

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Automatique
Laboratoire de commande des processus



Thèse de Doctorat en Automatique

Présentée par **BOUCHELOUKH Abdelghani**

Magister en Automatique de l'Ecole Militaire Polytechnique

Intitulée

Systeme de vision pour la commande d'un UAV « *Unmanned aerial vehicle* », type quadrotor

Membres du jury :

Président	M. BOUKHETALA Djamel	Professeur, ENP
Directeur de thèse	M. BOUDJEMA Farès	Professeur, ENP
Co-directeur de thèse	M. NEMRA Abdelkrim	MCA, EMP
Examineur	M. LARABI Slimane	Professeur, USTHB
Examinatrice	Mme. ACHOUR Nouara	Professeur, USTHB
Examineur	M. LARBES Chérif	Professeur, ENP
Examineur	M. DJEKOUN Abdeloualid	Directeur de Recherche, CDTA
Invité	M. YACEF Fouad	MRA, CDTA

ENP 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Automatique
Laboratoire de commande des processus



Thèse de Doctorat en Automatique

Présentée par **BOUCHELOUKH Abdelghani**

Magister en Automatique de l'Ecole Militaire Polytechnique

Intitulée

Systeme de vision pour la commande d'un UAV « *Unmanned aerial vehicle* », type quadrotor

Soutenue publiquement le 03/07/2022 devant le jury composé de:

Président	M. BOUKHETALA Djamel	Professeur, ENP
Directeur de thèse	M. BOUDJEMA Farès	Professeur, ENP
Co-directeur de thèse	M. NEMRA Abdelkrim	MCA, EMP
Examineur	M. LARABI Slimane	Professeur, USTHB
Examinatrice	Mme. ACHOUR Nouara	Professeur, USTHB
Examineur	M. LARBES Chérif	Professeur, ENP
Examineur	M. DJEKOUN Abdeloualid	Directeur de Recherche, CDTA
Invité	M. YACEF Fouad	MRA, CDTA

ENP 2022

REMERCIEMENTS

Je tiens tout d'abord à exprimer ma reconnaissance envers les membres du jury, dont la renommée et la qualité scientifique honore grandement ce travail :

- M **BOUKHETALA Djamel**, Professeur à l'ENP pour m'avoir fait l'honneur de présider le jury de ma thèse.
- M. **LARABI Slimane**, Professeur de l'USTHB, pour m'avoir fait l'honneur d'être examinateur de ce travail ;
- Mme **ACHOUR Nouara** , Professeur de l'USTHB, pour m'avoir fait l'honneur d'être examinatrice de ce travail.
- M. **LARBES Chérif**, EMP, Professeur à l'ENP, pour m'avoir fait l'honneur d'être examinateur de ce travail ;
- M. **DJEKOUN Abdeloualid**, Directeur de Recherche, CDTA, pour m'avoir fait l'honneur d'être examinateur de ce travail ;
- M. **YACEF Fouad**, Maître des conférences A, CDTA, pour m'avoir fait l'honneur d'être invité.

Je tiens à exprimer toute ma gratitude aux rapporteurs Professeur **BOUDJEMA Farès** et **Docteur. NEMRA Abdelkrim** d'avoir proposé le sujet sur lequel j'ai travaillé, et qui ont assuré la direction et l'encadrement du travail présenté dans cette thèse.

Je remercie aussi mon Département ainsi que les responsables de ma Direction de m'avoir donné la chance de continuer mes études.

Je tiens aussi à remercier le Chef de département Automatique ainsi que le Directeur de la Recherche de la Formation Post Graduée et les responsables de l'ENP ;

Ce travail a été réalisé au sein du Laboratoire Commande des processus. Je tiens également à remercier M. Tadjine, Chef dudit Laboratoire à l'ENP pour leurs aides, leurs générosités, leurs compréhensions et leurs patiences le long de ce projet de thèse.

Je tiens aussi à remercier les responsables de l'EMP, particulièrement, le Dr A. BAZOULA chef d'U.E.R Automatique, le Chef du Laboratoire Robotique et Véhicules Intelligents et le Directeur de la Recherche de la Formation Post Graduée de l'Ecole Militaire Polytechnique, de m'avoir accueilli au niveau de leur laboratoire.

Je tiens aussi à remercier le Professeur DJEMAI Mohamed, ainsi que toute l'équipe du laboratoire LAMIH de l'université de Valenciennes en France, particulièrement, les docteurs ELASSAYH Mohamed et ANGGRAENI Pipit.

Je tiens aussi à remercier le Professeur ROCHDI Merzouki, ainsi que toute l'équipe de l'université de Lille 1 et du Centre CRISL, particulièrement, les docteurs BELAROUSSI Abdelkader, BENSARKRENE Ismail, OUZIAL Mahdi, TERMACH Adel, CHEMALI Riadh et ADHIE Wahyu.

Mes vifs remerciements vont à mes co-auteurs Dr Fouad YACEF et Dr. DEMIME Fethi, Mr OUSSAR Abdellatif et LOUALI Rabeh.

Mes vifs remerciements vont aussi à tous mes collègues de l'UER Automatique de l'EMP, Drs : GUIATNI Mohamed, MANICH, SAKHI Samir, HANK Majid, ALLAM Ahmed, KOBZILLI ELhouari, BOUZID Yasser, KHALLAL Othman, ARAAR Oualid, Choukri, M. ZAMOUM et M. BESGHIR.

Dédicace

À ma mère,

À mon père,

Ma femme et mes enfants Ilaf-Zineb et Adem

À mes frères,

À mes Sœurs,

*À toutes les Familles, BOUCHELOUKH, BOUGHRIRA,
SOUYAD, YAKHELEF, BOUNEMEUR.*

À mes amis et collègues.

ملخص

يدرس هذا العمل تتبع المسار ثلاثي الأبعاد لطائرة بدون طيار- درون أو طائرة مسيرة ذات أربع أجنحة دوارة من نوع رباعيات الأجنحة باستخدام نظام تحديد المكان المرئي والذي تم دمجه في الحلقة المغلقة للتحكم. أولاً، يُقترح قياس المسافات البصري للجسم وكيف لحل مشكلة تحديد الموقع للطائرات بدون طيار. يعتمد الحل المقترح أساساً على تطبيق أقرب نقاط تكرارية على ميزات صورة بطريقة مسرعة (SURF). حيث نأخذ بعين الاعتبار العمق العكسي لهذه النقاط ثلاثية الأبعاد (المعلمة) لإعطاء أهمية أكبر للنقاط المعلمة القريبة. ثانياً، يعتمد تنفيذ خوارزمية تتبع المسار على الرؤية باستخدام وحدة التحكم بتراجع الخطوات Backstepping، أين يتم تحسين قيم هذا التحكم باستخدام خوارزمية تحسين سرب الجسيمات (Particle Swarm Optimization-PSO). أخيراً، ولتجنب محدودية تحديد الموقع البصري (من خلال: البيئة المظلمة، منطقة موحدة الشكل، المكان كثير حركة، إلخ). يتم دعم تحديد الموضع المرئي بنظام موضع العطالة وكذا باستخدام مرشح كالمان وكيف بمبدأ التحكم الغموض (Fuzzy Adaptive Kalman Filter). استناداً إلى متوسط عمق عدد النقاط ثلاثية الأبعاد المقدر تضبط خوارزمية FAKF بشكل تكيفي معلمات EKF لإعطاء أهمية أكبر للمستشعر المناسب لتحديد الموضع (عن طريق الرؤية المجسمة أو العطالة أو كليهما في نفس الوقت) من أجل توفير أفضل تقدير للموضع. تم التحقق من صحة الخوارزميات المقترحة باستخدام المحاكاة والبيانات التجريبية. تم تصور عدة سيناريوهات لمسارات مختلفة. وتم تقييم النهج المقترح باستخدام عدد مختلف من الميزات ثلاثية الأبعاد في الصور المجسمة مع الأخذ في الحسبان التشويش الكبير في القياس. تم الحصول على أداء جيد في النتائج الذي يؤكد كفاءة النهج المقترح.

الكلمات المفتاحية - الطائرات بدون طيار، تتبع المسارات، التحكم بتراجع الخطوات، تحسين عناصر السرب، مرشح كالمان، النظام الضبابي وقياس المسافات البصري.

Abstract

This work studies the 3D trajectory tracking of an Unmanned Aerial Vehicle (UAV) type Quadrotor using the visual localization system, which is integrated in a feedback control loop. First, an adaptive stereo visual odometry (SVO) is proposed to solve the problem of UAV localization. The proposed solution is mainly based on applying iterative closest point (ICP) on Speeded Up Robust Features (SURF). The inverse depth of 3D points is considered to give more importance to the nearest features. Secondly, a vision based trajectory tracking algorithm is implemented using Backstepping controller, the parameters of this latter are optimized using Particle Swarm Optimization (PSO) algorithm. Finally, and in order to avoid the limitation of visual localization (dark environment, uniform area, dynamic scene, etc). The visual pose is aided by inertial pose using a Fuzzy Adaptive Kalman Filter (FAKF), based on the number and the average depth of the estimated 3D points, the FAKF algorithm tune adaptively the EKF parameters to give more importance to the suitable sensor (SVO or IMU or both at the same time) in order to provide the best pose estimation. The proposed algorithms are validated using simulation and experimental data. Many scenarios are considered with different trajectories. The proposed approach is evaluated using different number of 3D features with significant measurement noise. Good performances are obtained which confirms the efficiency of the proposed approach.

Keywords — Unmanned aerial vehicles, Trajectory tracking, backstepping control, PSO, adaptif Kalman filter, Fuzzy System and Visual odometry

RESUME

Ce travail étudie le suivi de trajectoire 3D d'un Quadrotor de type UAV (Unmanned Aerial Vehicle) en utilisant un système de localisation visuelle, qui s'intègre dans une boucle fermée de commande. Tout d'abord, une odométrie visuelle stéréo adaptative (OVS) est proposée pour résoudre le problème de la localisation des drones. La solution proposée est principalement basée sur l'application du point le plus proche itératif (ICP) et sur les caractéristiques robustes accélérées (SURF), avec l'hypothèse que la profondeur inverse des points 3D donne plus d'importance aux entités les plus proches. Deuxièmement, un algorithme de suivi de trajectoire basé sur la vision est implémenté à l'aide du contrôleur Backstepping, les paramètres de ce dernier sont optimisés à l'aide de l'algorithme Particle Swarm Optimization (PSO). Enfin, afin d'éviter la limitation de la localisation visuelle (environnement sombre, zone uniforme, scène dynamique, etc.), la pose visuelle est facilitée par la pose inertielle à l'aide d'un filtre de Kalman adaptatif flou (FAKF), sur la base du nombre et de la profondeur moyenne des points 3D estimés, l'algorithme FAKF ajuste de manière adaptative les paramètres EKF pour donner plus d'importance au capteur approprié (OVS ou IMU ou les deux en même temps) afin de fournir la meilleure estimation de pose. Les algorithmes proposés sont validés à l'aide de simulations et de données expérimentales. De nombreux scénarios ont été appliqués avec des différentes trajectoires. L'approche proposée est évaluée en utilisant un nombre différent d'entités 3D avec un bruit de mesure significatif. De bonnes performances sont obtenues qui confirment l'efficacité de l'approche proposée.

Mots clés - Véhicules aériens sans pilote, suivi de trajectoire, commande Backstepping, PSO, Filtre de Kalman adaptatif, système flou et odométrie visuelle.

Table des matières

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

INTRODUCTION GENERALE..... 16

CHAPITRE I: NAVIGATION VISUELLE DES QUADRIROTORS 21

I.1	INTRODUCTION.....	21
I.2	APERÇU SUR LES VEHICULES AERIENS SANS PILOTE.....	21
I.3	CONCEPT NAVIGATION	24
I.4	CLASSIFICATION DES DRONES.....	25
I.5	MATERIELS DES QUADRIROTORS	27
I.6	LOGICIELS (SOFTWARE) DES MINI-DRONES.....	29
I.7	CAPTEURS DES MINI-DRONES.....	29
I.7.1	Capteurs proprioceptifs.....	30
I.7.1.1	Capteurs de déplacement.....	30
a)	Accéléromètres	30
b)	Radars Doppler	31
I.7.1.2	Capteurs d'attitude.....	31
a)	Gyroscope, Gyromètre et Gyrocompas.....	31
b)	Centrale inertielle.....	32
I.7.2	Capteurs extéroceptifs	33
I.7.2.1	Système de positionnement global (GPS)	33
I.7.2.2	Télémètres	34
I.7.2.3	Vision	36
I.7.2.4	Caméra de profondeur	37
I.8	APPLICATIONS DES QUADRIROTORS :.....	37
I.9	CONTROLE ET COMMANDE VISUELLE DES MINI-DRONES.....	38
I.9.1	Asservissement visuel:	38
I.9.2	Commande et suivi trajectoire par vision	39
I.10	LOCALISATION ET CARTOGRAPHIE	40
I.10.1	Localisation du robot volant	40
I.10.1.1	Localisation relative	40
I.10.1.2	Localisation absolue	41
I.10.1.3	Localisation hybride	42
I.10.2	Cartographie	42
I.10.2.1	La carte métrique :.....	42

I.10.2.2 La carte d'occupation (occupancy grid) :	43
I.10.2.3 Carte de points d'intérêt (Feature Maps) :	43
I.10.2.4 Carte topologique :	44
I.10.3 SLAM	44
I.10.4 Odométrie visuelle	45
I.10.5 Méthodes d'estimation	46
I.10.5.1 Méthodes basées sur les points d'intérêt	46
I.10.5.2 Méthodes directes	47
I.10.6 Méthodes avec apprentissage	47
I.11 CONCLUSION	48
CHAPITRE II: MODELISATION DU QUADROTOR ET DE LA CAMERA	50
II.1 INTRODUCTION	50
II.2 MODELISATION DYNAMIQUE DU QUADROTOR	50
II.3 MODELISATION DE LA CAMERA STEREO (BANC STEREOSCOPIQUE)	58
II.4 MODELISATION ET GEOMETRIE DES CAMERAS EMBARQUEES	59
II.4.1 Modèle géométrique de la caméra	60
II.4.2 Paramètres intrinsèques	60
II.4.3 Paramètres extrinsèques	62
II.4.4 Calibrage	64
II.5 RECONSTRUCTION 3D	66
II.5.1 Évaluation d'algorithme de triangulation	68
II.6 CONCLUSION	70
CHAPITRE III: ODOMETRIE VISUELLE	72
III.1 INTRODUCTION	72
III.2 HISTOIRE DE L'ODOMETRIE VISUELLE	74
III.2.1 Odométrie visuelle Stéréoscopique	74
III.2.2 Odométrie visuelle monoculaire	75
III.2.3 V-SLAM	76
III.2.4 Odométrie Visuelle contre V-SLAM	76
III.3 FORMULATION GENERALE DE L'OV	76
III.4 EXTRACTION DE POINTS D'INTERET	78
III.4.1 Détecteurs de points d'intérêt	78
III.4.1.1 Détecteur d'Harris	79
III.4.1.2 SURF (Speeded Up Robust Features)	79
III.5 MISE EN CORRESPONDANCE (MATCHING)	81
III.5.1 Mise en correspondance par SURF	82
III.5.2 Suivi de points d'intérêt (tracking)	83

III.5.2.1 Tracking par SURF.....	84
III.6 ESTIMATION DU MOUVEMENT.....	86
III.6.1 Principe d'odométrie visuelle 3D/3D	88
III.6.2 OVS adaptative à base d'ICP (Iterative Closest Points).....	90
III.6.2.1 Algorithme ICP	90
III.6.2.2 Association des points	91
III.6.2.3 Estimation des paramètres de la transformation.....	93
III.6.2.4 Validation de l'algorithme :.....	94
III.6.3 Algorithme SVD (Singular Value Decomposition).....	96
III.6.3.1 Généralité	96
III.6.3.2 Estimation de la transformation homogène Par l'algorithme SVD	97
III.6.4 Algorithme de Quaternion	99
III.6.4.1 Généralités sur les Quaternions :.....	100
III.6.4.2 Diverses représentations des rotations.....	101
a) Représentation classique par trois angles	101
b) Représentation par Quaternion	101
III.6.4.3 Estimation de la transformation homogène	102
III.7 ÉLIMINATION DES FAUSSES ASSOCIATIONS	105
III.7.1 RANSAC	106
III.7.2 Filtrage.....	107
III.8 CONCLUSION	107
CHAPITRE IV: COMMANDE BACKSTEPPING ADAPTATIVE BASEE VISION DU QUADRIROTOR.....	110
IV.1 INTRODUCTION.....	110
IV.2 COMMANDE PAR BACKSTEPPING	111
IV.3 COMMANDE PAR BACKSTEPPING OPTIMALE POUR LE SUIVI DE TRAJECTOIRE.....	113
IV.3.1 Essaim de particules (Particle Swarm Optimization, PSO).....	114
IV.3.2 Réglage des paramètres du backstepping par PSO.....	115
IV.4 LOCALISATION DU QUADRIROTOR PAR FUSION DE DONNEES IMU/SVO.....	116
IV.5 FILTRE DE KALMAN ETENDU	117
IV.6 LOCALISATION ADAPTATIVE FLOUE DU QUADROTOR.....	119
IV.7 CONCLUSION	121
CHAPITRE V: RESULTATS ET DISCUSSIONS	123
V.1 INTRODUCTION.....	123
V.1 VALIDATION DU SYSTEME DE LOCALISATION VISUELLE (SVO).....	123
V.1.1 Mise en œuvre de la plate-forme expérimentale	124
V.1.2 SVO pour environnement intérieur (indoor)	124

V.1.2.1 Localisation du robot par l’Odomètre Visuel Stéréoscopique (trajectoire rectiligne).....	125
V.1.2.2 Trajectoires quelconques	126
V.1.3 Environnement extérieur (outdoor)	127
V.2 SUIVI DE TRAJECTOIRE PAR UNE COMMANDE BACKSTEPPING	128
V.3 COMMANDE BACKSTEPPING BASEE VISION	130
V.3.1 Scénario 1 : navigation d’un Quadrotor sans incertitude.....	131
V.3.2 Scénario 2 : Effet de l’exactitude des points 3D sur la précision de l’OVS	133
V.3.3 Effet de nombre de points d’intérêts sur la précision de l’OVS	136
V.4 COMPARAISON DES COMMANDES A LOGIQUE FLOUE ET BACKSTEPPING.....	140
V.5 CONCLUSION	141
CONCLUSION GENERALE.....	143
BIBLIOGRAPHIE	146

LISTE DES FIGURES

Figure I-1 : (a) Kettering Bug, une torpille aérienne sans pilote expérimentale. (b) avion automatique sans pilote de Hewitt-Sperry 1917.....	22
Figure I-2 : (a) Gyroplane No. 1 de Bréguet Richet (b) Quadrotor conçu par George De Bothezat, 21 février 1923 Rumerman (2002).....	22
Figure I-3 : Oehmichen No.2.	22
Figure I-4 : (a) Convert wings Model A (b) Curtiss-Wright VZ-7.	23
Figure I-5 : Divers configuration VTOL.....	23
Figure I-6 : Logiciel de configuration des cartes d'autopilote « ArduPilot Project».....	24
Figure I-7 : Comportements envisagés d'un UAV.	25
Figure I-8 : Concept général de guidage, contrôle et commande GNC[1].	25
Figure I-9 : Exemple de mini-drones à voilure tournante.	27
Figure I-10 : Composants matériels d'une plateforme de recherche de mini-drones.	30
Figure I-11 : Exemples de Gyroscope(a) et gyromètre(b).	32
Figure I-12 : Centrale inertielle : (a) L'AHRS 3DM-GX3-25 de MicroStrain, (b) MTi 10-series de Xsens.	33
Figure I-13 : Un GPS 6M avec un contrôleur d vol APM 2.8 pour drone	33
Figure I-14 : Télémètre laser : (a) Cartographie (b) détection d'objets (c) SF02/F.....	34
Figure I-15 : Technologie LIDAR OnyxScan embarquée sur drone - AltiGator Drone & UAV Technologies.....	35
Figure I-16 : Caméra <i>stéréoscopique Bumblebee 2 à gauche et Caméra ZED montée sur drone à droite</i>	36
Figure I-17 : Drone DJI Inspire Black XT avec sa caméra de suivi de type Electro-Optical Infrared (EO/IR) FLIR 336 30Hz.....	36
Figure I-18 : Caméra Intel RealSense Tracking Camera T265.....	37
Figure I-19 : applications des drones (a) Matrice 300 RTK et ses caméras hybrides multicapteurs Zenmuse H20 (b) livraison Amazon.....	38
Figure I-20 : Schéma de suivi de trajectoire[38].....	39
Figure I-21 : Principe de la localisation relative.	41
Figure I-22 : Principe de la localisation absolue.	41
Figure I-23 : (a) Carte métrique, (b) Carte topologique.....	42
Figure I-24 : Cartes d'occupation[42].....	43
Figure I-25 : Cartes de points d'intérêt.	44
Figure I-26 : Comparaison des méthodes d'estimation itératives.	47
Figure II-1 : Configuration du quadrotor.	52
Figure II-2 : Mouvement du Quadrotor.	53
Figure II-3 : (a) Parrot Bebop 2.0 et (b) Ardrone 2.	53
Figure II-4 : Principe de calibrage stéréoscopique.	60

Figure II-5 : Projection perspective.	61
Figure II-6 : Banc stéréoscopique et la mire de calibrage.....	63
Figure II.7 : Mire de calibrage.....	65
Figure II.8 : Simulation de position de la mire par rapport banc stéréoscopique.....	65
Figure II-9 : Principe de la triangulation.....	67
Figure II-10 : Erreurs de triangulation suivant Z	69
Figure II-11 : Erreurs de triangulation suivant X	70
Figure III-1 . Principe de l'algorithme d'odométrie visuelle.	73
Figure III-2 . Organigramme de l'algorithme SVO.....	73
Figure III-3 : (a) Géométrie de l'initialisation d'une caméras [85], (b) une plateforme expérimentale[86].....	75
Figure III-4 . Principe de l'algorithme SVO.....	77
Figure III-5 : Points d'intérêt : (a) coins, (b) jonctions en T et (c) forte variation d'intensité.	78
Figure III-6 : Résultats de détection des points d'intérêt.....	80
Figure III-7 : Résultats des descripteurs des points d'intérêt.	80
Figure III-8 : Images rectifiées acquises.....	81
Figure III-9 : détection des points d'intérêt.	81
Figure III-10 : Mise en correspondance gauche/droite à l'instant k et $k + 1$ des points d'intérêt.....	83
Figure III-11 : Principe du suivi de points.	84
Figure III-12 : Mise en correspondance droite/droite, gauche/gauche de l'instant k à $k + 1$ des points d'intérêt.	84
Figure III-13 : Pseudo algorithme d'association des points d'intérêt par SURF[106].	85
Figure III-14 : Suivi des points d'intérêt par la méthode du SURF.....	85
Figure III-15 : mise en correspondance des points d'intérêt horizontal et vertical.	86
Figure III-16 : Recalage de deux nuages de points.....	91
Figure III-17 : Le code d'association par la distance euclidienne.	93
Figure III-18 : organigramme de l'algorithme d'ICP.	94
Figure III-19 : Résultat de simulation de l'algorithme ICP.	95
Figure III-20 : Évolution de l'erreur de simulation.	96
Figure III-21 : Pseudo code de l'algorithme d'optimisation par SVD.	99
Figure III-22 : Les angles d'Euler	101
Figure III-23 : Pseudo code de l'algorithme d'optimisation par Quaternion.	104
Figure III-24 : Les fausses associations par la méthode de plus proche voisin.	105
Figure III-25 : Les fausses Associations par le descripteur SURF.	106
Figure III-26 :Pseudo-code de l'algorithme RANSAC.	107
Figure IV-1 .: Structure de la commande quadrirotor pour le suivi de trajectoire.....	113
Figure IV-2 .: Schéma vectoriel du déplacement d'une particule.....	115

Figure IV-3 : Réglage des paramètres du Backstepping.	115
Figure IV-4 : Fitness function.	116
Figure IV-5 : Fusion des données IMU/OVS.	117
Figure IV-6 : Système de commande adaptative flou.	120
Figure IV-7 : Variables d'entrée et tracés des fonctions d'appartenance.	120
Figure V-1 : Constitution de notre plate-forme.	124
Figure V-2 : Environnement interne[16].	125
Figure V-3 : Résultats de la localisation par différentes méthodes pour la trajectoire rectiligne.	126
Figure V-4 . Carte construite des deux salles de laboratoire.	127
Figure V-5 : Trajectoire sous forme « infini » à l'aide de SVO.	127
Figure V-6 : Matching des points d'intérêt dans l'environnement outdoor.	128
Figure V-7 : Résultat de localisation avec ICP.	128
Figure V-8 : Différentes formes de trajectoires réalisées par la commande backstepping.	129
Figure V-9 : Erreur de suivi de trajectoire : (a) trajectoire circulaire (b) trajectoire rectangulaire.	129
Figure V-10 : Points 3D de référence aux instants $i=1$ et $i=10$	130
Figure V-11 : Suivi de trajectoire par odometre visual idéal.	133
Figure V-12 . Evolution de l'erreur quadratique moyenne MSE selon la variation de l'amplitude du bruit.	134
Figure V-13 . Effet de bruit (bruit = 0,1 (10%), nombre de points d'intérêts= 50 et nombre d'itérations ICP = 10).	136
Figure V-14 . MSE en fonction du nombre des points d'intérêts (bruit = 0,2).	137
Figure V-15 : Effet de points d'intérêts (nombre de points d'intérêts = 10, bruit = 0,1 et nombre d'itérations ICP = 10).	139
Figure V-16 : Trajectoire du quadrirotor par le backstepping vs la commande FLC.	140

LISTE DES TABLEAUX

Tableau I.1. Critères de classification des drones.	26
Tableau II.1. Paramètres du quadrotor.....	58
Tableau II.2 : Résultats de calibrage stéréoscopique par Matlab.....	66
Tableau II.3 : Erreurs de triangulation suivant l'axe Z	69
Tableau II.4 : Erreurs de triangulation suivant X	69
Tableau III.1: Résultats de mise en correspondance.	83
Tableau III.2: Tableau d'évolution des erreurs de simulation de l'ICP.	95
Tableau IV.1. Règles floues pour la fusion de données adaptative.	121
Tableau V.1 : Erreurs des différentes méthodes pour la trajectoire rectiligne.....	125
Tableau V.2. Effet des bruits sur la précision de l'OVS et la précision de la poursuite de la trajectoire.....	133
Tableau V.3. Nombre de points d'intérêts effectifs (bruit = 0,2 ; nombre d'itérations ICP = 10).....	137
Tableau V.4: Comparaison des contrôleurs Backstepping vs FLC.....	140

LISTE DES ABRÉVIATIONS

Abrév.	Désignation	Abrév.	Désignation
UAV	Unmanned aerial vehicle	UAS	Unmanned aircraft systems
RPV	remotely piloted vehicles	RPAS	Remotely Piloted Aircraft System
CCD	Charge Coupled Device	Kd-tree	K-dimensional tree
DDL	Degrés de Liberté	KF	Kalman Filter
DoH	Déterminant of Hessienne	KLT	Kanade, Lucas, Tomasi
EKF	Extended Kalman Filter	LoH	Difference of Hessienne
GPS	Global Position System	DoG	Difference of Gauss
INS	Inertial Navigation System	SLAM	Simultaneous Localization And Mapping
IMU	Inertial Measurement Unit	VSLAM	Visual Simultaneous Localization And Mapping
AHRS	Attitude and Heading Reference System	PTAM	Parallel Tracking and Mapping
ICP	Iterative Closest Point	SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition	SIFT	Scale Invariant Feature Transform
IDP	Inverse Depth Paramétrisation	BRIEF:	Binary Robust Independent Elementary Features
OVS ICP, SVD et Quaternion	Odomètre Visuel Stéréoscopique par ICP, SVD et Quaternions	ORB	Oriented Rotated BRIEF
OV	Odomètre Visuel	RANSAC	RANdom SAMple Consensus
SDK	Kit de développement logiciel	SSD	Sum of Squared Differences
SfM	Structure from Motion	ZNCC	Zero Normalized Cross Correlation
MALE	Moyenne Altitude Longue Endurance	SVSF	<i>Smooth Variable Structure Filter,</i>
HALE	Haute Altitude Longue Endurance	PSO	Optimisation de l'essaim de particules
UCAV	Unmanned Combat Air Vehicles	FAKF	Fuzzy Adaptive Kalman Filter
TUAV	Tactical Unmanned Aerial Vehicle	LiPo	Lithium Polymère
MAV	Mini Air Vehicule	VIO	Odométrie visuelle-inertielle
ROS	Robot operating system	LIDAR	LIght Detection And Ranging).
EO/IR	Electro-Optical Infrared	MSE	Erreur quadratique moyenne
RANSAC (<i>Random Sample Consensus</i>	FLC	Commande à logique floue
RMSE	<i>Rooth Mean Square Error</i>	FPGA	Field-Programmable Gate Array

Introduction Générale

INTRODUCTION GENERALE

Aujourd'hui, les drones constituent un potentiel technologique et un domaine émergent dans différents secteurs, où l'on pourrait même parler d'une prochaine révolution de l'intelligence artificielle, qui verrait les drones déferler dans notre vie quotidien. Les drones, qu'ils soient autonomes ou télécommandés, sont conçus pour effectuer des tâches de navigation qui présenteraient un risque élevé pour les pilotes humains, afin d'exécuter ou de fournir une extension des capacités humaines pour des missions spécifiques. Les applications des drones se généralisent, de l'usage militaire à des applications civiles, telles que les géolocalisations (état de la végétation, pollution, etc.), la sécurité, le sauvetage mais aussi la collecte d'informations, l'exploration, l'inspection, la surveillance de vastes zones, etc. Récemment, ils sont déployés dans les rues sous la surveillance des autorités en utilisant des caméras de drone pour assurer le respect du confinement et faire face à une épidémie de coronavirus rampante.

En général, ces véhicules aériens se distinguent par leur capacité à voler à différentes vitesses, à stabiliser leurs positions, à survoler une cible et à effectuer des manœuvres à proximité d'obstacles et à effectuer des vols intérieurs ou extérieurs [1]. Ils ont divers noms techniques : *unmanned aerial vehicles* UAV, véhicules télépilotes *remotely piloted vehicles* (RPV), systèmes d'aéronefs sans pilote, *unmanned aircraft systems* (UAS) et systèmes d'aéronefs télépilotes, *Remotely Piloted Aircraft System* (RPAS). Dans cette étude, le quadrirotor est préféré pour sa robustesse et sa modularité du fuselage, étant moins sujet aux dommages et plus facile à réparer.

Ces systèmes doivent avoir une certaine autonomie pour accomplir trois tâches principales dans le domaine de la robotique: le guidage, la commande et la navigation [2, 3]. Ce dernier comprend la perception, la cartographie et la localisation du robot [4, 5], poursuite de cible [6], évitement d'obstacles [7] et odométrie visuelle-inertielle (VIO) [8].

Dans notre cas, nous nous intéresserons aux drones à faible coût, à petite échelle et légers qui peuvent être utilisés de manière fiable à l'intérieur comme à l'extérieur. Les critères qui correspondent à ces catégories se trouvent dans les plateformes Parrot AR.Drone et Bebop [7], les plateformes Arducopter [8] ainsi que du DJI Quadrotor [9] et d'autres.

Ces drones sont équipés de capteurs extéroceptifs et proprioceptifs afin d'estimer leur position et leur orientation dans l'espace. Les principaux capteurs utilisés étaient le système de positionnement global (GPS) pour la position et le système de navigation inertielle (INS) pour

l'estimation de l'attitude. Ces capteurs, cependant, ont des limitations qui affectent les performances du système. Par conséquent, des processus d'estimation développés sont essentiels pour assurer la stabilité du système. Ces dernières années, l'évolution rapide des systèmes embarqués et leur miniaturisation ont amené des modules de caméras puissants et peu coûteux et des unités de mesure inertielle (IMU) qui pourraient être montés sur des drones pour extraire des informations utiles à bord et remonter les données nécessaires de navigation [10].

Dans la navigation UAV, deux niveaux de commande peuvent être distingués, premièrement, le haut niveau est structuré avec plusieurs techniques telles que: la détection et la description des caractéristiques, le flux optique pour l'estimation du mouvement, l'asservissement visuel basé sur l'image et les approches de cartographie [11], d'autre part, le bas niveau concerne la stabilisation de l'UAV, la commande d'attitude, le guidage et le suivi de trajectoire [12, 13].

Plusieurs méthodes de commande ont été présentées dans la littérature pour résoudre le problème de la stabilité et du suivi de trajectoire des quadrirotors. Les techniques de commande PID et LQ ont réussi à stabiliser l'attitude du quadrirotor autour de la position de vol stationnaire en présence de perturbations minimales. La technique de commande Backstepping a donné de bons résultats sur les systèmes non linéaires fortement couplés de commande et variant dans le temps tels que les UAV. L'application de cette technique a une certaine capacité découplage, et elle est largement utilisée dans les systèmes non linéaires en temps continu et en temps discret comme c'est le cas de [14, 15].

C'est dans ce contexte que s'inscrit notre objectif de recherche de développer un système de navigation UAV basé sur un contrôleur adaptatif par Backstepping flou utilisant l'odométrie visuelle. Tout d'abord, afin d'obtenir la localisation la plus précise et la plus rapide d'un quadrotor, un algorithme d'odométrie stéréo-visuelle adaptatif (SVO) est appliqué pour estimer les bonnes informations de pose d'un UAV en vue de maîtriser sa navigation. [16]. Nous avons utilisé le filtre de Kalman étendu (EKF) [17] et/ou le filtre à structure variable glissant (*Smooth Variable Structure Filter*, SVSF) [18, 19] pour la fusion de données SVO / IMU pour améliorer la localisation et fournir une pose précise de l'UAV. Deuxièmement, un contrôleur flou adaptatif est proposé pour fournir la pose d'un UAV appropriée en considérant le nombre et la gamme des caractéristiques 3D. Enfin, une stratégie de commande non linéaire utilisant le contrôleur Backstepping a été développée pour assurer une convergence rapide et un suivi précis de la trajectoire de référence souhaitée. Les paramètres du Backstepping sont réglés à l'aide de l'algorithme d'optimisation de l'essaim de particules (PSO) pour améliorer les performances de

la commande en boucle fermée.

Les principales contributions de cette thèse de doctorat sont présentées comme suit:

- L'implémentation d'un algorithme d'odométrie visuelle basé sur la stéréovision (SVO) utilisant les caractéristiques visuelles pour la navigation intérieure / extérieure à l'aide d'un algorithme adaptatif et itératif des points plus proche en utilisant des points d'intérêts *Adaptive Itérative Closest SURF* ;
- Une fusion SVO / IMU utilisant l'algorithme Fuzzy Adaptive Kalman Filter (FAKF) est proposée pour remédier la limitation de la localisation visuelle. Elle prend en compte le nombre et la profondeur moyenne des points 3D estimés. Cet algorithme ajuste de manière adaptative les paramètres EKF pour donner plus d'importance au capteur approprié (SVO ou IMU, ou utilisant les deux capteurs en même temps) pour fournir une estimation précise et robuste de la pose d'UAV ;
- La technique de suivi de trajectoire en utilisant un contrôleur Backstepping, dont les paramètres sont optimisés par la stratégie Particle Swarm Optimization (PSO) ;
- Les algorithmes proposés sont validés en simulation et à l'aide de données expérimentales (scénarios réels) et comparés à deux autres approches de localisation et de navigation.

ORGANISATION DE LA THÈSE

Pour bien mener ce travail, la thèse est organisée comme suit :

- Le ***premier chapitre*** est dédié aux généralités sur les drones, leurs catégories et l'ensemble des capteurs les plus utilisés pour la navigation, nous explorons leurs différents aspects de localisation et cartographie pour la navigation autonome. Finalement, nous avons évoqué l'exploitation de la vision pour la commande et la poursuite des trajectoires en utilisant la vision, en occurrence, l'*odométrie visuelle*.
- Le ***deuxième chapitre*** illustre en premier lieu le principe de la modélisation dynamique et la représentation des forces et des moments importants qui agissent sur le quadrotor ainsi que ses paramètres adaptés. Puis, nous avons jugé nécessaire et utile la modélisation du capteur de caméra stéréoscopique pour la perception tridimensionnelle, pour cela nous présentons le modèle géométrique d'une caméra stéréo, ainsi que son calibrage et enfin la technique de reconstruction 3D.
- Le ***troisième chapitre*** est consacré à l'état de l'art sur l'odométrie visuelle, monoculaire ou stéréoscopique, leurs principes généraux pour l'estimation de la pose. Nous proposons en plus

de la formulation du problème, les différentes approches d'estimation de mouvement, le filtrage de fausses valeurs, ainsi que les méthodes d'optimisation. Aussi, nous avons formulé les principes des algorithmes implémentés, à savoir, ICP (Itérative Closest Points), SVD (Singular Value Décomposition) et Quaternion pour améliorer performances d'optimisation. A la fin, nous avons présenté quelques algorithmes de filtrage de points pour enlever les mauvaises associations : RANSAC (RANdom SAmple Consensus) et filtre de Zhang.

- Le *quatrième chapitre* traite la technique adaptée de commande Backstepping. Par la suite un contrôleur flou adaptatif est proposé pour fournir la pose d'UAV appropriée en considérant le nombre et la gamme des caractéristiques 3D. Enfin, une stratégie de commande non linéaire utilisant le contrôleur Backstepping a été développé pour assurer une convergence rapide et un suivi précis de la trajectoire de référence souhaitée. Les paramètres de synthèse de commande par backstepping sont réglés à l'aide de l'algorithme d'optimisation de l'essaim de particules (PSO) pour obtenir des performances de contrôleur optimales. L'algorithme de fusion de données OVS/INS par l'EKF suivi par un contrôleur flou pour une sélection automatique de la pose adéquate a été présenté.

- Le *cinquième chapitre* présente, en premier lieu, la mise en œuvre de l'odométrie visuelle sur une plate-forme expérimentale (le robot mobile, l'équipement logiciel et matériel, et l'environnement). Deuxièmement les résultats de simulation, d'expérimentation ainsi qu'une analyse profonde de précision et de robustesse sont présentés pour plusieurs scénarios de localisation et de navigation et de commande backstepping.

La **conclusion générale** fait apparaître les grandes étapes de la thèse ainsi que les apports de en termes de méthodologies proposées et des résultats produits. De nombreuses perspectives sont données démontrant l'intérêt réel des solutions proposées. La bibliographie est très abondante, bien construite.

Chapitre I : Navigation visuelle des quadrirotors

CHAPITRE I: NAVIGATION VISUELLE DES QUADRIROTORS

I.1 Introduction

Nous allons présenter dans ce chapitre un bref aperçu sur le domaine de la robotique lié aux quadrirotors, l'évolution, le développement et les applications des véhicules aériens sans pilote, ainsi que leurs travaux remarquables proposés dans la littérature. Par la suite, notre étude se focalisera sur un type de mini-drone, à savoir, le quadrirotor, où nous présenterons quelques travaux de recherche des laboratoires qui travaillent sur ces plates-formes de recherche dans le domaine de la navigation visuelle. Cette étude nous permettra de situer notre travail par rapport aux travaux antérieurs. Nous allons présenter aussi quelques approches de stratégies de commande utilisées dans les quadrirotors.

I.2 Aperçu sur les véhicules aériens sans pilote

D'une manière générale, le terme de drones aériens regroupe l'ensemble des aéronefs sans pilotes. En anglais, le mot « drone » désigne le faux bourdon (mâle de l'abeille). Le nom a été baptisé dans les années 1930 au Royaume-Uni à la version automatisée d'avions-cibles à cause de leur vol lent et assourdissant ressemblant plus à celui du bourdon. Comme déjà cité, nous nous concentrons dans ce travail sur les drones miniaturisés à voilure tournante. Dans cette catégorie, on s'intéresse, particulièrement aux multi-rotors, en occurrence, les quadrotors. Ces derniers sont cités sous différents noms, à savoir, le quadrirotor, le quadcopter et le plus fréquemment utilisé est le terme en anglais quadrotor.

Leurs développements ont commencé au début des années 1900, leurs premières missions considéraient à faire la fourniture de cible à la formation militaire où la première tentative de drone motorisé fut comme cible aérienne pour l'entraînement "cible aérienne" de l'A. M. Low en 1916[20]. Low a confirmé que le monoplane de Geoffrey de Havilland était celui qui a volé qui a été contrôlé en 1917 à l'aide de son système radio[21]. D'autres développements sans pilote ont suivi pendant et après la Première Guerre mondiale qui sont entrées en service en 1935 en Grande Bretagne, menant à la flotte des cibles aériennes de Havilland 82 Queen Bee[21].

Une flotte de véhicules aériens sans pilote pour le combat a été décrite par Nikola Tesla en 1915. Ces développements ont également inspiré la construction du Kettering Bug par Charles Kettering et de l'avion automatique connu comme « bombe volante » de Hewitt-Sperry (figure 1, a-b), a fait son premier vol, démontrant le concept d'un aéronef sans pilote.

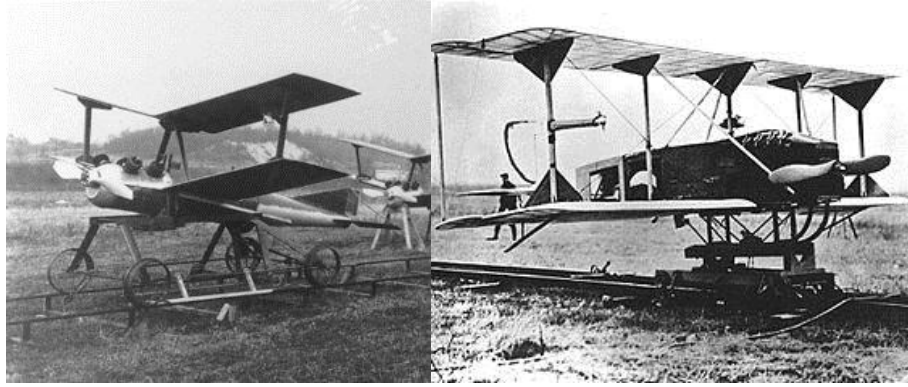


Figure I-1 : (a) Kettering Bug, une torpille aérienne sans pilote expérimentale. (b) avion automatique sans pilote de Hewitt-Sperry 1917.

Le Gyroplane No.1 est l'un des premiers quadrirotors avec pilote qui a décollé verticalement. C'est un hélicoptère à quatre rotors conçu par Louis Breguet (voir figure 2-a), et le premier vol a eu lieu en 1956. Il a également été le premier hélicoptère à quatre rotors. Par la suite dans les années 1920 en France, Étienne Oehmichen a conçu plusieurs Quadrotors dont le Oehmichen No.2 est probablement le premier hélicoptère fiable (voir figure 1.3a). L'américain George de Bothezat a construit le Convertwings Model A (voir la figure 1.2b), rotors qui a démontré un vol stationnaire [22].

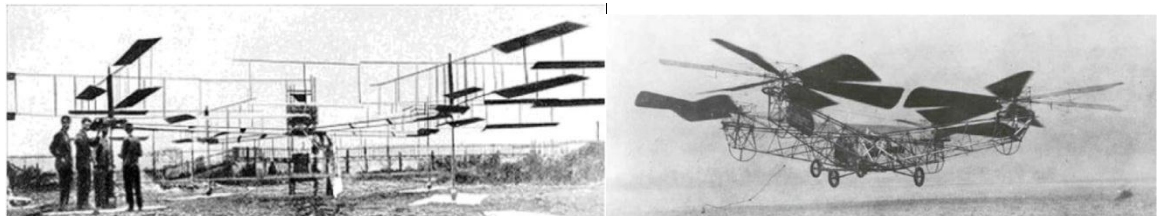


Figure I-2 : (a) Gyroplane No. 1 de Bréguet Richet (b) Quadrotor conçu par George De Bothezat, 21 février 1923 Rumerman (2002).

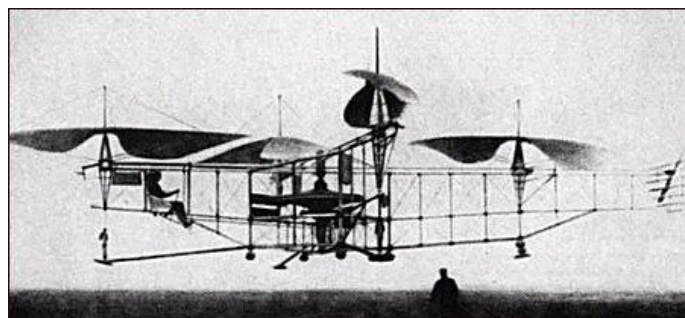


Figure I-3 : Oehmichen No.2.

