IV.13) suivante montre l'interface de cet méthode et l'information fournie par ce dernier concernant le Raspberry Pi :



FIGURE IV.13 – Adresse IP de la carte Raspberry Pi

Connexion à distance via VNC

Virtual Network Computing est un système de visualisation et de contrôle d'un ordinateur distant. Il permet au logiciel client VNC de transmettre les informations de saisie du clavier et de la souris à l'ordinateur distant, possédant un logiciel serveur VNC à travers un réseau informatique.

Une fois le VNC est activé sur raspberry pi, le logo VNC apparait en haut à droite du bureau, accolé aux habituels Wi-Fi et Bluetooth.

L'interface de VNC Viewer est on ne peut plus claire. Une fois lancée, il suffira d'entrer l'adresse IP de raspberry pi, puis de renseigner vos identifiant et mot de passe pour accepter cette toute première connexion au serveur VNC.

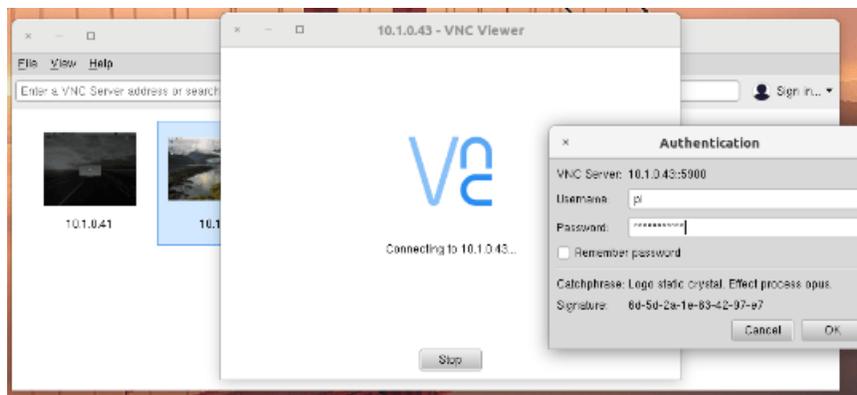


FIGURE IV.14 – Connexion à distance de raspberry avec VNC

IV.4.2 Langage de programmation (python)

La langage de programmation que nous devons utiliser c'est le python qui est un langage recommandé pour les nouvelles technologies comme par exemple le Raspberry.

Python est un langage de programmation polyvalent et interprété, connu pour sa lisibilité et sa simplicité syntaxique. Elle est souvent utilisé pour le développement d'applications, le scripting, l'analyse de données, l'apprentissage automatique et bien d'autres domaines.

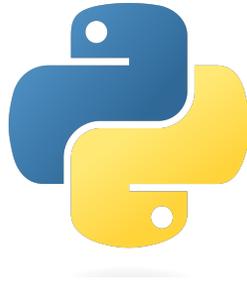


FIGURE IV.15 – Logo de python

IV.5 Calibration de MPU6050

La calibration est importante pour garantir que les mesures fournies par les capteurs sont précises et cohérentes. Elle vise à minimiser les erreurs systématiques afin d'obtenir des résultats plus fiables. Le processus de calibration du MPU6050 utilisé est expliqué dans les étapes suivantes :

Calibration du gyroscope La calibration du gyroscope du MPU6050 est essentielle pour obtenir des mesures précises de la vitesse angulaire sans biais. Pour commencer, on a placé le capteur immobile sur une surface plane pour éviter tout mouvement. Ensuite, on a pris plusieurs mesures de la vitesse angulaire dans les trois directions (X, Y et Z) pendant quelques secondes. Ces valeurs moyennes correspondent aux décalages du gyroscope. Lorsque la collection des données est en temps réel, on soustrayez ces décalages des mesures brutes du gyroscope pour obtenir des valeurs corrigées et plus précises.

Calibration de l'accéléromètre le module est maintenu dans une position immobile, et les valeurs de l'accéléromètre à l'intérieur sont lues et enregistrées. Ces valeurs servent de référence pour compenser tout biais ou dérive dans les mesures ultérieures. La calibration de l'accéléromètre du MPU6050 est importante pour obtenir des mesures précises de l'accélération linéaire.

Les données recueillies lors des étapes précédentes sont utilisées pour calculer les coefficients de calibration. Une fois que les coefficients de calibration sont déterminés, ils sont utilisés pour corriger les mesures en temps réel provenant des capteurs. Cela garantit que les mesures fournies sont aussi précises que possible.

```
MPU6050 module connection successful.
Old MPU6050 offsets: aX = -752, aY = -147, aZ = 622, gX = 0, gY = 0, gZ = 0
New MPU6050 offsets: aX = -906, aY = -83, aZ = 773, gX = 97, gY = 34, gZ = 20

aX = 0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.0 °/s, gY = 0.0 °/s, gZ = 0.2 °/s, T = 24.6 °C
aX = 0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.1 °/s, gY = 0.1 °/s, gZ = 0.1 °/s, T = 25.0 °C
aX = -0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = -0.1 °/s, gY = -0.1 °/s, gZ = 0.0 °/s, T = 25.0 °C
aX = -0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.2 °/s, gY = -0.0 °/s, gZ = 0.2 °/s, T = 25.2 °C
aX = -0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.1 °/s, gY = -0.1 °/s, gZ = -0.0 °/s, T = 25.3 °C
aX = 0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.1 °/s, gY = -0.1 °/s, gZ = -0.0 °/s, T = 25.4 °C
aX = 0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = -0.0 °/s, gY = -0.1 °/s, gZ = -0.1 °/s, T = 25.5 °C
aX = 0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.2 °/s, gY = -0.1 °/s, gZ = 0.1 °/s, T = 25.5 °C
aX = -0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.1 °/s, gY = -0.0 °/s, gZ = -0.1 °/s, T = 25.5 °C
aX = 0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.1 °/s, gY = -0.2 °/s, gZ = 0.2 °/s, T = 25.5 °C
aX = 0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = 0.0 °/s, gY = -0.1 °/s, gZ = -0.1 °/s, T = 25.6 °C
aX = -0.0 g, aY = -0.0 g, aZ = 1.0 g, gX = -0.0 °/s, gY = -0.2 °/s, gZ = 0.1 °/s, T = 25.5 °C
```

FIGURE IV.16 – Résultats d'interface de la calibration de MPU6050

IV.6 Localisation du robot par flux optique

Dans le cadre de notre projet, nous avons développé un script Python dédié à la localisation du robot à l'aide de la caméra Raspberry Pi. Notre objectif était de capturer une vidéo en temps réel et d'effectuer une analyse de flux optique afin de détecter les mouvements des points d'intérêt et de calculer les positions cartésiennes dans la scène. Cette méthode, combinée à une mesure de distance ultrasonique pour le déplacement vertical, nous a permis d'obtenir une localisation précise du robot. De plus, la figure suivante présente une image réelle du processus de détection des points d'intérêt.