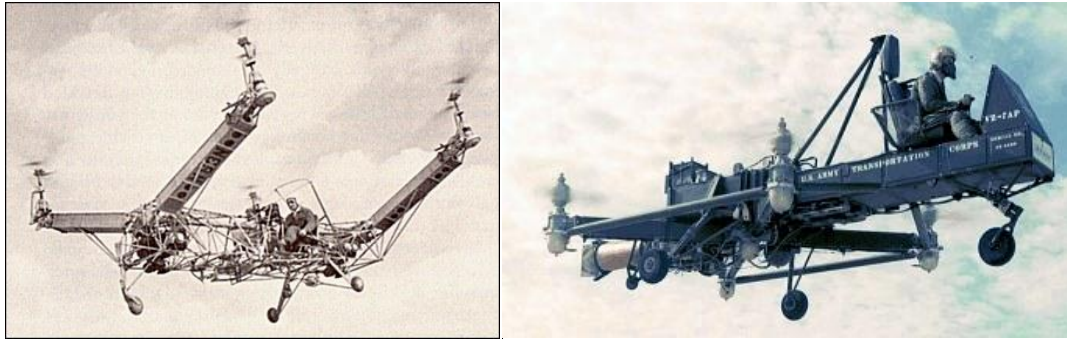


Figure I-4 : (a) Convert wings Model A (b) Curtiss-Wright VZ-7.

Le Curtiss-Wright VZ-7 a été conçu par la société Curtiss-Wright (voir figure I-2 (b)) pour l'armée américaine en 1958 (voir la figure 1.4a). Il a été contrôlé en modifiant la poussée des quatre hélices.

Au cours des dernières décennies, divers drones miniatures à voilure tournante ont été conçus par des chercheurs scientifiques pour des applications civiles. Il existe de nombreux drones commercialisés avec succès, tels que Draganflyer, Parrot de type AR.Drone 2 et Bebop 2, DJI par ses versions Phantom et Mavic, Gaii 330X, etc.

Leurs impacts sur la société civile ont donné beaucoup d'avantage de facilité et de bénéfice pour les conditions de vie humaine. Le sujet a attiré l'attention des industries, les centres de R&D et universités du monde entier pour faire face à de nouveaux défis, tels que détection et évitement d'obstacles, commande, atterrissage par vision, techniques de localisation, fusion de données, etc. Une étude récente a donné les différentes plateformes expérimentales de drones[23]. En effet, plusieurs gammes de prototype ont réalisés grâce aux progrès technologiques des moteurs électriques, batteries et systèmes électroniques embarqués et la communication sans fil.



OS4 de l'EPFL[24]



Parrot ARdrone 2



Parrot Bebop Drone 2



Gaii 330X-S



Quadrotor basculant - McART3- (Tilt-Wing)[25]



Prototype de Tail-sitter [3]

Figure I-5 : Divers configuration VTOL.

I.3 Concept navigation

La navigation désigne l'ensemble des techniques mises en œuvre pour récupérer le mouvement d'un véhicule (plus particulièrement sa trajectoire). Les premières méthodes de localisation ont été menées à partir d'observations visuelles fondées sur des points de repère dont la position était connue. Les corps célestes, tôt utilisés comme « balises » de positionnement, présentent malgré leur éloignement l'intérêt particulier en tant que référence fixe par rapport à un repère terrestre. Ces approches empiriques ont posé les jalons de la navigation.

Aujourd'hui, avec l'avènement de technologie, les systèmes de positionnement les plus utilisés pour la navigation sont les systèmes navigation par satellite, où nous pouvons distingués principalement les systèmes suivants : GPS, GLONASS, BeiDou et GALILEO. Aussi, d'autres systèmes qui sont basés sur les capteurs, tels que les systèmes de navigation à l'estime (navigation inertielle) et/ou combinés, par des réseaux radios et de télécommunications (RTK : Real Time Kinematic, PKK ; Post Processed Kinematic).

Tout d'abord, les drones en général sont conçus à base des microcontrôleurs et microprocesseurs implantés sur une carte électronique. A ce titre, Plusieurs des logiciels de control au sol ont été développés pour effectuer des missions par les drones. Aussi, ils permettent la navigation et planification des missions. Actuellement, plusieurs firmes proposent des solutions de control au sol des drones, à titre d'exemple voir fig.I-6.



Figure I-6 : Logiciel de configuration des cartes d'autopilote « ArduPilot Project».

D'une manière générale le quadrotor autonome est conçu à remplir une tâche sans l'intervention de l'opérateur, il a des comportements ou des fonctionnalités de naviguer autonome afin s'interagir avec son environnement. Parmi ses fonctionnalités on peut distinguer la perception, la locomotion, l'autonomie décisionnelle et l'interaction intelligente avec l'environnement dans lequel il effectue des tâches (figure I.7). L'architecture de guidage,

contrôle et navigation est montrée dans la figure I.8.

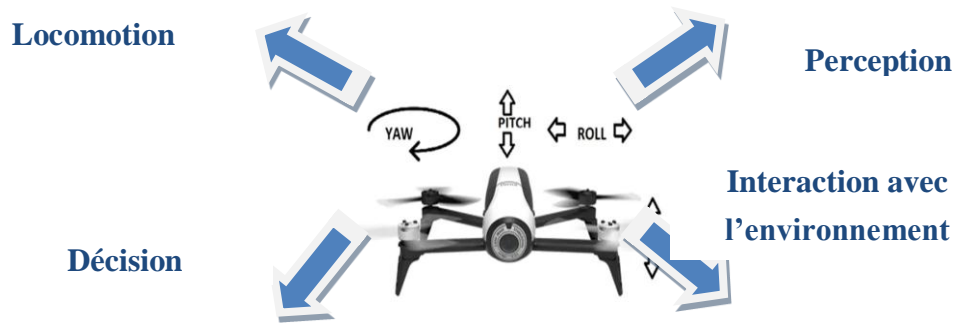


Figure I-7 : Comportements envisagés d'un UAV.

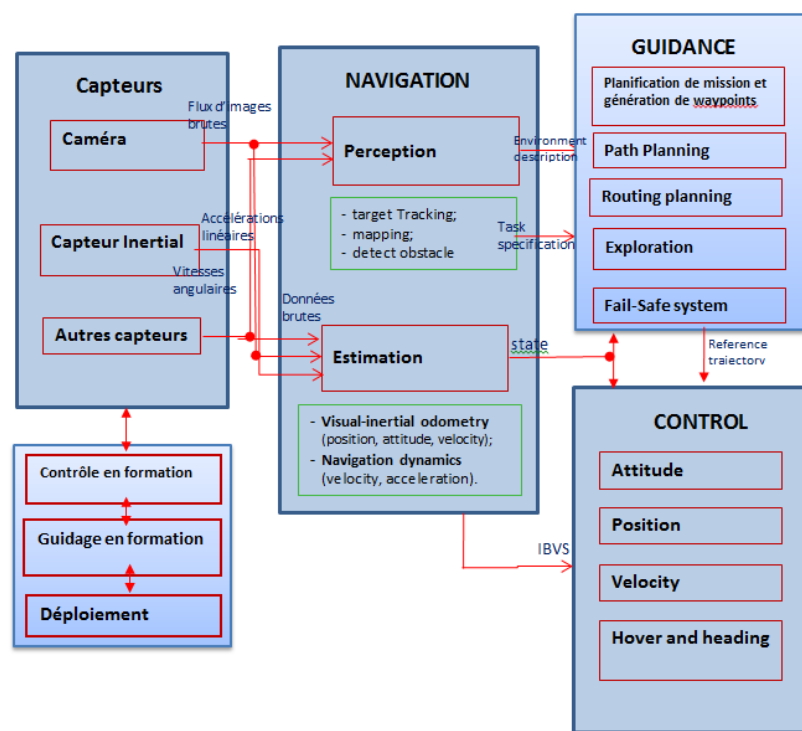


Figure I-8 : Concept général de guidage, contrôle et commande GNC[1].

Actuellement des firmes fournissent des logiciels comme station de control au sol pour la planification des missions, avec des radiocommandes intelligentes, à savoir, les logiciels, Mission Planner (voir la figure I-6), APM Planner 2.0, MAVProxy, QGroundControl ... etc.

I.4 Classification des drones

La classification des drones aériens ou aéronefs sans pilote peut être reparté selon différentes catégories et plusieurs critères :

Critères	Catégories
voilure	Fixe, tournante, hybride ou battante
Poids	Nano (<250 g), micro (250 à 2kg), mini (2-25 kg), moyen (25-150 kg), et grand (>150 kg).
Altitude/endurance	<ul style="list-style-type: none"> - Porté à main (600m) et 2km de portée, - de proximité (<1500m) et inférieur de 10km de portée. - drones tactiques (5.5km et 160km de portée) ; - MALE (5.5km et plus 200km de portée), - HALE (9100m et portée indéfinie), - UCAV (<i>Unmanned Combat Air Vehicles</i>).
Taille	Longueur, envergure, etc
Capacité à emporter des charges	drones de surveillance / drones armés / drones de combats
Caractère de poids	Classification sur cinq groupes pour les systèmes de véhicules sans pilote.
Hors catégories	Supersonique (Mach 1 à 5), hypersonique (Mach ¹ 5+) (15,200 m), suborbital altitude ou portée supérieur à 200 km, ainsi que la basse orbitale

Tableau I.1. Critères de classification des drones.

D'une manière générale, leur classification peut être résumée selon les catégories suivantes :

- Drones volant à haute altitude et de longue endurance (HALE : Haute Altitude Longue Endurance) ;
- Drones volant à moyenne altitude et de longue endurance (MALE : Moyenne Altitude Longue Endurance) permettant d'utiliser une charge de mission pouvant atteindre 500 kg ;
- Drones de combat UCAV (*Unmanned Combat Air Vehicles*)
- Drones tactiques TUAV (*Tactical Unmanned Aerial Vehicle*)
- Nano, Micro et Mini drones MAV (*Mini Air Vehicule*).

La figure suivante montre quelques exemples des plateformes ciblées qui peuvent être inclus dans notre contexte de thèse :

¹ Le nombre de Mach est un nombre sans dimension, noté Ma, qui exprime le rapport de vitesse locale d'un fluide à la vitesse du son dans ce même fluide.



DJI MAVIC 2 PRO



WL V911 4 CH



DYS D800 X8



DJI INSPIRE 2



Infotron IT180



HexaKopter



Drone Skydio 2



Black Hornet UK



DJI Ryze Tello



DJI Phantom 3



Parrot Anafi USA



Vapor 55

Figure I-9 : Exemple de mini-drones à voilure tournante.

I.5 Matériels des quadrirotors

Il est utile de mentionné dans la thèse, pour notre cas d'étude, le concept de la configuration générale des quadrirotors, qui sert pour éventuel implémentations. Cette configuration est constituée par des éléments suivants :

- **le châssis** (sur les quadrirotors, 4 bras en X, H ou reconfigurable) ;
- **les moteurs**, quatre moteurs brushless, se caractérisent par le diamètre de leur cage tournante et par le nombre de tours/volt ou KV. Un moteur ayant un KV de 1000 tr/V

fonctionnera à 12000 tours/min s'il est alimenté en 12V;

- **les contrôleurs**, les moteurs sont commandés par des interfaces de puissance ESC ou Electronique Speed Controller en anglais ;
- **les hélices**, sont souvent en plastique ou matériaux composites, identifiées par leur diamètre et leur pas indiqués en pouces, exemple : 9x4,5)
- **les batteries** sont utilisées essentiellement en « Lithium Polymère ». Elles sont issues d'une technologie qui permet d'avoir un très bon rapport poids/puissance. Un élément LiPo (1S) fournit une tension de 3,7V. Sur un drone, on utilise en général des batteries à 3 ou 4 éléments (3S ou 4S ...etc.).
- **la radiocommande**, pour piloter le drone, il faut un émetteur radio pour le pilote et un récepteur dans le drone. Il existe plusieurs technologies pour les radiocommandes, les radios FM en 41MHz (de moins en moins utilisées) et les radios en 2,4GHz ou 5GHz, souvent programmables. Une radiocommande doit idéalement comporter 6 voies pour piloter un drone, 4 voies sur les manettes et 2 voies sur des interrupteurs pour actionner certaines fonctions.
- **le contrôleur de vol**, c'est une carte électronique (exemple : carte mère de type ARM9 montée sur quadrirotor AR-drone, ArduPilot, DJI A3, PX4), microcontrôleur (STM32 de la famille basé sur le processeur ARM, tel que NAZE32) ou nano-ordinateur (Raspberry PI avec Navio2), équipée de capteurs très précis, qui va traiter les consignes du pilote envoyées à l'émetteur ainsi que les informations envoyées par ses capteurs et va transmettre des impulsions électriques aux contrôleurs des moteurs pour faire varier leur vitesse.

Les contrôleurs de vol sont équipés de gyroscopes et d'accéléromètres pour mesurer et compenser les déplacements (détaillés ci-après). Certaines contrôleurs de vol évolués sont équipés de GPS-GLONASS et d'altimètre afin de maintenir un point fixe ou une altitude ou même de retourner au point de départ du drone en cas de perte de signal radio (RTH : *return to home*).

- **la caméra**, (détaillée dans la section suivante), elle permet soit de d'enregistrer le vol pour le visionner plus tard, soit de faire du FPV (vol en immersion : *First Person View*) ou le mode follow me.

Dans ce chapitre nous allons nous intéresser aux différents outils (capteurs) qui servent à exploiter l'information de l'environnement, ainsi que les différentes techniques de perception de l'environnement (cartographie) et de localisation.

I.6 Logiciels (Software) des mini-drones

Outre les logiciels fournis par les firmes des contrôleurs de vol suscités, le système d'exploitation pour robots, ROS « robot operating system » est un ensemble d'outils informatique sous formes de logiciels libre-open source, permettant de développer des logiciels pour la robotique. A l'origine, né en 2000, à l'université de Sandford, puis porté à maturité en 2007 par la société américaine Willow Garage. Aujourd'hui, il est mené par l'open robotics (sur www.openrobotics.org, consulté le 16 avril 2022), anciennement Open Robotics Fountation ou OSRF. ROS est officiellement supporté par 75 robots.

ROS est un méta système d'exploitation, quelque chose entre le système d'exploitation le middleware. Il fournit des services proches d'un système d'exploitation, à savoir, l'abstraction du matériel, la gestion de la concurrence et la gestion des processus. Il fournit également des appels synchrones et asynchrones, des basses de donnes et système de paramétrage du robot.

L'intérêt dudit système d'exploitation pour la robotique est de consolider les efforts des chercheurs et de minimiser le temps passait en conception des logiciels embarqués associés. L'idée principale d'un OS robotique est d'éviter de réinventer la roue à chaque fois et de proposer des fonctionnalités standardisées faisant l'abstraction du matériel. ROS se positionne comme un facilitateur de projets robotique.

L'écosystème ROS comporte aujourd'hui des dizaines de milliers d'utilisateurs dans le monde et est soutenu par des acteurs industriels majeurs.

I.7 Capteurs des mini-drones

La figure (I-10) montre l'intégration des divers capteurs ainsi que l'instrumentation dans les quadrotors La localisation d'un véhicule aérien est basée sur des informations issues d'une diversité de capteurs embarqués, le nombre important des capteurs disponibles dans les quadrirotors offre une multitude de méthodes de localisation selon leur type. Ces méthodes de localisation peuvent être classées en trois types, à savoir, la localisation absolue, la localisation par estimation et la localisation dite hybride. Cette classification est basée essentiellement sur les sources de l'information et la nature des capteurs utilisés, qui se divisent en deux catégories :

Note:

- GNSS: système de navigation;
- Odroid : carte embaqué nano-ordinateur Samsung ;
- Carte de contrôleur : Pixhawk.

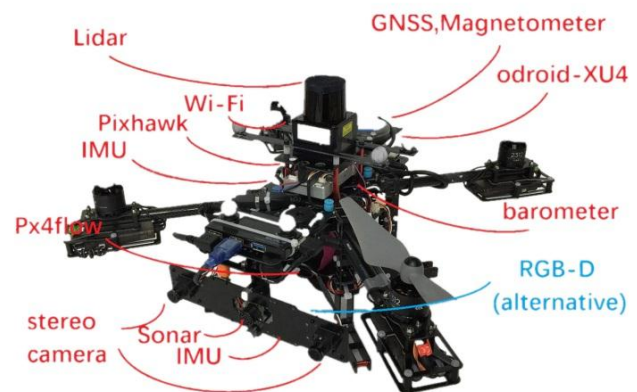


Figure I-10 : Composants matériels d'une plateforme de recherche de mini-drones.

a. Capteurs proprioceptifs : ils fournissent des informations sur l'état interne du robot volant (baromètre, centrales inertielles...).

Capteurs extéroceptifs : ils fournissent des informations relatives à l'environnement du robot volant (GPS, télémètres, Caméras...).

I.7.1 Capteurs proprioceptifs

Les Capteurs *proprioceptifs* fournissent des informations propres aux capteurs de drone, c'est-à-dire sur son état à un instant donné. Les informations sensorielles gérées dans ce cadre sont généralement des vitesses, des accélérations, des angles de rotation, des angles d'attitude. Cependant, ils ne peuvent pas fournir les informations lors de l'arrêt du quadrirotor. On peut regrouper les capteurs proprioceptifs en deux familles [26].

- Les capteurs de déplacement qui comprennent les accéléromètres et les radars Doppler, etc. Cette catégorie permet de mesurer des déplacements élémentaires, des variations de vitesse ou d'accélération sur des trajectoires rectilignes ou curvilignes.
- Les capteurs d'attitude, qui mesurent deux types de données : les angles de cap (direction de déplacement), et les angles d'Euler (lacet, roulis et tangage). Ils reposent principalement sur les mesures inertielles.

I.7.1.1 Capteurs de déplacement

a) Accéléromètres

L'accéléromètre est un capteur qui mesure l'accélération linéaire en un point donné. En pratique, la mesure de l'accélération est effectuée à l'aide d'une masse d'épreuve, de masse m , reliée à un boîtier du capteur [27]. Le principe de ce capteur est de mesurer l'effort massique

non gravitationnel qu'on doit appliquer à *la masse* pour le maintenir en place dans le boîtier lorsqu'une accélération est appliquée au boîtier. Le calcul du déplacement élémentaire du robot est obtenu par double intégration de ces informations. Cette double intégration conduit généralement à des accumulations importantes d'erreurs. Ce capteur est plus coûteux que les odomètres.

b) Radars Doppler

Au lieu d'estimer le déplacement par des mesures sur les roues, il est possible d'utiliser un radar pointé vers le sol qui permet de mesurer la vitesse du véhicule par effet Doppler.

I.7.1.2 Capteurs d'attitude

Les capteurs d'attitude mesurent deux types de données : les angles de cap et les angles de roulis et de tangage. Ils sont principalement constitués par les gyroscopes, les gyromètres, les gyrocompas, les capteurs inertiels composites, les inclinomètres et les magnétomètres. Ces capteurs sont en majorité de type inertiel. Ces capteurs ont pour point commun d'être généralement coûteux et sensibles au bruit, d'où une intégration moins fréquente dans les systèmes embarqués que les odomètres.

a) Gyroscopie, Gyromètre et Gyrocompas

Les gyroscopes permettent la mesure de l'orientation du robot mobile (figure I.4.a). Ces mesures sont intégrées pour calculer sa position. Ces capteurs sont particulièrement répandus en positionnement de robots aériens, en particulier dans le cas de fusion de mesures.

Le gyromètre (figure I.4.b) est un capteur qui permet de mesurer la vitesse angulaire du robot. Il existe plusieurs types de gyromètres dont les gyromètres laser, les gyromètres optiques, et les gyromètres à structures vibrantes, capacitifs, etc.

Le gyrocompas est un capteur qui permet de mesurer le cap. Il est composé d'un gyroscope et d'un compas magnétique. Le gyrocompas conserve le nord magnétique durant tout le déplacement du véhicule, après l'avoir initialement déterminé de façon autonome.

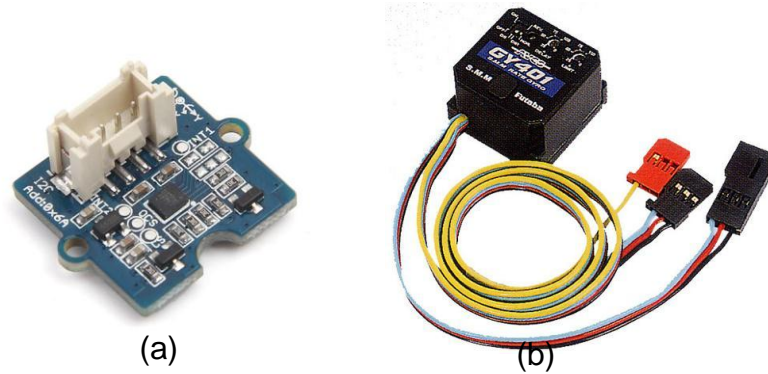


Figure I-11 : Exemples de Gyroscope(a) et gyromètre(b).

(a) Grove - 6-Axis Accelerometer & Gyroscope.

(b) Gyromètre Futaba.

b) Centrale inertielle

Le capteur inertielle (centrales inertielles figure I.12), ou IMU (de l'anglais Inertial Measurement Unit), est un dispositif muni de trois gyroscopes, de trois accéléromètres et d'un calculateur en temps réel [28]. Ce dispositif donne l'information sur la vitesse ou sur l'accélération, l'intégration de l'un des deux peut nous conduire au calcul de la distance. Son principe est que les informations issues des accéléromètres de chaque axe ainsi que les informations d'altitude issues des gyroscopes (tangage, roulis et lacet) sont traitées dans l'unité de mesure inertielle, les systèmes qui déterminent la pose de quadrirotor à partir des accéléromètres et des gyromètres sont appelés système de navigation inertielle.

Deux architectures de centrales inertielles existent dans la littérature : les centrales à plateforme stabilisée, ou INS (de l'anglais Inertial Navigation System), et les centrales à plateforme liée, ou IRS (de l'anglais Inertial Reference System).

Les capteurs inertielles sont utilisés pour former des systèmes munis d'un calculateur et d'une électronique de mise en œuvre capables de délivrer des informations utiles à la navigation et au pilotage. Parmi ces systèmes on cite : les INS (Inertial Navigation System), les INS/GPS (INS/Global Positioning System), les IRS (Inertial Reference System), les AHRS (Attitude and Heading Reference System). Les AHRS sont des combinaisons d'instruments capables de donner une estimation précise du roulis, tangage, lacet d'un véhicule en mouvement [29].

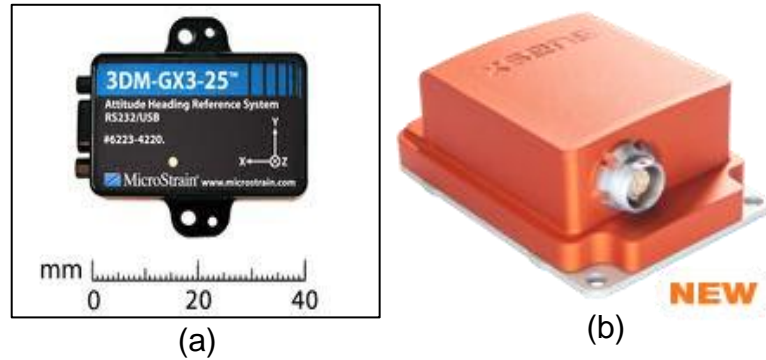


Figure I-12 : Centrale inertielle : (a) L'AHRS 3DM-GX3-25 de MicroStrain, (b) MTi 10-series de Xsens.

I.7.2 Capteurs extéroceptifs

Les capteurs extéroceptifs sont employés en robotique mobile pour collecter des informations sur l'environnement d'évolution du système mobile. Ils sont le complément indispensable aux capteurs proprioceptifs présentés précédemment. Des méthodes de fusion de données sont alors utilisées pour conditionner et traiter les informations sensorielles de natures différentes. Les principaux capteurs utilisés en robotique mobile sont : les capteurs télémétriques (les ultrasons, les lasers et les infrarouges), le GPS et les caméras[30].

I.7.2.1 Système de positionnement global (GPS)

Les besoins de localisation étant omniprésents dans de très nombreux secteurs de la vie actuelle, d'où l'idée d'avoir un système de localisation le plus universel possible qui a donné lieu à l'apparition du Global Positioning System (GPS) (figure I-13). C'est un système de balises utilisant les ondes radio, pour garantir une accessibilité de n'importe quel point de la surface du globe, placé sur des satellites en orbite terrestre. Ce système permet donc d'avoir une mesure de sa position dans un repère global couvrant la terre avec une précision variant de quelques dizaines de mètres à quelques centimètres suivant les équipements.



Figure I-13 : Un GPS 6M avec un contrôleur d vol APM 2.8 pour drone

Ce système est cependant loin de résoudre tous les problèmes de localisation des robots mobiles. Il fonctionne en effet difficilement dans des environnements urbains, et n'est pas utilisable à l'intérieur des bâtiments, car les ondes employées sont fortement atténuées en traversant les murs. Sa précision est de plus souvent trop faible pour qu'un robot terrestre puisse utiliser ces informations seules. En pratique, il est souvent couplé à un système inertielle qui permet de pallier aux pertes du signal GPS et il ne remplace de toute façon pas les capteurs du robot qui lui permettent de percevoir son environnement immédiat [31].

I.7.2.2 Télémètres

Les télémètres les plus utilisés actuellement pour des applications de cartographie et de localisation sont les télémètres laser à balayage (figure I-14). Ils utilisent un faisceau laser mis en rotation afin de balayer un plan, ce qui permet de mesurer la distance des objets qui coupent ce plan. Cette mesure peut être réalisée selon différentes techniques soit en mesurant le temps de vol d'une impulsion laser, soit par triangulation.

On trouve plusieurs technologies, à savoir, les SONAR, qui sont principalement utilisés dans le domaine maritime. En robotique, les capteurs à ultrasons, de par leur simplicité, leur compacité et leur coût, sont les plus fréquemment utilisés pour la détection d'objets relativement proches. En effet, les ondes ultrasonores étant très évanescentes, la portée des capteurs se limite à quelques mètres.

Deuxième catégorie de capteurs réunit les télémètres à ondes optiques, ou LIDAR (de l'anglais **L**ight **D**etection **A**nd **R**anging). Leur fonctionnement est le même que celui des SONAR, la différence étant le domaine spectral dans lequel ces appareils travaillent et la nature du signal utilisé. En effet, alors que les SONAR fonctionnent avec des ondes sonores, les LIDAR opèrent avec des ondes optiques dans le domaine du visible, des ultraviolets et des infrarouges.).

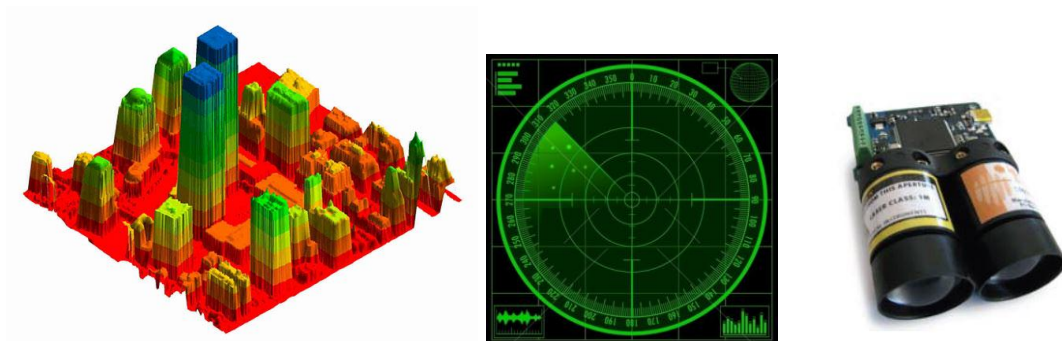


Figure I-14 : Télémètre laser : (a) Cartographie (b) détection d'objets (c) SF02/F.

Enfin, la dernière classe rassemble les télémètres à ondes radioélectriques, ou RADAR (de l'anglais RAdio Detection And Ranging). Ces éléments, via un oscillateur et une antenne, peuvent émettre de puissantes ondes radioélectriques de différentes façons. Les plus utilisées étant les ondes pulsées et les ondes continues. En analysant le signal réfléchi, il est possible de localiser et même d'identifier l'objet responsable de la réflexion et grâce à l'effet Doppler, les RADAR peuvent également calculer sa vitesse de déplacement. Le principal avantage des ondes radioélectriques est qu'elles peuvent se propager avec une faible atténuation à travers l'air et divers obstacles (nuage, brouillard, fumée ...) pouvant altérer un signal sonore ou lumineux. Cet instrument rend alors possible la détection et le pistage d'objets dans des conditions qui paralyseraient les autres technologies.

Ces capteurs à ultrason sont historiquement les premiers à être utilisés. Ils sont basés sur la mesure du temps de vol d'une onde sonore réfléchi par les obstacles pour estimer la distance. Par contre les télémètres à infrarouge utilisent une lumière infrarouge au lieu d'une onde sonore pour la détection et peuvent être basés sur différentes techniques qui permettent de recueillir plus ou moins d'informations.

Leurs Avantages réside dans la fourniture directe et précise des distances avec une haute précision ainsi que la mesure des distances à partir d'un grand de points fixes tels que: sols, murs, plafonds... etc.

Leurs inconvénients en plus de leurs coûts élevés et de leurs inutilisations sur les terrains accidentés (outdoor), ils réfléchissent la lumière lorsque le rayon traverse une plaque de verre ou de plastique. Ainsi, ces instruments concernent la zone aveugle, de quelques centimètres de longueur, en dessous de laquelle la détection d'objets n'est pas possible et la lenteur de propagation des ondes ultrasonores. Pour un drone miniature, leur utilisation est envisageable pour la mesure de la distance au sol dans les phases de décollage et d'atterrissage.



Figure I-15 : Technologie LIDAR OnyxScan embarquée sur drone - AltiGator Drone & UAV Technologies.

I.7.2.3 Vision

Les systèmes de vision en robotique sont basés sur l'utilisation d'une ou plusieurs caméras. Ils sont très utilisés en robotique mobile, car ils semblent proches des méthodes utilisées par les humains [32]. Ils sont très performants en termes de portée, précision, quantité d'informations exploitables et la facilité d'intégration, mais leur inconvénient majeur se situe dans la gestion de flux important des données exploitables. On peut distinguer deux grandes classes de caméras, les caméras CCD et les caméras CMOS.

Une caméra utilisée seule ne peut fournir qu'une information 2D (si elle n'est pas en mouvement). Les techniques qui permettront d'obtenir des informations 3D à partir d'un tel capteur sont généralement liées à l'adjonction d'un autre capteur. Dans ce cadre, nous pouvons identifier les caméras stéréoscopiques (figure I-16), utilisés dans les drones. À partir de deux points de vue de l'environnement différents, il est possible d'estimer la distance des objets et d'avoir ainsi une image de profondeur, cela simule la façon dont nous voyons et crée donc l'effet 3D vu. Qui peut être utilisée pour l'évitement d'obstacles ou la cartographie.



Figure I-16 : Caméra stéréoscopique Bumblebee 2 à gauche et Caméra ZED montée sur drone à droite.



Figure I-17 : Drone DJI Inspire Black XT avec sa caméra de suivi de type Electro-Optical Infrared (EO/IR) FLIR 336 30Hz.

I.7.2.4 Caméra de profondeur

Les caméras RGB-D, existent depuis quelques années, mais représentent un coût prohibitif pour la plupart des applications robotiques. Cependant, un nouveau type de matériel permettant d'atteindre cet objectif a fait son apparition récemment : la Kinect (figure I-18) de Microsoft/PrimeSense. Initialement destinée au monde de jeu vidéo, la Kinect représente une véritable révolution dans le domaine de la robotique. Pour une somme modique (130 euros environ/2015), elle permet de disposer d'un capteur fournissant à la fois des données 3D et des images[33].

Elle est composée de trois systèmes optiques : une caméra dans l'infrarouge, une caméra RGB et un laser. La caméra RGB permet d'accéder aux images qui seront traitées avec les outils usuels en vision par ordinateur. Le couple caméra IR/Laser sert à l'estimation de distance. Le laser projette dans la scène des patterns qui ne sont visibles que par la caméra IR. L'image IR permet alors, connaissant la distance de la base entre le laser et la caméra, d'estimer la distance d'un point de l'espace par triangulation. La résolution de la caméra IR permet d'obtenir un nuage de point de taille 640x480, qui correspond à la résolution de l'image. Il est alors possible d'attribuer une couleur précise à chaque point en faisant correspondre les points du nuage issu des mesures 3D avec l'image RGB [33].

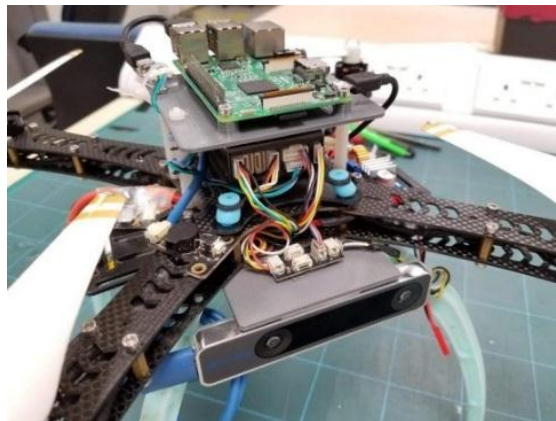


Figure I-18 : Caméra Intel RealSense Tracking Camera T265.

Après avoir présenté les différents capteurs utilisés en robotique, nous allons nous intéresser aux problèmes de la localisation pour la navigation et la perception de l'environnement réel.

I.8 Applications des quadrirotors :

Généralement, les quadrirotors sont utilisés pour la surveillance, l'inspection, la cartographie, la recherche et le sauvetage, etc. actuellement, ils sont impliqués dans des

applications plus avancées qui attirent l'attention des chercheurs. Ils sont également utilisés pour l'inspection des réseaux de hautes tensions telles que les lignes électriques, les gazoducs ou les voies ferrées et livraison (figure I-19). Un système UAV autonome basé sur la vision, qui cartographie et explore des environnements inconnus est décrit dans [88].



Figure I-19 : applications des drones (a) Matrice 300 RTK et ses caméras hybrides multicapteurs Zenmuse H20 (b) livraison Amazon.

I.9 Contrôle et commande visuelle des mini-drones

Au cours des dernières décennies, de nombreuses recherches ont étudié les capteurs pour des spécifiques applications pour la navigation visuelle et la commande visuel des drones. Dans cette étude, nous nous concentrons sur les méthodes basées sur la vision qui utilisent une ou plusieurs caméras comme capteur principal. Dans cette section, nous présentons brièvement les techniques de commande basées sur la vision de haut et de bas niveau et les méthodes de leurs chevauchements. Deuxièmement, les méthodes de commande basées sur la vision, telles que le PID, le flou et le backstepping.

La commande de vol à bas niveau couvre les tâches de contrôle qui sont déjà implémentées dans les drones, à savoir, la capacité de vol stationnaire et rejet de perturbation réalisées par la commande d'attitude, le suivi de trajectoire et la commande de position. D'autre part, la commande haut niveau de vol met en œuvre la planification des missions et de la trajectoire, y compris la cartographie et l'évitement d'obstacles[10].

I.9.1 Asservissement visuel:

Malgré que dans ce travail nous nous concentrons sur le suivi des trajectoires par l'odométrie visuelle en vue d'améliorer la commande, nous citons un autre volet d'étude qui utilise le même concept.

En général, l'asservissement visuel a deux directions principales : l'asservissement visuel basé sur l'image (IBVS) et l'asservissement visuel basé sur la position (PBVS)[34]. IBVS

utilise les caractéristiques d'image 2D pour le calcul des valeurs de commande, tandis que PBVS utilise la pose 3D d'une cible pour corriger la trajectoire de mouvement [35] [36]. Plusieurs approches et techniques ont été utilisées en combinant des capteurs dans différentes applications sur les quadrirotors, à savoir : système d'atterrissage et de navigation automatique pour générer des trajectoires robustes et stables.

I.9.2 Commande et suivi trajectoire par vision

Contrairement aux techniques d'asservissement visuel, les méthodes de commande basées sur la vision sont structurées par la configuration de caméras montées et l'utilisation d'autres capteurs de fusion, afin de mettre en œuvre des algorithmes, tels que la localisation, la commande et le filtrage, par exemple l'utilisation des unités de mesures inertielle (IMU) dans un Kalman étendu (EKF) pour les tâches de navigation autonome, telles que le survol sur des altitudes spécifiques, le suivi de trajectoires, les manœuvres et l'atterrissage ainsi que la manipulation d'objets[37]. Dans la figure suivante, l'auteur a conçu une commande adaptative par mode de glissement, un cas d'étude similaire.

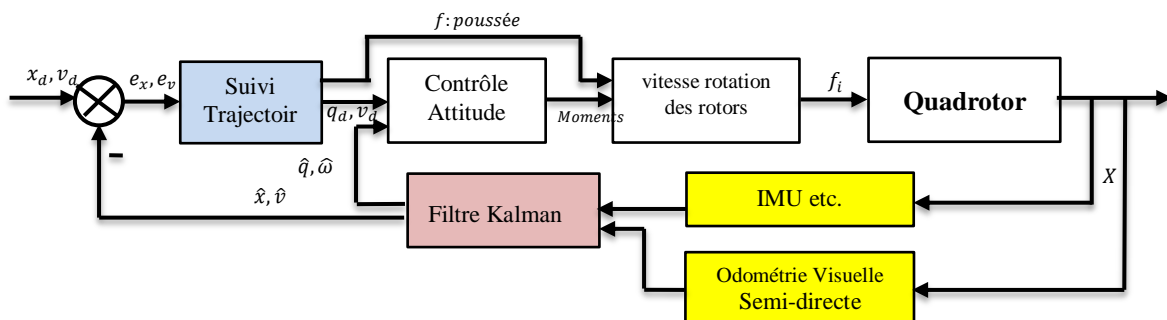


Figure I-20 : Schéma de suivi de trajectoire[38].

La navigation autonome est considérée comme très difficile et nécessite de nombreuses ressources pour les applications en temps réel basées sur le suivi et la planification de trajectoire [39]. En particulier, les méthodes à faible coût basées sur la vision deviennent de plus en plus complexes pour améliorer les capacités de navigation des drones. Cependant, cela devient une nécessité dans les véhicules aériens autonomes pour mettre en œuvre des algorithmes qui génèrent des trajectoires sans collision tout en augmentant considérablement l'autonomie des drones, notamment dans les missions où aucune ligne de vue n'est disponible.

I.10 Localisation et cartographie

Aujourd'hui le capteur le plus répandu pour la navigation des véhicules est bien le système GPS (*Global Positioning System*), car il fournit une position géo-référencée du récepteur en mesurant le temps de réception des signaux émis par des satellites. Cependant, la précision du capteur est dépend de la licence obtenue des récepteurs GPS par des fabricants, le grand public est de bas coût et l'erreur est de l'ordre de quelques dizaines de mètres, ce qui est bien trop important pour localiser un véhicule autonome et planifier des trajectoires sur des espaces navigables et ainsi en ajoutant le phénomène de multi-trajet des signaux GPS qui peuvent être réfléchis dans les zones urbaines.

Pour se localiser sans GPS, certains systèmes utilisent des capteurs proprioceptifs, qui mesurent une information sur le mouvement du drone. Les centrales inertielles embarquées à bord du véhicule mesurent de vitesses et accélérations, ils fournissent des mesures aberrantes et bruitées. La centrale inertielle fournit des mesures de vitesses et d'accélérations qu'il faut intégrer une ou deux fois pour obtenir un déplacement, ce qui entraîne des erreurs relativement importantes sur de longues distances (dérive). Donc il n'est pas envisageable d'utiliser uniquement des capteurs proprioceptifs pour la localisation.

Alternativement, les caméras sont des capteurs extéroceptifs à bas coût, fournissant une information très riche, précise, utilisable pour la localisation. C'est ce type de capteur qui va être utilisé dans la suite de ce mémoire. Contrairement aux capteurs laser, une image fournit une information projective de l'environnement, pour obtenir une information de localisation ou de géométrie, il est nécessaire de traiter les images.

I.10.1 Localisation du robot volant

La fonction localisation intégrée sur tout robot aérien tente de trouver une réponse à la question de base « où suis-je ? », pour cela il récupère les informations de l'environnement grâce à ses capteurs en déduisant sa pose (orientation et position). On distingue trois types de localisation :

I.10.1.1 Localisation relative

La localisation relative, également appelée localisation à l'estime, est basée sur l'utilisation des capteurs proprioceptifs[40], elle consiste à évaluer la pose, la vitesse et l'accélération du robot mobile. Cette évaluation est réalisée en intégrant les informations fournies par les capteurs proprioceptifs et la pose à l'instant $t - 1$ (figure I.11). Ces informations

peuvent être récoltées à partir des capteurs suivants : l'odométrie pour le déplacement, la centrale inertielle pour la pose...etc. Le flux d'informations issu par les capteurs est relativement important à l'estimation de la position ce qui favorise ce type de localisation, mais son inconvénient majeur est l'intégration (l'accumulation) d'erreur.

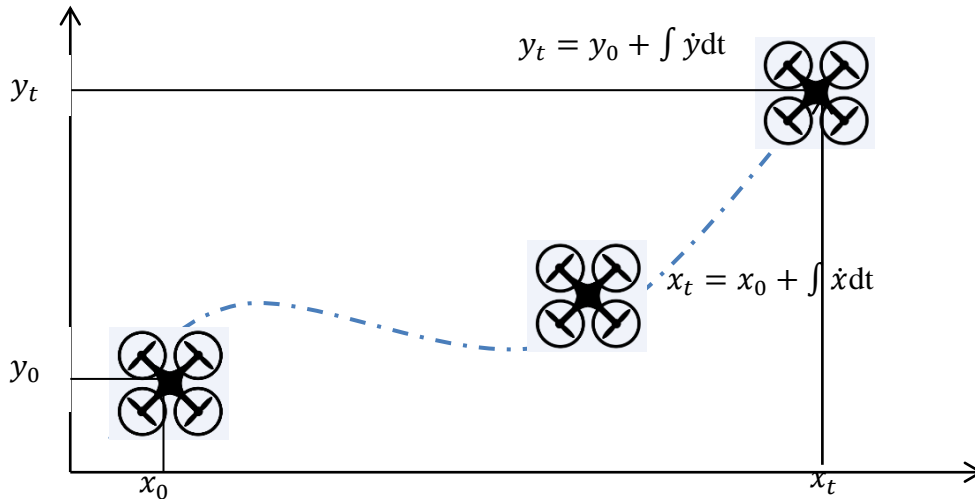


Figure I-21 : Principe de la localisation relative.

I.10.1.2 Localisation absolue

La localisation absolue est une technique qui permet à un robot de se repérer dans un repère global (figure I.12) lié à l'environnement[40]. Ces capteurs peuvent être différenciés en deux catégories. D'une part, il existe des capteurs dédiés à la localisation permettant de détecter des balises artificielles actives ou passives qui sont connues dans l'environnement[28], qui permet de mesurer soit la position soit la distance, parmi ces capteurs on trouve le GPS. D'autre part, il existe des capteurs dédiés à la perception qui réalisent des mesures sur l'environnement proche d'un robot mobile. Parmi ces capteurs on trouve les caméras, les télémètres laser. La localisation par ces capteurs est basée sur la détection des amers naturels ou artificiels.

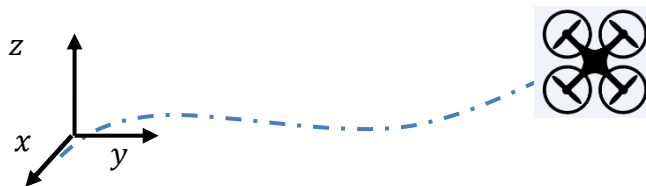


Figure I-22 : Principe de la localisation absolue.

I.10.1.3 Localisation hybride

La localisation hybride combine les données issues des capteurs extéroceptifs, et celles issues des capteurs proprioceptifs pour estimer la pose du robot. Cette technique va permettre de pallier les inconvénients de la localisation absolue et relative, et d'envisager de prendre les avantages de chacun des deux types pour trouver une meilleure solution, que ce soit en termes de précision ou de fiabilité pour la localisation du robot. Les données issues des différents capteurs sont fusionnées en utilisant soit un filtre de Kalman (KF), un filtre de Kalman étendu (EKF) ou un filtre à particule.

I.10.2 Cartographie

La cartographie est l'ensemble des études et des opérations, scientifiques, artistiques et techniques, intervenant à partir des résultats d'observations directes ou de l'exploitation d'une documentation, en vue de l'élaboration et de l'établissement de cartes, plans et autres modes d'expression. La cartographie en robotique mobile est un outil qui permet au robot de décrire son environnement d'évolution. Plusieurs types de carte ont été établies, parmi eux on cite les cartes métriques, topologiques, carte d'occupation, carte de points d'intérêt. Cette dernière a été utilisée dans notre travail.

I.10.2.1 La carte métrique :

La (Figure I.13-(a)) est une représentation géométrique plus ou moins détaillée de l'environnement à partir des données perceptuelles ; où on peut trouver des informations de longueur, de distance, de position...etc. elle offre une relation bien définie avec le monde réel[41].

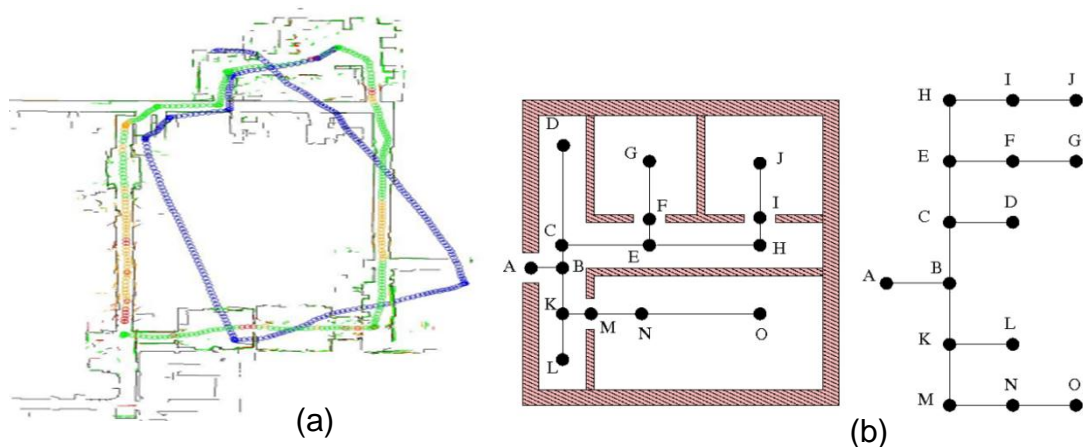


Figure I-23 : (a) Carte métrique, (b) Carte topologique.

I.10.2.2 La carte d'occupation (occupancy grid) :

Dans cette représentation (Figure I.14) l'environnement est divisé en secteur, et on attribue un vecteur à chaque secteur pour représenter ses propriétés : souvent, il s'agit du degré d'encombrement par un obstacle (indice indiquant que la cellule correspondante est occupée ou non par un obstacle). À chaque cellule est associée une probabilité d'occupation avec une valeur estimée à partir des mesures fournies par les capteurs[42].

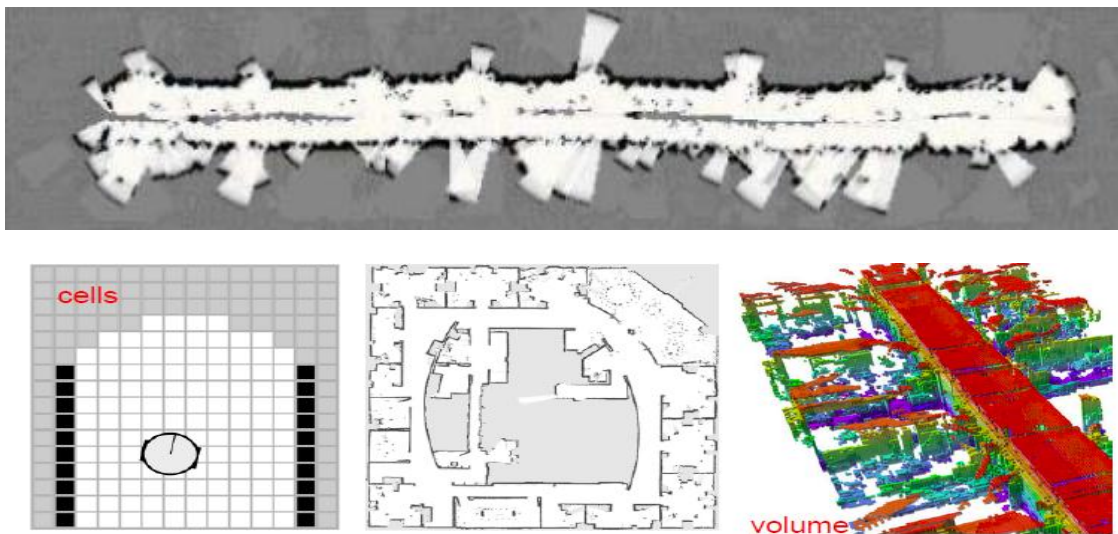


Figure I-24 : Cartes d'occupation[42].

I.10.2.3 Carte de points d'intérêt (*Feature Maps*) :

C'est une carte construite à partir des primitives (points clés) extraites d'une séquence d'images en utilisant la vision lors du déplacement d'un robot mobile[43] (Figure I.15). Les amers (points d'intérêt) sont des points de repère fixes et identifiables sans ambiguïté. Ces points doivent vérifier les caractéristiques suivantes : pouvoir discriminant, domaine de visibilité important, stabilité, invariance et bonne adaptation à la métrologie.

La carte de points d'intérêt est une représentation de ces points dans un repère fixe. Cette carte est mieux adaptée pour résoudre le problème du SLAM, elle offre une description plus compacte des larges espaces ouverts que la carte d'occupation. Mais elle est moins adaptée aux environnements encombrés.

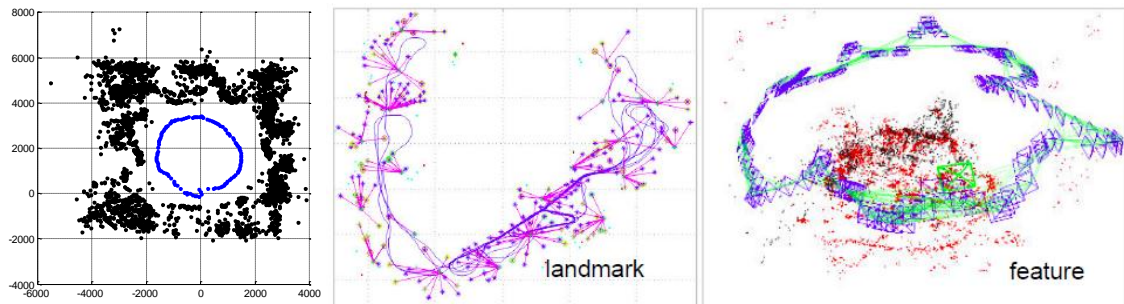


Figure I-25 : Cartes de points d'intérêt.

I.10.2.4 Carte topologique :

La carte topologique, figure I-26 (b), est une représentation décrivant les relations entre les éléments de l'environnement, sans utiliser un repère de référence absolu. Elles se présentent sous forme de graphes, dont les sommets correspondent à des lieux, souvent associés à des informations perceptuelles (histogrammes de couleurs, images, données télémétriques...etc.)[41].

I.10.3 SLAM

La localisation et cartographie simultanée (SLAM : *Simultaneous Localisation and Mapping*).est une technique utilisée par les robots mobiles et les véhicules autonomes pour construire une carte de l'environnement inconnu sans connaissance a priori, ou de mettre à jour une carte dans un environnement connu (avec une connaissance a priori d'une carte donnée), tout en gardant un suivi de leur emplacement actuel[18].

Ces dernières années le problème SLAM a été un axe de recherche particulièrement actif dans le domaine de la robotique Montemerlo et al. [44]; Durrant-Whyte & Bailey [45]. A partir d'une position de départ, le principe consiste à construire incrémentalement une carte de l'environnement, d'utiliser cette carte pour la localisation et de mettre la carte à jour lorsque de nouvelles mesures sont effectuées. Grâce à l'ajout de nouvelles données, l'incertitude sur la carte diminue. Classiquement, les techniques de SLAM visuel sont basées sur des points d'intérêts et un filtre de Kalman étendu. Cependant, ces méthodes ont une efficacité calculatoire limitée, due à l'inversion d'une matrice de covariance dont les dimensions augmentent avec la taille de la carte reconstruite. Récemment, des techniques de SLAM dense basées sur des méthodes directes ont été proposées. Tykkala & Comport [46] proposent d'intégrer temporellement les cartes de disparités obtenues par mise en correspondance stéréo afin d'améliorer la reconstruction, en réduisant la zone de recherche des disparités.

Cependant, toutes les approches présentées précédemment restent incrémentales, et

intègrent une erreur de dérive, non négligeable sur de longues distances. Certains algorithmes proposent de détecter les lieux où le robot est déjà passé, en fonction de l'apparence des images, on parle alors de fermeture de boucle (cf. Cummins & Newman [47]). Ce genre d'information permet d'ajouter des contraintes sur la trajectoire obtenue, qui peut être ré-optimisée efficacement par certains algorithmes dédiés à l'optimisation de graphes tels que Grisetti et al. [48].

Dans le domaine de la vision par ordinateur, le problème de *SLAM* est plus connu sous le nom de *Structure from Motion* (SfM) Martinec et al. [49]. Bien que l'objectif soit similaire, c'est à dire reconstruire le mouvement des caméras et la structure de la scène, les contraintes sont différentes. En effet, contrairement au *SLAM* où la reconstruction est effectuée de manière incrémentale et en ligne, toutes les images nécessaires à la reconstruction sont disponibles à l'initialisation. En général, les reconstructions sont effectuées hors ligne, ce qui permet d'utiliser des techniques d'ajustement de faisceaux (Triggs et al. [50]), consistant à optimiser simultanément toutes les variables du système (*i.e.* position des images et modèle 3D). Bien que certains algorithmes fournissent des reconstructions sur des régions de taille relativement importante (Furukawa & Ponce [51]), la reconstruction réaliste à très grande échelle reste pratiquement irréalisable. Lorsqu'une seule caméra est utilisée pour le *SLAM* visuel, on parle de *SLAM* monoculaire ou *Bearing Only SLAM*.

I.10.4 Odométrie visuelle

Finalement lorsque, seulement la trajectoire de la caméra est considérée, où s'inscrit notre thèse on parle alors d'odométrie visuelle Nistér et al. [52]; Howard [53]; Comport et al. [54] qui consiste à estimer incrémentalement la position de la caméra, le long de la trajectoire. L'algorithme odométrie visuelle pour estimer le mouvement de la caméra image par image à partir des paires d'images stéréo successives, sans aucune hypothèse a priori sur mouvement de la caméra, se base sur les primitives visuelles prépondérantes sélectionnées dans les deux images. Ensuite, les primitives des images sont mises en correspondance de telle sorte qu'à partir d'une primitive de la première image, on puisse connaître où elle se trouve dans la seconde et vice-versa. Alors, les images sont appariées via leurs primitives visuelles et l'étape suivante consiste à estimer les paramètres de déplacement de la caméra d'une façon incrémentale. Ces algorithmes ont été testés sur de nombreuses plateformes, y compris les véhicules à roues, et se sont avérés être rapides, précis et robustes [53].

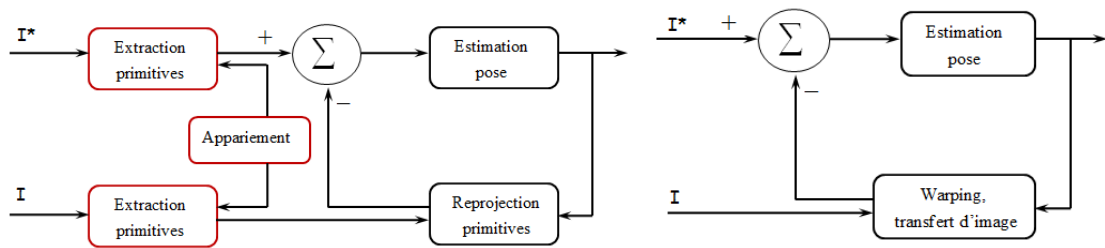
I.10.5 Méthodes d'estimation

La localisation visuelle peut être considérée comme l'estimation de la trajectoire d'une caméra par rapport à un repère initial. Ce repère peut être par exemple la première image d'une séquence, ou alors un objet connu, suivi dans les images et permettant d'extraire une information sur la position relative entre la caméra et l'objet. Tout d'abord, les techniques de localisation visuelle peuvent être divisées en deux groupes [55] : les méthodes basées sur les points d'intérêts et les méthodes directes. Ensuite le problème de localisation est étroitement lié à l'estimation de la géométrie de la scène, on parle alors de localisation et cartographie simultanée.

I.10.5.1 Méthodes basées sur les points d'intérêt

Les méthodes basées sur l'extraction de points d'intérêts (*feature-based*), cherchent à extraire des primitives visuelles locales dans les images, telles que les points de Harris & Stephens [56]. Une étape d'appariement est ensuite effectuée, c'est à dire mettre en correspondance un à un les points d'intérêts de chaque image. Pour cela il est nécessaire d'utiliser une mesure de similarité entre les primitives. En général le coût est évalué entre les intensités des pixels extraits dans voisinage des points d'intérêts, par une SSD (*Sum of Squared Differences*), une SAD (*Sum of absolute differences*) ou une ZNCC (*Zero Normalized Cross Correlation*). Pour rendre plus robuste cette étape, souvent critique pour ce genre de méthodes, il est possible d'utiliser des descripteurs locaux, robustes aux changements d'échelle et aux rotations, tels que Lowe [57]; Bay et al. [58]. Le mouvement relatif entre les images est en général obtenu soit par une méthode robuste de type RANSAC (*Random Sample Consensus* Fischler & Bolles) [59], soit par une méthode itérative de minimisation non linéaire de l'erreur de re-projection des points d'intérêts.

Ce genre d'approches, sont les plus fréquemment utilisées pour l'estimation de pose entre images, car elles permettent de réduire l'information contenue dans les images à quelques centaines de points d'intérêts. Cependant, elles nécessitent une étape intermédiaire d'extraction et d'appariement de points d'intérêts entre les images, basée en général sur des seuils de détection. Cette étape de prétraitement est souvent mal conditionnée, bruitée et non robuste, ce qui nécessite d'utiliser des techniques d'estimation et filtrage.



(a) Méthode basé points d'intérêts. (b) Méthode directe.

Figure I-26 : Comparaison des méthodes d'estimation itératives.

(a) Une méthode basée points d'intérêts nécessite l'extraction de primitives dans les images, puis une étape d'appariement. La pose entre les images peut alors être estimée itérativement en minimisant la re-projection des points d'intérêts.

(b) Une méthode directe minimise directement une erreur d'intensité entre les deux images, en utilisant une technique de synthèse d'image.

I.10.5.2 Méthodes directes

Les méthodes directes (*image-based*), quant à elles, n'ont pas de phase de sélection de points d'intérêts ou de primitives visuelles. Le mouvement de la caméra est directement obtenu en minimisant les erreurs d'intensités communes aux deux images à l'aide d'une transformation paramétrique (figure I.6). Dans ce cas l'estimation du mouvement et la mise en correspondance des pixels s'effectuent simultanément lors de l'optimisation. Dans la majorité des cas, ce type de technique est utilisé pour le suivi d'une surface plane Lucas & Kanade [60], ou alors le suivi d'un ensemble de surfaces planes Mei et al. [61]; Shi & Tomasi [62]; Caron et al. [63]. Quelques travaux, notamment ceux présentés dans Comport et al. [54] proposent une généralisation de l'algorithme au suivi de modèles 3D obtenus par mise en correspondance dense stéréo. Tous les pixels des images sont alors utilisés dans la boucle d'estimation, permettant une estimation robuste et précise du mouvement.

I.10.6 Méthodes avec apprentissage

Un algorithme de SLAM en temps réel n'étant pas envisageable à grande échelle pour la localisation d'un robot, il est possible de découpler le problème en deux parties[55] :

1. La cartographie, la partie la plus complexe, peut-être traitée hors ligne lors d'une phase d'apprentissage.
2. La carte obtenue peut alors être utilisée en ligne, pour localiser efficacement une caméra naviguant à l'intérieur du modèle.

Ce type de méthodes peut également être classé en deux groupes : l'utilisation de

modèles 3D, obtenus soit par conception assistée par ordinateur (CAO), soit avec une méthode de reconstruction automatique, et les modèles de type « mémoires images », consistant à distribuer dans l'environnement des images acquises lors de la phase d'apprentissage, sans reconstruire explicitement le modèle 3D global.

I.11 Conclusion

Au cours de ce chapitre, nous avons brièvement présenté le contexte général des mini-drones à voilure tournantes. Nous avons également vu les principales architectures et leurs applications. Nous avons présenté un aperçu sur les véhicules aériens sans pilote et leurs classifications. Nous avons détaillé l'ensemble des capteurs utilisés pour la localisation et la commande. Ensuite, nous avons introduit le problème de localisation visuelle comme un problème essentiel pour la navigation autonome. Enfin un état de l'art sur les techniques de localisation visuelle a été donné. Notre souci est donc de développer un module d'odométrie visuelle à base des informations issues du capteur de vision permettant le suivi de trajectoires.

Dans le chapitre suivant, nous allons présenter la modélisation du quadrirotor et ainsi que du capteur caméra stéréoscopique pour lesquels nous allons définir des stratégies de commande et de navigation autonome.

Chapitre II : Modélisation du quadrirotor et de la caméra

CHAPITRE II: MODELISATION DU QUADROTOR ET DE LA CAMERA

II.1 Introduction

La modélisation des véhicules aériens est complexe et requiert la connaissance de l'aérodynamique du véhicule étudié, de ses caractéristiques mécaniques ainsi que de la dynamique des actionneurs. La complexité des phénomènes aérodynamiques impliqués induisent d'une part une plus forte sensibilité aux perturbations pour les systèmes de petite échelle, et d'autres parts des limitations sur la charge utile qui génèrent des contraintes de dimensionnement et de poids pour les capteurs embarqués.

Dans la littérature, la modélisation et le développement de nouvelles techniques de commande des quadrotors a été un sujet de plusieurs travaux de recherche tels que : les travaux de Madani et Hamel [64, 65], Kendoul et Wang et al [66, 67] sur les multi-rotors. La première étape de la modélisation consiste à caractériser le comportement d'un engin évoluant dans un fluide quelconque. Cette modélisation nécessite également un travail spécifique à chaque configuration de véhicule étudié. La deuxième étape consiste à déterminer les relations entre les variables pertinentes d'entrée et de sortie en se basant sur les concepts fondamentaux de la physique.

Dans ce chapitre, nous présentons le modèle dynamique d'un véhicule à voilure tournante en utilisant le formalisme de Newton-Euler. Un véhicule à voilure tournante peut être considéré comme un corps rigide à six degrés de liberté pour lequel on y ajoute les forces aérodynamiques générées par ses actionneurs. Ensuite, nous présenterons la géométrie de la caméra stéréoscopique, la modélisation, le calibrage et le principe de la reconstruction 3D.

II.2 Modélisation Dynamique du Quadrotor

Nous considérons ici le cas des aéronefs sans pilote à voilure tournante (des UAV's) de type quadrotor ou plus particulièrement de taille petite des MAV (*Micro Aerial Vehicle*), le choix de ce type d'UAV a pour but de réaliser des missions prédéfinies en assurant sa bonne localisation et ses meilleures performances en termes de commande et de stabilité, et se justifie par :

- leurs discrétions, car ils sont équipés par des moteurs de moyenne puissance (~50 W) qui ne font pas trop de bruit ;

- leurs agilités : ce type d'UAV parviennent à faire des manœuvres que d'autres de gabarit plus important ne peuvent pas se permettre ;
- leurs coûts : ils ont un coût si relativement bas, qu'on peut se permettre d'en former un escadron pour assurer l'accomplissement d'une certaine mission ;
- ce type de missions citées ne nécessite pas une charge utile importante, juste l'instrumentation de bord (GPS, IMU et caméra) qu'un MAV peut très bien porter.

Comme déjà évoqué précédemment, un modèle du quadrotor sera donné en vue de sa commande en poursuite de trajectoires. A cette fin, l'utilisation d'une base de données visuelle est adoptée grâce aux capteurs appropriés embarqués d'une part, et des techniques de commandes avancées telles que le backstepping d'autres parts.

La configuration géométrique d'un quadrotor n'est pas autre que celui d'un hélicoptère sans pilote à quatre rotors, sur lequel des repères lui seront associés. Chaque rotor est constitué d'un moteur électrique et d'une hélice. Les hélices 1 et 3 tournent dans le sens opposé de celui des hélices 2 et 4, voir figure II-1. Le mouvement de cet UAV est contrôlé par la variation des vitesses de rotation des rotors pour varier le couple et la portance produits par chacun d'eux.

La modélisation des robots volants ou des drones est une tâche minutieuse car la dynamique du système est fortement non-linéaire et pleinement couplée particulièrement pour les robots volants légers.

En pratique, le vol des robots volants est régi par un grand nombre d'effets physiques. Cependant, il est nécessaire dans un premier temps d'effectuer des hypothèses afin d'éliminer ceux de plus faible importance. Pour cela le modèle dynamique est dérivé en utilisant le formalisme de Newton–Euler [64, 68] sous les hypothèses suivantes:

- la structure mécanique de l'UAV est supposée rigide ;
- le quadrotor possède une structure parfaitement symétrique ce qui suppose que la matrice d'inertie sera diagonale;
- le centre de masse et le repère se coïncident ;
- Les hélices sont rigides ;
- la portance et la traînée de chaque rotor sont proportionnelles au carré de sa vitesse angulaire, ce qui est une approximation très proche du comportement aérodynamique.

Pour la modélisation du quadrotor, deux repères sont à considérer (figure II-1), un repère fixe lié à la terre R^a et un second mobile R^b lié au robot volant ayant pour origine le centre de masse de ce dernier.

Comme le montre la figure II-3, un châssis fixé au corps, \mathcal{R}^b , est considérée comme alignée avec les axes transversaux du quadrotor et du référentiel inertiel, \mathcal{R}^a . \mathcal{R}^b est décrit dans \mathcal{R}^a par un vecteur position, r , au centre de masse, c , et la matrice de rotation est notée R . Pour éviter les singularités associées au paramétrage, la matrice de rotation complète est utilisée pour décrire les orientations. La vitesse angulaire du quadrotor dans le cadre du corps est donnée par $\Omega = R^T \dot{R}$, où $R \in SO3$. Pour décrire l'orientation du système, les angles d'Euler— le roulis, le tangage et le lacet (notés respectivement, ϕ , θ et ψ)— sont utilisés [24].

Les quatre rotors sont numérotés de 1 à 4, comme le montre la figure II-1. Celle-ci montre également que le pas des rotors impairs est opposé à celui des rotors pairs. La vitesse angulaire du rotor est ω_i ; la portance résultante (F_i) et le moment de réaction (M_i) sont donnés par la formule suivante :

$$M(\eta) = \begin{cases} F_i = k_f \omega_i^2 \\ M_i = k_M \omega_i^2 \end{cases} \quad (\text{II.1})$$

La coordination des vitesses des moteurs permet de piloter le quadrotor à six degrés de liberté à l'aide de seulement quatre commandes (qui sont les couples appliqués par les moteurs sur chaque hélice). La figure II-2 montre les mouvements possibles qu'un drone peut effectuer. Plus précisément, un quadrotor a cinq mouvements principaux (vertical, roulis, tangage, lacet et translations horizontales). Par contre la figure II-3 illustre les mouvements et rotations sur un exemple de deux plateformes volantes.

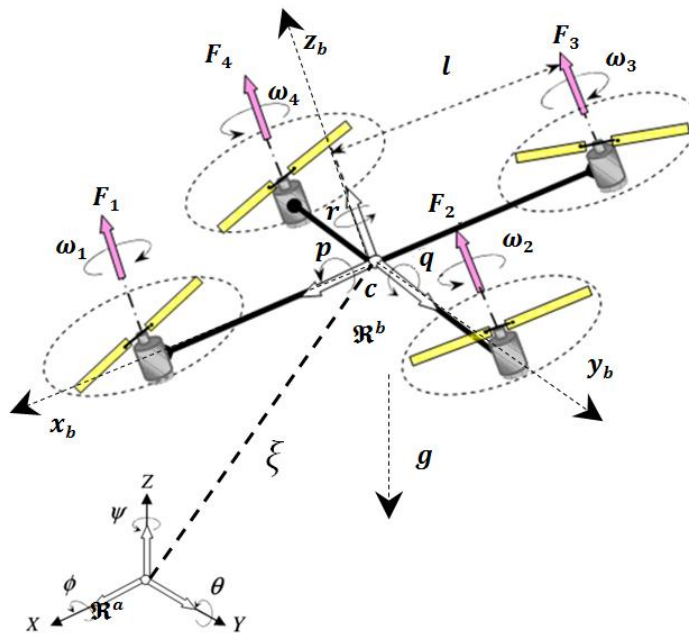


Figure II-1 : Configuration du quadrotor.

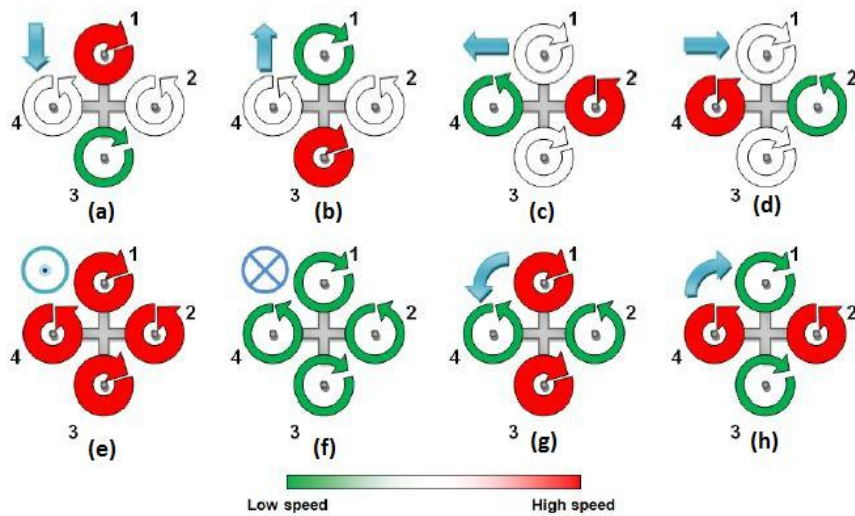


Figure II-2 : Mouvement du Quadrotor.

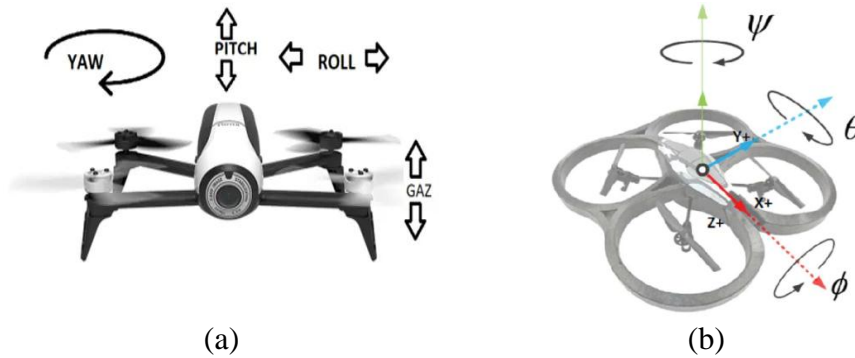


Figure II-3 : (a) Parrot Bebop 2.0 et (b) Ardrone 2.

La position absolue d'un quadrotor par rapport au référentiel d'inertie et son attitude sont décrites par $\xi = [x, y, z]^T$ et les angles d'Euler, $\eta = [\varphi, \theta, \psi]^T$, respectivement. Les angles d'attitude sont le lacet (ψ , rotation autour du z-axis), tangage (θ , rotation autour du y-axis), et roulis (φ , rotation autour du x-axis) angles; Plus précisément, ($-$ resp), ($-\frac{\pi}{2} < \theta < \frac{\pi}{2}$), et ($-\frac{\pi}{2} < \varphi < \frac{\pi}{2}$).

Les angles d'Euler ont un total de 12 séquences possibles suivant les deux orientations. Cependant, le problème majeur est que tous les angles présentent des singularités et des erreurs d'orientation. Ces angles sont souvent appelés angles de roulis, de tangage et de lacet (ZYX dans cette étude).

La relation du référentiel corps-fixe, \mathcal{R}^b , par rapport au référentiel initial fixé à la terre, \mathcal{R}^a , satisfait $\mathcal{R}^b = R \cdot \mathcal{R}^a$, et la matrice de rotation, R , est défini comme suit, où s et c

désignent respectivement les opérateurs trigonométriques « sin » et « cos ».

$$R(\eta) = \begin{pmatrix} c\psi c\theta & s\phi s\theta c\psi - s\psi c\phi & c\phi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\phi s\theta s\psi + c\psi c\phi & c\phi s\theta s\psi - s\psi c\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \quad (\text{II.2})$$

Soit $V = [u, v, w]^T \in \mathfrak{R}^b$ le vecteur vitesse linéaire, et $\Omega = [p, q, r]^T \in \mathfrak{R}^b$ le vecteur vitesse angulaire de la structure aérienne exprimée dans le repère lié au corps-fixe [65].

La relation entre les vecteurs de vitesse $^{(V, \Omega)}$ et $^{(\dot{\xi}, \dot{\eta})}$ est donnée par la suivante:

$$\begin{cases} V = R^T(\eta) \dot{\xi} \Rightarrow \dot{\xi} = R(\eta) V \\ \Omega = M(\eta) \dot{\eta} \end{cases} \quad (\text{II.3})$$

où

$$M(\eta) = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi \cos\theta \\ 0 & -\sin\phi & \cos\phi \cos\theta \end{pmatrix} \quad (\text{II.4})$$

Les équations du mouvement utilisant le formalisme de Newton–Euler donnent le modèle dynamique suivant [64]:

$$\begin{cases} m \ddot{\xi} = F_f + F_t + F_g \\ J \cdot \dot{\Omega} = -\Omega \wedge J \cdot \Omega + M_f - M_a - M_g \end{cases} \quad (\text{II.5})$$

où

$$\begin{cases} \dot{R} = R \cdot S(\Omega) \\ \dot{\xi} = V \end{cases} \quad (\text{II.6})$$

Où Ω représente le vecteur de vitesse angulaire exprimée dans le repère lié à la mobilité. Il est donné par:

$$\Omega = M(\eta) \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (\text{II.7})$$

Où v désigne la vitesse linéaire exprimée dans la référence inertielle. Ci-dessus, m est la masse totale de la structure, et $J \in \mathfrak{R}^{3 \times 3}$ est une matrice d'inertie constante définie positive symétrique du quadrotor par rapport à \mathfrak{R}^b , où la matrice d'inertie peut être écrite comme

$$J = \text{Diag}(J_{xx}, J_{yy}, J_{zz}).$$

$s(\Omega)$ désigne la matrice antisymétrique du vecteur vitesse angulaire, $\Omega = (\Omega_1 \ \Omega_2 \ \Omega_3)^T$ est définie comme suit:

$$S(\Omega) = \begin{pmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{pmatrix} \quad (\text{II.8})$$

F_f désigne la force de portance générée par les quatre rotors et est donnée par :

$$F_f = \begin{pmatrix} c \varphi s \theta c \psi + s \psi s \varphi \\ c \varphi s \theta s \psi - s \varphi c \psi \\ c \varphi c \theta \end{pmatrix} \cdot \left(\sum_{i=1}^4 F_i \right) \quad (\text{II.9})$$

Où F_i est la force de portance générée par le rotor i ; il est proportionnel au carré de la vitesse angulaire, ω_i , comme suit:

$$F_i = b \cdot \omega_i^2 \quad (\text{II.10})$$

Où b est la constante ou le facteur de portance..

F_t : désigne la force de traînée le long des axes $(\vec{Ox}, \vec{Oy}, \vec{Oz})$ et est donné par :

$$F_t = K_t \cdot \dot{\xi} \quad (\text{II.11})$$

Où $K_t = \text{Diag}(-k_{tx}, -k_{ty}, -k_{tz})$ est la matrice de translation du coefficient de traînée. F_g représente la force de gravité exprimée par :

$$F_g = m \cdot (0 \quad 0 \quad -g)^T \quad (\text{II.12})$$

M_f : désigne le couple produit par les forces de poussée et de traînée et est exprimé par:

$$M_f = \begin{pmatrix} \ell (F_3 - F_1) \\ \ell (F_4 - F_2) \\ d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{pmatrix} \quad (\text{II.13})$$

Où ℓ est la distance entre le centre du quadrirotor et l'axe du rotor et $d > 0$ est le facteur de traînée.

M_a : représente le couple résultant du frottement aérodynamique et s'exprime sous la forme

$$M_a = \begin{pmatrix} k_{fax} & 0 & 0 \\ 0 & k_{fay} & 0 \\ 0 & 0 & k_{faz} \end{pmatrix} \cdot \Omega^2 \quad (\text{II.14})$$

Où $k_{fax}, k_{fay}, k_{faz}$ sont les coefficients de frottement aérodynamique. M_g : représente le couple gyroscopique des hélices et est donné par:

$$M_g = \sum_{i=1}^4 J_r \cdot \Omega \wedge \begin{pmatrix} 0 \\ 0 \\ (-1)^{i+1} \cdot \omega_i \end{pmatrix} \quad (\text{II.15})$$

Où J_r est l'inertie des rotors.

Sur la base des équations ci-dessus, le modèle suivant est obtenu [68, 69]:

$$\left\{ \begin{array}{l} \ddot{\varphi} = \left(\frac{J_y - J_z}{J_x} \right) \dot{\theta} \dot{\psi} - \frac{J_r}{J_x} \Omega \dot{\theta} - \frac{k_{fax}}{J_x} \dot{\varphi}^2 + \frac{\ell}{J_x} u_2 \\ \ddot{\theta} = \left(\frac{J_z - J_x}{J_y} \right) \dot{\varphi} \dot{\psi} + \frac{J_r}{J_y} \Omega \dot{\varphi} - \frac{k_{fay}}{J_y} \dot{\theta}^2 + \frac{\ell}{J_y} u_3 \\ \ddot{\psi} = \left(\frac{J_x - J_y}{J_z} \right) \dot{\theta} \dot{\varphi} - \frac{k_{faz}}{J_z} \dot{\psi}^2 + \frac{1}{J_z} u_4 \\ \ddot{x} = -\frac{k_{fx}}{m} \dot{x} + \frac{1}{m} u_x u_1 \\ \ddot{y} = -\frac{k_{fy}}{m} \dot{y} + \frac{1}{m} u_y u_1 \\ \ddot{z} = -\frac{k_{fz}}{m} \dot{z} - g + \frac{1}{m} (\cos \varphi \cos \theta) u_1 \end{array} \right. \quad (\text{II.16})$$

Où u_1, u_2, u_3 et u_4 sont les entrées de commande du système, qui sont exprimées sur la base des vitesses angulaires des quatre rotors comme suit:

$$\left\{ \begin{array}{l} u_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 = b(\omega_4^2 - \omega_2^2) \\ u_3 = b(\omega_3^2 - \omega_1^2) \\ u_4 = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \\ u_x = (\cos \varphi \cos \psi \sin \theta + \sin \varphi \sin \psi) \\ u_y = (\cos \varphi \sin \psi \sin \theta - \sin \varphi \cos \psi) \\ \Omega = \omega_1 - \omega_2 + \omega_3 - \omega_4 \end{array} \right. \quad (\text{II.17})$$

La dynamique de translation décrite par les dernières expressions de l'équation (II.17) montre que les variables (x, y, z, ψ) et (φ, θ) sont couplés. De ces équations de la dynamique de traduction, les expressions suivantes pour les contraintes non holonomiques d'ordre supérieur peuvent être extraites:

$$\left\{ \begin{array}{l} \tan \theta = \frac{\left(\ddot{x} - \frac{k_{tx}}{m} \dot{x} \right) \cos \psi + \left(\ddot{y} - \frac{k_{ty}}{m} \dot{y} \right) \sin \psi}{\ddot{z} + g - \frac{k_{tz}}{m} \dot{z}} \\ \sin \varphi = \frac{-\left(\ddot{x} - \frac{k_{tx}}{m} \dot{x} \right) \sin \psi + \left(\ddot{y} - \frac{k_{ty}}{m} \dot{y} \right) \cos \psi}{\sqrt{\left(\ddot{x} - \frac{k_{tx}}{m} \dot{x} \right)^2 + \left(\ddot{y} - \frac{k_{ty}}{m} \dot{y} \right)^2 + \left(\ddot{z} + g - \frac{k_{tz}}{m} \dot{z} \right)^2}} \end{array} \right. \quad (\text{II.18})$$

Ces deux contraintes reflètent la dépendance existante entre les différentes variables du modèle dynamique du quadrotor, qui peut être exploitée pour atténuer le sous-actionnement du système car elle permet une dérivation simple.

Chaque rotor contient un moteur à courant continu (CC) actionnant son hélice via un réducteur. Le moteur à courant continu est régi par la dynamique suivante:

$$\left\{ \begin{array}{l} V_r = ri + L \frac{di}{dt} + k_e \omega_i \\ k_m i = J_r \frac{d\omega_i}{dt} + C_s + k_r \omega_i^2 \end{array} \right. \quad (\text{II.19})$$

Où V_r est l'entrée du moteur, ω_i est la vitesse de rotation du $i^{\text{ème}}$ hélice, k_e et k_m sont les constantes de couple électrique et mécanique, respectivement, k_r définit le couple à charge constante, r représente la résistance interne du moteur, et J_r et C_s sont respectivement l'inertie du rotor et le frottement solide.

Ainsi, le modèle choisi pour le rotor s'exprime comme suit:

$$\dot{\omega}_i = b V_{ri} - \beta_0 - \beta_1 \omega_i - \beta_2 \omega_i^2, i \in [1, 4] \quad (\text{II.20})$$

$$\text{Où } \beta_0 = \frac{C_s}{J_r}, \beta_1 = \frac{k_e k_m}{r J_r}, \beta_2 = \frac{k_r}{J_r} \text{ et } b = \frac{k_m}{r J_r}$$

Tableau II.1. Paramètres du quadrotor.

Caractéristiques	Valeur
Masse du quadrirotor	$m=0.5$ kg
Bras de levier latéral	$l=0.2$ m
Coefficient de poussée	$b=4.74e-5$ N s ²
Coefficient de traînée	$d=2.35e-7$ N m s ²
Moment d'inertie des moteurs	$J_r=1e-4$ kg m ²
Moment d'inertie de roulement	$I_x=6.22e-2$ kg m ²
Moment d'inertie de tangage	$I_y=7.33e-2$ kg m ²
Moment d'inertie de lacet	$I_z=9.64e-2$ kg m ²
vitesse angulaire maximale	$\omega_{max}=220$ rad/s

II.3 Modélisation de la caméra stéréo (banc stéréoscopique)

La vision humaine est une source importante d'inspiration des algorithmes de perception artificielle de l'environnement. En se basant sur la vision binoculaire, M. Taira [70] a étudié le rôle de certains neurones dans la perception de l'orientation des surfaces tridimensionnelles. En vision monoculaire, la troisième dimension n'est pas facile à estimer. Une seule projection montre le problème d'avoir un point sur le plan image, de plusieurs points de l'espace réel. Il manque d'autres informations pour pouvoir estimer les coordonnées de chaque point de l'espace. En vision par ordinateur, la composante tridimensionnelle peut être obtenue de différentes façons. S.T. Barnard [71] a distingué deux techniques permettant la vision des reliefs : la vision active (laser, radar, lumière structurée), et la *vision passive*² (la stéréo photométrie, la stéréovision, et la vision monoculaire en présence du mouvement).

La localisation visuelle utilise une ou plusieurs caméras à des points de vues différentes. Par inspiration de l'être humain, les caméras stéréoscopiques sont considérées comme un meilleur moyen pour percevoir l'environnement. Ce capteur délivre un flux d'information important qui nécessite une connaissance des différents requis (calibrage, géométrie épipolaire, points d'intérêt...etc.).

Pour extraire l'information 3D à partir des points d'intérêt nous faisons appel à un algorithme de triangulation. Dans ce chapitre nous allons essayer de présenter quelques

² En vision passive, une scène est simplement observée par une ou plusieurs caméras sans interaction entre les capteurs utilisés et l'environnement.

techniques de vision par ordinateur pour l'extraction des points d'intérêts, l'appariement des primitives, ainsi de reconstruire des points 3D à partir des images acquises par un banc de caméras stéréoscopiques.

Nombreuses sont les applications de vision par ordinateur qui requièrent la mise en correspondance de primitives entre deux prises de vue. On peut citer à titre d'exemple, les applications de reconstruction 3D, suivi d'objet ou de cible mobile, ou encore notre sujet la localisation visuelle. Dans ce cadre de travail, il s'agit de calculer la position et l'orientation du robot mobile dans un environnement réel.

II.4 Modélisation et géométrie des caméras embarquées

La vision binoculaire (stéréoscopique) est une vision passive inspirée de la vision humaine permettant d'obtenir l'information de relief d'une scène à partir de deux projections bidimensionnelles de la même scène. La position tridimensionnelle des points objet est déduite à partir de deux images d'une même scène prises de deux points de vue légèrement différents. Les deux images stéréoscopiques doivent être prises en même temps dans le cas où la scène observée est dynamique. Le rendu 3D d'une scène est obtenu en se référant aux trois étapes fondamentales suivantes : le calibrage, la mise en correspondance ou l'appariement et la reconstruction 3D par triangulation. Il est crucial de bien choisir les paramètres du modèle géométrique (entraxe, focales, etc.) du capteur stéréoscopique selon le dilemme suivant :

- Plus l'entraxe, défini par la distance entre les deux centres optiques des deux caméras, est faible et les axes optiques parallèles, plus le champ visuel couvert communément par les deux caméras est grand, et plus les images sont proches au sens de l'appariement des indices.
- Plus les caméras sont écartées, meilleure est la reconstruction tridimensionnelle obtenue.

Calculer la position 3D d'un point réel à partir d'un capteur stéréoscopique consiste à passer par les étapes suivantes :

- Définition de l'architecture géométrique du capteur stéréoscopique (calibrage, rectification).
- Appariement des primitives (la mise en correspondance) des images stéréoscopiques.
- Reconstruction tridimensionnelle par triangulation géométrique.

II.4.1 Modèle géométrique de la caméra

Le modèle sténopé modélise une caméra idéale (simple projection perspective). Le système optique induit des distorsions géométriques qui affectent la projection des points. Ainsi, un point projeté dans une image ne correspond pas au modèle sténopé. La distorsion est d'autant plus élevée que le champ de vue de la caméra est grand. Cependant, il est possible de modéliser la distorsion en enrichissant le modèle sténopé par des termes supplémentaires (le modèle devient non linéaire).

La calibration permet de déterminer les paramètres intrinsèques et extrinsèques d'une ou plusieurs caméras en utilisant une mire dont le modèle est connu à l'avance. Pour réaliser la calibration du banc stéréoscopique, les paramètres à estimer sont (figure II.4):

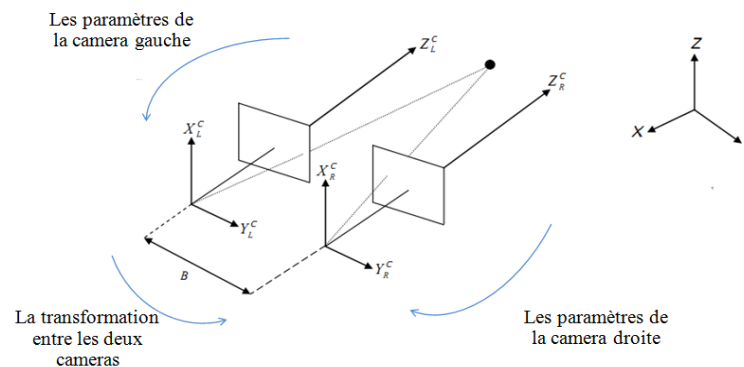


Figure II-4 : Principe de calibration stéréoscopique.

Pour déterminer les paramètres du modèle de la caméra, on la placera devant une mire (objet étalon) de calibration. C'est un ensemble de points dont les coordonnées sont parfaitement connues dans un repère de la mire qui est différent du repère caméra. Chaque point de la mire se projette dans l'image puis on mesure ses coordonnées dans le repère image. Le point Q est défini par ses coordonnées (x_w, y_w, z_w) dans le repère universel. Il est également précisé par ses coordonnées (x_c, y_c, z_c) dans le repère caméra.

II.4.2 Paramètres intrinsèques

Le changement de repère qui lie les coordonnées du point q dans le repère image (figure II.5) avec le repère lié à la caméra est défini par les paramètres intrinsèques de la caméra.