FIGURE IV.17 – Visualisation des points d'Intérêt

Le processus débute par la capture successive d'images à partir de la caméra. Chaque image est ensuite convertie en niveaux de gris, ce qui permet de réduire la complexité de l'analyse tout en conservant les informations essentielles de la scène.

La détection de points d'intérêt, tels que les coins, est réalisée sur l'image en niveaux de gris. Ces points d'intérêt sont des régions caractéristiques de l'image, susceptibles de servir de points de suivi pour mesurer les mouvements.

Le flux optique est calculé à l'aide de l'algorithme Lucas-Kanade, qui permet de suivre le déplacement de ces points d'intérêt d'une image à l'autre. Ce calcul nous donne des informations sur la direction et l'amplitude des mouvements entre les images successives.

Afin de garantir la stabilité des points détectés, une vérification est effectuée sur la magnitude du mouvement. Les points dont le mouvement est en dessous d'un seuil prédéfini (variable stability threshold) sont filtrés. Cette étape permet de s'assurer que seuls les points d'intérêt stables et significatifs sont pris en compte.

De plus, nous avons rencontré des problèmes liés à la détection de mouvements indéterminés par la caméra. Pour remédier à cela, nous avons ajouté une étape de recherche de nouveaux points d'intérêt. Si la caméra ne détecte pas suffisamment de points d'intérêt stables, une nouvelle détection est effectuée, et le processus de calcul du flux optique est répété. Cela garantit que nous disposons toujours de points de suivi fiables.

Enfin, les déplacements horizontaux (" dx ") et verticaux (" dz ") calculés à partir du flux optique, ainsi que le déplacement vertical mesuré par le capteur ultrasonique (" dy "), sont cumulés au fil du temps pour obtenir des coordonnées cartésiennes. Ces coordonnées représentent la position du robot dans la scène par rapport à son point de départ.

Ce script offre ainsi une méthode robuste pour localiser notre robot mobile en temps réel, en utilisant la puissance du flux optique, de la caméra Raspberry Pi et du capteur ultrasonique. Cela ouvre des perspectives passionnantes pour des applications telles que la navigation autonome du robot, le suivi d'objets en mouvement dans l'environnement et bien d'autres applications nécessitant une localisation précise.

Présenté de manière claire et structurée dans le diagramme suivant, qui illustre les étapes essentielles de

la localisation du robot par flux optique à l'aide de la caméra Raspberry Pi et du capteur ultrasonique :

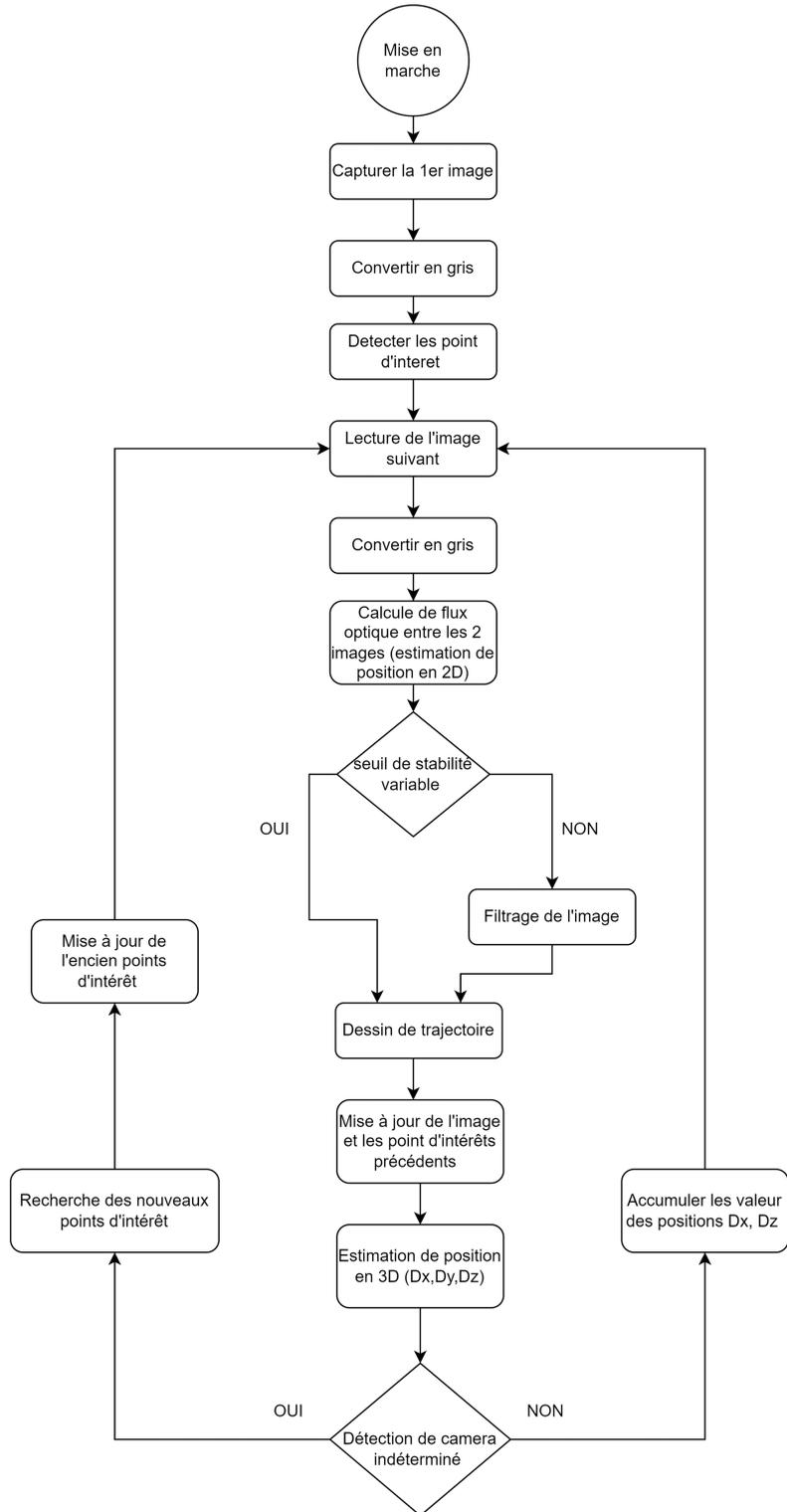


FIGURE IV.18 – Diagramme de flux optique

IV.7 Résultats obtenus par le MPU6050

Dans cette section, nous présentons les résultats obtenus à l'aide du capteur MPU-6050 pour compléter la pose du robot, notamment en termes de position obtenue par flux optique et d'orientation.