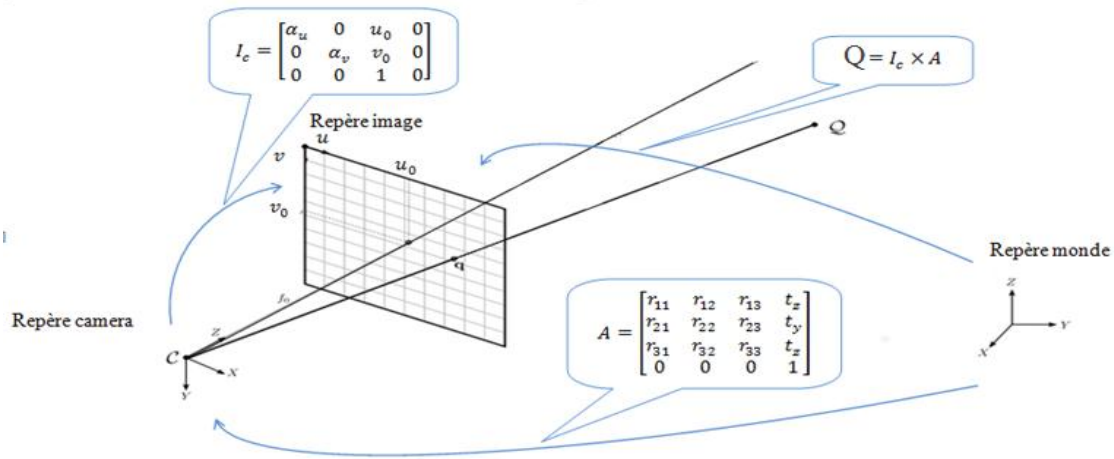


Figure II-5 : Projection perspective.

En projection perspective, un point Q est transformé en son homologue q dans le plan image par:

$$\begin{cases} u = u_0 - k_u f \frac{x_c}{z_c} \\ v = v_0 + k_v f \frac{y_c}{z_c} \end{cases} \quad (\text{II.21})$$

Avec : (u, v) : sont les coordonnées dans le repère image, (u_0, v_0) : est le centre d'image, f : la focale, (k_u, k_v) : sont des facteurs d'échelle, (x_c, y_c) : sont les coordonnées du point dans le repère camera.

Soit en notation matricielle :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} -k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (\text{II.22})$$

$$= \begin{bmatrix} -k_u f & 0 & u_0 & 0 \\ 0 & k_v f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (\text{II.23})$$

Avec : $u = U/W$ et $v = V/W$.

On pose : $I_c = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ avec : $\alpha_u = -k_u f$ et $\alpha_v = k_v f$.

C'est une application linéaire de l'espace projectif vers le plan projectif exprimant la transformation perspective :

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = I_c \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (\text{II.24})$$

Dans ce modèle figure quatre paramètres α_u, α_v, u_0 et v_0 . Ce sont les paramètres qui vont être estimés par le calibrage. Notons que la distance focale ne peut être calculée explicitement. En introduisant des coordonnées caméra normalisée par rapport à Z telles que :

$$\begin{cases} x_c = \frac{x}{z} \\ y_c = \frac{y}{z} \\ z_c = 1 \end{cases} \quad (\text{II.25})$$

À partir de cela la relation entre les coordonnées image et les coordonnées caméra est :

$$\begin{cases} u = \alpha_u x_c + u_0 \\ v = \alpha_v y_c + v_0 \end{cases} \quad (\text{II.26})$$

II.4.3 Paramètres extrinsèques

Les paramètres extrinsèques permettent de définir la transformation géométrique entre le repère caméra et la scène (repère objet) voir figure II.6, ils sont représentés par la matrice A de l'équation II.27.

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.27})$$

Afin de déterminer ces paramètres du modèle de la caméra, nous allons placer devant la caméra une mire (un objet étalon) : Un ensemble de points dont les coordonnées sont parfaitement connues dans un repère de la mire qui est différent du repère caméra, figure II-6. Chaque point de la mire se projette dans l'image et on mesure ses coordonnées dans le repère image.

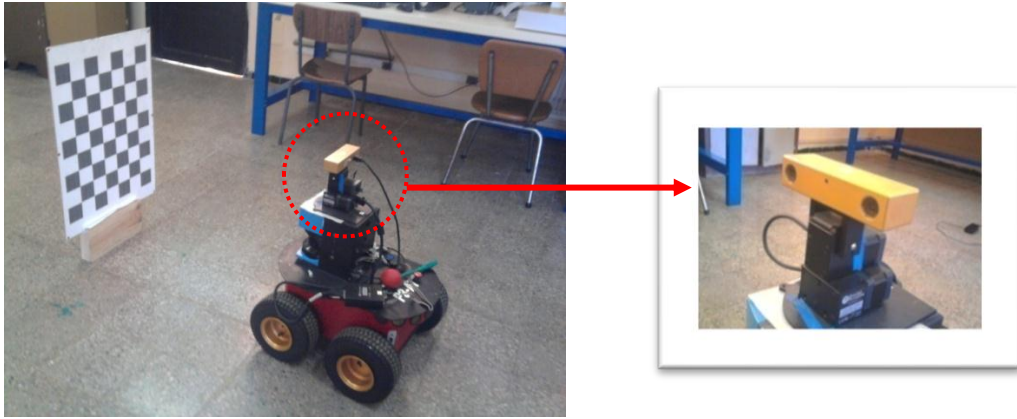


Figure II-6 : Banc stéréoscopique et la mire de calibration.

La transformation mire/image se décompose donc en une transformation mire/caméra suivie d'une projection et suivie enfin d'une transformation caméra/image. La transformation mire/caméra se compose d'une rotation et d'une translation

Nous pouvons écrire la transformation mire/image sous la forme d'une matrice 3×4 appelée matrice de projection perspective et qui peut se décomposer comme suit :

$$M = I_c \cdot A \quad (\text{II.28})$$

$$M = \begin{bmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{bmatrix} \quad (\text{II.29})$$

Avec $r_1 = (r_{11} \ r_{12} \ r_{13})$.

M est la matrice de projection perspective et elle peut, en général, s'écrire sous la forme suivante:

$$\begin{bmatrix} Su \\ Sv \\ S \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{23} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (\text{II.30})$$

Dans cette formule X , Y et Z sont les coordonnées d'un point Q de la mire dans le repère mire. Cette matrice peut également s'écrire sous une forme plus simple :

$$M = \begin{bmatrix} m_1 & m_{14} \\ m_2 & m_{24} \\ m_3 & m_{34} \end{bmatrix} \quad (\text{II.31})$$

De même $m_1 = (m_{11} \ m_{12} \ m_{13})$.

En identifiant M avec $I_c \cdot A$, en tenant compte des propriétés d'orthonormalité de la

rotation on obtient un ensemble d'équations qui permettent de calculer les paramètres intrinsèques et extrinsèques en fonction des coefficients de M .

En identifiant M avec $I_c.A$ on a :

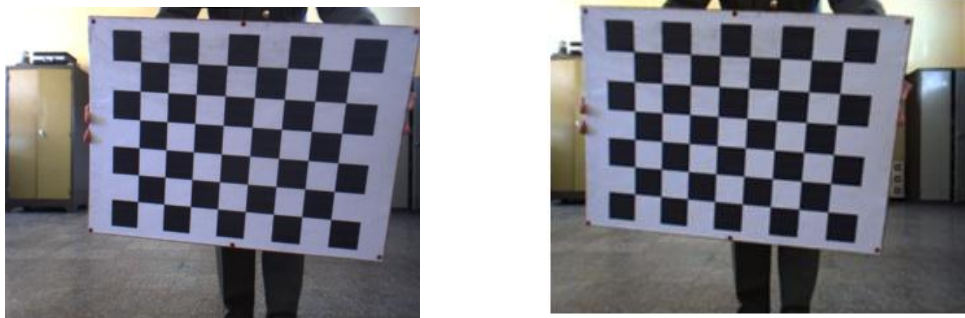
$$\left\{ \begin{array}{l} r_3 = m_3 \\ u_0 = m_1 \cdot m_3 \\ v_0 = m_2 \cdot m_3 \\ \alpha_u = - \|m_1 \wedge m_3\| \\ \alpha_v = - \|m_2 \wedge m_3\| \\ r_1 = \frac{1}{\alpha_u} (m_1 - u_0 m_3) \\ r_2 = \frac{1}{\alpha_v} (m_2 - v_0 m_3) \\ t_x = \frac{1}{\alpha_u} (m_{14} - v_0 m_{34}) \\ t_y = \frac{1}{\alpha_v} (m_{24} - v_0 m_{34}) \\ t_z = m_{34} \end{array} \right. \quad (\text{II.32})$$

II.4.4 Calibrage

Le calibrage stéréoscopique consiste donc à déterminer la matrice de transformation entre le repère caméra gauche et le repère caméra droite. Cette matrice qui est une matrice composée d'une matrice de rotation et une matrice de translation. Pour la déterminer nous commençons par calibrer chaque caméra par rapport à une mire unique. Ceci nous fournit les coefficients de deux matrices M et M' de calibrage. Ensuite on extrait les paramètres intrinsèques et extrinsèques de chaque caméra. A l'aide des paramètres extrinsèques on construit deux matrices de transformations, une étant la transformation du repère mire au repère de la caméra gauche, et l'autre du repère mire/caméra droite. Enfin, on calcule la transformation de repère de la caméra gauche/ repère de la caméra droite, la distance entre les deux caméras joue un rôle important pour la perception de profondeur.

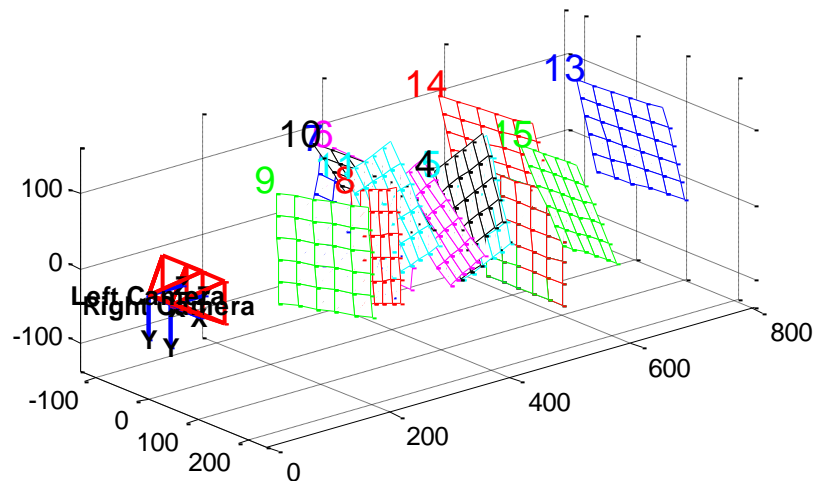
Cette relation entre les deux caméras s'appelle la géométrie épipolaire, elle est totalement indépendante de la structure de la scène. Tout point de l'image gauche leur correspondant se trouve sur des droites dans l'image droite qui passe par un point s'appelle épipole droite (de même gauche). Une configuration particulière mène les droites épipolaires horizontales dit, la rectification d'images. La rectification d'une paire d'images stéréoscopiques permet de simplifier considérablement le processus de mise en correspondance. Pour notre travail, nous avons utilisé le banc stéréoscopique Bumblebee2 stéréo. Ce qui permet d'acquérir des images rectifiées, donc sans distorsion avec différentes résolutions.

Pour calibrer notre banc stéréoscopique nous avons utilisé un toolbox « stéréo-calib-gui » matlab [72] et un kit de programme opencv [73] les résultats sont comparés et validés et jugés acceptables par la reconstruction de l'ensemble de points de la mire, la position 3D de la mire par rapport au banc est simulée par la figure II.7.



(a) Image gauche de la mire.

(b) Image droite de la mire.

Figure II.7 : Mire de calibrage.**Figure II.8 :** Simulation de position de la mire par rapport banc stéréoscopique.

Les résultats de calibrage obtenus sont représentés dans le tableau II.1 (figure II-8), ils montrent que la matrice de rotation entre les deux cameras est égale à l'identité, et la première composante du vecteur de translation (selon x) est égale à 120.24090 mm, dont on sait a priori qu'elle est égale à 120 mm, donc nous pouvons dire que notre estimation des paramètres est juste avec une erreur petite de 0.24090 mm.

Caméra droite	La focale	$\alpha_u = 234.23025$ $\alpha_v = 234.75334$
	Le point central	$u_0 = 159.76682$ $v_0 = 124.25891$

Caméra gauche	La focale	$\alpha_u = 234.57305$ $\alpha_v = 235.14684$
	Le point central	$u_0 = 161.52446$ $v_0 = 125.22095$
Transformation entre les deux caméras	Vecteur de rotation	$\alpha = -0.00315$ $\beta = 0.00659$ $\gamma = 0.00004$
	Vecteur de translation	$t_x = -120.24090$ $t_y = 0.05382$ $t_z = -0.60776$

Tableau II.2 : Résultats de calibrage stéréoscopique par Matlab.

II.5 Reconstruction 3D

La reconstruction tridimensionnelle consiste à calculer la géométrie de la scène observée par le système d'acquisition. Elle est obtenue à partir des résultats de la phase de mise en correspondance en utilisant les paramètres de calibrage du système des caméras utilisées.

La triangulation est la technique qui va nous permettre de tirer l'information 3D, c'est-à-dire la profondeur. Nous supposons que les étapes de rectification et de mise en correspondance sont déjà faites. Donc on a les images dans le même plan et les axes optiques sont parallèles.

Soit deux caméras dans le même plan avec des axes optiques parallèles (images rectifiées) et soit le point physique P dont les coordonnées sont (X_g, Y_g, Z_g) dans le repère caméra gauche et (X_d, Y_d, Z_d) dans le repère caméra droite, et O_g et O_d sont les centres optiques de la caméra gauche et droite (figure II.9).

Soit le point C le centre du repère de la scène et qui correspond au milieu entre O_g et O_d , alors si nous localisons le point P par rapport au point C , les coordonnées réelles de l'objet P seront $(X_g, Y_g - \frac{b}{2}, Z_g)$ avec : $b = \overline{O_g O_d}$.

Considérons P_g, P_d les projections du point P sur les images gauche et droite qui ont respectivement les coordonnées (u_g, v_g, f_g) et (u_d, v_d, f_d) . Avec $f = f_d = f_g$ désigne la distance focale. Comme les images sont rectifiées alors : $v_g = v_d$.

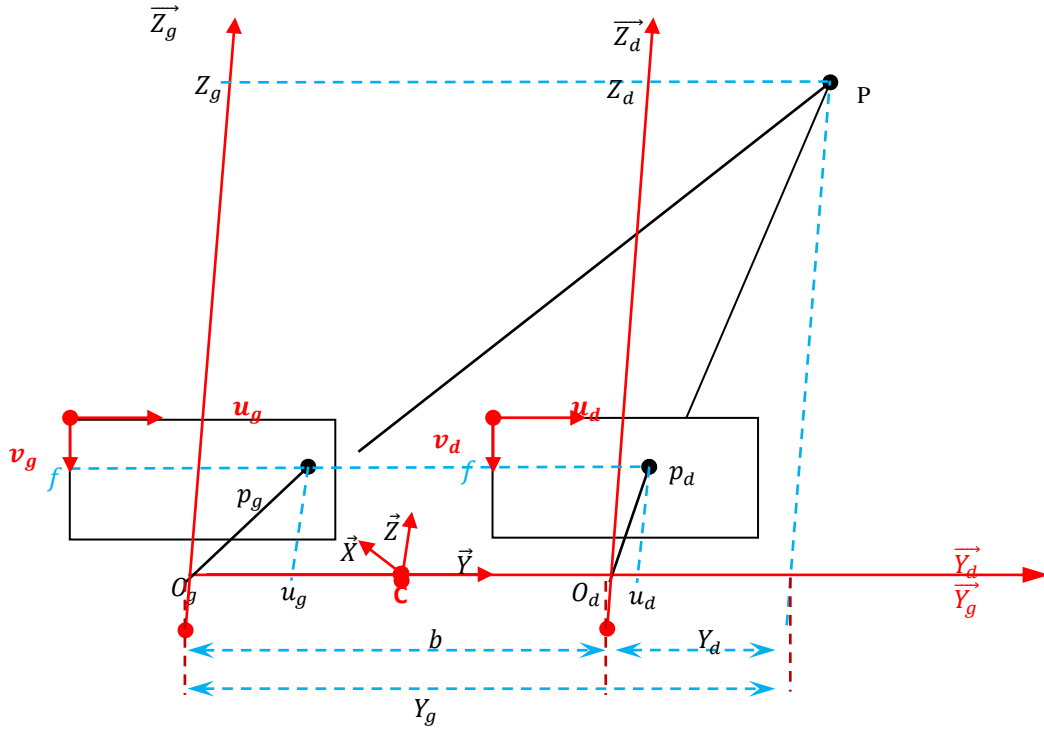


Figure II-9 : Principe de la triangulation.

On a :

$$\begin{bmatrix} s \cdot u_g \\ s \cdot v_g \\ s \end{bmatrix} = I_{CG} \cdot \begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix} \quad (\text{II.33})$$

Et

$$\begin{bmatrix} s \cdot u_d \\ s \cdot v_d \\ s \end{bmatrix} = I_{CD} \cdot \begin{bmatrix} X_d \\ Y_d \\ Z_d \\ 1 \end{bmatrix} \quad (\text{II.34})$$

Avec I_{CG}, I_{CD} sont les matrices des paramètres intrinsèques des caméras gauche et droite, $s = z_G/f$ et $s' = z_D/f$ [74].

Notons que la distance focale ne peut être calculée explicitement. En introduisant des coordonnées des deux caméras sans dimension normalisée par Z telles que :

$$\begin{cases} X_C = \frac{X_g}{Z_g} \\ Y_C = \frac{Y_g}{Z_g} \\ Z_C = 1 \end{cases} \quad (\text{II.35})$$

$$\begin{cases} X'_c = \frac{X_d}{Z_d} \\ Y'_c = \frac{Y_d}{Z_d} \\ Z'_c = 1 \end{cases} \quad (\text{II.36})$$

En remplaçant dans (II.27) et (II.28), nous obtenons :

$$\begin{cases} \begin{bmatrix} X_C \\ Y_C \\ 1 \end{bmatrix} = I_{CG}^{-1} \cdot \begin{bmatrix} u_g \\ v_g \\ 1 \end{bmatrix} \\ \begin{bmatrix} X'_c \\ Y'_c \\ 1 \end{bmatrix} = I_{CD}^{-1} \cdot \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} \end{cases} \quad (\text{II.37})$$

À partir de cette dernière équation, nous pouvons déterminer X_C, X'_c, Y_C et Y'_c donc il reste Z_g à déterminer. D'après la figure II.9 nous avons :

$$Y_g = Y_d + b \quad (\text{II.38})$$

Avec b est la ligne de base entre les deux caméras. D'après les équations (II.29) et (II.32), nous pouvons écrire:

$$Z_g \times Y_C = Z_d \times Y'_c + b \quad (\text{II.39})$$

Si nous supposons que $Z = Z_d = Z_g$ (la distance entre un point de la scène et les deux caméras est presque la même, car les deux caméras sont proches l'une à l'autre, alors l'équation II.31 devient :

$$Z = \frac{b}{Y_C - Y'_c} \quad (\text{II.40})$$

D'après les relations précédentes, les coordonnées réelles de l'objet P seront

$$\begin{cases} X = X_g \\ Y = Y_g - \frac{b}{Z} \\ Z = \frac{b}{Y_C - Y'_c} \end{cases} \quad (\text{II.41})$$

II.5.1 Évaluation d'algorithme de triangulation

Le calcul des coordonnées 3D des points d'intérêt est une étape très importante dans l'algorithme d'odométrie visuelle, une erreur induite par cette étape va engendrer une grande erreur dans ce dernier. Pour cela, nous présentons ci-dessous les erreurs d'estimation des coordonnées (X, Z) des points d'intérêt (vue que le travail est effectué dans le plans (X, Z)).

Nous avons effectué ce test dans un environnement intérieur où nous calculons les différentes valeurs de Z et de X . Le premier test consiste à faire varier la valeur de Z de $500mm$ jusqu'à $4000mm$, tout ça en fixant les valeurs de X et de Y .

$Z(mm)$	500	750	1000	1500	2000	2500	3000	3250	3500	4000
$Z(mm)$ <i>estimé</i>	503.9	753.6	989.5	1507.2	1984.1	2519.8	3023.8	3219.4	3532.7	3962
$dZ(mm)$	3.97	3.61	10.42	7.21	15.86	19.83	23.8	30.6	32.75	37.45

Tableau II.3 : Erreurs de triangulation suivant l'axe Z .

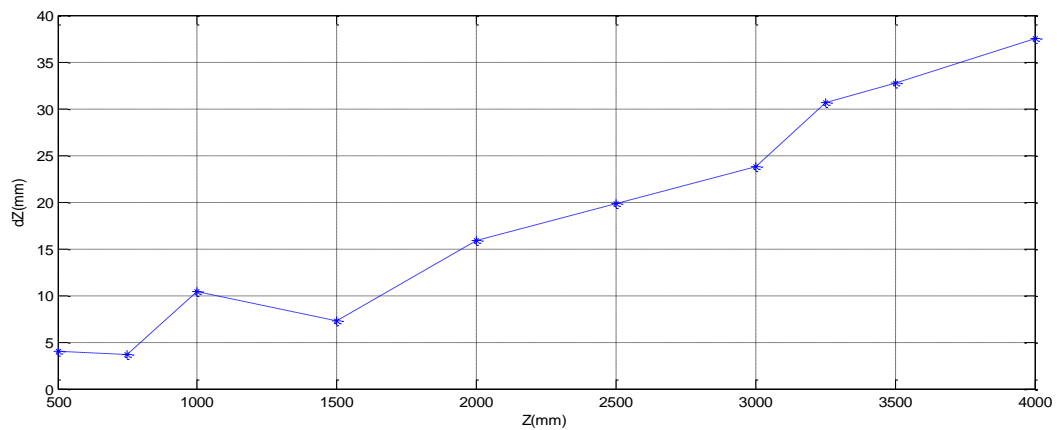


Figure II-10 : Erreurs de triangulation suivant Z .

D'après le tableau (II.3), nous remarquons que l'erreur d'estimation sur Z augmente progressivement en fonction de la profondeur réelle du point détecté (figure II.10). Cela est justifié par la faiblesse de la résolution des caméras pour estimer la profondeur, les erreurs sur les paramètres de calibrage et la valeur de la ligne de base. C'est pour cette raisons que nous avons choisi de filtrer les points qui se trouvent à plus de $3500mm$.

Le deuxième test consiste à faire varier la valeur de X du $-350mm$ jusqu'à $500mm$, tout ça en fixant les valeurs de Z et de Y .

$X(mm)$	-350	-300	-200	100	200	300	400	500
$dX(mm)$	12.03	7.5	5.53	3.66	3.91	8.52	12.41	12.83

Tableau II.4 : Erreurs de triangulation suivant X .

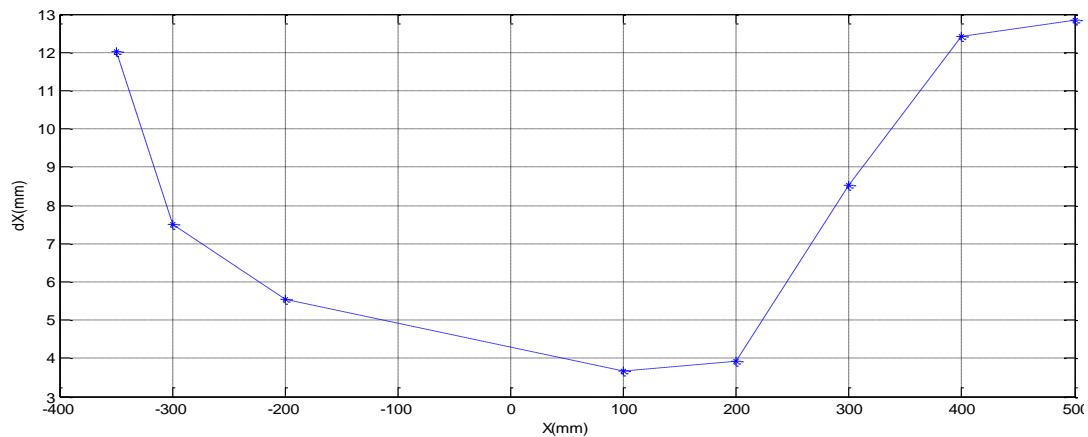


Figure II-11 : Erreurs de triangulation suivant X .

Le tableau II.4 et la figure II.11 présentent la variation de l'erreur d'estimation suivant l'axe X . Cette erreur diminue au voisinage de zéro et augmente ailleurs. Alors, la bonne estimation est au voisinage de l'axe optique des deux caméras.

II.6 Conclusion

Dans ce chapitre, nous avons présenté les concepts de la modélisation du quadrotor et du banc stéréoscopique de la camera. En premier lieu, les forces et moments qui agissent sur le quadrotor sont expliqués. Aussi le modèle dynamique du Quadrotor a été établi par le formalisme de Newton-Euler. Nous avons présenté un modèle qui décrit presque tous les phénomènes physiques agissant sur le Quadrotor par un vecteur des équations différentielles non linéaires de deuxième ordre. Ce modèle montre la nature couplée, complexe, non linéaire, multi variable et sous-actionnée, ce qui rend sa commande relativement difficile. Nous avons aussi présenté un modèle simplifié proposé dans la littérature

En second lieu, nous avons rappelé quelques principes de la stéréovision ; où nous avons commencé par un aperçu général sur la perception de relief (profondeur) par la vision binoculaire, puis nous avons abordé la géométrie des caméras stéréoscopiques (le calibrage du banc stéréoscopique et la géométrie épipolaire).

Dans le chapitre qui va suivre nous allons entamer notre approche pour résoudre le problème d'odométrie visuelle 3D, en utilisant deux nuages de coordonnées 3D issus de la mise en correspondance et la triangulation..

Chapitre III : Odométrie Visuelle

CHAPITRE III: ODOMETRIE VISUELLE

III.1 Introduction

L'Odométrie Visuelle (OV) est le processus d'estimation de *l'egomotion*³ [75] (égo-mouvement), d'un agent mobile (véhicule, robot ou drone) en utilisant uniquement une seule ou plusieurs caméras embarquées. Ses domaines d'application incluent la robotique mobile, l'informatique embarquée, la réalité augmentée et l'automobile.

Le terme odomètre visuel a été introduit pour la première fois en 2004 par Nister [52]. Il a utilisé le terme odomètre pour sa ressemblance avec l'odomètre des roues dans la robotique mobile, qui estime de manière incrémentale le mouvement d'un véhicule en intégrant le nombre de tours de ses roues au cours du temps. De la même manière l'OV fonctionne par une estimation incrémentale de la pose du quadrirotor par estimation du mouvement induit par le changement des images acquises. Afin que l'OV travaille efficacement, il faut que les trames d'images acquises présentent un éclairage et un chevauchement suffisamment dans la scène, il devrait y avoir une texture suffisante pour permettre l'extraction de mouvement apparent.

Tout d'abord, une revue historique récente de la recherche dans le domaine d'odométrie visuelle et ses principes fondamentaux, a été présentée [76], ensuite nous décrivons les principaux techniques d'estimation de mouvement avec un système de vision monoculaire et binoculaire, en exposant les avantages et les inconvénients de chaque système. Ensuite nous traiterons les différents algorithmes pour l'élimination des fausses associations. Une importance particulière sera accordée à l'algorithme RANSAC (RANdom Sample And Consensus). D'autres aspects d'étude vont être traités dans ce chapitre, où on cite essentiellement la fermeture de la boucle (loop-closing detection), et l'ajustement de faisceaux (Bundle adjustment). Il n'existe pas de solution idéale et unique pour l'OV, ainsi la solution optimale doit être choisie avec soin en fonction de l'environnement spécifique de navigation et compte tenu des ressources de calculs.

Le principe de l'algorithme SVO est illustré sur la figure III-1. La figure montre qu'en position initiale (k), le robot (drone) observe le nuage de points, $P_{k/camera\ frame}$. Lorsque ce drone se déplace en effectuant une rotation et une translation (respectivement R et t), il observera le même nuage de

³ L'egomotion est défini comme le mouvement 3D d'une caméra à l'intérieur d'un environnement. Dans le domaine de la vision par ordinateur, il se réfère à estimer le mouvement relatif de celle-ci dans une scène rigide.

points, $P_{k+1/camera\ frame}$, encore. Pour estimer le mouvement du drone (R et t), la transformation entre les deux nuages de points, $P_{k/camera\ frame}$ et $P_{k+1/camera\ frame}$, doit être déterminé.

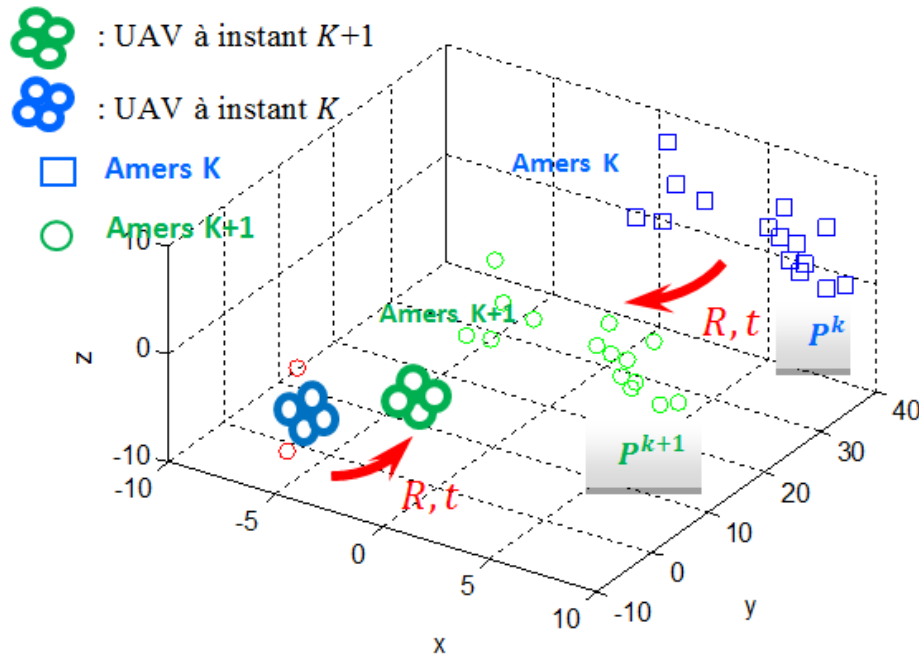


Figure III-1 . Principe de l'algorithme d'odométrie visuelle.

L'organigramme de la figure III-2 montre qu'à chaque itération, l'algorithme SVO commence par acquérir les images de gauche et de droite. Par la suite, l'algorithme SURF est appliqué pour détecter, extraire et faire correspondre les caractéristiques de l'image ; ces étapes sont répétées à l'itération suivante. La correspondance et le suivi des caractéristiques détectées des images successives permettent la triangulation 3D, la construction de cartes et l'estimation de la pose, qui sont expliquées plus en détail dans les sous-sections suivantes.

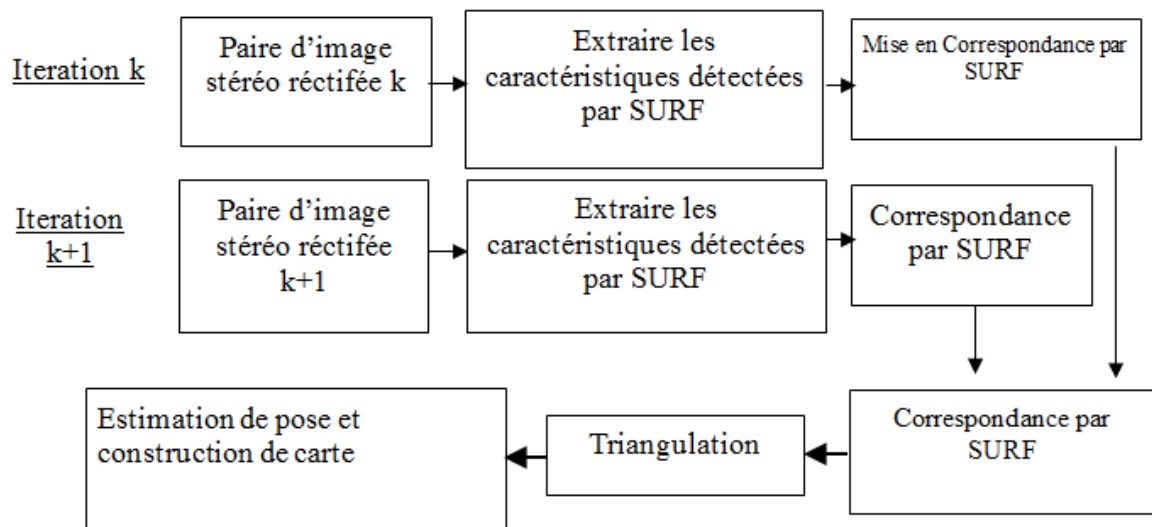


Figure III-2 . Organigramme de l'algorithme SVO.

III.2 Histoire de l'odométrie visuelle

Le problème d'estimation de la pose relative d'une caméra et la structure tridimensionnelle (3D) à partir d'un ensemble d'images (calibré ou non calibré), est connu dans la communauté de vision par ordinateur par la structure de mouvement (SFM). Ses origines remontent à des travaux tels que [77] et [78]. L'OV est un cas particulier de la SFM [76].

L'algorithme SFM est plus général et aborde le problème de la reconstruction 3D, il détermine à la fois la structure et la pose de caméra à partir d'un ensemble d'images ordonnées ou non ordonnées. La structure finale et la pose de la caméra sont généralement raffinées avec une optimisation hors ligne (tel que l'ajustement de faisceaux [79] [50]), dont le temps de calcul augmente avec le nombre d'images. Inversement, l'OV se concentre sur l'estimation du mouvement 3D de la caméra de manière séquentielle pour chaque image acquises et en temps réel. L'ajustement de faisceaux peut être utilisé pour affiner l'estimation locale de la trajectoire.

Le problème de l'estimation de l'égomotion d'un véhicule à partir de l'entrée visuelle seulement a commencé dans les années 1980 par Moravec [80]. Il est intéressant de constater que la plupart des premières recherches sur l'OV [80, 81] ont été appliquées sur les Rovers planétaires utilisés dans le programme d'exploration de Mars de la NASA.

III.2.1 Odométrie visuelle Stéréoscopique

La plupart des recherches effectuées sur l'OV ont été réalisées en utilisant des caméras stéréoscopiques s'appuyant sur le travail de Moravec [80]. Matthies et Shafer [82, 83], ont utilisé un système binoculaire renforcé par la procédure de Moravec pour détecter et suivre les coins. Olson, Matthies *et al* [81, 84] ont montré que l'utilisation d'une seule caméra pour estimer l'égomotion provoque des erreurs d'estimation qui augmente en fonction de la distance parcourue.

La figure III-3 illustre un exemple récent des deux approches de travaux de recherche utilisant des caméras stéréo, de profondeur et carte de calcul graphique Jetson GPU de la firme Nvidia [85, 86].

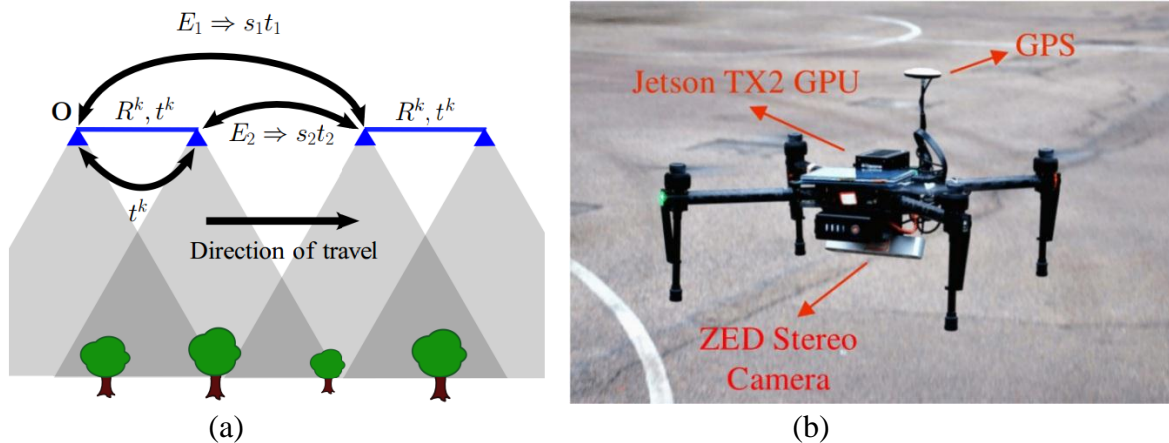


Figure III-3 : (a) Géométrie de l'initialisation d'une caméras [85], (b) une plateforme expérimentale[86].

Une approche différente d'estimation de mouvement et d'élimination des valeurs aberrantes a été proposée par Milella et Siegwart [87]. Ils ont utilisé l'approche Shi-Tomasi [62] pour la détection de coin, ils ont conservé ces points avec une grande confiance dans la carte de la disparité. L'estimation de mouvement ensuite a été résolu en utilisant d'abord la méthode des moindres carrés, puis l'algorithme itératif de plus proche (ICP) de [88]. Ce dernier est un algorithme populaire pour les enregistrements 3D à partir de balayages laser, et pour raison de robustesse, une étape d'élimination des valeurs aberrantes (fausse association) a été intégrée dans l'ICP.

Les travaux mentionnés auparavant présentent un aspect commun : la triangulation de points 3D pour chaque paire d'image stéréoscopique, et le mouvement relatif obtenue par un algorithme d'alignement de point 3D/3D. Une approche d'implémentation complètement différente a été proposée en 2004 par Nister et al [52] qui présentent la première validation de l'OV pour une longue distance en temps réel, et avec une robuste rejection des valeurs aberrantes. Le mouvement relatif calculé vu comme un problème un problème 3D/2D. Un système d'estimation de mouvement différent a été introduit après par Comport et al. [89]. Au lieu d'utiliser l'alignement de points 3D/3D ou les techniques d'estimation de la pose de caméra 3D/2D, ils ont opté pour le tenseur quadri-focal, pour calculer le mouvement à partir de la transformation des correspondances 2D/2D d'images sans faire la triangulation. L'avantage d'utiliser directement les points 2D bruts résident dans un calcul plus précis de mouvement.

III.2.2 Odométrie visuelle monoculaire

La différence avec le système stéréoscopique est que dans l'OV monoculaire, le mouvement relatif et la structure 3D doit être calculée simultanément à partir des images acquises par la caméra. Et comme l'échelle absolue est inconnue, la distance entre les deux premières poses de la

caméra est généralement fixé à un. Lorsqu'une nouvelle image arrive, l'échelle relative et la pose de la caméra sont déterminés par les deux premières images en utilisant soit la connaissance de la structure 3D ou le tenseur tri-focal [79].

III.2.3 V-SLAM

Bien que ce travail se concentre sur l'OV, il est utile de mentionner les travaux de recherches de Durrant-Whyte et Tim Bailey [45], sur le problème de localisation et cartographie simultanée visuelle (V-SLAM). Le problème de VSLAM peut être résolu de deux façons. Soit avec des méthodes de filtrage en fusionnant les informations provenant de tous les capteurs (caméras, odomètre ou IMU) [90] ou soit avec des méthodes basées sur le scan-matching (alignement des nuages de points observés à deux instants de temps successives) en utilisant des techniques d'optimisation. Cette dernière est très similaire à l'algorithme d'odométrie visuelle.

III.2.4 Odométrie Visuelle contre V-SLAM

Dans cette partie, nous analysons la relation de l'OV avec le V-SLAM. Le but de l'algorithme SLAM en général (V-SLAM en particulier) est d'obtenir une approche globale pour une estimation cohérente de la trajectoire du robot. Le V-SLAM consiste à la fois, à localiser le robot et de construire une carte d'environnement. Et lorsque le robot passe par un endroit déjà visité, (fermeture de la boucle) l'algorithme VSLAM va réduire les erreurs de la dérive de la carte et de la pose.

Inversement, l'OV vise à retrouver le chemin progressivement, pose après pose, en utilisant une technique d'optimisation. Cependant en OV la philosophie est différente, on ne s'intéresse qu'à la cohérence locale de la trajectoire et la carte locale. Tandis que le SLAM s'intéresse à la cohérence de la carte globale. L'OV peut être utilisé comme un bloc de construction pour un algorithme de SLAM pour estimer le mouvement incrémental de la caméra. Néanmoins, pour faire une méthode SLAM complète, il faut ajouter une certaine façon pour détecter la fermeture de la boucle et éventuellement une étape d'optimisation globale pour obtenir une carte métrique cohérente.

III.3 Formulation générale de l'OV

Un agent se déplace dans un environnement quelconque, et prend des images avec un système de vision fixé rigidement sur le robot volant, à des instants de temps k . Dans le cas d'un système monoculaire, l'ensemble des images prises à l'instant k est noté $I_{0:n} = \{I_0, \dots, I_n\}$. Dans le cas d'un système stéréo, il ya une image gauche et une image droite, à chaque instant, noté

$I_{l,0:n} = \{I_{l,0}, \dots, I_{l,n}\}$ et $I_{r,0:n} = \{I_{r,0}, \dots, I_{r,n}\}$, la figure III-4 montre une illustration de ce scénario.

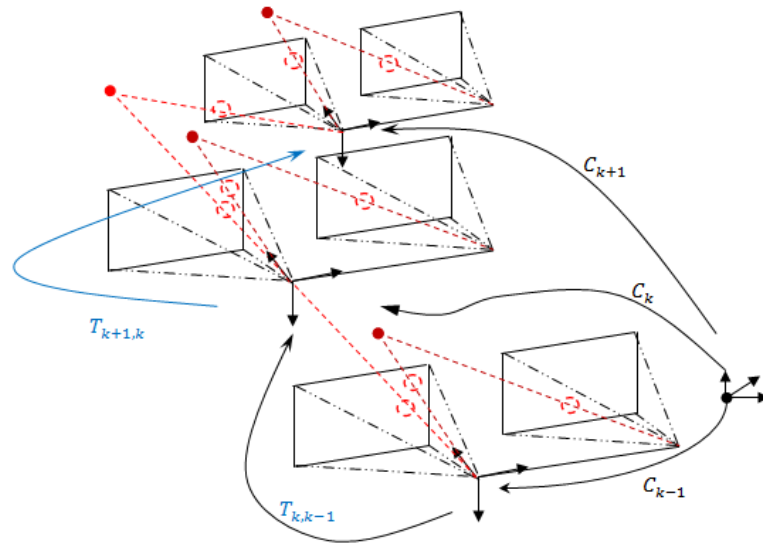


Figure III-4 . Principe de l'algorithme SVO.

Par simplicité, le repère de la caméra est supposé confondu avec le repère de quadrotor. Dans le cas d'un système stéréo, sans perte de généralité, le repère de la caméra gauche peut être utilisé comme origine. Deux positions successive du banc de caméras stéréoscopiques à des instants $k - 1$ et k sont reliées par la transformation du corps rigide $T_{k,k-1} \in \mathbb{R}^{4 \times 4}$ de la forme suivante :

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad (\text{III.1})$$

Où $R_{k,k-1} \in SO(3)$ est la matrice de rotation⁴, et $\mathbf{t}_{k,k-1} \in \mathbb{R}^{3 \times 1}$ est le vecteur de translation. L'ensemble $T_{1:n} = \{T_{1,0}, \dots, T_{n,n-1}\}$ contient tous les mouvements subséquents. Pour simplifier la notation, désormais, T_k sera utilisée à la place de $T_{k,k-1}$. Finalement, l'ensemble des poses de la caméra $C_{0:n} = \{C_0, \dots, C_n\}$ contient les transformations de la caméra par rapport aux coordonnées initiales du repère à l'instant $k = 0$. La pose courante C_n peut-être calculer en concaténant les transformations $T_k (k = 1 \dots n)$, et par conséquent, $C_n = C_{n-1}T_n$ avec C_0 étant la pose de la caméra à l'instant $k = 0$ qui peut être réglé arbitrairement.

La tâche principale de l'OV est de calculer les transformations relatives T_k à partir des images (des amers visuels) I_k et I_{k-1} et puis les concaténer, pour récupérer la trajectoire complète $C_{0:n}$ de la caméra. Cela signifie que l'OV récupère le chemin progressivement, pose après pose. Une technique de d'optimisation robuste et rapide doit être utilisée après cette étape, pour obtenir une estimation plus précise de la trajectoire locale.

⁴ Le groupe orthogonal $SO(3)$ est souvent appelé groupe des rotations (vectorielles) dans l'espace (tridimensionnel).

III.4 Extraction de points d'intérêt

Un point d'intérêt [91] est un point qui doit avoir des caractéristiques plus significatives que les autres. Ces points correspondent à des changements bidimensionnels du signal. Nous pouvons en citer, les coins en 'L' et les points de jonctions en 'T' (figure III.5). Plusieurs travaux ont été menés pour l'extraction de ces points dans l'image, en développant de tels détecteurs, il est bon de préciser que leur champ d'application n'est pas limité. En effet, on les retrouve dans bien des domaines de la vision par ordinateur, que ce soit en reconnaissance, en analyse des formes et du mouvement ou encore dans la reconstruction 3D d'une scène. C'est l'analyse du mouvement qui nous intéresse particulièrement. Les points ont plusieurs avantages [92], parmi lesquels :

1. Sources d'informations plus stables que les contours car plus de contraintes sur la fonction d'intensité.
2. Robuste aux occultations (soit occulté complètement, soit visible).
3. Présents dans une grande majorité d'images.

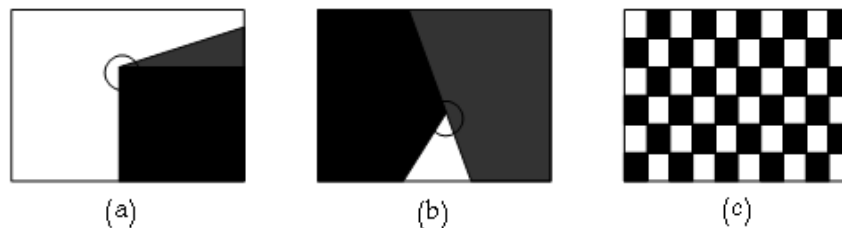


Figure III-5 : Points d'intérêt : (a) coins, (b) jonctions en T et (c) forte variation d'intensité.

III.4.1 Détecteurs de points d'intérêt

L'algorithme d'odométrie visuelle est principalement basé sur la précision de la mise en correspondance des caractéristiques dans les images successives. Les caractéristiques de mise en correspondance caractérisent une partie d'une image à l'aide d'un descripteur unique [93]. De nombreuses études ont été menées sur l'extraction des points caractéristiques d'une image. A cet effet, l'algorithme le plus connu est le détecteur de coin de Harris [94]. En [95], Lowe a proposé un algorithme de transformation de caractéristiques invariants à l'échelle (SIFT) pour l'extraction et la correspondance de caractéristiques.

La technique SURF, qui est une approximation du SIFT, est plus rapide que le SIFT sans réduire la qualité des points détectés [96]. Une autre alternative pour SIFT est les caractéristiques élémentaires indépendantes robustes binaires (BRIEF)[97], qui sont moins complexes que SIFT mais ont des performances de correspondance similaires [98]. Rublee et al. ont proposé BRIEF orienté FAST et tourné comme substitut des méthodes SIFT et SURF [99]. Récemment, une

analyse du temps de calcul et une comparaison des performances de ces méthodes ont été menées dans plusieurs études [100].

Dans cette étude, le détecteur SURF est utilisé, car il est robuste et rapide. L'algorithme SURF se rapproche de la matrice de Hesse en utilisant l'intégrale d'une image, ce qui réduit le temps de calcul et maintient une précision acceptable. Par la suite, la matrice Hessienne est ensuite utilisée pour sélectionner les caractéristiques significatives. Dans ce cas, l'information qui caractérise la région environnante des caractéristiques est représentée par un vecteur descripteur.

A partir de l'étude des détecteurs existants, nous pouvons conclure que les détecteurs basés sur la matrice Hessienne sont les plus stables et donnent des résultats reproductibles plus que ceux basés sur le détecteur d'Harris. D'ailleurs, l'utilisation du déterminant de l'Hessienne est préférée à sa trace (le Laplacien). Egalement les approximations comme la Différence de Gauss (DoG) proposées par Lowe [101], elles peuvent améliorer le temps de calcul au détriment de la précision.

III.4.1.1 Détecteur d'Harris

La méthode d'Harris [56] a été choisie pour la détection des points d'intérêt. Ce détecteur de coins se base sur une matrice de structure liée à la fonction d'auto corrélation calculée à partir des gradients de l'image à traiter.

$$M = w \otimes \begin{bmatrix} \left(\frac{dI}{dx}\right)^2 & \frac{dI}{dx} \frac{dI}{dy} \\ \frac{dI}{dx} \frac{dI}{dy} & \left(\frac{dI}{dy}\right)^2 \end{bmatrix} \quad (\text{III.2})$$

Pour une meilleure stabilité du détecteur de Harris Stephens, une version dite précise utilise pour le calcul des dérivées de l'image une fonction de lissage gaussienne. Le but de ce lissage est d'éliminer les contours pour ne garder que les coins qui représentent les réels points d'intérêt.

III.4.1.2 SURF (Speeded Up Robust Features)

Un point d'intérêt est un point qui caractérise de façon unique une partie de l'image. Plus précisément, il s'agit d'un point localisé finement, augmenté de son voisinage. Pour être utile, ce point doit être très bien identifié à l'aide d'un descripteur unique et il doit être possible de le retrouver facilement. Il existe plusieurs méthodes de détection/description de points d'intérêt (SIFT «Scale Invariant Feature Transform», SURF« Speeded Up Robust Feature », Harris...etc.) [33]. Nous avons choisi la méthode SURF pour sa robustesse, son efficacité, et son invariance à l'échelle, la rotation, l'illumination et les transformations de type affine. La méthode SURF est à la fois un détecteur et un descripteur de points d'intérêts.

Les résultats de la détection de points d'intérêt sur des images réelles prises sont représentés par les figures III.6, III.7, III.8 et III.9 :

La figure III-9 montre un exemple réel qui a utilisé la méthode SURF pour l'extraction de caractéristiques à partir d'images stéréoscopiques (gauche et droite) à des itérations successives (k et $k+1$) .

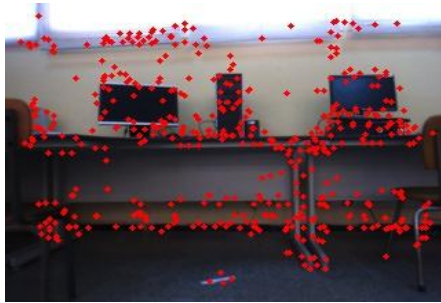


Image gauche



Image droite

Figure III-6 : Résultats de détection des points d'intérêt.

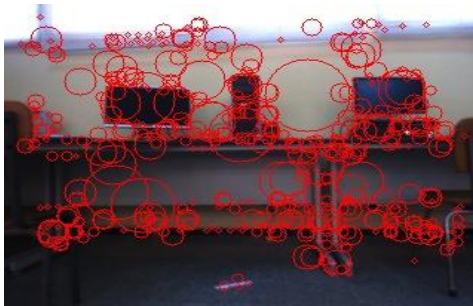


Image gauche

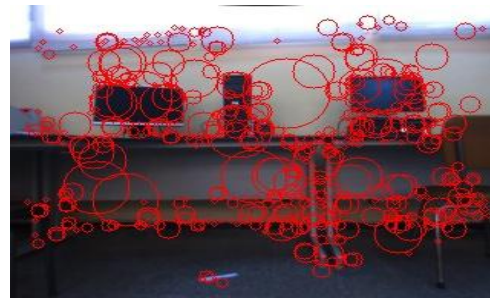


Image droite

Figure III-7 : Résultats des descripteurs des points d'intérêt.

Dans la figure III.11 les cercles représentent les normes des descripteurs.

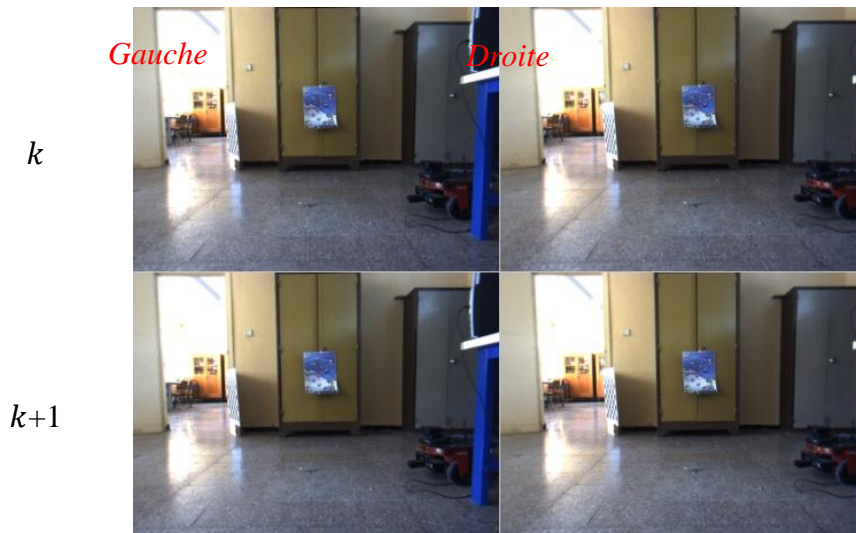


Figure III-8 : Images rectifiées acquises.

Détection des points d'intérêt à l'aide du détecteur SURF est présenté dans la figure (III.9).

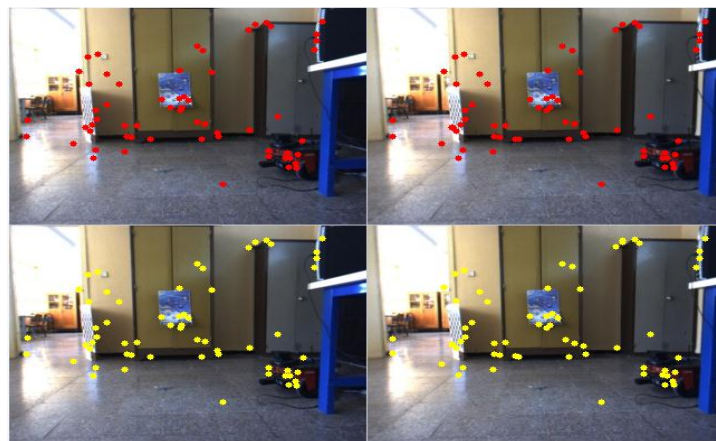


Figure III-9 : détection des points d'intérêt.

III.5 Mise en correspondance (matching)

En vision par ordinateur, le problème d'appariement ou de mise en correspondance d'images numériques consiste à déterminer un ensemble de couples de points, où chaque couple étant formé d'un point d'une image et d'un point de l'autre image. Les points d'images sont les projections de points physiques de la scène. L'enjeu de l'appariement est d'identifier correctement dans chacune des deux images la projection du même point physique. Cette mise en correspondance est d'une grande importance pour les étapes suivantes de reconstructions 3D et de mesure des déplacements (de l'estime de la position). Une mise en correspondance imprécise peut engendrer des erreurs importantes dans les reconstructions 3D, et donc dans les mesures.

Plusieurs méthodes d'appariement sont présentées dans la littérature. Elles se différencient

par le choix des points caractéristiques présents dans la scène observée et qui seront appariés dans les images. Le choix de la méthode repose principalement sur la nature de la scène photographiée et sur la détection de points caractéristiques dans les images. Dans le cas d'images de scènes polyédriques ou des scènes d'extérieurs telles que des rues ou des bâtiments, l'appariement est généralement basé sur la détection de coins, de jonctions ou de contours caractéristiques. L'étape d'appariement des images est alors basée sur l'extraction de ces points de références.

III.5.1 Mise en correspondance par SURF

Pour chaque nouvelle paire d'images, un nouveau nuage visuel est produit et mis en correspondance avec les points de nuage existants. Cette correspondance s'effectue en plusieurs étapes. Tout d'abord, chaque point 3D du nuage visuel précédent est associé à son point correspondant dans le nouveau nuage visuel à l'aide du descripteur SURF (la distance euclidienne entre les deux descripteurs est utilisée pour vérifier cette association). Deuxièmement, les nuages associés sont utilisés pour estimer une pose initiale, (R_0, t_0) , entre les nuages visuels précédents et nouveaux en utilisant l'approche des moindres carrés [102].

Dans la troisième étape, la correspondance visuelle des nuages est affinée en rejetant les valeurs aberrantes à l'aide de l'algorithme itératif du point le plus proche (ICP) pour calculer une pose précise (R et t) entre deux scans. Pour estimer R et t , l'algorithme ICP minimise un critère d'optimisation [103].

La figure III-14 montre un exemple réel d'appariement de caractéristiques dans des images stéréoscopiques (gauche et droite) en itérations successives (k et $k+1$).

Cette étape consiste à déterminer une relation entre les points détectés dans l'image droite et leurs points correspondants de l'image gauche. Cette relation doit être robuste face au changement d'échelle, conditions d'acquisition d'images, rotation, translation...etc.

Pour cela nous avons choisi de faire l'association par SURF par ce qu'elle est robuste face au changement d'échelle, la rotation...etc. L'association des points d'intérêt est basée sur la comparaison entre leurs vecteurs descripteurs. Son principe est de comparer la distance (euclidienne, mahalanobis...etc.) [104] avec un seuil, on associe les deux points où la distance est inférieure au seuil. La variation de seuil nous permet d'éliminer les fausses matching. Pour notre cas nous avons opté pour la distance euclidienne et l'algorithme implémenté est définie comme suit :

- n_g : nombre de points détectés dans l'image gauche ;

- n_d : nombre de points détectés dans l'image droite ;
- $d(i, j)$: distance euclidienne entre le descripteur du $i^{\text{ème}}$ point d'intérêt de l'image gauche et le $j^{\text{ème}}$ point d'intérêt de l'image droite
- S : un seuil pour associer les points d'intérêt (deux points i et j sont associés si et seulement si $d(i, j) < S$)

Le tableau III.1 montre le nombre de points matchés aux différents seuils, et avec une valeur Hessienne de 64 éléments[105], c.-à-d. seulement les points d'intérêts ayant une valeur Hessienne plus grande sont extraits, ce nombre diminue avec la diminution de la distance descripteur.

seuil	1	0.8	0.6	0.5	0.4	0.2
Nombre de points matchés	451	259	181	139	102	29

Tableau III.1: Résultats de mise en correspondance.

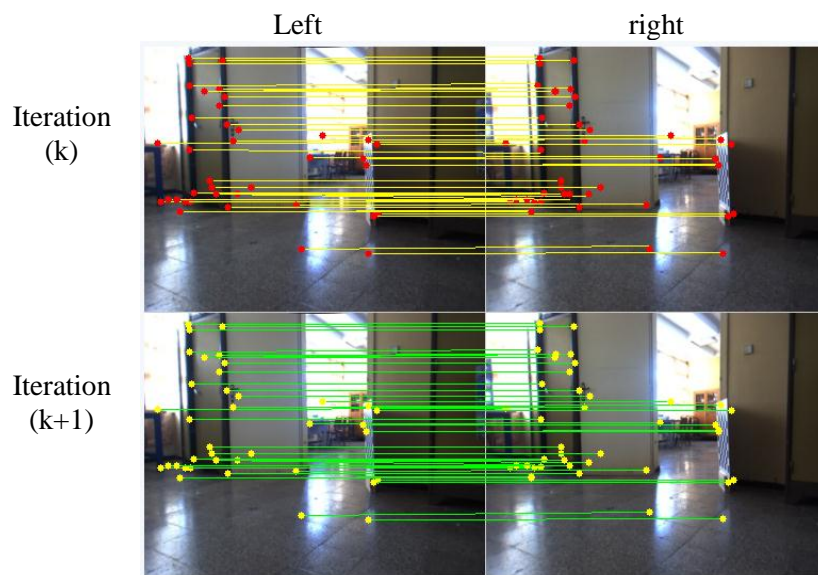


Figure III-10 : Mise en correspondance gauche/droite à l'instant k et $k + 1$ des points d'intérêt.

III.5.2 Suivi de points d'intérêt (tracking)

Une fois les points détectés dans les deux images, il faut les apparier, c'est-à-dire mettre en correspondance chaque point situé au même endroit de la scène dans les deux images, en vue de calculer la transformation qui permet de passer d'un ensemble de points à l'autre (changement de point de vue). Pour apparier deux points, on peut travailler sur l'intensité des pixels du voisinage ou sur des descripteurs locaux plus complexes utilisant des dérivées premières et parfois seconde

voire plus [58]. Le suivi de points ou "tracking" est une méthode d'estimation du déplacement entre deux images successives à partir d'un modèle de mouvement et en présupposant qu'un point doit se retrouver dans une fenêtre de taille donnée (w_x, w_y), dans l'image suivante. Le principe est illustré dans la figure (III.11). On rappelle que f_i^{t-1} est le point de l'image I^{t-1} , f_i^t le point correspondant dans l'image I^t et d la translation liant I^{t-1} et I^t .

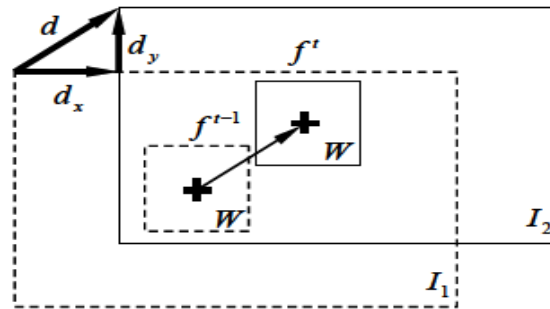


Figure III-11 : Principe du suivi de points.

Ensuite on fait une deuxième mise en correspondance entre les points d'intérêt de l'instant k avec ceux de $k + 1$ (matching vertical) en ne gardant que les points qui se trouvent dans les quatre images (voir figure III.12).

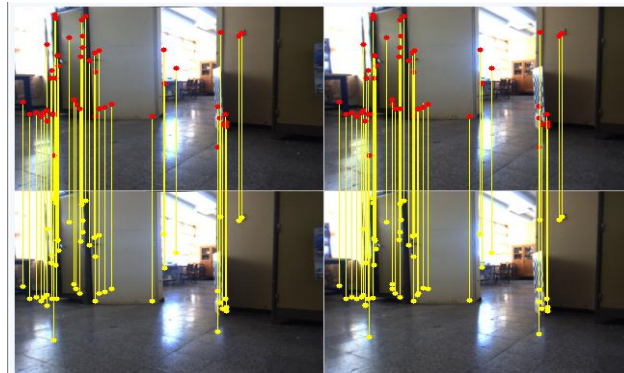


Figure III-12 : Mise en correspondance droite/droite, gauche/gauche de l'instant k à $k + 1$ des points d'intérêt.

III.5.2.1 Tracking par SURF

L'association des points d'intérêt est basée sur la comparaison entre leurs vecteurs descripteurs (Section II.4.3.3) La distance entre les descripteurs est ensuite calculée. Si cette distance est inférieure à un certain seuil, les deux points d'intérêt sont associés. Plusieurs distances peuvent être utilisées ; la distance euclidienne, la distance de Mahalanobis et la norme infinie. Dans notre travail nous utilisons la distance euclidienne pour des raisons de simplicité. La figure (III.14) montre le tracking des points d'intérêt entre la première image et la vingtième image.

Algorithme : Technique d'association les descripteurs

```

Entrée : Deux ensembles de descripteurs à associer
Sortie : vecteur d'indice des descripteurs appariés
Début
  Pour i de 1 à nbre de descripteur de l'image 1
    Pour j de 1 à nbre de descripteur de l'image 2
      Calcule la distance d entre les descripteurs
      Si  $d > \text{seuil}$ 
        Vect_ind=[ Vect_ind i j]
      Fin Si
    Fin pour
  Fin pour
  
```

Figure III-13 : Pseudo algorithme d'association des points d'intérêt par SURF[106].



Figure III-14 : Suivi des points d'intérêt par la méthode du SURF.

L'avantage de l'algorithme KLT réside dans la rapidité ; le temps de calcul est très réduit à cause de la réduction de l'espace de recherche (seulement une fenêtre), de plus cette méthode est connue par sa robustesse face au changement d'échelle (la pyramide d'échelle). Cependant cette technique exige une cadence d'environ vingt-cinq (25) images par seconde pour assurer un bon tracking. L'association des points d'intérêt détectés par le SURF entre deux images est basée sur la distance entre leurs descripteurs, ces derniers se distinguent par leurs robustesses face aux changements de luminance et d'échelle, la taille du descripteur influe directement sur la robustesse et le temps de calcul.

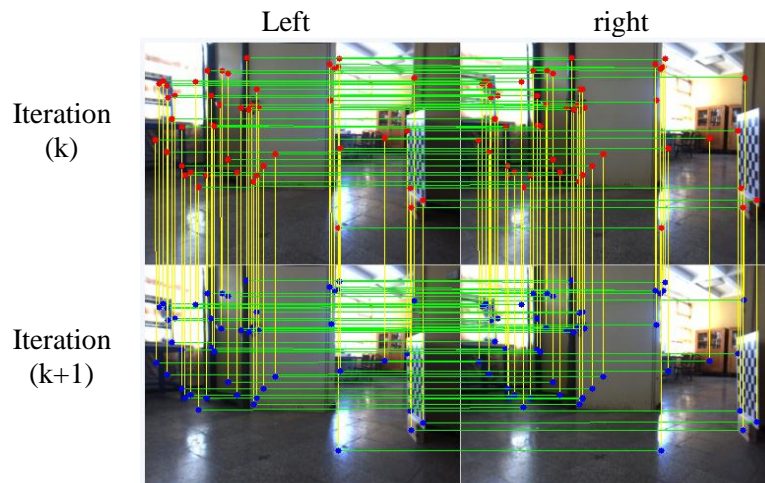


Figure III-15 : mise en correspondance des points d'intérêt horizontal et vertical.

Enfin une mise en correspondance entre les ensembles de vues successives gauche/droite de l'instant k avec ceux de $k + 1$ (voir figure III.19).

Dans la fig. III-19, les projections, P_l et P_r , du point P sont donnés par les coordonnées (u_l, v_l, f_l) et (u_r, v_r, f_r) , respectivement, ils sont considérés sur les images de gauche et de droite, où:

$f = f_d = f_g$ désigne la distance focale; d est la disparité, où $d = u_r - u_l$; et b est la ligne de base. De plus, parce que les images stéréo sont rectifiées, $v_l = v_r$.

Notez que la distance focale ne peut pas être explicitement calculée. Les coordonnées des deux caméras sont saisies sans normaliser les dimensions par Z .

Le calcul des coordonnées 3D des points d'intérêt est une étape cruciale dans l'algorithme d'odométrie visuelle, et une erreur induite par cette étape conduit à une grande erreur dans les résultats finaux

III.6 Estimation du Mouvement

L'estimation du mouvement est une étape primordiale. En effet, le mouvement de la caméra est calculé entre deux observations successives (f_{k-1} et f_k). Par concaténation de tous les mouvements, on peut rétablir la trajectoire complète de la caméra (du robot mobile) (on suppose que la caméra est montée de façon rigide).

Selon les types de correspondances des points caractéristiques en 2D ou 3D, il existe trois méthodes différentes [76],[107].

- 2D/2D: Dans ce cas, les deux f_{k-1} et f_k sont spécifiés en coordonnée image 2D (en pixel).

- 3D/3D: Dans ce cas, les deux f_{k-1} et f_k sont précisées en 3D (métrique). Pour ce faire, il faut passer par une étape de triangulation à chaque instant, par exemple, en utilisant un système de stéréovision.

- 3D/2D: Dans ce cas f_{k-1} sont précisées en 3D et f_k sont leurs correspondants en 2D. Dans le cas monoculaire, la structure 3D a besoin d'être triangulé à partir de deux vues de caméra adjacentes (par exemple, I_{k-2} et I_{k-1}), puis apparier les caractéristiques 2D de l'image dans un troisième point de vue (par exemple I_k). Dans le schéma monoculaire, l'appariement nécessite au moins trois points de vue.

On note que les primitives (ou caractéristiques) peuvent être des points ou des lignes. En général, en raison de l'absence de lignes dans des scènes non structurées, les entités ponctuelles (points d'intérêts) sont utilisées en OV. Ces trois techniques ont été bien détaillé dans [107].

- 3D/2D: Le mouvement à partir de la structure 3D et des correspondances des primitives.

On se réfère sur le travail de Nister et al. [52]. La transformation T_k est calculée à partir des correspondances 3D/2D de X_{k-1} et de p_k . X_{k-1} peut être estimée à partir de données stéréo ou, dans le cas monoculaire, à partir la triangulation des mesures d'image p_{k-1} et p_{k-2} . Cependant ce dernier, nécessite des correspondances d'image à travers trois points de vue. La formulation générale dans ce cas est de trouver T_k qui minimise l'erreur de reprojction l'image.

$$\arg \min_{T_k} \sum_i \|p_k^i - \hat{p}_{k-1}^i\|^2, \quad (\text{III.3})$$

Où \hat{p}_{k-1}^i est la reprojction de point 3D de X_{k-1}^i dans l'image I_k par la transformation T_k . Ce problème est connu par *la perspective de n points* (abréviation *PnP*) [59].

Dans le cas 3D/2D, le P3P est une méthode standard pour une estimation de mouvement robuste en présence de valeurs aberrantes [59]. Une solution simple et directe au problème *PnP* pour $n \geq 6$ de points est l'algorithme de transformation linéaire directe [79]. Une correspondance de point 3D/2D fournit deux contraintes de la forme suivante pour les entrées de $P_k = [R|t]$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -x & -y & -z & -1 & x\tilde{v} & y\tilde{v} & z\tilde{v} & \tilde{v} \\ x & y & z & 1 & 0 & 0 & 0 & 0 & -x\tilde{u} & -y\tilde{u} & -z\tilde{u} & \tilde{u} \end{bmatrix} \begin{bmatrix} P^1 \\ P^2 \\ P^3 \end{bmatrix} = 0 \quad (\text{III.4})$$

Où chaque vecteur de quatre P_j^T (le $j^{\text{ème}}$ ligne de P_k) et x, y, z sont les coordonnées des points 3D de X_{k-1} . Remplir les contraintes de correspondances de six-points donne un système linéaire d'équations de la forme $AP = 0$. Les entrées de P peut être calculée à partir du vecteur nul de A,

par exemple, de la SVD. Les parties rotation et translation peuvent facilement être extraites de $P_k = [R|t]$.

L'estimation de mouvement 3D/2D suppose que les points de l'image 2D arrivent d'une seule caméra. Cela signifie que dans le cas d'une caméra stéréo, les points de l'image 2D sont ceux de la caméra gauche ou de la caméra droite. Il est souhaitable d'utiliser les points 2D des deux caméras en même temps. Une solution généralisée de l'algorithme d'estimation de mouvement 3D/2D (par exemple, les points 2D d'image à partir de plusieurs caméras) a été proposée par Nister dans [108] pour les caméras calibrées (par exemple, la position et l'orientation mutuelle entre les caméras est connu).

Dans le cas monoculaire, il est nécessaire de trianguler les points 3D et estimer la pose à partir des appariements (*matches*) 3D/2D en alternance. Ce schéma alternatif est souvent désigné comme la SFM. Commencant par deux vues, l'ensemble initial de points 3D et la première transformation sont calculés à partir des appariements des primitives 2D/2D. Les transformations consécutives sont alors calculées à partir des appariements des primitives 3D/2D. Pour ce faire, les primitives (points d'intérêts) doivent être appariées (ou suivis) sur plusieurs images (au moins trois). Une nouvelle primitive 3D (points d'intérêts) est à nouveau triangulée lorsqu'une nouvelle transformation est calculée et ajoutée à l'ensemble des points d'intérêts 3D.

Le défi principal de cette méthode est de maintenir un ensemble cohérent et précis des primitives 3D triangulées, et de créer des appariements 3D/2D (*matches*) des primitives pendant au moins trois images adjacentes.

Dans notre travail on a opté d'utiliser l'approche 3D/3D car elle est simple et efficace et elle permet non seulement d'estimer la pose du robot mais aussi la carte 3D de l'environnement.

III.6.1 Principe d'odométrie visuelle 3D/3D

L'odométrie visuelle est une technique utilisée pour la détermination de la position et l'orientation d'un mobile dans un repère globale par estimation de mouvement à partir d'images obtenues par une ou plusieurs caméras portées par ce dernier. Cette estimation se fait par reconstruction incrémentale.

Le principe de fonctionnement de l'odomètre visuel repose sur l'estimation du déplacement de la caméra entre deux positions, ou bien entre deux vues différentes qui engendrent deux images différentes, dans lesquelles, une trame de points spécifiques connus sous le nom de points d'intérêt est généralement suivie. Les points d'intérêt connus aussi sous le nom d'amers sont des points qui

doivent avoir des caractéristiques plus significatives que les autres points de l'image et qui correspondent à des changements bidimensionnels [109] et ils sont représentés par un descripteur. C'est un vecteur qui représente l'information caractéristique du point d'intérêt ou une région qui l'entoure.

Dans un premier temps, des points d'intérêts prépondérants sont sélectionnés dans les deux images. Ensuite, une mise en correspondance entre les deux ensembles de points (ensemble des points d'intérêt de la première image et ensemble des points d'intérêt de la deuxième image) est nécessaire afin d'apparier les deux images. La dernière étape consiste à estimer les paramètres de déplacement de la caméra grâce aux suivis des points d'intérêt dans une séquence d'images.

Soient P_k et P_{k+1} des ensembles de points 3D appartenant à P_k and P_{k+1} associés aux caractéristiques extraites par la méthode SURF, à partir de deux images acquies en deux étapes consécutives ; donc, pour chaque pas de temps, $k = 1, 2, 3, \dots, N$. Chaque point, $P_k^i \in \mathfrak{R}^{3 \times 1}$, correspond au point, $P_{k+1}^i \in \mathfrak{R}^{3 \times 1}$, au pas de temps suivant comme suit :

$$P_{k+1}^i = R P_k^i + t \quad (\text{III.5})$$

Par la suite, l'algorithme d'odométrie visuelle utilise ces points 3D estimés pour déterminer l'orientation (R) et la translation (t) du robot en minimisant l'erreur quadratique moyenne (MSE) entre deux nuages de points successifs (équation (31)). Ce critère est non linéaire et couplé du fait de la matrice de rotation.

$$\underset{R, t}{\operatorname{arg\,min}} F(R, t) = \frac{1}{N} \sum_{i=1}^N \left\| (R P_{k+1}^i + t) - P_k^i \right\|^2 \quad (\text{III.6})$$

The pose estimation criterion can be formulated by minimizing the MSE of the defined distance between the reconstructed 3D points obtained from two successive iterations.

Le critère à minimiser se réécrit comme suit:

$$F(R, t) = \frac{1}{N} \sum_{k=1}^N \left[(X_k^i - X_k')^2 + (Y_k^i - Y_k')^2 + (Z_k^i - Z_k')^2 \right] \quad (\text{III.7})$$

where (X_k, Y_k, Z_k) and (X_k', Y_k', Z_k') are the coordinates of the reference and transformed scans in the camera frame, respectively.

où (X_k, Y_k, Z_k) et (X_k', Y_k', Z_k') sont les coordonnées des scans de référence et transformés dans le cadre de la caméra, respectivement.

L'algorithme ICP, introduit par et Medioni [110] et McKay [88], est la technique d'enregistrement (alignement de balayage) la plus couramment utilisée en raison principalement de

sa simplicité et de son efficacité. L'algorithme de base est le suivant : Soit \mathbf{P}_{k+1}^i le nouveau balayage (c'est-à-dire le nuage de points) acquis à l'instant $k+1$, et \mathbf{P}_k^i le balayage de référence à l'instant k . L'objectif de l'algorithme est de trouver une transformation géométrique pour aligner le nouveau scan avec le scan de référence ; cette transformation est composée d'une translation et d'une rotation. Dans l'algorithme proposé ici, l'algorithme ICP est combiné avec le descripteur SURF. Dans ce cas, l'estimation de la pose se compose de deux étapes. La première étape consiste à trouver la correspondance entre les deux scans et à minimiser le MSE [111]. Dans cette étape, l'estimation de pose initiale $\mathbf{R}_0, \mathbf{t}_0$ est fournie par le descripteur SURF. L'utilisation du descripteur SURF pour l'association de points diminue considérablement le temps de calcul de l'algorithme ICP, et la pose estimée satisfait rapidement le critère de convergence.

III.6.2 OVS adaptative à base d'ICP (Iterative Closest Points)

Pour augmenter la précision de l'algorithme d'odométrie visuelle proposé, la pondération inverse des profondeurs des points (distance caméra/point) est prise en compte en plus de la correspondance et du filtrage des points SURF. L'analyse de précision des capteurs de vision stéréo a montré que la précision d'un point reconstruit est proportionnelle à sa profondeur inverse.

En conséquence, les critères ICP adaptatifs utilisant le point SURF sont obtenus comme suit ::

$$F(R, t) = \frac{1}{N} \sum_{k=1}^N (1/d_i^2) [(X_k^i - X_k')^2 + (Y_k^i - Y_k')^2 + (Z_k^i - Z_k')^2] \quad (\text{III.8})$$

Où d_i désigne la profondeur de la i^{th} feature. Equation (24) est résolue par trois approches : ICP, décomposition en valeur singulière (SVD) et algorithme de quaternion [16, 102, 112].

De plus, pour minimiser l'Eq. (24), un algorithme SURF-ICP adaptatif est implémenté. La présence d'outliers entre les deux nuages de points peut ralentir la convergence de l'algorithme d'estimation de (R, t) . Pour éviter ce problème, un consensus d'échantillon aléatoire (RANSAC) et des filtres de Zhang [103, 113, 114] sont utilisés pour le rejet des valeurs aberrantes.

Mise en correspondance entre les points d'intérêt détecté dans l'image gauche et l'image droite de chaque instant k et $k + 1$ (matching horizontal).

III.6.2.1 Algorithme ICP

L'algorithme ICP (Itérative Closest Points) a été introduit par Chen et Medioni [110] et Besl et McKay [115] ; c'est la méthode courante de recalage entre deux nuages de points 3D. Elle

nécessite de fournir une estimée initiale de la transformation rigide entre ces deux nuages (figure III.21) et l'affiné itérativement en minimisant la somme des distances au carré entre un point et son plus proche voisin dans l'autre modèle. De nombreuses variantes ont été proposées afin d'améliorer la convergence de l'algorithme telle que l'initialisation...etc. Les résultats de l'algorithme ICP dépendent fortement de la qualité de l'estimée initiale fournie. Celle-ci peut être déterminée en estimant la position du robot par un autre capteur tel que l'Odomètre dans notre cas.

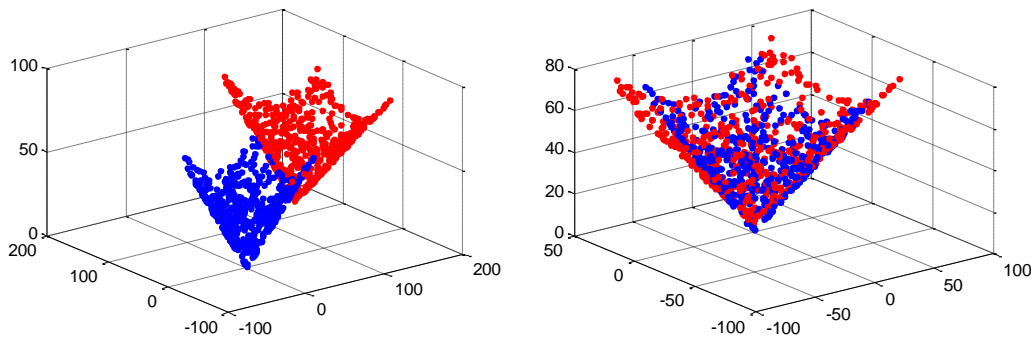


Figure III-16 : Recalage de deux nuages de points.

L'exploitation des données inertielles (gyromètre) permet d'obtenir une estimée de l'orientation R_0 du robot au moment de la prise de vue et donc de la rotation à appliquer au nuage de points 3D à recaler. Dans ce cas, une paire de points 3D suffit pour estimer la composante translation de la transformation rigide entre les deux points de vue. En exploitant un plus grand nombre de paires, il est possible aussi de déterminer à nouveau l'orientation afin de la rendre plus précise et robuste face aux dérives dues aux mesures inertielles.

Les étapes principales de l'algorithme ICP sont les suivantes :

- Association des points par le critère du plus proche voisin.
- Estimation des paramètres de transformation en utilisant une fonction de coût quadratique moyenne.
- Transformation des points en utilisant les paramètres estimés.
- Calculer l'erreur globale aux moindres carrés .si cette erreur est inférieure à un seuil, alors arrêter, sinon réitérer.

III.6.2.2 Association des points

La convergence de l'algorithme ICP dépend beaucoup de la qualité des associations utilisées. En effet, la présence de fausses associations, dans le meilleur des cas, ralentit la convergence de l'algorithme et peut au pire des cas causer sa divergence. Une méthode de mise en correspondance

robuste est donc indispensable. La distance géométrique euclidienne "classique" ne peut pas être suffisante pour établir des appariements relativement corrects et nécessaires à la convergence d'ICP. Ainsi, d'autres critères peuvent être utilisés. Il est possible dans notre cas de combiner des caractéristiques géométriques telles que la distance euclidienne et l'association des points par les descripteurs SURF à partir des deux paires d'images gauche et droite de première capture et celles de deuxième capture avant la reconstruction des nuages 3D. Dans ce cas, les paires de points sont classées une première fois selon leur distance et leur couleur qui ne doit pas dépasser un certain seuil, ensuite parmi ces points qu'on appelle compatibles, on calcule les points les plus proches en utilisant une distance géométrique.

L'algorithme de mise en correspondance de deux nuages de points 2D en utilisant les descripteurs SURF. Dans ce paragraphe nous allons présenter seulement l'algorithme d'association géométrique qui est basé sur la distance euclidienne. Son principe est le suivant : déterminer pour chaque point de nuage de points 3D à l'instant $k + 1$ (P^{k+1}) son correspondant dans le nuage de points 3D à l'instant (P^k). Pour cela nous calculons d'abord la distance entre chaque point P_i^{k+1} et tous les points P^k , ensuite le point qui présente la distance minimale correspond à l'associé de P_i^{k+1} . La Figure III.22 présente le code d'association par la distance euclidienne.

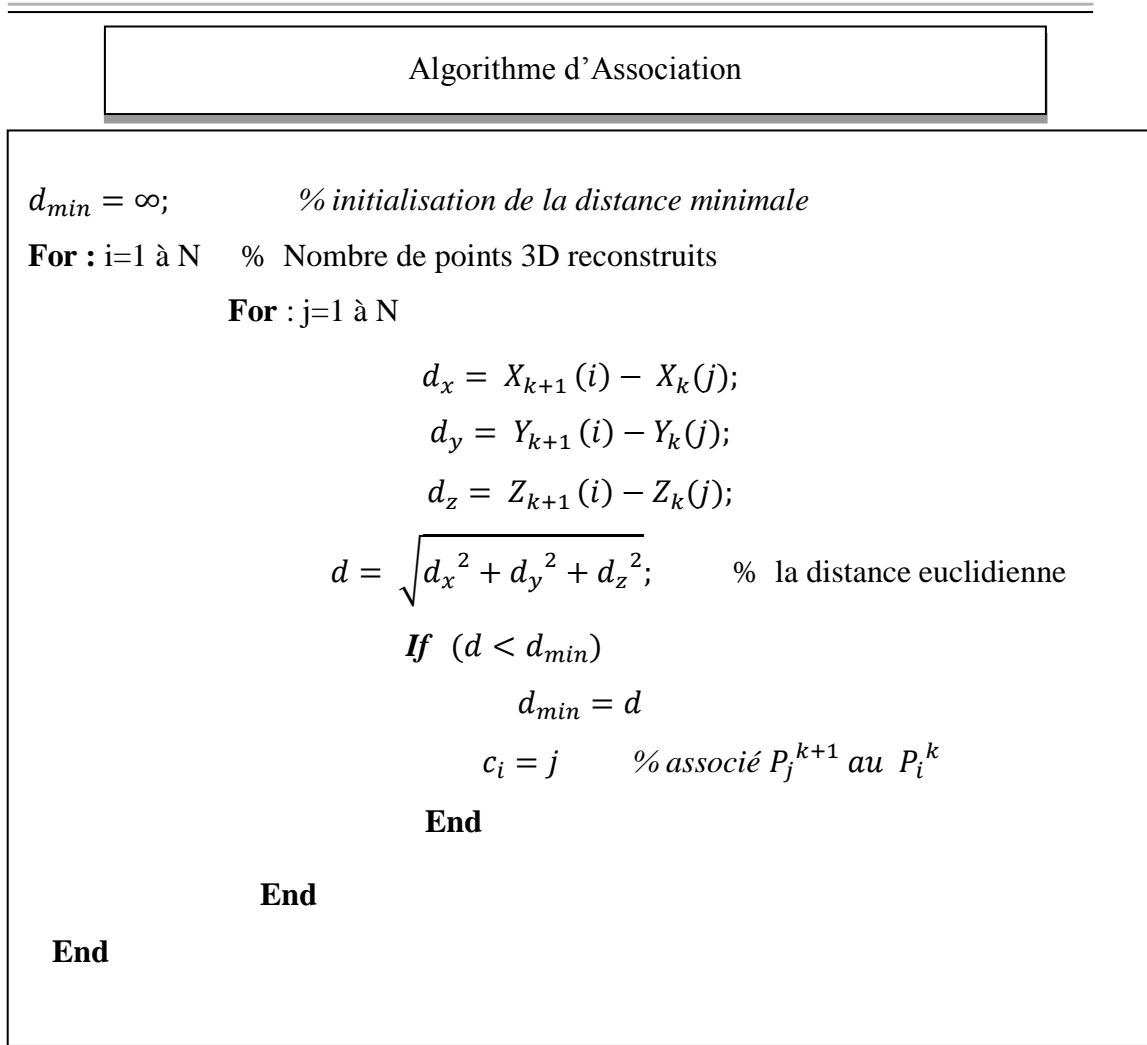


Figure III-17 : Le code d'association par la distance euclidienne.

III.6.2.3 Estimation des paramètres de la transformation

Dans l'étape d'estimation de la matrice de transformation dans l'algorithme ICP (Itérative Closest Points) nous allons utiliser le système d'équations (III.15) voir figure (III.23).

$$e = \frac{1}{N} \sum_{i=1}^N [(X_i^k - X'^k)^2 + (Y_i^k - Y'^k)^2 + (Z_i^k - Z'^k)^2] \quad (\text{III.9})$$

Les principes de l'algorithme ICP sont assez simples, mais l'implémentation pour une application temps réel peut être délicate du fait notamment que le temps de convergence ne peut pas être maîtrisé. C'est notamment l'étape de recherche du plus proche voisin qui rend l'algorithme plus lent. Le plus souvent on utilise pour cette tâche une organisation des données sous une forme particulière nommée kd-tree [33].