Mesure des Accélérations et des Vitesses Angulaires avec le MPU6050 le code Python ci-dessus présente la méthode pour obtenir les données d'accélération et de vitesse angulaire à partir du capteur MPU6050. Ce capteur est capable de fournir des informations essentielles sur les mouvements angulaires et les rotations autour de ses axes.

```
from mpu6050 import mpu6050
import time

mpu = mpu6050(0x68)

while True:

    accel_data = mpu.get_accel_data()
    print("Acc_X: " + str(accel_data['x']))
    print("Acc_Y: " + str(accel_data['y']))
    print("Acc_Z: " + str(accel_data['z']))
    print()

    gyro_data = mpu.get_gyro_data()
    print("Gyro_X: " + str(gyro_data['x']))
    print("Gyro_Y: " + str(gyro_data['y']))
    print("Gyro_Z: " + str(gyro_data['z']))
    print()
    print("-----")
    time.sleep(1)
```

La première partie du code commence par initialiser le capteur MPU-6050 en définissant son adresse I2C, qui est 0x68 dans cet exemple. Ensuite, la boucle principale lit en continu les données provenant du capteur et les affiche.

Les résultats obtenus sont les suivants :

Accélérations (Acc X, Acc Y, Acc Z) : Ces données représentent les accélérations le long des axes X, Y et Z respectivement. Elles sont mesurées en unités d'accélération gravitationnelle (g).

Vitesses angulaires (Gyro X, Gyro Y, Gyro Z) : Ces données représentent les vitesses angulaires autour des axes X, Y et Z. Elles sont mesurées en degrés par seconde ($^{\circ}/s$).

La boucle continue d'acquérir ces données et les affiche une fois par seconde. Ces informations sont essentielles pour comprendre les mouvements et les orientations de l'appareil où le capteur est installé. Les résultats peuvent être utilisés pour la navigation, la stabilisation ou la surveillance des mouvements dans diverses applications, telles que la robotique et la réalité augmentée.

Orientation selon l'axe Z le code Python ci-dessus illustre le processus d'acquisition et de calcul de cet angle. Le MPU-6050 est utilisé pour lire les données du gyroscope, permettant ainsi de détecter les changements d'orientation.

```
from mpu6050 import mpu6050
import time
import math

# Initialize MPU-6050 sensor
mpu = mpu6050(0x68) # Replace 0x68 with the correct I2C address of your MPU-6050
```

```

# Set initial values
angle_z = 0.0
num_readings = 100
offset_angle_z = 0.0
time_interval = 0.1 # Define the time interval between readings (in seconds)

# Take initial readings for offset calculation
for _ in range(num_readings):
    gyro_data = mpu.get_gyro_data()
    offset_angle_z += gyro_data['z']

# Calculate average offset
offset_angle_z /= num_readings

# Reset offset value
angle_z = 0.0

# Main loop
while True:
    # Read gyroscope data
    gyro_data = mpu.get_gyro_data()

    # Calculate change in angle around z-axis
    delta_angle_z = (gyro_data['z'] - offset_angle_z) * time_interval

    # Integrate to obtain cumulative angle
    angle_z += delta_angle_z

    # Normalize angle to be within 0-360 degrees range
    angle_z = angle_z % 360

    # Print the current angle
    print("Change in angle around Z-axis: {:.2f} degrees".format(angle_z))

    # Add a small delay between readings
    time.sleep(time_interval)

```

Le script commence par initialiser le capteur MPU-6050 et définir des valeurs initiales, telles que le nombre de lectures pour le calcul de la compensation du gyroscope, l'angle initial autour de l'axe Z et l'intervalle de temps entre les lectures. Ensuite, il effectue plusieurs lectures pour déterminer l'offset initial, qui est ensuite soustrait des lectures ultérieures pour obtenir les changements d'angle.

Le résultat obtenu est une mesure de l'angle de rotation autour de l'axe Z, exprimée en degrés. Le code effectue cette mesure en temps réel et l'affiche à chaque itération de la boucle principale. L'angle est normalisé pour être compris dans la plage de 0 à 360 degrés.

L'utilisation du capteur MPU-6050 permet d'obtenir une estimation de l'orientation du dispositif selon l'axe Z, ce qui peut être utile dans diverses applications, notamment dans le domaine de la robotique et de la navigation.

IV.8 Filtre de Kalman

Nous sommes en mesure d'obtenir des estimations plus précises de la position et de l'orientation du robot. Le capteur MPU-6050 nous fournit des informations précises sur l'orientation du robot dans l'espace. En intégrant ces données avec les informations de mouvement que nous obtenons à partir de l'analyse de flux optique, nous créons une vue plus complète et cohérente de la trajectoire du robot.

Pour mettre en œuvre l'utilisation du filtre de Kalman, nous avons utilisé la classe `KalmanFilter` de la bibliothèque `filterpy`, ce qui a grandement simplifié notre travail. Cette classe nous a permis d'efficacement définir un modèle de prédiction et d'observation, ainsi que les matrices de transition appropriées pour notre système.

Le fondement du filtre de Kalman se trouve sa capacité à optimiser la combinaison des données mesurées avec les prédictions issues du modèle. Dans notre cas, cela signifiait intégrer de manière efficace les données de déplacement (dx et dz) provenant de l'analyse de flux optique, ainsi que les données d'orientation du capteur MPU-6050. Cette intégration nous a permis d'obtenir des estimations plus robustes et précises de la position et de l'orientation du robot, tout en atténuant les perturbations et en réduisant les bruits.

Les figures suivantes présentent une comparaison détaillée entre les résultats de position et d'orientation obtenus avec et sans l'utilisation du filtre de Kalman. Les données représentent les mouvements dans les trois axes de translation (x , y , z) ainsi que les angles d'orientation autour de l'axe z .

Les déplacements enregistrés sont les suivants :

- Translation selon l'axe x : un déplacement de 20 cm dans la direction positive de l'axe x .
- Translation selon l'axe y : un déplacement de 20 cm dans la direction négative de l'axe y .
- Translation selon l'axe z : un déplacement de 20 cm dans le sens positive de l'axe z .
- Rotation d'angle de 90 degrés autour de l'axe z : une rotation de 90 degrés dans le sens des aiguilles d'une montre.

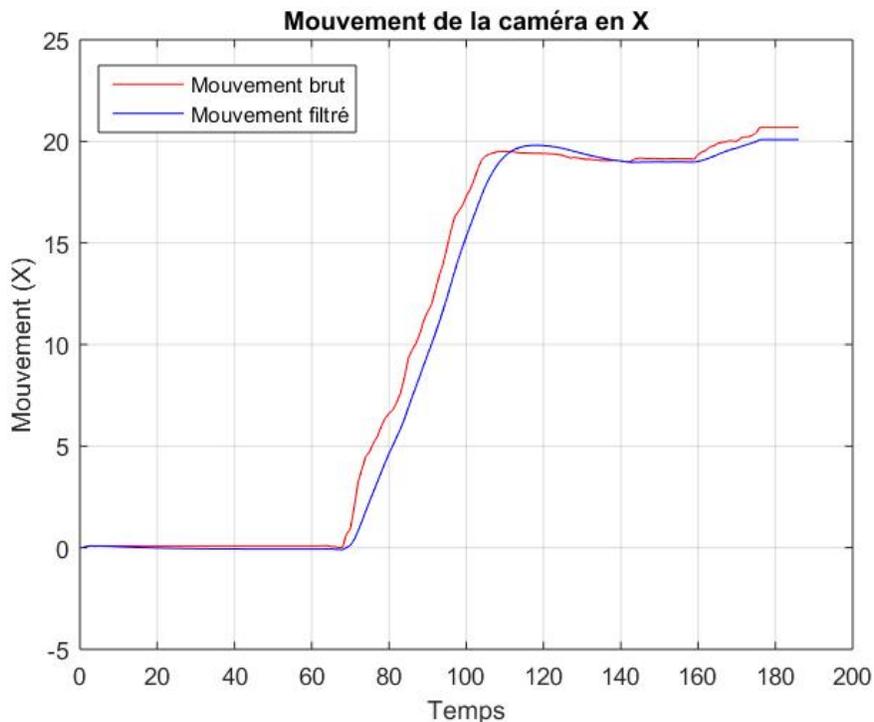


FIGURE IV.19 – Résultats des mouvements de bruit et de filtrage selon X par rapport au temps

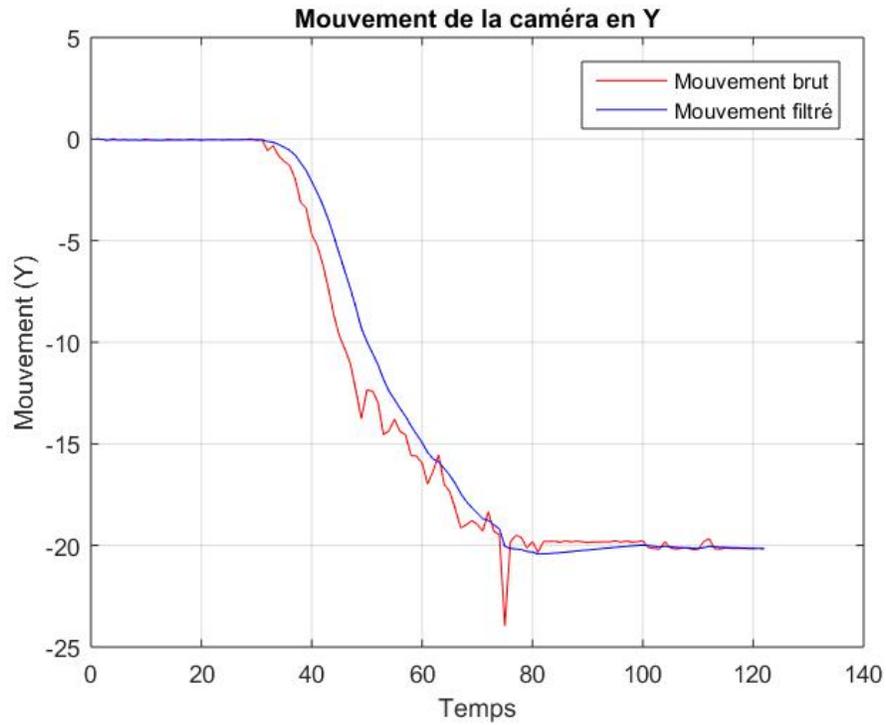


FIGURE IV.20 – Résultats des mouvements de bruit et de filtrage selon Y par rapport au temps

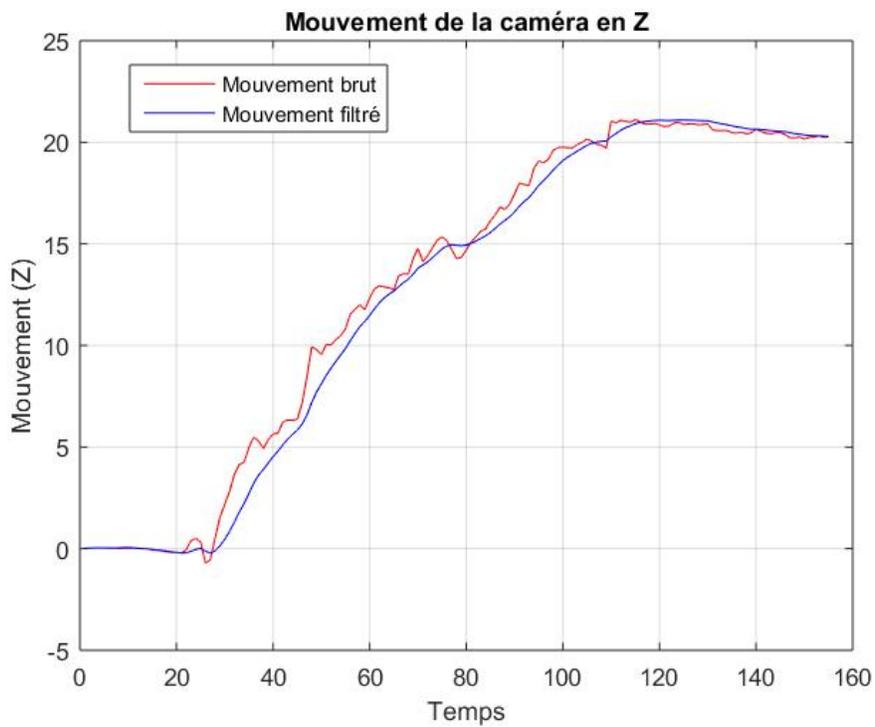


FIGURE IV.21 – Résultats des mouvements de bruit et de filtrage selon Z par rapport au temps