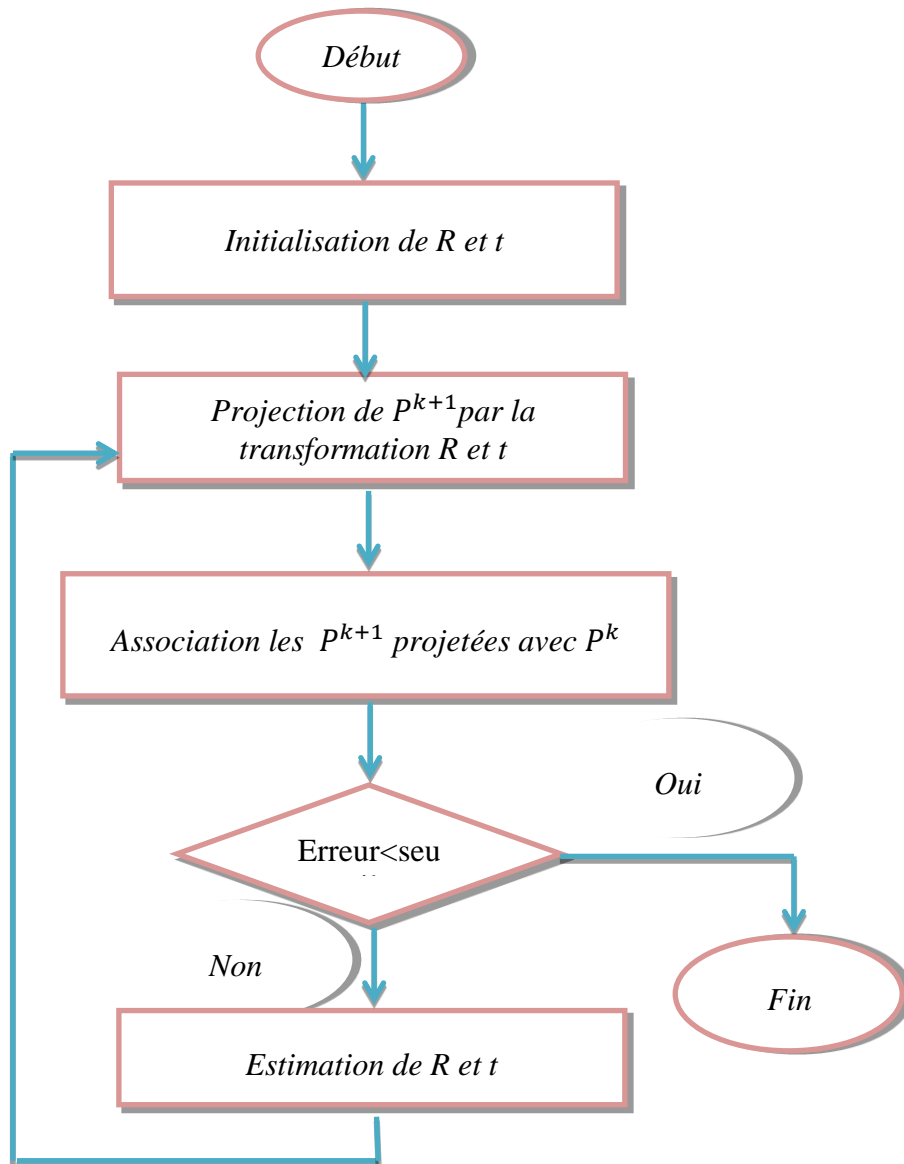


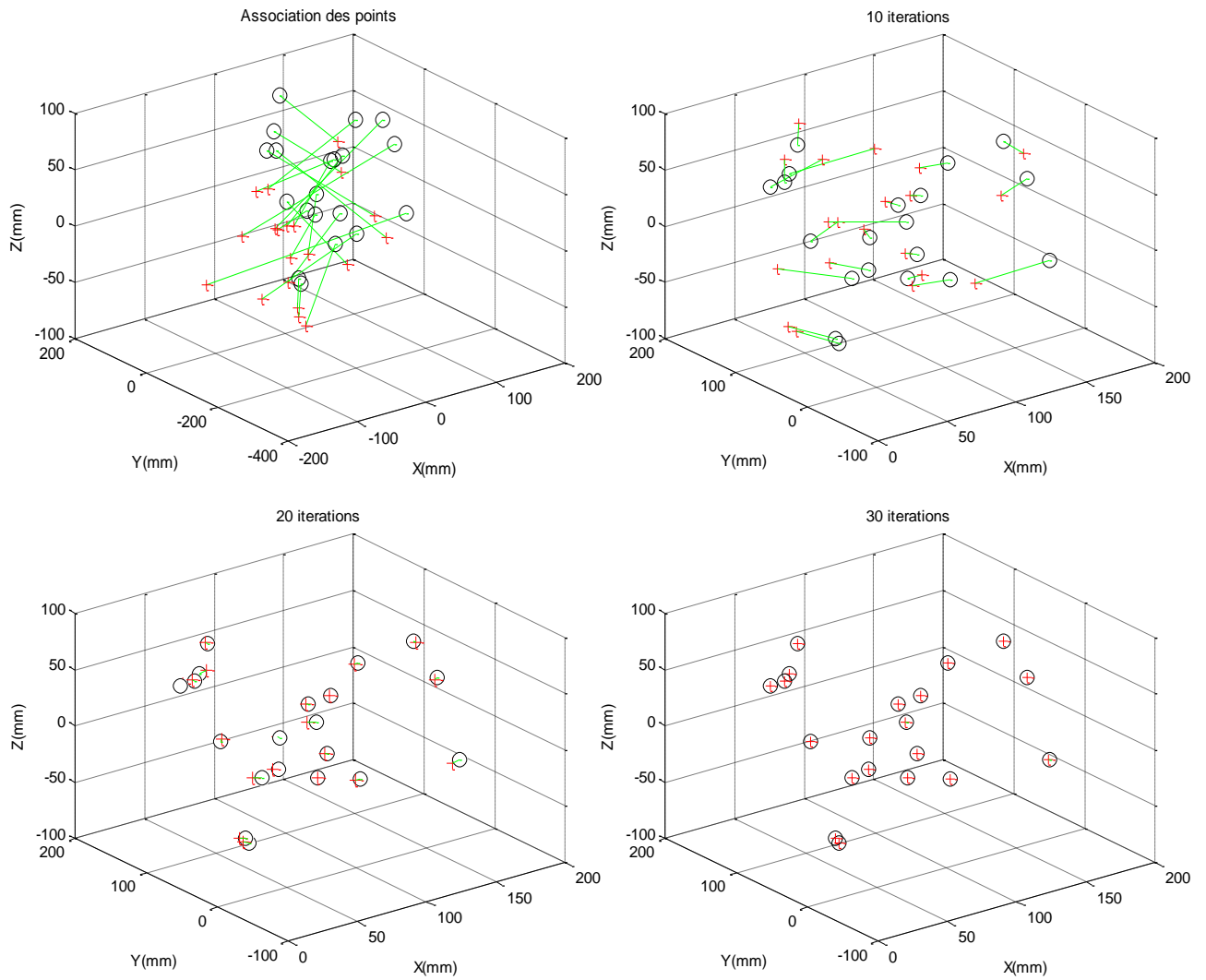
Figure III-18 : organigramme de l'algorithme d'ICP.

III.6.2.4 Validation de l'algorithme :

Pour valider l'algorithme d'ICP implémenté, nous avons effectué une simulation par des données synthétiques, avant le passage à la pratique. Qui a pour but l'évaluation des performances de cet algorithme.

Dans cette simulation nous avons généré un nuage de points 3D noté P^k . Ce nuage a subi une transformation (rotation et translation) à paramètre connu pour avoir un deuxième nuage de points P^{k+1} . Ces deux nuages vont être considérés comme l'entrée de l'algorithme,

L'objectif est l'estimation de la transformation qui permet la superposition des deux nuages de points P^k et P^{k+1} projeté, le résultat de ce test pour des différentes itérations est illustré dans la figure III.24. Dans la figure III.25 et le tableau III.2 nous avons montré



évolution de l'erreur au cours d'exécution de l'algorithme ICP :

Figure III-19 : Résultat de simulation de l'algorithme ICP.

Itération	0(initial)	10	20	30
Erreur (mm)	5.1641e+003	712.6441	73.8509	0.8047

Tableau III.2: Tableau d'évolution des erreurs de simulation de l'ICP.

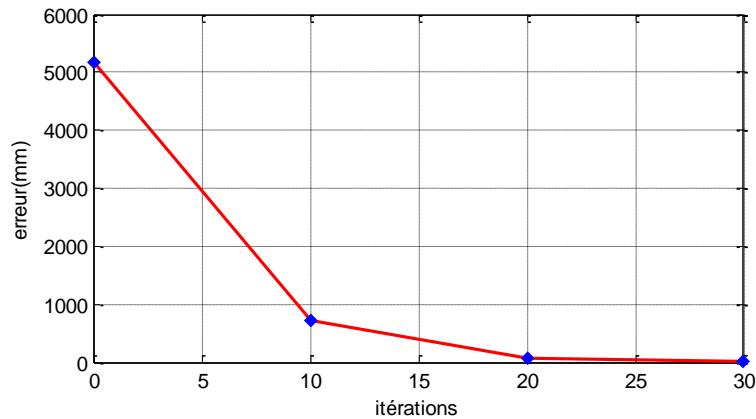


Figure III-20 : Évolution de l'erreur de simulation.

Nous remarquons que l'erreur d'estimation diminue quand le nombre d'itérations augmente, une meilleure superposition des points a été obtenue après 30 itérations

L'algorithme ICP est couramment utilisé pour l'estimation de la transformation rigide entre deux ensembles de points. Besl et McKay[115] ont prouvé qu'en déterminant les points les plus proches et en cherchant la transformation entre les deux ensembles de points appariés, la solution converge obligatoirement vers un minimum. Mais l'inconvénient majeur de cet algorithme réside dans le temps nécessaire à la convergence qui est généralement le problème de toutes les méthodes itératives ce qui rend leur implémentation en temps réel délicate. C'est pour cette raison on a pensé aux autres algorithmes plus rapides tels que SVD

III.6.3 Algorithme SVD (Singular Value Decomposition)

III.6.3.1 Généralité

La méthode de la décomposition en valeur singulière (SVD) a été utilisée dans plusieurs applications tel que le recalage 3D par shinji Umeyama[116], l'ajustement de deux ensembles de nuage de points par Arun [117] et l'évaluation des méthodes d'alignement des points d'intérêt par Ballesta [118]. Le principe de cette méthode est basé sur l'idée qu'une matrice M , de dimension $m \times n$ peut-être décomposée en une matrice U de dimension $m \times m$, une matrice diagonale S de dimension $m \times n$ qu'est composée de réels positifs, qui sont les valeurs singulières de la matrice M , et une matrice notée V^t de dimension $n \times n$.

$$M = U * S * V^t \quad (\text{III.10})$$

$$\text{Avec: } S = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m \end{bmatrix}$$

III.6.3.2 Estimation de la transformation homogène Par l'algorithme SVD

En supposant que l'on dispose de deux nuages de points 3D appariés à partir desquels on veut calculer la transformation rigide (rotation et translation) qui les lie. Ceci revient à minimiser une mesure de similarité en minimisant le critère de l'équation (III.6).

Arun, Huang et Blostein[117, 118] proposent une méthode de minimisation au sens des moindres carrés basée sur la décomposition SVD d'une matrice représentant l'ensemble des appariements. La méthode propose de résoudre le problème en deux temps, en séparant la rotation et la translation. On peut se rendre indépendant à la translation en exprimant chaque point par rapport au centre de masse du nuage dont il est issu.

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \mu_k = \frac{1}{N} \sum_{i=1}^N P_i^k \\ Q_i^k = P_i^k - \mu_k \end{array} \right. \\ \left\{ \begin{array}{l} \mu_{k+1} = \frac{1}{N} \sum_{i=1}^N P_i^{k+1} \\ Q_i^{k+1} = P_i^{k+1} - \mu_{k+1} \end{array} \right. \end{array} \right. \quad (III.11)$$

Ainsi on peut exprimer l'équation (III.6) en fonction de R uniquement (III.12) :

$$e = \frac{1}{N} \sum_{i=1}^N [(RQ_i^{k+1} + R\mu_{k+1} + t) - Q_i^k - \mu_k]^2 \quad (III.12)$$

Par conséquent $R\mu_{k+1} - \mu_k + t = 0$ donc on peut exprimer la quantité à minimiser par :

$$e = \frac{1}{N} \sum_{i=1}^N [RQ_i^{k+1} - Q_i^k]^2 \quad (III.13)$$

$$= \frac{1}{N} \sum_{i=1}^N [RQ_i^{k+1} - Q_i^k]^t [RQ_i^{k+1} - Q_i^k] \quad (III.14)$$

$$= \frac{1}{N} \sum_{i=1}^N (Q_i^{k+1t} R^t R Q_i^{k+1} - Q_i^{k+1t} R^t Q_i^k - Q_i^{kt} R Q_i^{k+1} + Q_i^{kt} Q_i^k) \quad (III.15)$$

$$= \frac{1}{N} \sum_{i=1}^N (Q_i^{k+1t} Q_i^{k+1} - 2 Q_i^{kt} R Q_i^{k+1} + Q_i^{kt} Q_i^k) \quad (III.16)$$

Par conséquent, minimiser l'équation (III.16) est équivalent à maximiser l'équation (III.17)[119]:

$$F = \sum_{i=1}^N Q_i^{kt} R Q_i^{k+1} \quad (III.17)$$

$$= \text{Trace} \left(\sum_{i=1}^N R Q_i^{k+1} Q_i^{k^t} \right) \quad (\text{III.18})$$

$$= \text{Trace}(R M) \quad (\text{III.19})$$

Avec : $M = \sum_{i=1}^N Q_i^{k+1} Q_i^{k^t}$.

Soit la décomposition SVD de M (équation III.10). Tel que : $S = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$.

D'où la matrice de rotation R :

$$R = U * V^T \quad (\text{III.20})$$

Il faut vérifier $\det(R) = 1$ que pour s'assurer qu'il s'agit bien d'une rotation, car l'algorithme peut échouer (rarement).

La translation est calculée par la formule (III.21) :

$$t = \mu_k - R * \mu_{k+1} \quad (\text{III.21})$$

Erreur minimale qu'on peut avoir dans l'estimation de (R, t) par cette méthode est égale :

$$E(R, t) = \sum_{i=1}^N \left(\|Q_i^k\|^2 + \|Q_i^{k+1}\|^2 \right) - 2(\sigma_1 + \sigma_2 + \sigma_3) \quad (\text{III.22})$$

La figure III.21 présente un pseudo algorithme pour l'estimation de la transformation homogène par la méthode SVD :

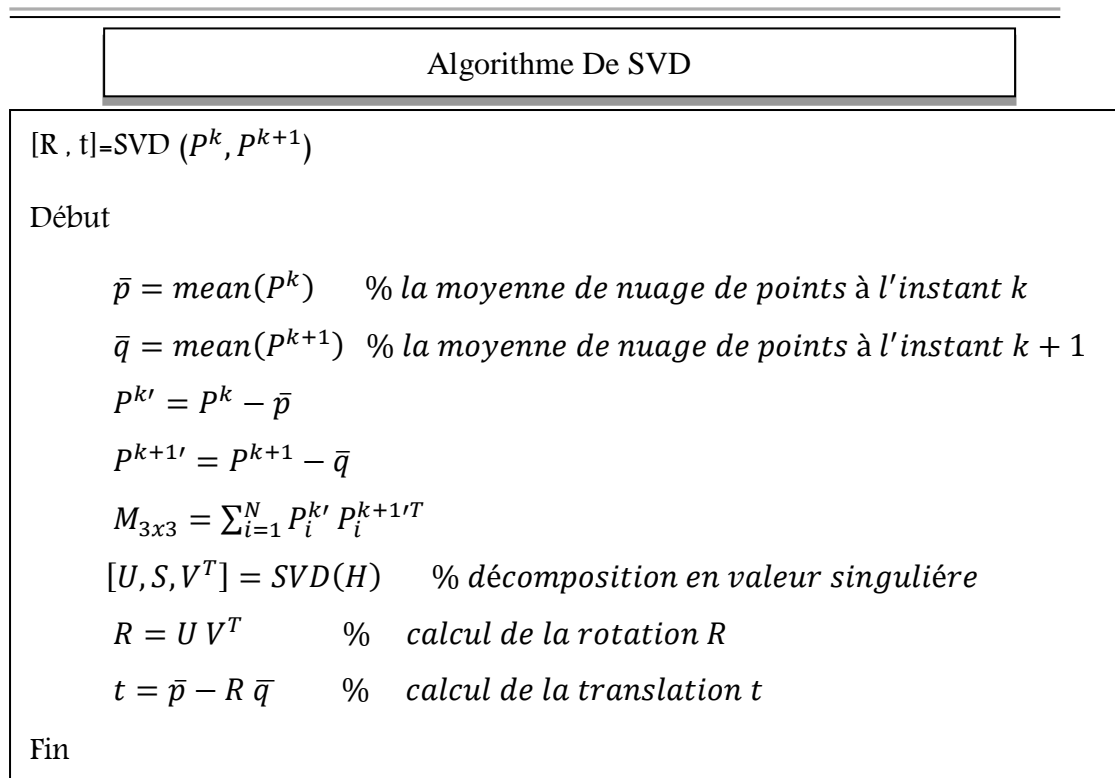


Figure III-21 : Pseudo code de l'algorithme d'optimisation par SVD.

L'algorithme d'estimation d'une transformation rigide par la décomposition en valeur singulières (SVD) est très utilisé actuellement pour sa simplicité et sa rapidité d'exécution, mais son inconvénient c'est qu'il peut donner des fausses matrices de rotation qui ont un déterminant différent de l'unité $\det(R) \neq 1$.

L'expression axe/angle présente deux défauts majeurs pour le traitement à l'ordinateur, la solution n'est pas unique (racine positive. ou négative.), la sinusoire $\sin(\theta)=0$ mène à une singularité (axe non-défini), En pratique, il y aura une mauvaise condition numérique pour tout angle proche de 0 ou de 180°, et une erreur pour ces deux cas, Pour remédier à ce problème de rotation, il est préférable de changer la représentation de la matrice rotation R (les angles d'Euler) par une autre représentation. Les deux inconvénients cités auparavant disparaissent de façon élégante en employant les paramètres d'Euler, ou quaternions ou paramètres de Rodriguez, Les quaternions sont employés en robotique industrielle.

III.6.4 Algorithme de Quaternion

Les quaternions sont un concept mathématique intéressant avec une relation profonde des bases de l'algèbre et la théorie des nombres, dans la pratique, ils sont le plus utile comme un moyen de représenter les orientations. Les quaternions ont été introduits par William Rowan Hamilton en 1843. Ils construisent un ensemble de quadruplets qui étend l'espace des propriétés

des nombres complexes dans le plan. Les quaternions Q sont ainsi des éléments d'un espace de dimension quatre, dont nous noterons (q_0, q_x, q_y, q_z) les quatre composantes d'une représentation non redondante de type cartésien. Ils peuvent être utilisés pour représenter une rotation R dans l'espace [120].

III.6.4.1 Généralités sur les Quaternions :

Les quaternions peuvent être vus comme des vecteurs de dimensions quatre ou encore comme des nombres complexes à trois parties imaginaires :

$$q = q_0 + iq_x + jq_y + kq_z = q_0 + v \quad (\text{III.23})$$

Tel que : $q_0, q_x, q_y, q_z \in R$ et $i^2 = j^2 = k^2 = ijk = -1$.

On peut remarquer qu'on obtient également :

$$\begin{cases} ij = -ji = k \\ jk = -kj = i \\ ki = -ik = j \end{cases} \quad (\text{III.24})$$

Grace à ces formules on peut facilement calculer le produit de deux quaternions, noté « * » (III.25) :

$$r * q = (r_0 + ir_x + jr_y + kr_z)(q_0 + iq_x + jq_y + kq_z) \quad (\text{III.25})$$

Le produit de deux quaternions n'est pas commutatif, comme le produit vectoriel, en effet ce produit peut s'écrire sous forme matricielle :

$$r * q = Q(r)q \quad (\text{III.26})$$

$$\text{Avec : } Q(r) = \begin{bmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & -r_z & r_y \\ r_y & r_z & r_0 & -r_x \\ r_z & -r_y & r_x & r_0 \end{bmatrix}$$

On peut également écrire le produit sous la forme suivante :

$$q * r = W(r)q \quad (\text{III.27})$$

$$\text{Avec : } W(r) = \begin{bmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & r_z & -r_y \\ r_y & -r_z & r_0 & r_x \\ r_z & r_y & -r_x & r_0 \end{bmatrix}$$

Nous allons établir maintenant quelques autres propriétés des quaternions :

Conjuguée : $\bar{q} = [q_0, -v]$;

Module : $\|q\| = \sqrt{q * \bar{q}} = \sqrt{q_0^2 + |v|^2}$;

Inverse : $q^{-1} = \frac{1}{q} = \frac{\bar{q}}{\|q\|^2}$ si $\|q\| = 1$ alors : $q^{-1} = \bar{q}$.

III.6.4.2 Diverses représentations des rotations

Dans ce qui suit nous considérons que l'orientation ou l'attitude d'un repère est représentée par la rotation qui fait passer d'un repère choisi comme repère origine au repère considéré, il existe plusieurs représentations des rotations [121], on cite les angles d'Euler, les vecteurs de rotation (axe/angle), La matrice des cosinus directeurs (DCM : Direction Cosine Matrix), et les quaternions...etc, nous avons choisi de présenter quelques une.

a) Représentation classique par trois angles

En pratique, pour l'interprétation humaine, les rotations sont représentées par les trois angles des trois rotations successives autour de trois axes de base comme les montre la figure III.26, parmi ces représentations nous trouvons:

les angles ψ de cap autour de z_0 , θ d'assiette longitudinale autour de y_1 et φ de roulis autour de x_2 sont utilisés en aéronautique ; les angles ψ de précession autour de z_0 , θ de nutation autour de x_1 et φ de rotation propre autour de z_2 sont utilisés pour l'étude des gyroscopes (angles d'Euler).

Bien que ces représentations n'aient que trois degrés de liberté, elles présentent toutes des redondances parasites. Dans le cas des angles aéronautiques on a l'équivalence suivante :

$$(\psi, \theta, \varphi) \equiv (\pi + \psi, \pi - \theta, \pi + \varphi) \quad (\text{III.28})$$

Et pour les angles d'Euler, on a l'équivalence :

$$(\psi, \theta, \varphi) \equiv (\pi + \psi, -\theta, \pi + \varphi) \quad (\text{III.29})$$

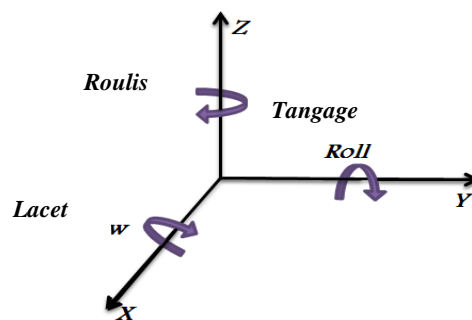


Figure III-22 : Les angles d'Euler

b) Représentation par Quaternion

On peut représenter une rotation avec un quaternion unitaire à condition de pouvoir trouver

une transformation qui transforme un vecteur (un quaternion purement imaginaire) en un vecteur de façon que la transformation préserve la longueur du vecteur transformé ainsi que le produit scalaire et le signe du Produit vectoriel.

Soit r un quaternion purement imaginaire et q un quaternion unitaire (le module égal à 1). On peut écrire le quaternion r' tel que [120]:

$$r' = q * r * \bar{q} \quad (III.30)$$

$$= (Q(q)r) * \bar{q} \quad (III.31)$$

$$= (W(q)^t Q(q)) r \quad (III.32)$$

Avec :

$$W(q)^t Q(q) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 0 & 2(q_x q_y + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 0 & 2(q_x q_y - q_0 q_z) & 2(q_y q_z + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \quad (III.33)$$

On peut maintenant établir l'expression de la matrice de rotation R en fonction de quaternion q :

$$R = \begin{bmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_y - q_0 q_z) & 2(q_y q_z + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \quad (III.34)$$

III.6.4.3 Estimation de la transformation homogène

Nous sommes maintenant en mesure d'écrire à nouveau le critère permettant de déterminer la transformation rigide optimale entre deux nuages de points 3D l'équation (III.6)

Les correspondances sont déjà établies par un autre critère, dans notre cas l'utilisation du descripteur SURF assure la mise en correspondance et le suivi (association) de ces points dans une séquence d'images. Après association, nous aurons un ensemble de paires de points de taille N , nous allons suivre les mêmes étapes de développements que nous avons déjà montré dans la section précédant « Estimation de la transformation homogène par SVD », jusqu'à l'équation à maximiser (III.17).

$$F = \max_R \sum_{i=1}^N Q_i^{kT} R Q_i^{k+1} \quad (III.35)$$

Nous représentons la rotation à l'aide du quaternion unitaire q [120] :

$$F = \sum_{i=1}^N (q * Q_i^k * \bar{q}) \cdot Q_i^{k+1} \quad (\text{III.36})$$

$$= \sum_{i=1}^N (q * Q_i^k) \cdot (Q_i^{k+1} * q) \quad (\text{III.37})$$

$$= \sum_{i=1}^N (\bar{\mathfrak{R}}_i^k q) \cdot (\mathfrak{R}_i^{k+1} q) \quad (\text{III.38})$$

Avec :

$$Q_i^{k+1} * q = \begin{bmatrix} 0 & -x_i'^{k+1} & -y_i'^{k+1} & -z_i'^{k+1} \\ x_i'^{k+1} & 0 & -z_i'^{k+1} & y_i'^{k+1} \\ y_i'^{k+1} & z_i'^{k+1} & 0 & -x_i'^{k+1} \\ z_i'^{k+1} & -y_i'^{k+1} & x_i'^{k+1} & 0 \end{bmatrix} \cdot q \quad (\text{III.39})$$

$$= \mathfrak{R}_i^{k+1} q \quad (\text{III.40})$$

$$q * Q_i^k = \begin{bmatrix} 0 & -x_i'^k & -y_i'^k & -z_i'^k \\ x_i'^k & 0 & z_i'^k & -y_i'^k \\ y_i'^k & -z_i'^k & 0 & x_i'^k \\ z_i'^k & y_i'^k & -x_i'^k & 0 \end{bmatrix} \cdot q \quad (\text{III.41})$$

$$= \bar{\mathfrak{R}}_i^k q \quad (\text{III.42})$$

$$= \sum_{i=1}^N (q^T \bar{\mathfrak{R}}_i^k T) \cdot (\mathfrak{R}_i^{k+1} q) \quad (\text{III.43})$$

$$= q^T \left(\sum_{i=1}^N \bar{\mathfrak{R}}_i^k T \mathfrak{R}_i^{k+1} \right) q \quad (\text{III.44})$$

$$= q^T N q \quad (\text{III.45})$$

La matrice N est une Matrice 4×4 et symétrique, elle possède quatre valeurs propres réelles $\lambda_1, \lambda_2, \lambda_3$ et λ_4 qui correspondent aux quatre vecteurs propres v_1, v_2, v_3 et v_4 .

Le quaternion q qui maximise l'équation (III.36) est égal au vecteur propre qui correspond à la valeur propre la plus élevée de N [120]. Pour retourner à la matrice de rotation il suffit de remplacer dans (III.34) par les composantes de quaternion q , et la translation est calculée par la formule suivante :

$$t = \mu_k - R * \mu_{k+1} \quad (\text{III.46})$$

Algorithme De Quaternion

```

[R , t]=Quaternion (Pk, Pk+1)
Début
% calcul de la moyenne de chaque nuage de points(Pk, Pk+1)
  p̄ = mean(Pk)
  q̄ = mean(Pk+1)
  Pk' = Pk - p̄
  Pk+1' = Pk+1 - q̄
  M3x3 = ∑i=1N Pik' Pik+1'T
% calcul de la matrice N
  delta = [M(2,3) - M(3,2); M(3,1) - M(1,3); M(1,2) - M(2,1)]
  N = [trace(M)delta'; delta (M+M'-trace(M)*eye(3))]
% calcul des valeurs et des vecteurs propres de N
  [Vecteurs_pro, valeur_pro] = eig(N)
  [index] = max(valeur_pro)
% conversion du Quaternion à la matrice rotation
  R = quaternionToMatrixR(Vecteurs_pro(index))
  t = p̄ - R q̄
Fin
  
```

Figure III-23 : Pseudo code de l'algorithme d'optimisation par Quaternion.

L'avantage majeur de ces deux derniers algorithmes SVD et Quaternion c'est qu'ils sont directement exploitables dans le cas de l'odométrie visuelle à six degrés de liberté parce qu'ils fournissent les six paramètres de la matrice de transformation (les trois rotations et les trois translations), par contre c'est délicat de rendre l'algorithme ICP exploitable dans tel environnement à cause des lourdes équations à résoudre.

III.7 Élimination des fausses associations

La présence des fausses associations (outliers) entre les deux nuages de points peut ralentir la convergence de l'algorithme d'estimation $de(R, t)$ et même elle peut causer la divergence de l'algorithme vers une mauvaise estimation. Les associations entre les deux nuages de points sont établies par les deux techniques citées antérieurement.

La technique de plus proche voisin : est basée sur l'association entre les points qui présentent la plus faible distance euclidienne entre eux, cette technique n'est pas robuste et peut facilement causer la divergence de l'algorithme d'estimation, comme le montre la figure suivante .

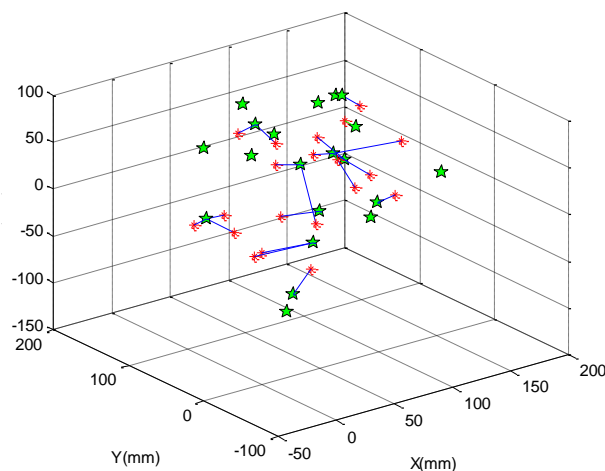


Figure III-24 : Les fausses associations par la méthode de plus proche voisin.

La deuxième technique d'association est basée sur le tracking, en utilisant le descripteur SURF, cette technique est la plus robuste par rapport aux autres techniques, mais elle présente aussi un certain nombre de fausse association, figure (III.29).

Pour remédier à ce problème, il existe dans la littérature plusieurs méthodes d'élimination des mauvaises associations. Dans notre projet nous avons implémenté deux méthodes, les plus populaires qui sont RANSAC et filtrage de Zhang.

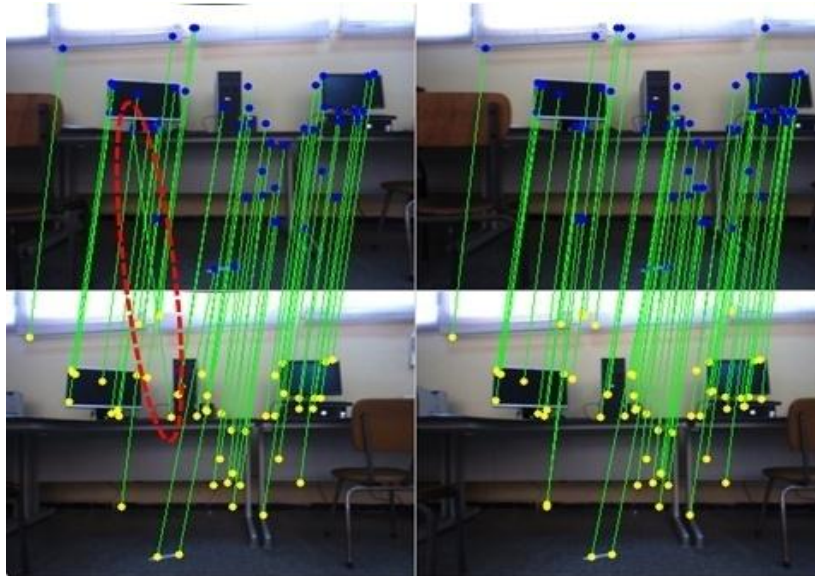


Figure III-25 : Les fausses Associations par le descripteur SURF.

III.7.1 RANSAC

RANSAC est une abréviation de "RANdomSAmple Consensus". Il s'agit d'une méthode itérative pour estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données observées qui contient des valeurs aberrantes (outliers). RANSAC c'est un algorithme non déterministe dans le sens où il produit un résultat correct avec une certaine probabilité, celle-ci augmentant à mesure que le nombre d'itérations est grand. L'algorithme a été publié pour la première fois par Fischler et Bolles en 1981.

Cette méthode est classiquement utilisée pour extraire un modèle d'un ensemble de données bruitées. Dans le cas présent, RANSAC est utilisé pour déterminer quels sont les points qui permettent d'estimer la transformation entre les deux nuages le plus correctement possible. Autrement dit, RANSAC permet d'éliminer les fausses associations.

RANSAC est basé sur le principe suivant (figure III.26) : un ensemble de points dans chaque nuage sont tirés au hasard pour déterminer une base. La transformation liant les deux bases est alors calculée par une des méthodes d'estimation que nous avons citées dans ce chapitre. Les nuages sont recollés suivant cette transformation et la distance entre chaque couple de points est estimée. Un score est alors attribué à la transformation : si un nombre important de points se superposent, le score est élevé, sinon il est faible. Finalement, la transformation qui possède le plus haut score est conservée, car elle permet de recoller plus de points de l'ensemble, donc elle a plus de chance d'être correcte. Cette méthode permet en plus d'évaluer la transformation liant les deux nuages de point et de filtrer les points qui n'entrent pas dans le modèle principal (outliers).

Algorithme de RANSAC

```

Début
For: it=1 à it Max
     $P_{aleatoire} = n \text{ valeurs choisies au hasard de } (P^k, P^{k+1})$ 
     $T = Transformation(P_{aleatoire})$ 
     $P'^{k+1} = Projection(P^{k+1}, T)$ 
    Fori=1 à N
        If ( $|P_i^k - P_i'^{k+1}| < seuil$ )
             $inliers = inliers \ \&\& (P_i^k, P_i'^{k+1})$  % Ajouter cette paire
        de point aux inliers
    End
    End
    If( $size(inliers) > size(best\_D)$ )
         $best_P = inliers$ 
         $best_T = T$ 
    End
End
     $return (best\_P, best\_T)$ 
Fin
  
```

Figure III-26 :Pseudo-code de l'algorithme RANSAC.

III.7.2 Filtrage

L'algorithme de filtrage que nous allons présenter dans ce paragraphe est introduit pour la première fois par Zhengyou Zhang en 1992 [122, 123] pour éliminer les fausses associations. Il consiste à calculer la distance entre deux points P_i^k et $P_i'^{k+1}$ projetées par la transformation (rotation et translation) estimée ; si cette distance est supérieure à un seuil, la paire va-t-êtré rejetée ce qui fait qu'elle n'intervienne pas dans l'estimation de la transformation (rotation et translation) de la prochaine itération (outlier).

Le critère dont il se base cet algorithme pour décider qu'une association est fausse ou non est d'utilisé un seuil adaptatif en fonction des caractéristiques statistiques de l'erreur (moyenne, médian, écart-type) tel que présenter dans l'algorithme de filtrage de Zhang[122].

III.8 Conclusion

Dans ce chapitre nous avons décrit le nouveau système de localisation visuelle en utilisant une caméra monoculaire ou stéréoscopique. Ce système sera embarqué sur le quadrotor pour réaliser des tâches de perception et navigation. Nous avons situé le domaine de l'OV dans la

robotique mobile en montrant sa relation avec le SLAM visuel, puis la formulation du problème est présentée, ainsi que les différentes approches d'estimation de mouvement à partir d'images. Dans ce chapitre nous avons présenté le problème OVS qui vise à estimer la pose (position et orientation) du quadrotor à partir de deux nuages de points d'intérêts successifs observés par le banc stéréoscopique. Les algorithmes présentés sont basés sur la validation de plusieurs techniques existantes : Itérative Closest Point (ICP), la décomposition en valeur singulière (SVD) et les Quaternions.

Ces outils permettent d'élaborer d'éventuels points référentiels en 3D pour le suivi des trajectoires par la commande Backstepping.

Chapitre IV : Commande
Backstepping adaptative basée
vision

CHAPITRE IV: COMMANDE BACKSTEPPING ADAPTATIVE BASEE VISION DU QUADRIROTOR

IV.1 Introduction

Le quadrirotor est une plateforme idéale pour explorer ou exécuter des tâches qui ne peuvent pas être exécutées par un robot terrestre, telles que, la surveillance des agricoles, les secours en cas des catastrophes et la livraison express. En outre, il présente certains avantages caractéristiques, liés par sa petite taille, son poids léger, sa grande flexibilité, sa maniabilité rapide et sa fabrication à faible coût. Pour exécuter des missions spécifiques avec des quadrirotors, une commande précise du suivi de trajectoire, ainsi que des capacités anti-perturbation efficaces sont nécessaires. Par conséquent, il est important d'améliorer les performances de la commande de suivi de trajectoire.

Le système de suivi de trajectoire peut-être divisé en deux parties. Le premier est le système de localisation visuelle (OVS) et le deuxième la commande de suivi de trajectoire. Le système de localisation visuelle fournit l'estimation du quadrirotor. On se basant sur cette estimation, la commande de suivi de trajectoire calcule le couple pour chaque rotor selon la loi de commande et le modèle du quadrirotor[2].

Le système de localisation quadrirotor traditionnel repose principalement sur les données de fusion des centrales inertielles (IMU) et du système de positionnement global (GPS). Cependant, le signal GPS est affecté par les interférences électromagnétiques. Ainsi, il est difficile d'acquérir la position dans des environnements complexes, en particulier dans un environnement intérieur. En conséquence, de nombreuses recherches ont été effectuées sur le système de localisation avec d'autres capteurs, tels que le capteur à ultrasons, le lidar et le capteur visuel. Comparé au lidar et à d'autres capteurs, le système de localisation visuelle est plus adapté aux petits quadrirotors. Le développement rapide de la vision par ordinateur favorise également l'utilisation de caméras embarquées. Par conséquent, un nombre croissant d'études sur la commande de vol autonome basée sur la vision pour les quadrirotors sont menées récemment. La plupart d'entre eux introduisent simultanément la technique de localisation et de cartographie (SLAM) comme système de localisation[124, 125].

Dans ce manuscrit de thèse nous avons utilisé une caméra stéréoscopique embarquée sur une plateforme mobile pour valider le système de localisation. D'autre part, la conception d'une commande adaptative basée vision joue un rôle plus important dans le système de suivi de

trajectoire.

Depuis quelques années, de nombreuses classes d’algorithmes de commande visuelle ont été implantées sur les quadrirotors, on cite dans la suite quelques exemples:

- basée sur la commande classique PD et PID[126] ;
- basée sur la commande par le Backstepping[3] ;
- basée sur la commande par mode glissant[127, 128];
- basée sur de la commande optimale et la commande linéarisante[129];
- basée sur la commande floue et réseau de nuerons [130]...etc.

L’objectif de ce chapitre est d’élaborer une stratégie de commande robuste et adaptative, basée principalement sur une localisation visuelle stéréoscopique adaptative à l’environnement de navigation suivi par une commande Backstepping optimal. Dans la suite, tout d’abord nous allons présenter la commande par backstepping et l’optimisation de ses paramètres.

IV.2 Commande par backstepping

Dans ce chapitre nous allons présenter la commande backstepping pour le suivi de trajectoire à l’aide du drone de type quadrotor considéré. Le modèle dynamique, exprimé dans l’équation (II.16), peut être décrit comme un modèle d’état, $\dot{X} = f(X, U)$, en introduisant le vecteur $X = (x_1 \ x_2 \ \dots \ x_{12})^T$, qui est un vecteur d’état du système, tel que $X = (\varphi \ \dot{\varphi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z})^T$. En conséquence, on obtient ce qui suit:

$$f(X, U) = \begin{pmatrix} x_2 \\ -a_1 x_4 x_6 - a_3 \bar{\Omega} x_4 - a_2 x_2^2 + b_1 u_2 \\ x_4 \\ a_4 x_2 x_6 + a_6 \bar{\Omega} x_2 - a_5 x_4^2 + b_2 u_3 \\ x_6 \\ -a_7 x_2 x_4 - a_8 x_6^2 + b_3 u_4 \\ x_8 \\ a_9 x_8 + \frac{1}{m} u_1 u_1 \\ x_{10} \\ a_{10} x_{10} + \frac{1}{m} u_y u_1 \\ x_{12} \\ a_{11} x_{12} - g + \frac{1}{m} (\cos x_1 \cos x_3) u_1 \end{pmatrix} \quad (IV.1)$$

$$\text{Où } \begin{cases} a_1 = \frac{J_z - J_y}{J_x} & a_2 = \frac{k_{fax}}{J_x} & a_3 = \frac{J_r}{J_x} \\ a_4 = \frac{J_z - J_x}{J_y} & a_5 = \frac{k_{fay}}{J_y} & a_6 = \frac{J_r}{J_y} \\ a_7 = \frac{J_y - J_x}{J_z} & a_8 = \frac{k_{faz}}{J_z} & \\ a_9 = -\frac{k_{fx}}{m} & a_{10} = -\frac{k_{fy}}{m} & a_{11} = -\frac{k_{fz}}{m} \\ b_1 = \frac{\ell}{J_x} & b_2 = \frac{\ell}{J_y} & b_3 = \frac{1}{J_z} \end{cases}$$

Les entrées de commande sont supposées être bornées comme suit [131]:

$$\begin{cases} 0 \leq u_{1i} \leq 4b \omega_{i \max}^2 \\ |u_{2i}| \leq \ell b \omega_{i \max}^2 \\ |u_{3i}| \leq \ell b \omega_{i \max}^2 \\ |u_{4i}| \leq 2d \omega_{i \max}^2 \end{cases} \quad (\text{IV.2})$$

où $\omega_{i \max}$ est la vitesse de rotation maximale de l'hélice.

La figure 4 montre la structure du système de commande quadrirotor. Les trois positions (x , y , et z) et l'angle de lacet, ϕ , sont choisis comme sorties, et les signaux d'attitude souhaités, φ_d et θ_d , pour φ et θ , respectivement, requis pour le suivi sont générés par le contrôleur de position. Ce contrôleur est également utilisé pour suivre les signaux de référence (x_d , y_d , et z_d) pour les trois positions. Un contrôleur d'attitude est utilisé pour suivre les signaux de référence pour les trois angles d'attitude

La technique de backstepping est utilisée pour traiter le problème de commande de suivi de trajectoire robuste pour une classe de systèmes non linéaires multiples incertains avec des incertitudes non linéaires, des perturbations externes et des couplages internes. La stabilité et la propriété de suivi robuste du système en boucle fermée sont prouvées par l'analyse de stabilité de Lyapunov. Les erreurs de suivi sont limitées pour être petites, comme requis pour le taux de convergence attendu.

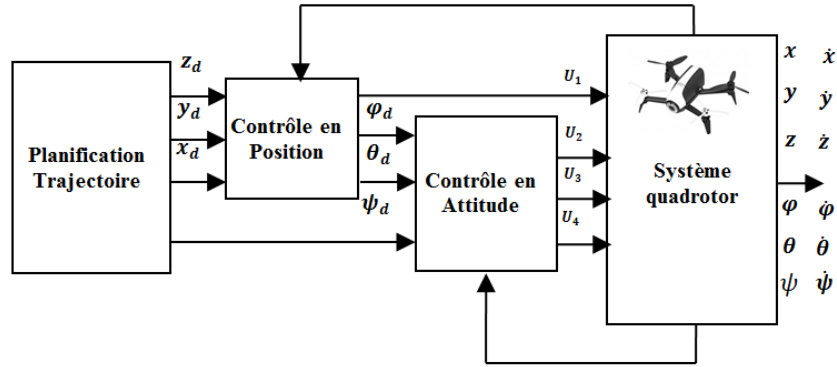


Figure IV-1 : Structure de la commande quadrirotor pour le suivi de trajectoire.

Les résultats de la simulation d'UAV vérifient l'efficacité du système de commande de surveillance proposé. Les fonctions de Lyapunov suivantes sont définies:

$$V_i = \begin{cases} \frac{1}{2} z_i^2 & / i \in \{1, 3, 5, 7, 9, 11\} \\ \frac{1}{2} (V_{i-1} + z_i^2) & / i \in \{2, 4, 6, 8, 10, 12\} \end{cases} \quad (\text{IV.3})$$

Où

$$z_i = \begin{cases} x_{id} - x_i & / i \in \{1, 3, 5, 7, 9, 11\} \\ x_i - \dot{x}_{(i-1)d} - \alpha_{(i-1)} z_{(i-1)} & / i \in \{2, 4, 6, 8, 10, 12\} \end{cases} \quad (\text{IV.4})$$

L'application de la technique du backstepping au modèle d'état du quadrotor dans l'équation (21) donne les commandes suivantes [24, 132]:

$$\begin{cases} u_1 = (m / \cos(x_1) \cos(x_3)) (-a_{11} x_{12} + g - \alpha_{11} x_{12} - \alpha_{12} z_{12} + z_{11}) \\ u_2 = b_1^{-1} (-a_1 x_4 x_6 - a_2 x_2^2 - a_3 x_4 \bar{\Omega} - \alpha_1 x_2 - \alpha_2 z_2 + z_1) \\ u_3 = b_2^{-1} (a_4 x_2 x_6 + a_5 x_4^2 - a_6 \bar{\Omega} x_2 - \alpha_3 x_4 - \alpha_4 z_4 + z_3) \\ u_4 = b_3^{-1} (a_7 x_2 x_4 + a_8 x_6^2 - \alpha_5 x_6 - \alpha_6 z_6 + z_5) \\ u_x = (m / u_1) (-a_9 x_8 - \alpha_7 x_8 - \alpha_8 z_8 + z_7) \\ u_y = (m / u_1) (-a_{10} x_{10} - \alpha_9 x_{10} - \alpha_{10} z_{10} + z_9) \end{cases} \quad (\text{IV.5})$$

$$\text{Où } \begin{cases} \alpha_i > 0, \forall i \in 1, 12 \\ \cos(x_1) \cos(x_3) \neq 0 \end{cases}$$

IV.3 Commande par backstepping optimale pour le suivi de trajectoire

Les paramètres de backstepping appropriés sont sélectionnés par des calculs empiriques et ajustés par plusieurs tests. Ces paramètres de commande sont optimisés par l'optimisation de

l'essaim de particules pour améliorer la qualité du suivi de trajectoire.