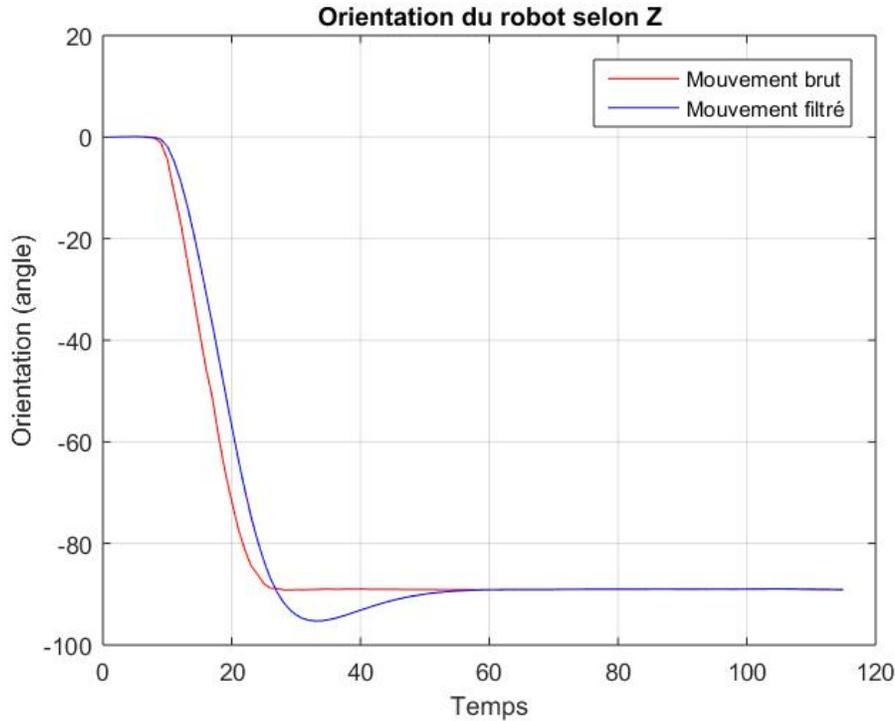


FIGURE IV.22 – Résultats des mouvements de bruit et de filtrage pour l’orientation du robot selon Z par rapport au temps

Les courbes de résultats présentent une comparaison entre les données filtrées par le filtre de Kalman et les données brutes, non filtrées. Cette comparaison permet de démontrer de manière saisissante la capacité du filtre de Kalman à améliorer la précision des estimations de position. De plus, le rôle fondamental du filtre de Kalman dans l’obtention de mesures plus précises pour l’orientation, surtout lorsque des mouvements compliqués et des rotations sont en jeu. Les données filtrées illustrent comment le filtre de Kalman optimise les informations fournies par les capteurs IMU (Unité de Mesure Inertielle), permettant ainsi d’obtenir des estimations plus fiables et stables de l’orientation, en prenant en compte les variations et les bruits présents dans les données brutes.

IV.9 Conclusion

Dans ce dernier chapitre, nous avons choisi avec attention les éléments essentiels pour notre robot mobile. Après avoir terminé le montage final, nous avons mené des expérimentations approfondies pour confirmer nos décisions et mesurer les performances de notre robot. De plus, nous avons configuré le Raspberry Pi et calibré le MPU6050 pour une meilleure précision.

Finalement, après de nombreux essais, nous avons réussi à déterminer avec précision la localisation de notre robot. Pour ce faire, nous avons utilisé deux sources de données cruciales : le flux optique pour la détermination de la position cartésienne et le MPU6050 pour l’acquisition des données d’orientation. L’élément clé de notre succès a été l’intégration du filtre de Kalman dans notre processus. Cet outil nous a permis d’atténuer les perturbations et les bruits, conduisant ainsi à des résultats de grande précision.

Conclusion générale et perspectives

L'accroissement de l'intérêt pour le domaine de la robotique a constitué la principale motivation de cette recherche. L'objectif de ce travail était de permettre la localisation précise des robots en milieu extérieur en intégrant les données issues de l'approche de flux optique et du capteur inertiel en temps réel. Grâce à cette approche, nous avons réussi à obtenir une estimation plus précise et robuste de la position et de l'orientation du robot, même dans des environnements complexes et exigeants.

Cependant, il convient de noter certaines limites associées à l'approche basée sur le flux optique. Le suivi des points d'intérêt peut être sensible aux variations de luminosité, ce qui peut entraîner des problèmes de fiabilité dans des conditions d'éclairage changeantes. De plus, les légères vibrations de la caméra en mouvement et les changements rapides des points d'intérêt en mouvement peuvent influencer la précision du positionnement.

D'autre part, bien que le capteur inertiel MPU-6050 ait été utilisé avec succès dans cette recherche, il présente également des limitations. Le capteur ne fournit pas directement les accélérations linéaires, ce qui limite son utilisation en tant que source d'estimation de la position. En outre, notre étude s'est principalement concentrée sur la détermination de l'orientation du robot plutôt que sur ses accélérations et vitesses angulaires. De plus, le MPU-6050 nécessite une calibration minutieuse pour garantir des résultats précis, ce qui peut être un processus récurrent au lieu d'être une étape unique.

Afin d'améliorer la précision de ces données, nous avons employé deux types de filtres. Tout d'abord, nous avons mis en place un filtre qui se base sur la stabilité des points pour sélectionner ceux dont la stabilité est inférieure à un seuil spécifié, tout en éliminant les points dont la stabilité dépasse ce seuil. Cette approche s'est révélée utile pour extraire les mouvements ou les variations lentes et significatives au sein de l'ensemble de données des points, tout en ignorant les fluctuations rapides ou le bruit.