IV.3.1 Essaim de particules (Particle Swarm Optimization, PSO)

L'algorithme PSO est une méthode de calcul qui optimise un problème en améliorant itérativement une solution candidate par rapport à une mesure de qualité donnée [133]. Au cours des années, plusieurs modifications de l'algorithme PSO d'origine ont été suggérées.

Les équations formalisant le mouvement des particules sont données par [134].

$$\begin{cases} v_i^{k+1} = w^k v_i^k + \alpha_1 \gamma_{1,i} (P_i - x_i^k) + \alpha_2 \gamma_{2,i} (G - x_i^k) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases} \quad (\text{IV.6})$$

Où vecteurs x_i^k et v_i^k sont la position et la vitesse actuelles du $i^{\text{ème}}$ particule dans le $k^{\text{ème}}$ génération, respectivement. L'essaim se compose de N particules, c'est-à-dire $i = \{1, \dots, N\}$. Par ailleurs, p_i est la meilleure position personnelle de chaque individu, et G est la meilleure position globale observée parmi toutes les particules jusqu'à la génération actuelle. Les paramètres $\gamma_{1,2} \in [0,1]$ sont des valeurs aléatoires uniformément distribuées et $\alpha_{1,2}$ ont des constantes d'accélération. La fonction w est l'inertie des particules, qui génère une certaine quantité de mouvement des particules.

L'algorithme PSO se base sur la collaboration des individus entre eux. Les particules utilisent leurs expériences individuelles et de l'expérience de la population pour trouver la source d'aliment (point de conception optimale). Grâce à des règles de déplacement simples (dans l'espace de recherche), les particules peuvent converger progressivement vers un minimum. De cette façon, une particule trouve la solution optimale attire à elle-même le reste de la population, jusqu'à ce que toute la population trouve la même solution. Essentiellement, chaque particule essaye de se positionner près de la population sans se heurter avec les membres de celle-ci.

Alors, L'algorithme de la PSO consiste dans les étapes suivantes :

- Créer une population initiale avec distribution aléatoire de particules et de leurs vitesses;
- Calculer le vecteur de vitesse pour chaque particule, en utilisant l'expérience de chaque particule et l'acquisition de vitesse de la population ;
- Mise à jour de la position de chaque particule en utilisant le vecteur de vitesse et la position précédente;
- Evaluer la convergence.

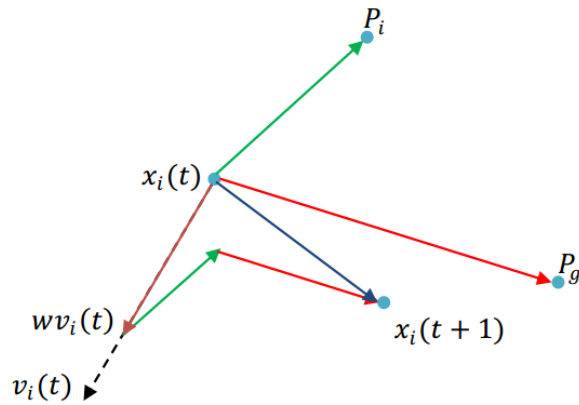


Figure IV-2 : Schéma vectoriel du déplacement d'une particule

La figure (IV.2) présente un schéma de principe résumant les explications ci-dessus.

IV.3.2 Réglage des paramètres du backstepping par PSO

Les composantes du vecteur de paramètres de réglage peuvent être définies en fonction des connaissances disponibles et des objectifs de commande. Dans la méthode de réglage PSO proposée pour le régulateur backstepping, le vecteur de réglage, P , est composé des paramètres du commande ($\alpha_i, i = 1, \dots, 12$).

$$P = [\alpha_1 \quad \dots \quad \alpha_{12}] \quad (IV.7)$$

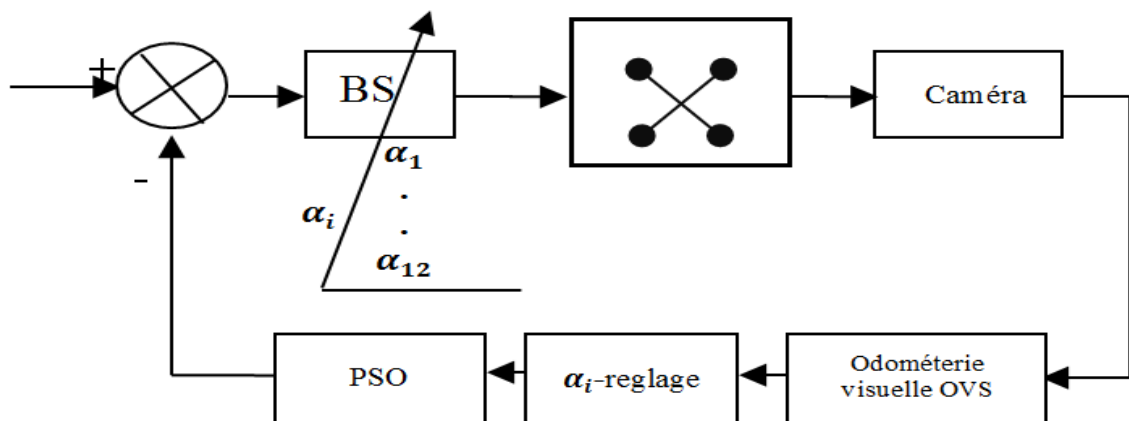


Figure IV-3 : Réglage des paramètres du Backstepping.

Une fonction de fitness à erreur quadratique (SE), telle qu'elle est exprimée dans l'équation (IV.8), est utilisée pour quantifier l'efficacité de la commande backstepping donnée. Pour une réponse indicielle du système en boucle fermée, elle est évaluée en ligne (Fig.IV-4).

$$SE = \frac{1}{4N} \sum_{k=1}^N e_{\phi}^2(k) + e_{\theta}^2(k) + e_{\psi}^2(k) + e_z^2(k) \quad (IV.8)$$

Où $e_{\phi} = \phi_{ref} - \phi, e_{\theta} = \theta_{ref} - \theta, e_{\psi} = \psi_{ref} - \psi, e_z = z_{ref} - z$

Les meilleures valeurs finales des paramètres réglés sont les suivantes (voir Fig. 6) :
 $P=[9.049 \ 1.855 \ 4.027 \ 2.699 \ 4.013 \ 2.467 \ 2.073 \ 4.423 \ 5.821 \ 1.708 \ 3.374 \ 7.755]$.

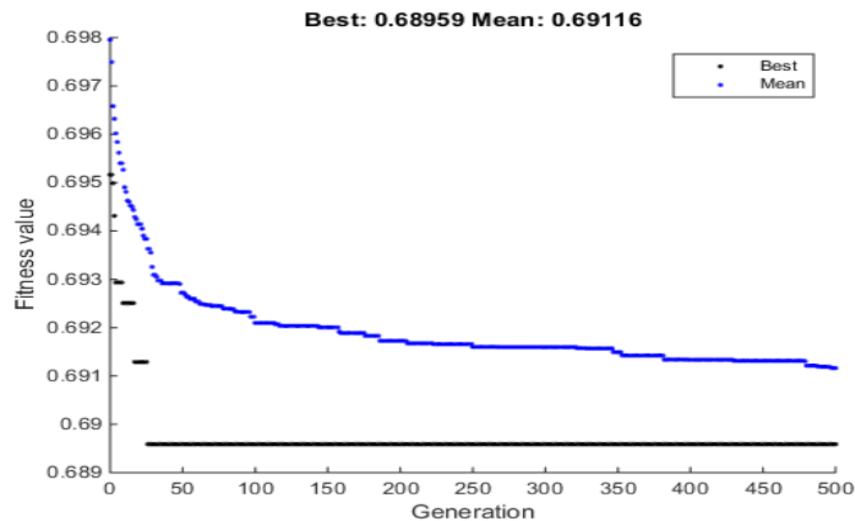


Figure IV-4 :. Fitness function.

Dans la simulation, les auteurs ont opté pour l'utilisation d'une méthode de réglage basée sur PSO pour l'optimisation des paramètres du contrôleur en marche arrière. Cette méthode est basée sur la fonction de fitness SE calculée par le suivi de trajectoire de référence du système. Pour améliorer les performances du système, certaines contraintes non linéaires sont ajoutées à la procédure d'optimisation. La méthode proposée est appliquée au quadrirotor choisi, et son efficacité est démontrée par les résultats de la simulation.

IV.4 Localisation du quadrirotor par fusion de données IMU/SVO

La localisation du quadrotor est la tâche principale pour n'importe quel algorithme de navigation autonome. Dans notre travail et afin d'assurer une localisation précise une solution de localisation basée sur la fusion de données a été proposée. La majorité des Quadrirotor se localise principalement par un système de localisation inertielle. Ce dernier, malgré les avantages qu'ils présentent, rapidité et localité mais ils dérivent dans le temps. Comme solution l'Odométrie Visuelle Stéréoscopique (OVS) présente une bonne solution de localisation complémentaire, car c'est un capteur extrinsèque et il ne dérive pas dans le temps. La fusion des deux capteurs inertielle (IMU) et visuelle (OVS) avec un filtre de Kalman étendu donne de meilleure performance rapidité/robustesse. La présence de l'IMU assure le maintien d'une bonne localisation même en absence de la donnée visuelle.

La figure (IV.5) montre un schéma global de l'approche proposée. La précision de la localisation est extrêmement dépendante du nombre de points et de la profondeur moyenne du nuage de points 3D.

- Lorsque le nombre d'extractions augmente, la précision augmente ; cependant, le temps de calcul augmente également.
- La précision diminue à mesure que la profondeur augmente, ce qui suggère que la précision est proportionnelle à la profondeur inverse (1/profondeur moyenne).
- Ces deux paramètres (nombre de points et 1/profondeur moyenne) sont utilisés comme entrées pour un système d'inférence floue pour déterminer une approche de localisation appropriée (IMU, OVS ou fusion OVS/IMU).

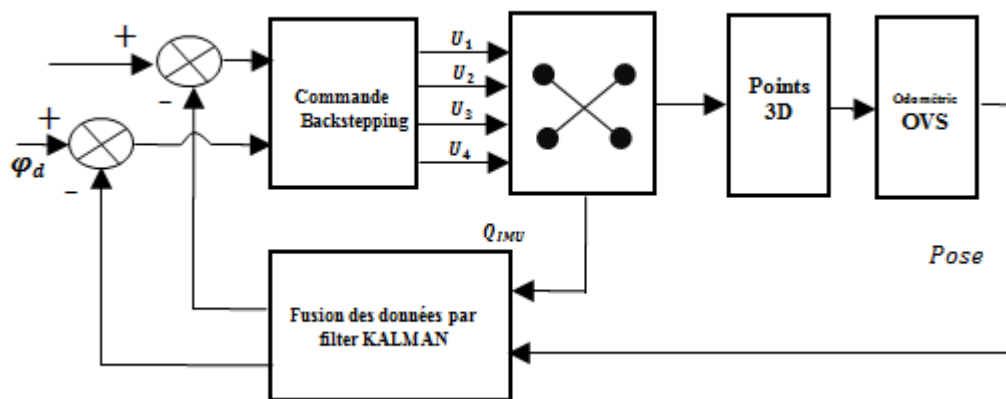


Figure IV-5 : Fusion des données IMU/OVS.

IV.5 Filtre de Kalman étendu

Le filtre de Kalman Étendu est l'un de meilleur filtre non-linéaire simple et sous-optimal. Dans cette thèse le EKF assure la fusion des données inertielles (IMU) et les données visuelles (OVS). Ces deux capteurs sont complémentaires, l'OVS assure une bonne localisation à long terme quand l'IMU maintient une bonne localisation quand l'information visuelle n'est pas disponible (court terme).

Dans différents modèles d'estimation, des filtres de navigation sont adoptés comme schémas de fusion de capteurs pour obtenir les états d'un véhicule aérien. Un tel modèle qui est proposé dans [135] est un modèle linéaire basé sur le modèle de navigation d'erreur avec l'hypothèse de petits angles d'Euler. Dans ce cas, l'utilisation d'un filtre de Kalman linéaire est suffisante. Cependant, l'hypothèse ci-dessus est généralement invalide, nécessitant l'utilisation d'approches de filtrage non linéaire, telles qu'un EKF, comme détaillé ci-dessous [136].

Le filtre de Kalman est un estimateur récursif, ce qui implique que pour estimer l'état courant du système, $X_k \in \mathcal{R}^n$, seul l'état précédent, X_{k-1} , et les mesures courantes, $Y_k \in \mathcal{R}^m$, sont nécessaires (et éventuellement l'entrée du filtre, $U_{k-1} \in \mathcal{R}^l$).

Le modèle d'état et l'observation du système non linéaire sont définis comme suit:

$$X_k = f_k(X_{k-1}, U_{k-1}, W_k) \quad (IV.9)$$

$$Y_k = h_k(X_k, V_k) \quad (IV.10)$$

Où X_k et Y_k sont respectivement l'état et l'observation à un pas de temps ; et W_k et V_k représentent des bruits additifs. Ces deux variables sont supposées indépendantes et suivent la loi de distribution normale. Dans cette étude, le modèle d'état est obtenu à l'aide de l'équation (21) basée sur les systèmes IMU [17]: Où :

$$p(W) \sim \mathcal{N}(0, Q) \text{ et } p(V) \sim \mathcal{N}(0, R) \quad (IV.11)$$

Le système d'odométrie visuelle proposé basé sur une caméra stéréo sert de modèle d'observation, fournissant des mesures à chaque fois en sortie Y_k :

$$Y_k = [x_k, y_k, z_k, \varphi_k, \theta_k, \psi_k] \quad (IV.12)$$

W_k et V_k sont supposés être des gaussiennes à moyenne nulle non corrélées avec des covariances connues, Q_k et R_k , respectivement. Par conséquent, l'objectif de la technique de filtrage est d'estimer X_k en utilisant l'observation existante, Y_k .

L'algorithme EKF est le suivant.

Algorithme de filtre de Kalman étendu

Initialisation:

$$\widehat{X}_0 = \bar{X}_0 = E[X_0]$$

$$R_0 = Q_0 = \text{cov}(X_0)$$

Prédiction:

$$\widehat{X}_k^- = f(\widehat{X}_{k-1}, \widehat{U}_{k-1})$$

$$P_k^- = A_k P_{k-1} A_k^T + B_k Q_{k-1} B_k^T + Q$$

$$\text{where } A_k = \frac{\partial f}{\partial x} |_{(\widehat{x}_{k-1}, \widehat{u}_{k-1})}, B_k = \frac{\partial f}{\partial u} |_{(\widehat{x}_{k-1}, \widehat{u}_{k-1})}$$

Correction:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R]^{-1}$$

$$\text{where } H_k = \frac{\partial h}{\partial x} |_{(\widehat{x}_{k-1}, \widehat{u}_{k-1})}$$

$$\widehat{X}_k = \widehat{X}_k^- + K_k (Y_k - H_k \widehat{X}_k^-)$$

$$P_k = (I - K_k H_k) P_k^-$$

Les mesures inertielle permettent une estimation complète du mouvement avec une fréquence élevée ; cependant, ces mesures dérivent avec le temps. En revanche, bien que l'estimation de pose d'odométrie visuelle nécessite un temps considérable pour l'extraction et la correspondance des caractéristiques, elle donne une estimation de pose robuste. Une solution

appropriée pour obtenir des positions précises en temps réel et éviter les dérives de pose IMU est une approche de fusion de données utilisant un filtre robuste et optimal. Dans cette étude, un EKF a été développé à cet effet.

IV.6 Localisation adaptative floue du Quadrotor

Le système de localisation par fusion de données (EKF-IMU/SVO) peut être non efficace dans quelques situations à cause des performances de l'OVS qui varient selon plusieurs critères. Par exemple un OVS dans une zone trop texturée donne une très bonne précision par rapport à un OVS dans un environnement non texturé. De même, l'OVS proposé est sensible aux environnements dynamiques. Donc si l'environnement est statique une bonne précision sera obtenue et dans le cas contraire la précision de l'OVS se dégrade. Dans la section précédente, l'algorithme de fusion proposé (EKF-IMU/OVS) ne prend pas en considération les variations des performances de l'OVS alors que c'est très important. Dans cette section, nous allons proposer une version adaptative de l'algorithme EKF-IMU/OVS en utilisant un contrôleur logique flou (FLC) pour sélectionner automatiquement les paramètres de fusion dans le but de maximiser les performances de la solution proposée (Figure IV-6).

Les performances SVO dépendent considérablement du nombre de points clefs. Les performances peuvent diminuer considérablement lorsque le nombre de fonctionnalités diminue. Pendant la navigation, des véhicules sans pilote pénètrent occasionnellement dans des endroits où le nombre d'éléments est faible ; par conséquent, SVO fournit des positions inexacts. De plus, dans certains scénarios, les points observés sont très éloignés de la caméra, et comme la précision d'une caméra de stéréovision diminue avec la profondeur (Fig. II.10), cela conduit à une localisation imprécise..

Pour éviter le problème SVO et réduire les erreurs de localisation, un système de localisation adaptatif basé sur la fusion de données visuelles et inertielles (SVO/IMU) est proposé dans cet article. À cette fin, un contrôleur de logique floue qui tiendra compte du nombre et de la profondeur des points 3D observés est suggéré. Il a deux entrées : l'inverse de la profondeur et du nombre d'entités, et sa sortie est un facteur de sélection. Le tableau d'inférence floue est résumé dans le tableau IV.1.

Le problème majeur de l'odométrie visuelle ou localisation visuelle est sa dépendance vis-à-vis de la qualité de l'image et du nombre et de la profondeur des points 3D observés, ce qui limite considérablement les performances de la localisation visuelle. Premièrement, il donne des résultats inexacts lorsque le nombre de points d'intérêt est petit. Deuxièmement, il est

extrêmement sensible à la profondeur des caractéristiques observées. Parallèlement, il atteint une meilleure précision pour les points d'intérêt à proximité (Fig. II.10). La figure (IV.5) illustre les principaux apports de cette étude pour éviter les limites de localisation visuelle. Dans les expériences, un système de logique floue est utilisé pour fusionner les données visuelles de la caméra et les données inertielles de l'IMU. Les entrées du bloc flou sont le nombre et la profondeur des points 3D observés (IV.6). Ces entrées, illustrées à la figure (IV.7), peuvent être utilisées comme une adaptation des points d'intérêts observés pour un cadre de fusion optimal. Par conséquent, un bloc flou sous ROS (*Robotic Operation System*) peut servir pour une implémentation réelle [137].

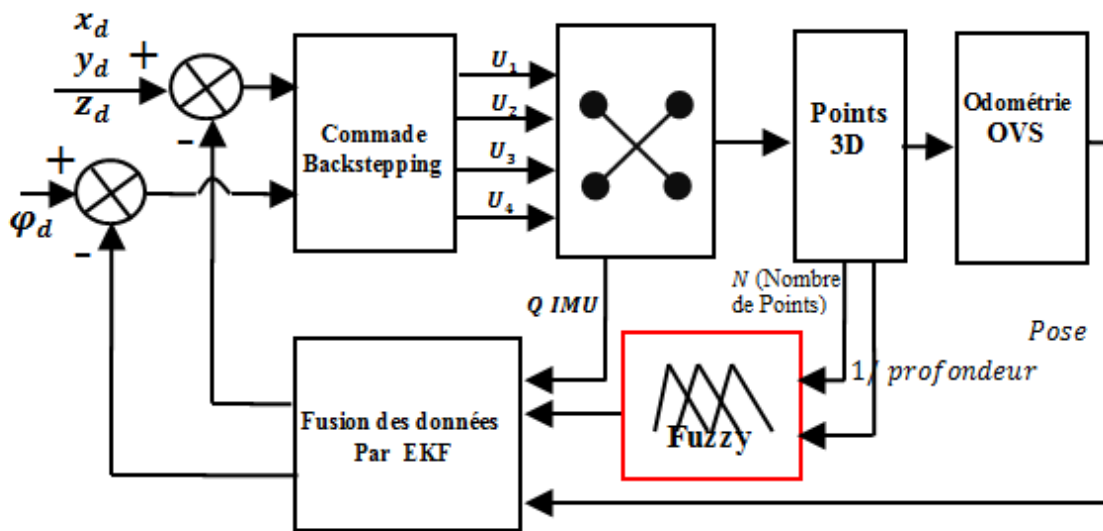


Figure IV-6 : Système de commande adaptative flou.

- Sous ensemble flous des entrées :

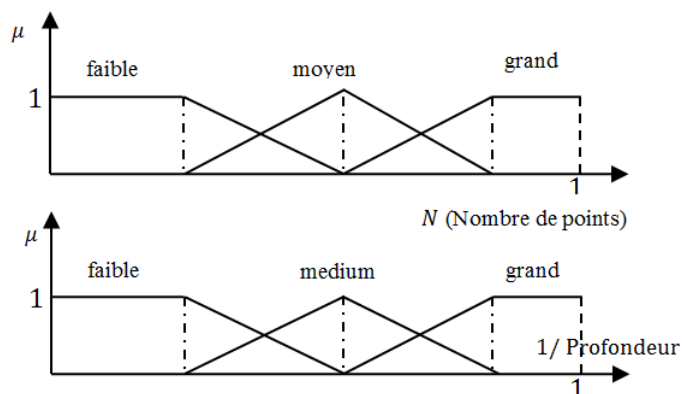


Figure IV-7 : Variables d'entrée et tracés des fonctions d'appartenance.

- Sous ensemble flous des sorties : la sortie de commande se caractérise par les variables linguistiques suivantes : IMU, OVS et Fusion.

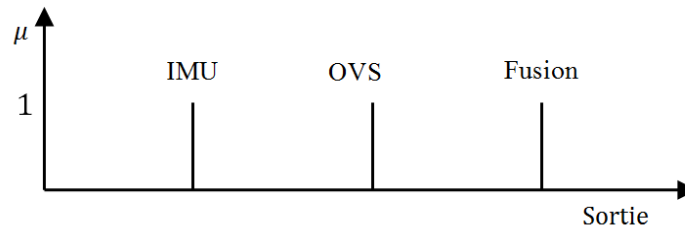


Figure IV-8 : Variables de sortie.

- Matrice d'inférence Floue :

1/ profondeur N (nombre de points)	faible	moyen	grand
faible	IMU	IMU	Fusion
moyen	IMU	Fusion	OVS
grand	Fusion	OVS	OVS

Tableau IV.1. Règles floues pour la fusion de données adaptative.

IV.7 Conclusion

Dans ce chapitre nous avons proposé une nouvelle stratégie de contrôle basée vision pour la poursuite de trajectoire par un quadrirotor. La stratégie proposée est basée principalement sur une approche de localisation par fusion de données inertielles et des données visuelle par filtre de Kalman étendu (EKF) et une approche de commande à base du contrôleur Backstepping à paramètre optimiser par l'algorithme PSO. La fusion IMU/OVS a permis d'assurer une localisation rapide et précise du quadrirotor.

Cependant, le système de localisation proposée est très sensible aux nombre et distances des points observés ainsi que la nature de l'environnement de navigation (statique/dynamique) ce qui affecte significativement la robustesse de l'EKF-IMU/SVO. Pour remédier à ce problème et pour rendre le système de localisation proposé efficace quel que soit l'environnement de navigation, une solution floue adaptative a été proposée pour prendre en considération les variations de l'environnement externe. Le contrôleur flou proposé consiste à ajuster automatiquement les paramètres de l'EKF pour pondérer correctement les performances de l'OVS vis-à-vis l'IMU.

L'approche proposée a été validée et de bons résultats ont été obtenus.

Chapitre V :

Résultats et Discussion

CHAPITRE V: RESULTATS ET DISCUSSIONS

V.1 Introduction

Les résultats obtenus sont présentés en trois parties. Tout d'abord, l'évaluation de l'approche de localisation visuelle sur une plateforme mobile a été implémentée, dans laquelle s'inscrit notre premier objectif pour valider le système de localisation proposé (OVS) en temps réel et avec des données réelles ainsi pour le généraliser par la suite dans le cas des quadrirotors. Deuxièmement, l'optimisation des paramètres de la commande par Backstepping par PSO, a été validée en simulation avec des comportements de suivi des trajectoires et ce, en utilisant l'odométrie visuelle pour localisation. Enfin, une analyse de robustesse du système de navigation a été réalisée (localisation et suivi de trajectoire) sous plusieurs scénarios (avec et sans perturbation) et différents types d'environnements ont été discutée pour valider les performances de notre approche proposée.

V.1 Validation du système de localisation visuelle (SVO)

Dans cette section, nous allons présenter, premièrement, quelques alternatives pour résoudre le problème d'estimation de la position et de l'orientation d'un Quadrirotor, avec un banc de caméra stéréoscopique en se basant sur un banc d'essai d'un robot mobile doté d'une caméra stéréoscopique et grâce à la mise en correspondances de points d'intérêt. Cependant, une évaluation expérimentale est nécessaire pour déterminer quelle est la méthode la plus adéquate pour une telle application sous certaines conditions spécifiques. Pour cela, des séries de tests ont été effectuées sur quelques cas avec des images réelles. En tenant compte du calibrage des deux camera dans le but d'estimer les trajectoires reconstituées du Quadrirotor, pour différentes trajectoires.

Deuxièmement, nous allons présenter les résultats de validation pour différents scénarios de navigation intérieure (*indoor*) et extérieure (*outdoor*) utilisant la localisation visuelle (Odométrie Visuelle Stéréoscopique) par les caméras *Bumblebee 2* embarquée sur le robot Pioneer 3AT[16].

Nous avons validé les algorithmes de l'Odomètre Visuel Stéréoscopique par ICP, SVD et Quaternion avec plusieurs scenarios : différentes trajectoires du banc de cameras stéréoscopiques *Bumblebee 2* à l'intérieure et à l'extérieure de la salle. Les premières validations ont été faites à l'arrêt (essai stationnaire), où le banc de cameras stéréoscopiques ne faisait aucun mouvement. Le deuxième test est effectué avec un mouvement ayant un unique

degré de liberté (une translation suivant l'un des axes du banc de la caméra). Dans le troisième test, nous avons choisi une trajectoire circulaire. Enfin, nous avons effectué des trajectoires de forme quelconque, aussi bien dans un environnement intérieur que dans un environnement extérieur.

V.1.1 Mise en œuvre de la plate-forme expérimentale

L'objectif est d'implémenter un système de localisation à moindre coût, robuste, précis et fiable, en utilisant uniquement la vision en évitant d'autre système de localisation tel que le GPS et les systèmes basés sur d'autres capteurs inertiels ou lasers. Notre but est de donner une fonctionnalité au robot mobile ou agent mobile pour faire la navigation autonome.

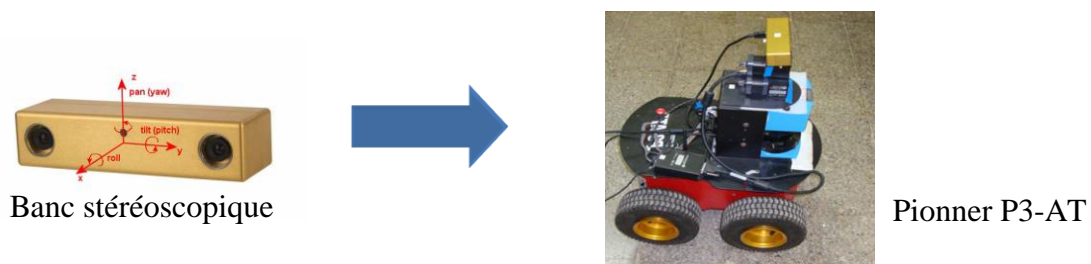


Figure V-1 : Constitution de notre plate-forme.

Le banc stéréoscopique Bumblebee®2 embarqué est doté d'un système logiciel pour la capture et le prétraitement des images peut être divisé en deux catégories, elles sont développées par la firme Point Grey Research. Le Digiclops™ est un système de stéréovision fournit en temps réel des images numériques pour des applications telles que le suivi visuel, la construction 3D, l'interface homme-machine et la robotique mobile. Le Triclops est un kit complet de développement logiciel (SDK) pour la vision 3D par ordinateur. Le SDK fonctionne avec Digiclops™[16].

V.1.2 SVO pour environnement intérieur (indoor)

Pour évaluer les performances de l'odomètre visuel proposé, nous avons choisi trois types d'environnement : un environnement interne les salles de laboratoire robotique (murs en béton, armoires métalliques...), un environnement externe (terrain accidenté).

Nous avons choisi comme environnement interne deux salles adjacentes. Les deux salles sont séparées par une porte au milieu elles s'étalent sur une superficie de 112 m² (14 m de longueur et 8 m de largeur).



Figure V-2 : Environnement interne[16].

Nous allons commencer à valider nos algorithmes dans un environnement intérieur, le robot mobile se déplacer dans un plan, le problème de calcul de la pose (OVS) se réduit à trois degrés de liberté deux translations et une orientation. Dans ce cas plusieurs scénarios sont envisagés pour tester les méthodes d'estimation de la pose, ainsi deux méthodes de filtrage sont élaborées pour éliminer les valeurs aberrantes (fausses valeurs) et affiner le calcul de la trajectoire.

V.1.2.1 Localisation du robot par l'Odomètre Visuel Stéréoscopique (trajectoire rectiligne)

Le tableau (V.1) montre les résultats d'estimation de la pose du robot par l'algorithme Odomètre Visuel Stéréoscopique (OVS) avec les différentes techniques d'optimisation implémentées (ICP, SVD et Quaternion) dans le cas de déplacement selon une trajectoire rectiligne suivant l'axe X , comparés au déplacement réel effectué. Les trajectoires estimées par ces méthodes sont présentées dans la figure (V-3).

		$X(mm)$	$Y(mm)$	$\theta(deg)$
Déplacement réel		16000	0	0
Odomètre	Déplacement estimé	15280	43	10
	Erreur	720	43	10
ICP	Déplacement estimé	15840	-182	4
	Erreur	160	182	4
SVD	Déplacement estimé	15639	-235	4
	Erreur	361	235	4
Quaternion	Déplacement estimé	15815	305	2
	Erreur	185	305	2

Tableau V.1 : Erreurs des différentes méthodes pour la trajectoire rectiligne.

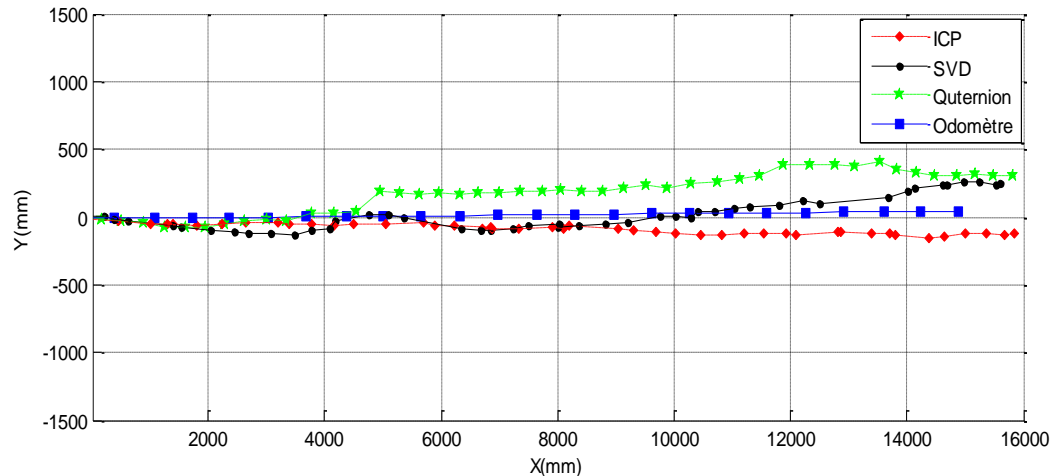


Figure V-3 : Résultats de la localisation par différentes méthodes pour la trajectoire rectiligne.

Dans cet essai, nous remarquons que la trajectoire donnée par l'odométrie dérive avec une erreur importante selon l'axe X par rapport aux algorithmes de localisation implémentés. Cela est dû à la grande distance parcourue, en outre l'algorithme ICP présente une meilleure estimation de la trajectoire effectué par le robot comparé aux autres algorithmes avec une erreur moins de 20 *cm* selon les deux axes.

Dans cet essai nous avons focalisé nos présentations seulement sur la localisation pour bien valider cette étape qu'est dite très importante dans la résolution du problème OVS. En fait une bonne localisation implique une bonne cartographie, cependant dans la suite nous allons montrer les résultats de la localisation et la cartographie simultanément (SLAM) pour des différents scénarios.

V.1.2.2 Trajectoires quelconques

Grâce à des expériences, l'algorithme d'odométrie visuelle proposée est validé sur une trajectoire complexe, la trajectoire accomplie par le robot a une forme de huit (∞). La figure (V-4) montre l'estimation de pose obtenue à l'aide d'un capteur linéaire (rouge) et de l'algorithme SVO (vert) ainsi que la carte 3D (points noirs) obtenue par le même algorithme. L'expérience est réalisée à l'intérieur dans deux salles de 12×7 m chacune. Les performances de l'algorithme SVO basé sur l'algorithme adaptatif SURF-ICP, en particulier lorsqu'un nombre important de bonnes caractéristiques 3D sont notées à chaque instant d'échantillonnage, sont illustrées à la Figure (V-5). L'algorithme adaptatif SURF-ICP a été comparé à deux autres algorithmes pour estimation de trajectoire : les algorithmes SVD et quaternion (Fig. V-5)[16].

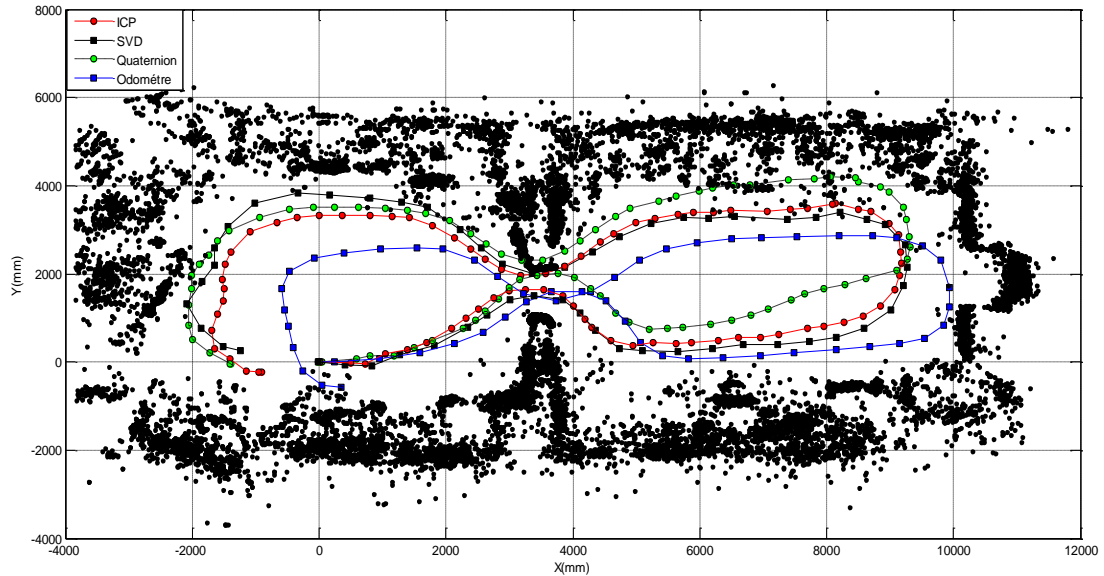


Figure V-4. Carte construite des deux salles de laboratoire.

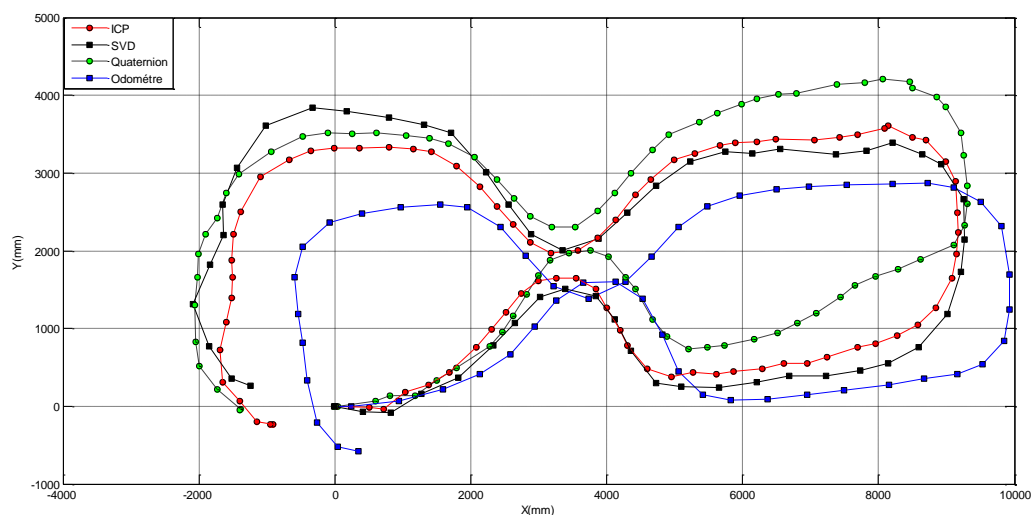


Figure V-5 : Trajectoire sous forme « infini » à l'aide de SVO.

V.1.3 Environnement extérieur (outdoor)

Pour l'environnement externe, nous avons travaillé dans un terrain irrégulier (figure V.6).

Dans cet environnement extérieur beaucoup d'hypothèses qui ont été fixées au préalable vont pas être respectées tels que : l'environnement statique, le nombre de degrés de liberté et les conditions d'acquisition des images. D'où l'augmentation des erreurs d'estimation de trajectoire par les algorithmes implémentés. Nous avons effectué l'essai dans un terrain de distance de 277m, avec des vitesses de translation et de rotation variables parce que nous avons commandé le robot avec un joystick. La figure (V.8) présente une vue de la scène outdoor prise par le robot. La figure illustre les deux étapes de détection et de mise en correspondance des points d'intérêt, ainsi que le suivi par SURF.

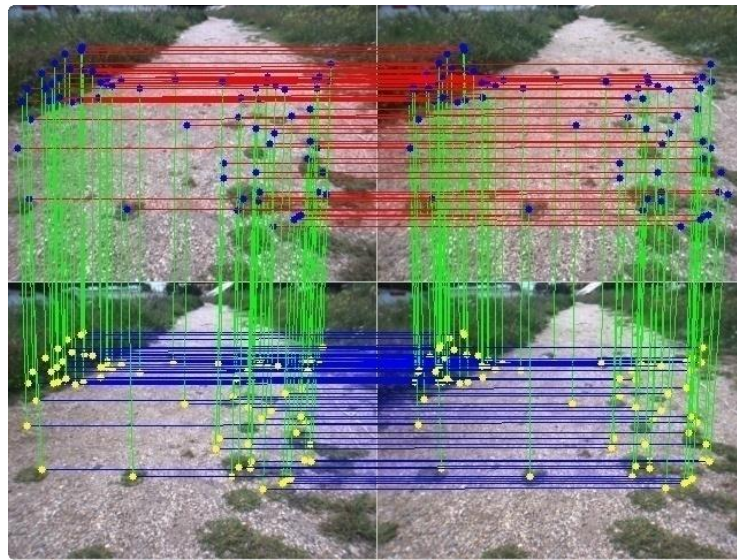


Figure V-6 : Matching des points d'intérêt dans l'environnement outdoor.

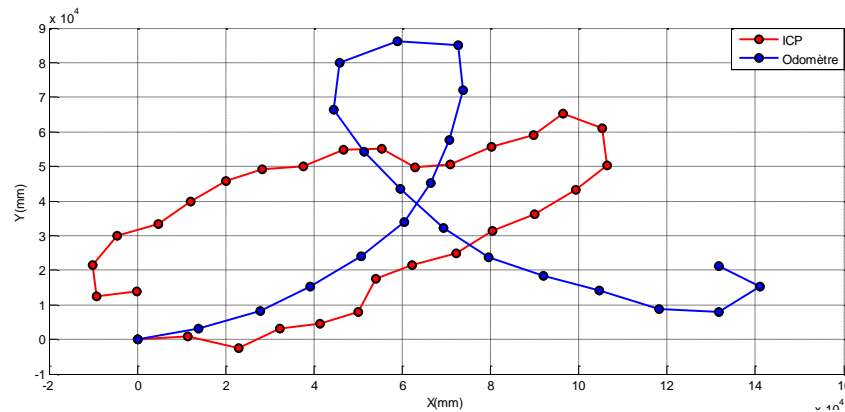


Figure V-7 : Résultat de localisation avec ICP.

V.2 Suivi de trajectoire par une commande backstepping

Pour évaluer les performances du suivi de trajectoire d'UAV à l'aide du commande backstepping et de la localisation visuelle, plusieurs formes de trajectoires et scénarios sont proposées (Figure V-8). Tout d'abord, Les formes suscitées auparavant telles que le carré, le cercle et la trajectoire sous forme quelconque sont validées et simulées par une commande backstepping avec réglage de ses paramètres optimisés par PSO. A ce titre, deux scénarios présentés par simulation dans cette thèse , figures V-8 et V-9, montrent un meilleur suivi et stabilité[2].

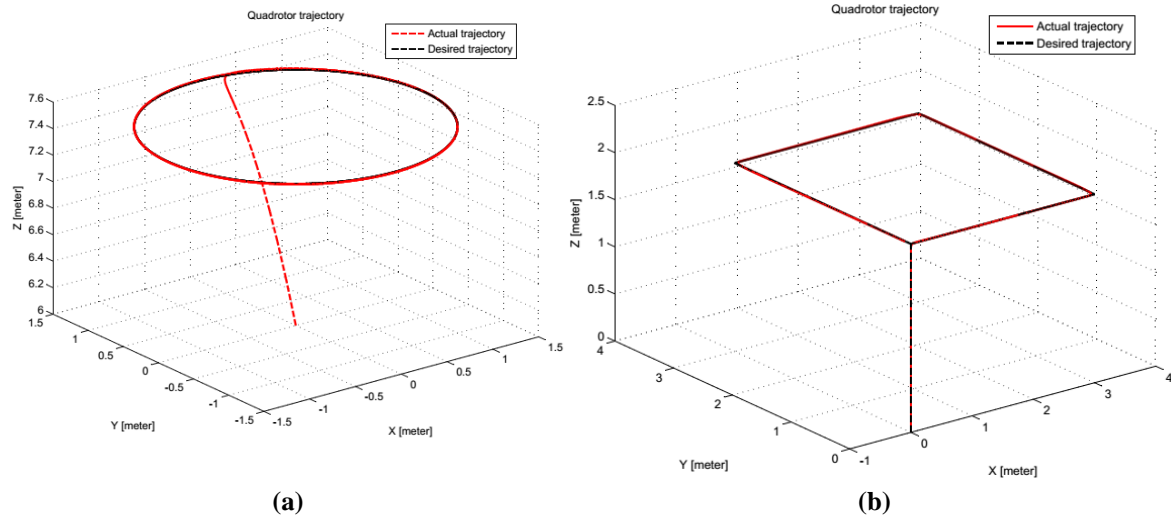


Figure V-8 : Différentes formes de trajectoires réalisées par la commande backstepping.

La figure V-8 montre que le quadrotor peut suivre la trajectoire souhaitée. Pour quantifier les résultats obtenus, les évolutions X, Y et Z pour les trajectoires circulaires et rectangulaires sont illustrées sur la Fig. V-9. En l'absence de perturbations, les erreurs de suivi obtenues sont similaires et acceptables.