En outre, l'utilisation du filtre de Kalman a joué un rôle essentiel dans l'estimation des mouvements de bruit et des perturbations dans notre système. Ce filtre a permis d'améliorer significativement la précision de la détection des variations de position à partir des données du flux optique, avec une estimation de l'orientation provenant du MPU-6050. Ces deux sources d'informations, nous avons pu obtenir une estimation de la localisation du robot plus fiable et cohérente, ce qui a renforcé notre capacité à naviguer et à interagir dans des environnements complexes.

En envisageant le futur de ce projet de robot mobile, d'importants développements se profilent afin d'amplifier son influence dans le domaine agricole. L'intégration d'un système de détection d'obstacles promet une navigation sécurisée et précise au sein d'environnements agricoles complexes. Cette capacité permettra au robot de détecter et d'éviter habilement les obstacles tels que les rochers, les arbres et les équipements, réduisant considérablement les risques de perturbations. Simultanément, l'intégration d'un capteur spécialisé dédié à l'analyse de la fertilité du sol représente une avancée majeure. En collectant des données relatives aux nutriments, au pH du sol et à d'autres paramètres clés, le robot deviendra un atout inestimable pour une gestion précise des cultures, favorisant ainsi des pratiques agricoles durables et optimisées. La convergence de ces évolutions renforcera davantage le rôle du robot mobile en tant qu'élément central pour une agriculture moderne et efficace.

.1 Annexes

.1.1 Raspberry Pi 3 model B /Annexes

.1.2 La carte mère Raspberry Pi 3

- Processeur intégré Quad-core ARM Cortex-A53 1.2 GHz (BroadcomBCM2837)
- GPU Dual Core Video Core IV Multimedia Co-Processor, RAM : 1024 Mo
- HDMI, 4x USB, RJ45, jack 3.5 mm, connecteurs pour APN et écran tactile
- Lecteur de cartes Micro SD
- Wi-Fi b/g/n et Bluetooth 4.1
- Support des distributions dédiées basées sur Linux et Windows 10 [42]

Les composants du Raspberry Pi 3

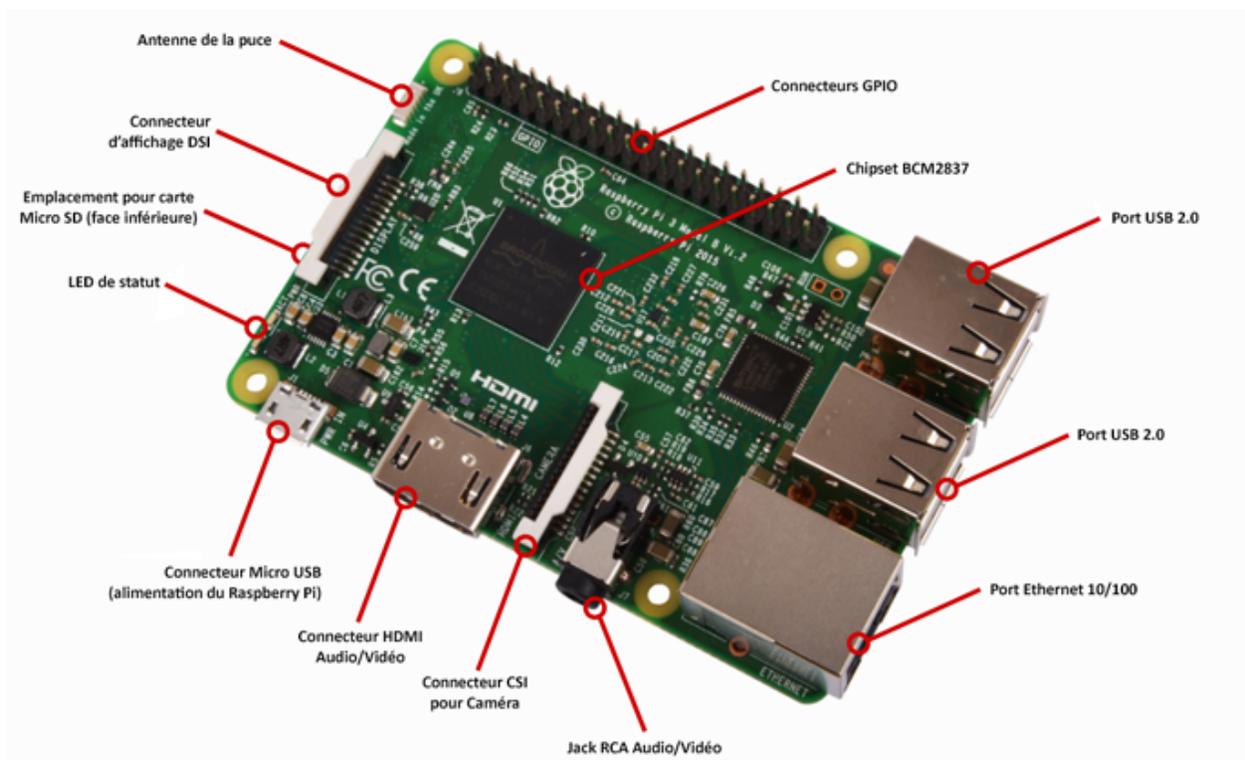


FIGURE 23 – Les composants du Raspberry Pi 3

Processeur Broadcom BCM2837 quad-core ARM Cortex-A53 64 bits cadencé à 1,2 GHz.

Mémoire vive (RAM) 1 Go LPDDR2 SDRAM.

Processeur graphique Broadcom VideoCore IV, prenant en charge les résolutions jusqu'à 1080p à 60 images par seconde.

Connectivité réseau Ethernet 10/100 BaseT, compatible avec le protocole PoE (Power over Ethernet) via un module PoE optionnel.

Connectivité sans fil Wi-Fi 802.11n et Bluetooth 4.1 BLE (Bluetooth faible consommation).

Ports d'entrée/sortie 4 ports USB 2.0, 1 port HDMI, 1 port Ethernet, 1 prise jack 3,5 mm pour la sortie audio/vidéo, 1 port CSI pour la caméra Raspberry Pi et 1 port DSI pour l'écran tactile Raspberry Pi.

Emplacement pour carte microSD utilisé pour stocker le système d'exploitation et les données utilisateur.

Les broches d'entrées-sorties (GPIO) un ensemble de broches d'E/S générales (GPIO) à usage général permettant d'interfacer la carte avec des composants électroniques externes.

Alimentation 5 V, via un port micro-USB ou une connexion GPIO à 40 broches.

.1.3 L298N

.1.4 Description du L298N

- Léger, petit
- Il est utilisé pour contrôler les moteurs à courant continu et les moteurs pas à pas unipolaires et bipolaires.
- Des diodes de roue libre intégrées pour protéger le circuit contre les surtensions induites par le moteur lorsqu'il est en arrêt.
- Un dissipateur pour dissiper la chaleur en cas de forte charge.
- Un sélecteur pour sélectionner la source d'alimentation
- Un sélecteur pour sélectionner la source d'alimentation
- 2 Sorties pour des moteurs a courant continu
- Indicateur LED pour alimentation 5V
- Indicateur LED pour sens de rotation moteur

Caractéristiques du L298N

- Il contient deux ponts en H complets
- Tensions : 5V
- Courant : de 0 à 36mA (courant maximal : 2A dans une seule branche)
- Puissance Max : 25W
- Poids : 48g
- Dimensions : 60mm x 54mm

Fonctionnement du double pont en H (L298N)

Le double pont en H permet de faire passer le courant électrique dans les deux sens à travers le moteur, en inversant la polarité de l'alimentation. Les ponts en H sont constitués de quatre transistors : deux transistors NPN et deux transistors PNP, qui sont configurés pour fonctionner ensemble en un seul bloc.

Les ports ENA et ENB permettent de gérer l'amplitude de la tension délivrée au moteur, grâce à un signal PWM, les ports IN1, IN2 pour le moteur A et les ports IN3, IN4 pour le moteur B permettent de contrôler le pont en H par conséquent le sens de rotation des moteurs [43]

— Contrôle du sens de moteur droit A :

	Sens avant	Sens arrière
IN1	BAS	HAUT
IN2	HAUT	BAS

— Contrôle du sens de moteur gauche B :

	Sens avant	Sens arrière
IN3	BAS	HAUT
IN4	HAUT	BAS

1.5 Ultrason

Caractéristiques du capteur HC-SR04

Les caractéristiques en détail du Capteur sonar à Ultrasons HC-SR04 sont présentées dans le tableau ci-dessous :

Distance de captation	2 cm à 5 m
Résolution (précision)	3mm
Tension d'exploitation (voltage d'entrée)	5 V
Courant (ampérage d'entrée)	15 mA
Fréquence d'opération	40 Hz
Angle de mesure	30 degrés
Angle efficace	15 degrés
Signal d'entrée trigger	10 μ s TTL impulsion
Dimensions (longueur x largeur x hauteur)	5 mm x 20 mm x 15 mm

Broches de connexion du l'ultrason

Capteur se compose d'un émetteur d'ultrasons de fréquence 40 kHz. Il peut être utilisé comme élément principal dans une télémessure ou utilisé pour déterminer la vitesse du son.

Le brochage est le suivant :

- VCC : 5V d'alimentation.
- GND : Masse.
- Trig : Borne de déclenchement de la mesure, connectée à un PIN numérique de raspberry réglé en sortie.
- Echo : Borne de récupération du signal de mesure, connectée à un PIN numérique de raspberry réglé en entrée.

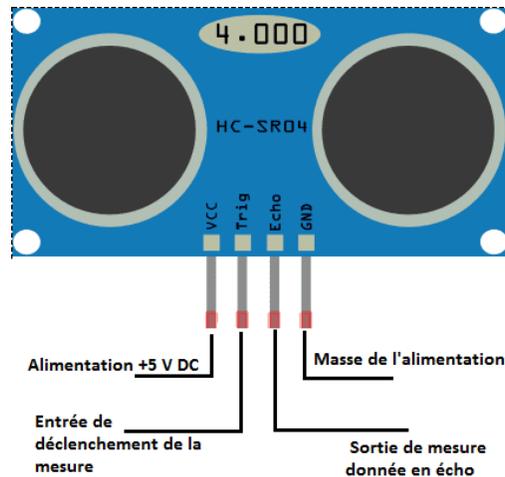


FIGURE 24 – Broches de connexion du HC-SR04

1.6 Les caractéristiques du 6WD

Matériel le châssis est en métal, ce qui le rend robuste et résistant.

Les roues le robot dispose de six roues motrices d'un courant évalué :350mA * 6, moteur de vitesse d'origine :17000 tr/min, et vitesse de l'arbre de sortie :500 tr/min, et d'un Couple nominal :5 kg/cm, ce qui permet une meilleure traction et une meilleure stabilité dans des environnements tout-terrain.

Dimensions sont d'environ 280 x 200 x 105 mm, ce qui permet de loger une variété de composants électroniques.

Poids Le poids du châssis est d'environ 1,7 kg.

Charge maximale La capacité de charge maximale est d'environ 5 kg.

Tension De fonctionnement la tension nominale du système est de 12V DC, et la tension d'entrée recommandée est de 12V à 24V DC.

Les caractéristiques de python

Syntaxe claire et lisible Il utilise une syntaxe facile à comprendre grâce à son indentation significative (l'alignement des blocs de code détermine leur exécution) plutôt que des accolades ou des mots-clés.

Syntaxe claire et lisible Il offre une abstraction élevée des détails de bas niveau, ce qui permet de se concentrer davantage sur la résolution des problèmes plutôt que sur les détails d'implémentation.

Large bibliothèque standard Il dispose d'une vaste bibliothèque standard qui fournit de nombreux modules et fonctionnalités prêts à l'emploi, couvrant divers domaines tels que l'entrée/sortie, la manipulation de chaînes, le traitement des fichiers, le réseau, les mathématiques, etc.

Portabilité Il est disponible sur de nombreuses plates-formes, telles que Windows, macOS et différentes distributions de Linux. Cela permet de créer des applications qui fonctionnent sur différents systèmes d'exploitation sans nécessiter de modifications importantes du code source.

Bibliographie

- [1] Pierre Andry, Philippe Gaussier, Sorin Moga, Jean-Paul Banquet, and Jacqueline Nadel. Learning and communication via imitation : An autonomous robot perspective. *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans*, 31(5) :431–442, 2001.
- [2] Simon Lavington. *Early British Computers : The Story of Vintage Computers and the People who Built Them*. Manchester University Press, 1980.
- [3] W. Grey Walter. The tortoise : A machine for stimulus-response research. Technical report, Cambridge University Engineering Department, 1949.
- [4] Robert L. Thorndike and Edward L. Thorndike. A professional and personal appreciation. In Gregory Kimble, Michael Wertheimer, and Charlotte White, editors, *Portraits of Pioneers in Psychology*. Lawrence Erlbaum, Hillsdale, 1991.
- [5] Francisco Rubio, Francisco Valero, and Carlos Llopis-Albert. A review of mobile robots : Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2), 2019.
- [6] Bernard Benet. Perception de l’environnement naturel pour des applications d’agriculture de précision. In *Journées nationales sur ROS 2018*, pages 89–p, 2018.
- [7] Hansueli Dierauer. Nouveaux systèmes de sarclage : Précis, mais pas encore utilisables partout. *Bioactualités*, (9) :14–15, 2018.
- [8] Siegfried Jahnke, Johanna Roussel, Thomas Hombach, Johannes Kochs, Andreas Fischbach, Gregor Huber, and Hanno Scharr. Pheno seeder-a robot system for automated handling and phenotyping of individual seeds. *Plant physiology*, 172(3) :1358–1370, 2016.
- [9] Richard Woodward. Robots agricoles – l’automatisation intelligente au service de l’agriculture. juillet 7, 2023.
- [10] Maroua Ladhari. *Architecture générique de fusion par approche Top-Down : application à la localisation d’un robot mobile*. PhD thesis, Université Clermont Auvergne [2017-2020], 2020.
- [11] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on robotics and automation*, 12(6) :869–880, 1996.
- [12] Vincent Frémont. Odométrie 3d vision/lidar pour les véhicules intelligents. In *Journées Nationales de la Recherche en Robotique, JNRR’09*, 2009.
- [13] Rodolphe Dubois. *Méthodes de partage d’informations visuelles et inertielles pour la localisation et la cartographie simultanées décentralisées multi-robots*. PhD thesis, École centrale de Nantes, 2021.
- [14] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Localisation par vision monoculaire pour la navigation autonome. *traitement du signal*, 23(1) :95–107, 2006.
- [15] Pierre Merriaux. *Contribution à la localisation robuste embarquée pour la navigation autonome*. PhD thesis, normandie université, 2016.
- [16] France Saint Etienne du Rouvray. Détection, localisation et suivi d’objets basé deep learning pour un fauteuil roulant intelligent. 2020.
- [17] Noureddine Slimane. *Système de localisation pour robots mobiles*. PhD thesis, Université de Batna 2, 2008.
- [18] Kamel HEDDOUCHE. *Etude et conception d’un robot marchant*. PhD thesis, Université Mohamed Khider Biskra, 2014.

- [19] Slimane Nouredine. *Système de localisation pour robots mobiles*. PhD thesis, Université de Batna, 2005.
- [20] Fares BOUSIF. *Conception, réalisation et implémentation d'un robot mobile dans un environnement de travail*. PhD thesis, Université Abou-Bekr Belkaid, 2020.
- [21] Damien LHOMME-DESAGES. *Commande d'un robot mobile rapide à roues non Directionnelles sur sol naturel*. PhD thesis, Université Paris 6, April 2013.
- [22] David Filliat. *Robotique mobile*. PhD thesis, EDX, 2011.
- [23] Bernard Bayle. *Robotique mobile*. PhD thesis, Université de Strasbourg, 2013.
- [24] J. Neimark and N. Fufaev. *Dynamics of Nonholonomic Systems*, volume 33 of *Translations of Mathematical Monographs*. 1972.
- [25] Abdelhakim Debib. *Modélisation et commande d'un robot mobile à 4 roues*. PhD thesis, 2016.
- [26] Hakan Temeltaş Sercan Arslan. Robust motion control of a four wheel drive skid-steered mobile robot., 2011.
- [27] Nouioua Islem Zekhref Oussama. Localisation et navigation d'un robot mobile par webcam. <https://di.univ-blida.dz/jspui/bitstream/123456789/1970/1/Untitled.pdf>, 2015.
- [28] Feth Allah Hadjila and Réda Bouabdallah. Reconnaissance des visages en utilisant les réseaux de neurones. Mémoire d'ingénieur, 2003.
- [29] François Guérin—Edouard Leclercq—Alain Faure and Marc Gorka. Commande d'un robot mobile par vision artificielle stéréoscopique.
- [30] Université de Picardie Jules Verne. Cours de traitement d'image. Licence de Physique S6, 2007-2008.
- [31] Maamar Meliani. *Segmentation d'Image par Coopération Régions-Contours*. PhD thesis, Ecole Nationale Supérieure d'Informatique, 2012.
- [32] JP Gastellu-Etchegorry. Acquisition et traitement d'image numérique. *Université Paul Sabatier IUT-Département de Mesures Physiques*, 2008.
- [33] Maïtine Bergounioux. Quelques méthodes de filtrage en traitement d'image. 2010.
- [34] Détermination de la position d'un objet à l'aide d'un couple de caméras., 2012.
- [35] Gary Bradski, Adrian Kaehler, et al. Opencv. *Dr. Dobb's journal of software tools*, 3(2), 2000.
- [36] Sherad Rawat. Raspberry pi camera module calibration using opencv. Feb 21, 2021.
- [37] B.K.P. Horn and B.G. Schunck. Determining optical flow. 1981.
- [38] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81 : 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.
- [39] Adam Heinrich. *An Optical Flow Odometry Sensor Based on the Raspberry Pi Computer*. PhD thesis, Master's thesis, Czech Technical University in Prague, 2017.
- [40] Michael J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 292–302, Maui, HI, 1991. IEEE Computer Society Press.
- [41] Hichem Belattar. Quelques considérations sur le filtre de kalman discret. *FACULTÉ DES SCIENCES UNIVERSITÉ DE SHERBROOKE Québec, Canada*, 2017.
- [42] Raspberry Pi Foundation. Teach, learn, and make with the Raspberry Pi Foundation — raspberypi.org. [Accessed 24-Apr-2023].
- [43] BEDADI Mehdi. *Intitulé : ETUDE ET REALISATION D'UN VEHICULE AUTONOME*. PhD thesis, UNIVERSITE BADJI MOKHTAR ANNABA, 2019.