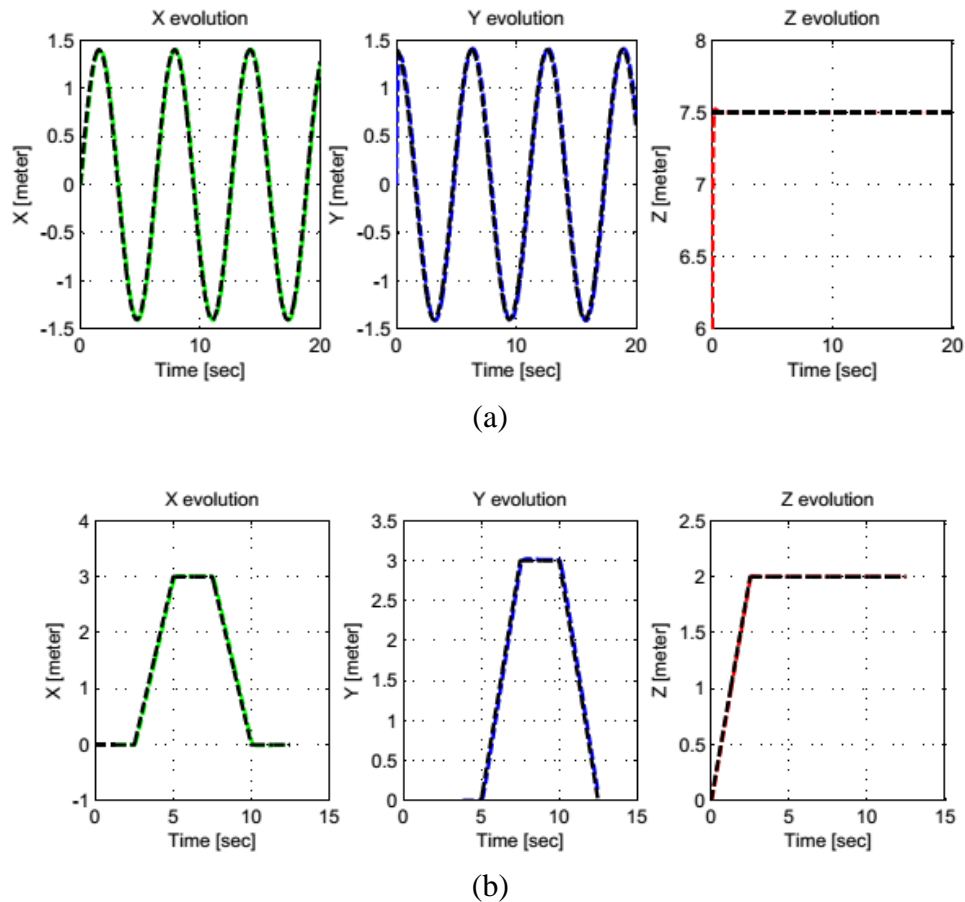


Figure V-9 : Erreur de suivi de trajectoire : (a) trajectoire circulaire (b) trajectoire rectangulaire.

V.3 Commande Backstepping basée vision

L'étude de cas présentée dans cette thèse vise à résoudre les défis rencontrés par les systèmes de vision dans la pratique, tels que le manque de points, les scènes homogènes (absence de texture), les scènes dynamiques et même le cas extrême quand l'information visuelle est complètement pas disponible. Dans ces cas critiques l'utilisation d'un capteur inertiel est très importante surtout pour l'estimation de l'attitude du quadrotor.

La pose visuelle est fusionnée avec la pose inertielle à l'aide d'un filtre EKF, et un système adaptatif flou est proposé pour sélectionner la meilleure stratégie de fusion (SVO/IMU) pour l'estimation de pose appropriée. La pose estimée est ensuite utilisée par un contrôleur de Backstepping Flou Adaptatif proposé pour suivre la trajectoire souhaitée.

Une analyse de robustesses profonde a été effectuée pour valider et évaluer les performances du Contrôleur Backstepping Flou Adaptatif utilisant un système de localisation EKF SVO/IMU. Des scènes de différents types sont considérées.

La figure V-10 illustre un exemple de deux nuages de points 3D aux instants $i = 1$ et $i = 10$; ces deux nuages sont supposés provenir du banc des caméras stéréoscopiques. En utilisant l'algorithme d'odométrie visuelle décrit ci-dessus, les points d'intérêt sont extraits ; par la suite, ils sont appariés à l'aide de l'algorithme SURF et l'estimation du mouvement réel est réalisée à l'aide de l'approche ICP Adaptatif.

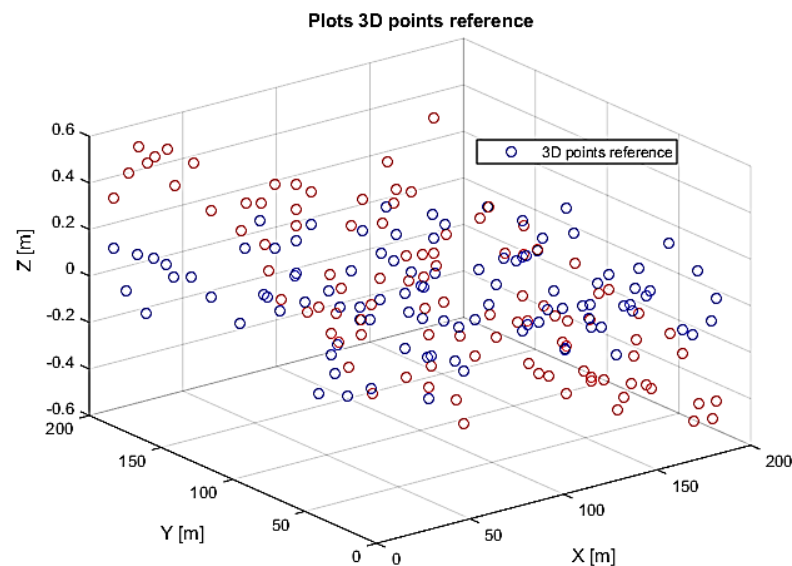
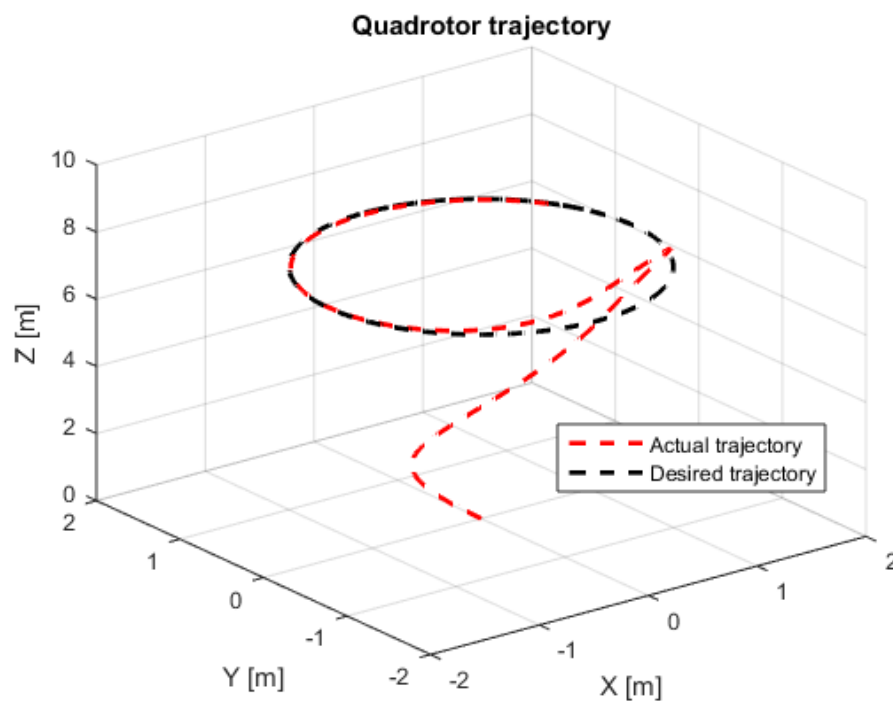


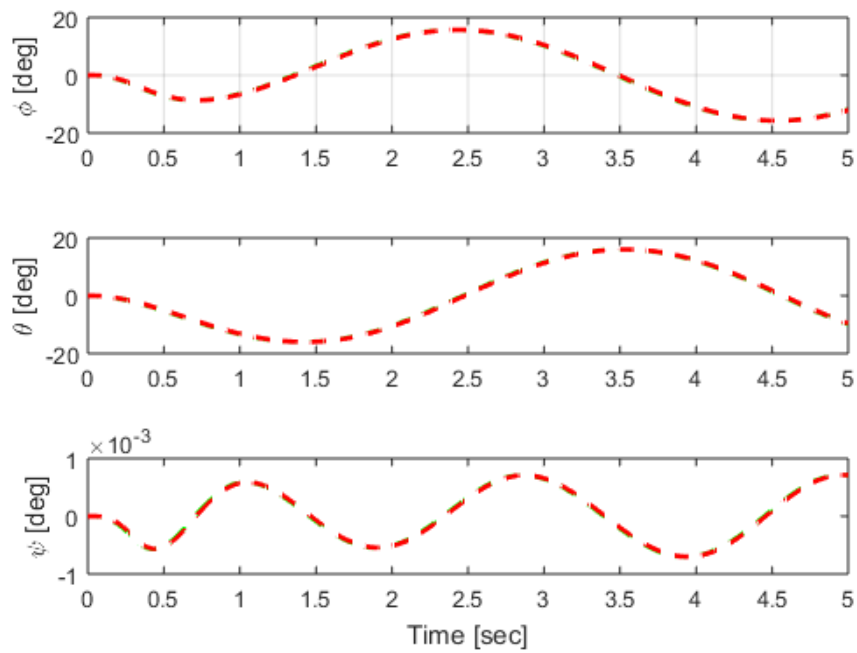
Figure V-10 : Points 3D de référence aux instants $i=1$ et $i=10$.

V.3.1 Scénario 1 : navigation d'un Quadrotor sans incertitude

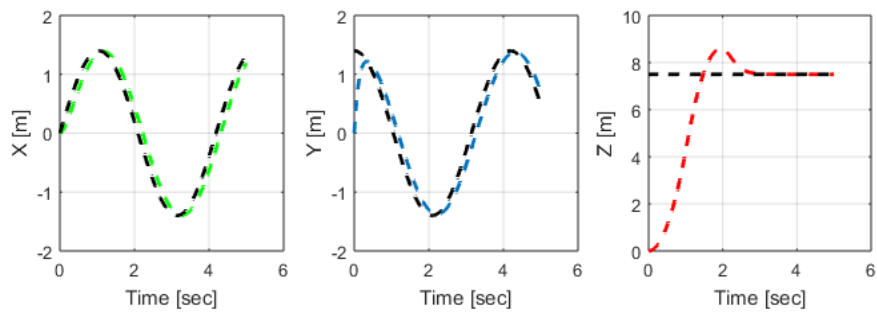
Les résultats de poursuite de trajectoire avec l'approche proposée utilisant 100 points d'intérêt sont présentés dans les figures V-11 (a) - (e). La figure V-11 montre les trajectoires souhaitées et réelles ; la figure V-11.b illustre les Angles de roulis, de tangage et de lacet de l'UAV ; la figure V-11.c illustre l'évolution des positions X, Y et Z ; les figures V-11.d et V-11.e illustre les entrées commande et les vitesses de rotation respectivement. Notez que l'angle de lacet a une plus petite variation par rapport aux angles de tangage et de roulis. En l'absence de bruit, un suivi de trajectoire fluide, une réponse parfaite du drone ainsi qu'une stabilité, précision et une vitesse satisfaisantes du quadrotor sont obtenus.



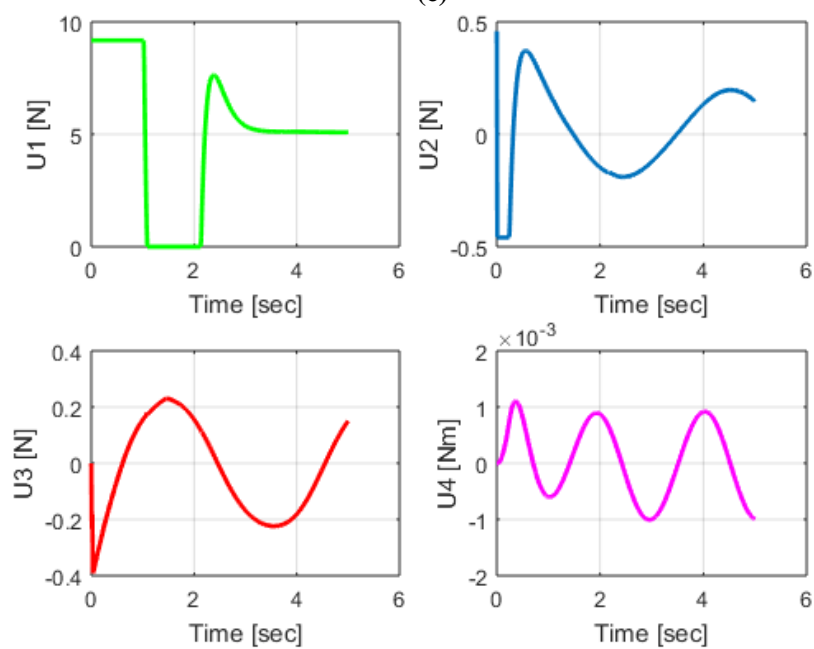
(a)



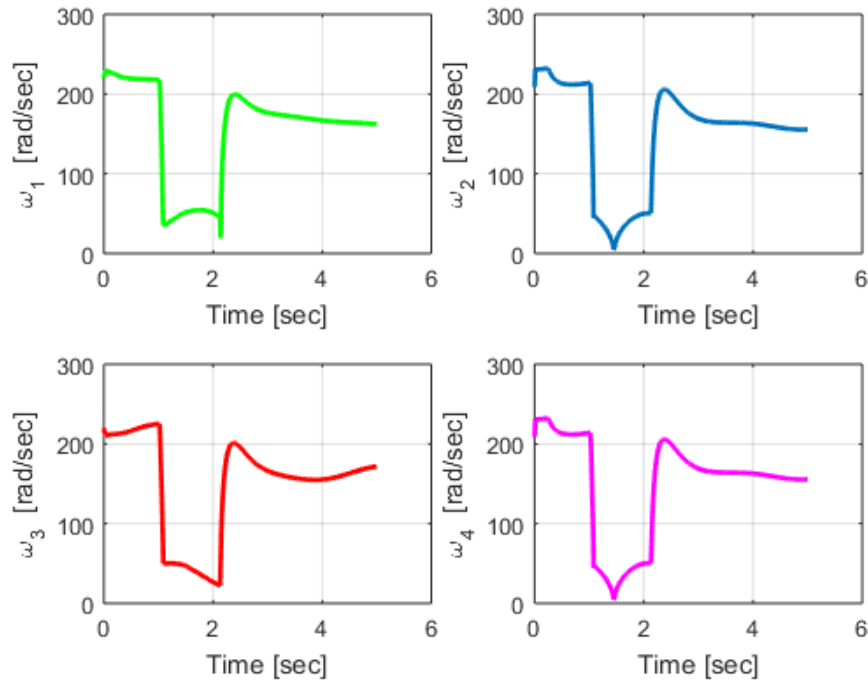
(b)



(c)



(d)



(e)

Figure V-11 : Suivi de trajectoire par odometre visuel idéal.

V.3.2 Scénario 2 : Effet de l'exactitude des points 3D sur la précision de l'OVS

Cette partie présente le suivi de trajectoire en considérant le bruit issu de la caméra embarquée sur le quadrirotor. Dans ce scénario, l'erreur quadratique moyenne (MSE) du suivi de trajectoire a été calculée en tenant compte que la variation du bruit et le calcul du temps de simulation écoulé. Les valeurs d'erreur sont résumées dans le tableau 3. Elles sont calculées en utilisant la formule suivante :

$$MSE(x_i) = \frac{1}{n} (x_i * x_i^T)^{1/2} \quad (V.1)$$

Où x_i est le vecteur d'erreur de suivi de chaque position et orientation. Le bruit de mesure considéré est un bruit blanc gaussien de moyenne μ et d'écart type $\sigma = bb$ (Equation V-2).

:

$$P_{out} = P_{cam_t} + bb * randn \quad (V.2)$$

Où $randn$ est le bruit blanc gaussien ; P_{out} est le point de référence dérivé du capteur de la caméra et affecté par les bruits ; et P_{cam_t} représente les points observés par le capteur de la caméra dans cette scène.

Tableau V.2. Effet des bruits sur la précision de l'OVS et la précision de la poursuite de la trajectoire.

Exp. No.	Taux de bruit (%)	MSE (x) (m)	MSE (y) (m)	MSE (z) (m)	Temps écoulé
1.	0.01 (1%)	0.0090	0.0115	0.1283	0.050132
2.	0.05 (5%)	0.0090	0.0115	0.1291	0.043893
3.	0.1 (10%)	0.0090	0.0115	0.1287	0.051088
4.	0.3 (30%)	0.0095	0.0118	0.1301	0.041462
5.	0.4 (40%)	0.0100	0.0131	0.1263	0.049947
6.	0.5 (50%)	0.0103	0.0124	0.1322	0.042025
7.	0.8 (80%)	0.0177	0.0149	0.1274	0.051184
8.	1 (100%)	0.0250	0.0186	0.1268	0.043292

Le tableau V-2 résume l'évolution de l'MSE face à différentes amplitudes de bruit. Dans l'expérience, le nombre de points d'intérêt est de 50 et le nombre d'itérations ICP est de 10.

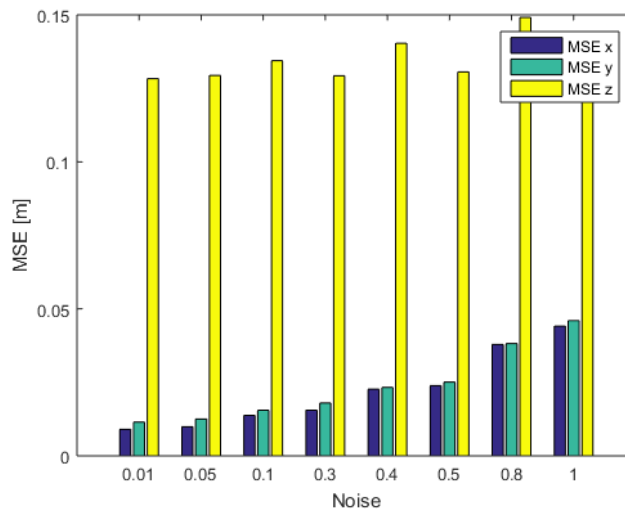
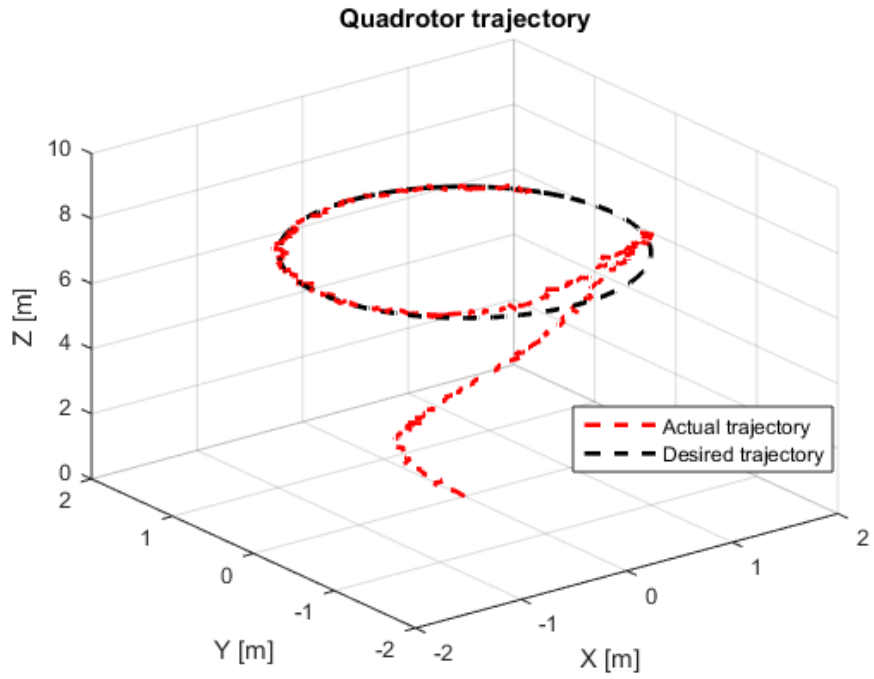


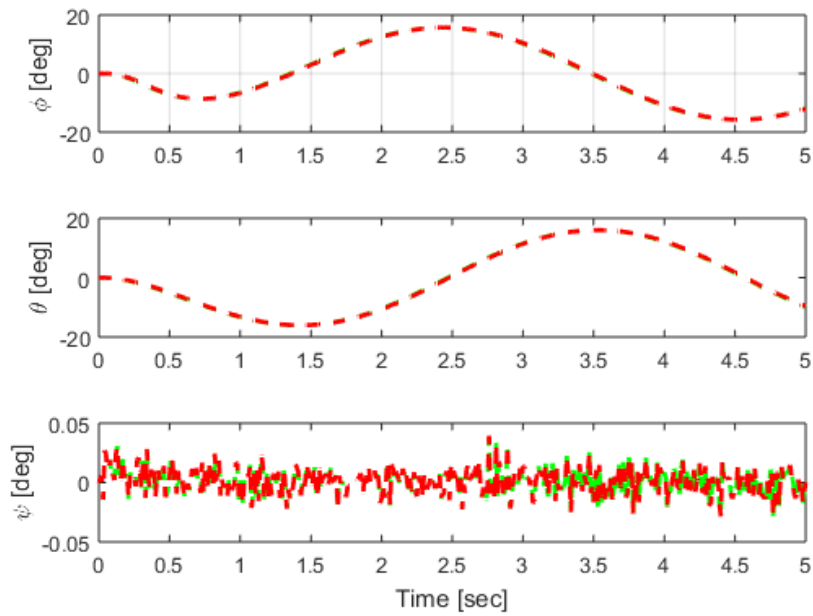
Figure V-12 . Evolution de l'erreur quadratique moyenne MSE selon la variation de l'amplitude du bruit.

Considérant les exemples suivants. Lorsque la valeur du bruit est de 10 % (ligne 3 du tableau V-2), la figure V-13, (a)–(e), montre que le quadrirotor suit la trajectoire souhaitée avec une précision appropriée. Comme l'illustre clairement la figure V-12, la meilleure précision est obtenue lorsque les axes X et Y sont suivis ($MSE(\text{en X et Y}) < 1,2 \text{ cm}$). En revanche, la MSE dépasse 12 cm lorsque le suivi est en l'axe Z. Comme le montre la Fig. V-13, (d) et (e), le bruit affecte de manière significative les moteurs sans balai (the brushless DC motors) et génère une vitesse considérable (ω_i) chattering.

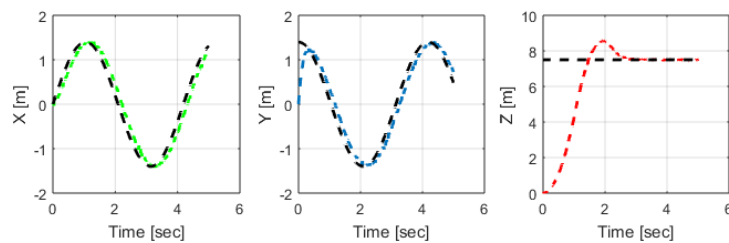
Ce résultat confirme l'importance du Superviseur flou utilisé qui a pu assurer le meilleur compromis entre SVO/IMU en ajustant adoptivement les paramètres de l'EKF pour obtenir une meilleure précision de localisation.



(a)



(b)



(c)

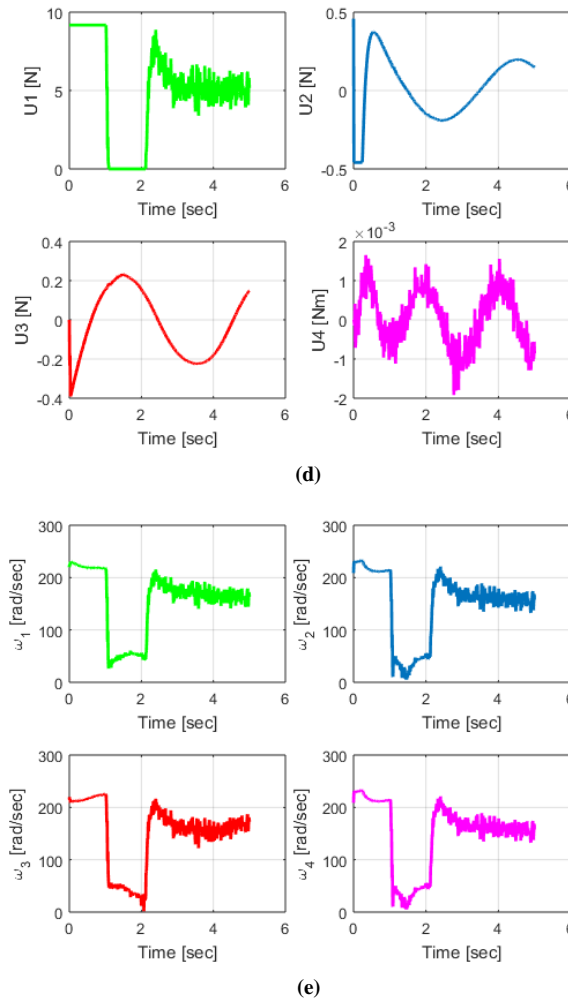


Figure V-13 . Effet de bruit (bruit = 0,1 (10%), nombre de points d'intérêts= 50 et nombre d'itérations ICP = 10).

V.3.3 Effet de nombre de points d'intérêts sur la précision de l'OVS

Le nombre de points caractéristiques 3D estimés obtenus par la caméra stéréovision a un impact significatif sur les performances de navigation du quadrotor. Le plus petit nombre de points d'intérêt 3D (minimum = 3) réduit le temps de calcul, mais diminue également la précision de la localisation. En revanche, une localisation satisfaisante et précise peut être obtenue avec un nombre important des points d'intérêt. Cependant, un temps de calcul substantiel est requis. Lors de la navigation, le nombre de véritables points 3D est inconnu car le quadrotor navigue dans différentes zones (par exemple, lorsque le quadrotor navigue devant un mur uniforme, le nombre de points SURF extraits est insuffisant et pas robuste. Des points précis et robustes sont nécessaires pour une localisation précise. Malheureusement, en pratique, ce n'est pas le cas, et plusieurs facteurs sont à considérer (scène lointaine, zone sombre, scène dynamique, etc.). L'utilisation de méthodes de filtrage, telles que RANSAC et filtre de Zhang,

devient essentiel pour la préparation des données avant l'utilisation. Dans ce but et pour surmonter la limitation SVO, les données inertielles du capteur IMU sont fusionnées avec la position visuelle à l'aide de l'algorithme EKF.

Le tableau V-3 résume l'effet du nombre des points d'intérêt 3D sur la précision de la navigation. Les résultats de simulation obtenus montrent la variation de l'erreur moyenne en fonction du nombre de points. Les erreurs obtenues selon X, Y et Z diminuent lorsque le nombre de points 3D augmente. La figure V-15, (a)–(e), montre les résultats obtenus pour le suivi de trajectoire à l'aide du contrôleur backstepping à base d'une localisation OVS/IMU Floue Adaptative.

Tableau V.3. Nombre de points d'intérêts effectifs (bruit = 0,2 ; nombre d'itérations ICP = 10).

Exp. No.	Nombre de points d'intérêts	MSE (x) (m)	MSE (y) (m)	MSE (z) (m)	Temps écoulé
1.	3	0.0277	0.0237	0.1588	0.042734
2.	5	0.0122	0.0147	0.1225	0.044257
3.	7	0.0108	0.0128	0.1275	0.044753
4.	10	0.0102	0.0141	0.1236	0.043687
5.	15	0.0099	0.0120	0.1242	0.043321
6.	20	0.0094	0.0120	0.1281	0.044868
7.	50	0.0088	0.0114	0.1283	0.045458
8.	100	0.0095	0.0116	0.1290	0.046161

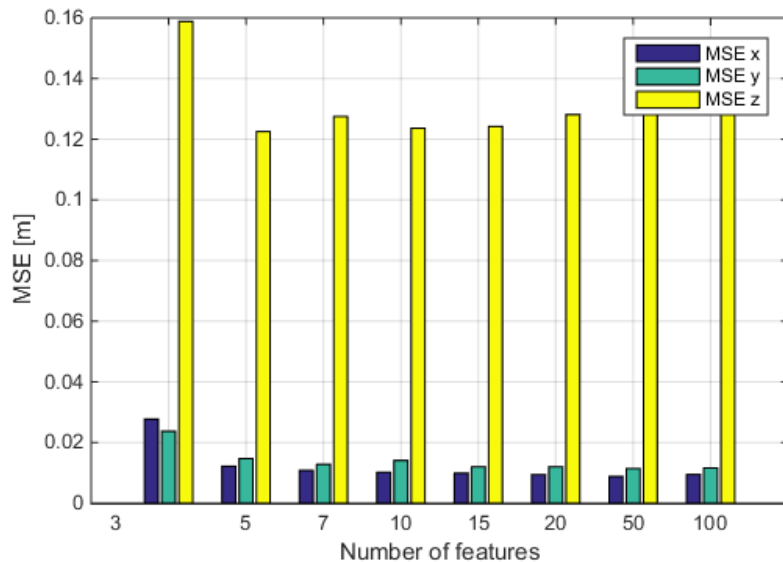
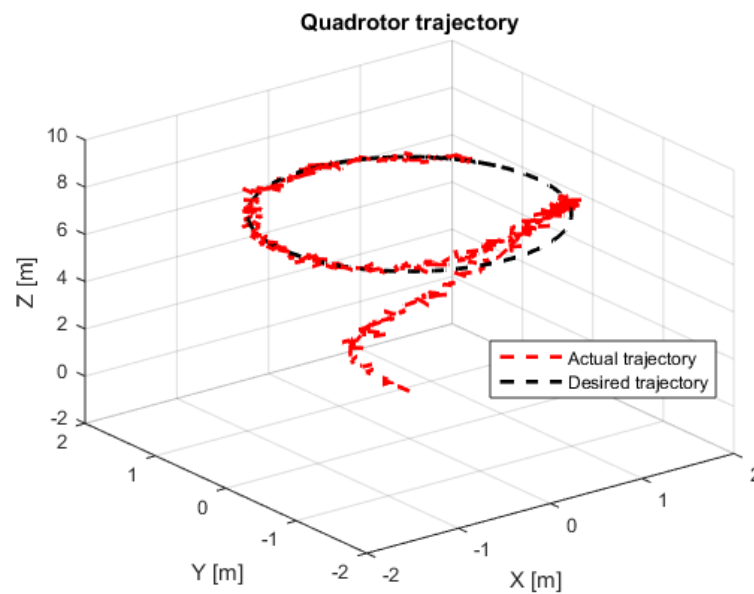


Figure V-14 . MSE en fonction du nombre des points d'intérêts (bruit = 0,2).

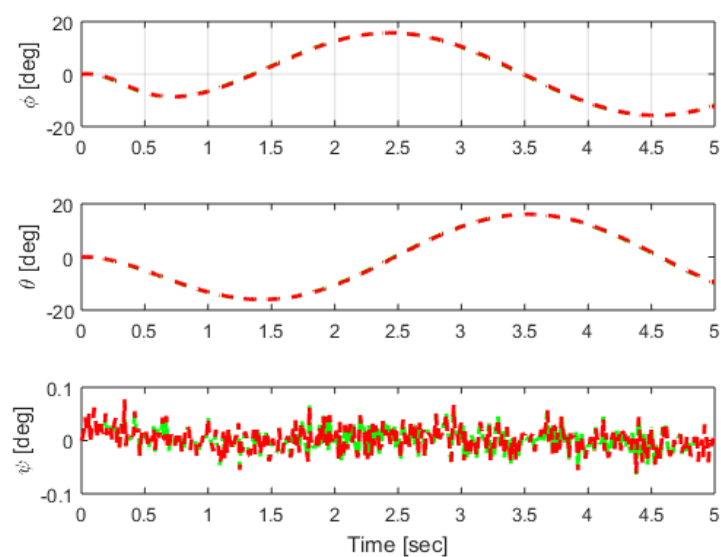
Le tableau 4 résume l'évolution de la MSE avec différents nombres de points caractéristiques ; 20 % du bruit affecte les points d'intérêts, estimé avec 10 itérations d'ICP. Ces résultats démontrent que l'algorithme proposé est robuste et ne nécessite qu'un nombre

suffisant de points (égal ou supérieur à trois) dans l'environnement pour reconstruire et estimer la trajectoire du drone. Comme l'illustre clairement la Fig. 24, la meilleure précision est obtenue lorsque les axes X et Y sont suivis ($MSE < 1,28$ cm).

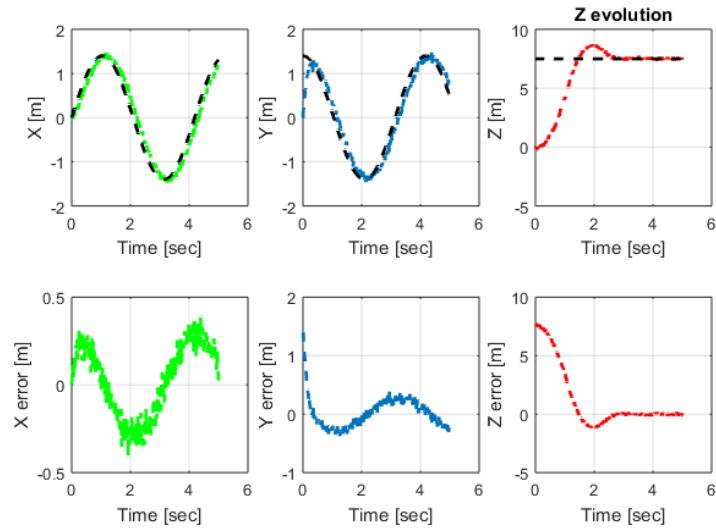
Le tableau V-3 résume l'évolution de l'MSE avec différents nombres de points 3D affecté par un bruit de 20 % et pour 10 itérations d'ICP. Ces résultats démontrent que l'algorithme proposé est robuste même avec un nombre réduit des points. Comme l'illustre clairement la Fig. V-14, la meilleure précision est obtenue lorsque les axes X et Y sont suivis (MSE (en X et Y) $< 1,2$ cm).



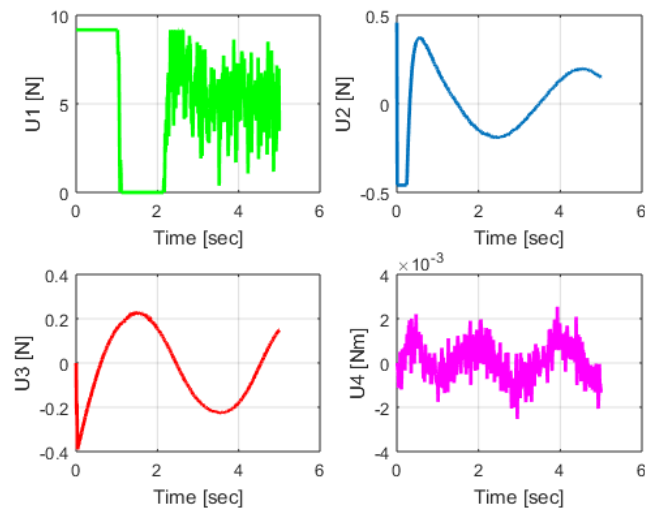
(a)



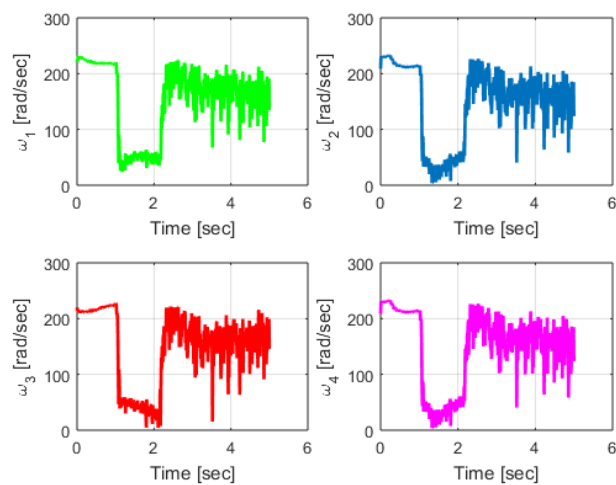
(b)



(c)



(d)



(e)

Figure V-15 : Effet de points d'intérêts (nombre de points d'intérêts = 10, bruit = 0,1 et nombre d'itérations ICP = 10).

V.4 Comparaison des commandes à Logique Floue et Backstepping

Les méthodes proposées pour une commande par backstepping optimal basé sur le suivi de trajectoire sont comparées à une commande à logique floue (FLC). Ce dernier ne nécessite aucune information sur le modèle, ce qui le rend adapté à une navigation incertaine. Quatre (04) contrôleurs flous de type Takagi Sugeno Type 1 sont mis en œuvre pour le contrôle du roulis, tangage, lacet et l'altitude. Les résultats de suivi de trajectoire à l'aide de la commande FLC sont donnés sur la Fig.V-16. La comparaison des RMSE (*Root Mean Square Error*) entre le PSO-Backstepping et la commande FLC est donnée dans le tableau.V-4. Comme on peut le voir, l'approche proposée est plus précise que l'approche floue selon l'axe Z (Figure V-16), des performances similaire pour les deux axes X et Y, en outre, moins de temps de calcul est nécessaire pour suivre la trajectoire par le Backstepping optimal. Pour les deux cas le système de localisation utilisé est basé sur EKF-SVO/IMU.

Tableau V.4: Comparaison des contrôleurs Backstepping vs FLC

Contrôleur	RMSE	X- RMSE	Z- RMSE	Z- RMSE	Temps du calcul [ms] ^o
Backstepping optimal		3.9587	4.2951	2.1749	26.00
Commande floue		3.9587	4.2951	3.0595	55.50

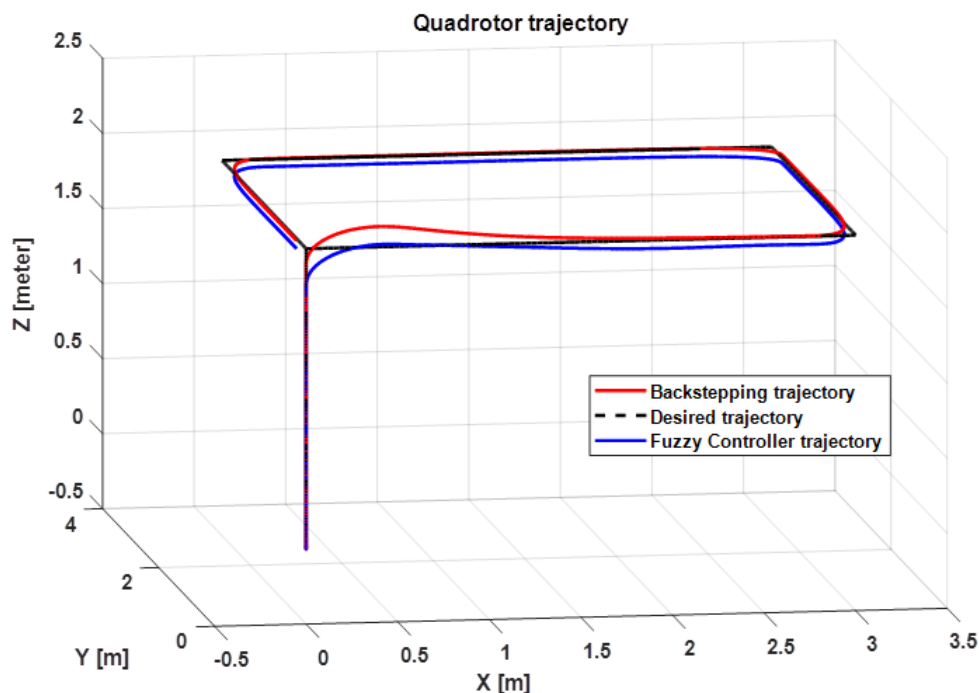


Figure V-16 : Trajectoire du quadrirotor par le backstepping vs la commande FLC

V.5 Conclusion

Dans ce chapitre plusieurs scénarios pour la validation des algorithmes d'odométrie visuelle stéréoscopique ont été élaborés afin de calculer le déplacement (position et l'orientation) d'un UAV. À partir de la mise en correspondance des points d'intérêts des images acquises et la reconstruction 3D par triangulation géométrique, nous avons validé les algorithmes d'OVS pour la localisation ainsi que la construction de la carte de l'environnement.

Dans un premier temps, nous avons quantifié et évalué les erreurs dans le cas statique (scène statique et le banc stéréoscopique est fixe). Les résultats obtenus des trois méthodes d'OVS par ICP, Quaternions et SVD sont satisfaisant.

Dans une deuxième étape les positions estimées dans l'étape précédente servent comme une localisation dans la boucle de retour pour la commande Backstepping dans le but de suivre une trajectoire par l'UAV.

Une analyse profonde de robustesse a été effectuée pour évaluer les performances de l'approche proposée face au nombre et incertitudes des points 3D observés. Nous avons confirmé la robustesse de l'approche même avec un nombre réduit des points 3D. Ce résultat confirme l'importance du Superviseur flou utilisé qui a pu assurer le meilleur compromis entre SVO/IMU en ajustant adoptivement les paramètres de l'EKF pour obtenir une meilleure précision de localisation.

Finalement, la commande Backstepping optimale a été comparée avec une commande basée sur la logique floue type Sugeno (FLC). Les résultats obtenus montrent une meilleure précision de la commande Backstepping surtout pour la poursuite du Z ainsi qu'un temps de calcul faible par rapport à l'FLC.

Conclusion générale

CONCLUSION GENERALE

L'objectif de ce travail est de développer un système de navigation UAV basé sur une commande Backstepping adaptative floue en utilisant l'odométrie visuelle. Tout d'abord, afin d'obtenir une localisation précise du quadrotor, un algorithme d'odométrie stéréo-visuelle adaptative (SVO) est proposé pour localiser l'UAV via les images acquises. Ensuite, et dans le but de renforcer la robustesse de cette localisation un filtre de Kalman étendu (EKF) / le filtre à structure variable glissant (*Smooth Variable Structure Filter*, SVSF) sont implémentés pour la fusion de données SVO / IMU pour fournir une pose précise de l'UAV même en cas d'absence des données visuelle. Deuxièmement, un contrôleur flou adaptatif est proposé pour fournir la pose d'UAV appropriée en considérant le nombre et la profondeur des partitives 3D observées. Enfin, une stratégie de contrôle non linéaire utilisant la commande par Backstepping a été développée pour assurer une convergence rapide et un suivi précis de la trajectoire de référence souhaitée. Les paramètres la commande par Backstepping sont réglées à l'aide de l'algorithme d'optimisation de l'essaim de particules (PSO) pour obtenir des performances de contrôleur optimales.

Le contrôleur flou proposé assure d'une façon adaptative un meilleur compromis précision/robustesse. Dans le cas où le nombre des points clefs est important dans des distances acceptable le FLC donne plus d'importance à l'OVS par rapport au capteur inertiel. Le FLC exécute ces actions en adaptant les matrices de covariance des bruits d'état et des bruits de mesure de sorte que l'EKF porte beaucoup plus sur le système de localisation le plus confident.

Les algorithmes proposés ont été validé en simulation et avec des données expérimentales. Premièrement, une analyse de précision de l'OVS a été faite pour l'exploiter par la suite par le contrôleur flou. L'OVS a été validé avec des images réelles sur par le robot mobile Pioneer 3AT.

Les résultats de suivi de trajectoire ont été évalués aussi à l'aide de données de simulation avec des scénarios réalistes ; des résultats satisfaisants ont été obtenus. Une analyse profonde de robustesse a été effectuée pour évaluer les performances de la solution proposée (Estimation de la pose et suivi de la trajectoire par l'UAV) face aux : Bruit sur les points clefs, variation du nombre des points, ainsi la précision des points 3D observés. Les résultats dérivés correspondant à différents scénarios qui ont montré l'efficacité de l'approche proposée et l'importance de l'hybridation visuelle/inertielle qui a permis de maintenir une localisation précise et robuste même en absence des données visuelle. Le suivi de la trajectoire par la

commande par Backstepping Optimale a donné de très bonnes performances par rapport aux commandes classiques.

PERSPECTIVES

Dans les travaux futurs, on vise à implémenter l'approche proposée sur système embarqué (Raspberry PI ou FPGA) pour valider simultanément la localisation et le suivi de la trajectoire de l'UAV. D'autres champs ouverts pour la continuation de ces travaux de thèse tournent autour les perspectives suivantes :

- Mettre en place une plateforme d'évaluation de l'algorithme OVS dans le cas des robots volants (UAVs) ;
- Optimiser les algorithmes et exploiter la bibliothèque OpenCV sous ROS pour les exécuter en temps réel ;
- Utiliser une autre variante des capteurs (de profondeur par exemple) qui fournit à la fois des images RGB et des images de profondeur, pour améliorer et minimiser les erreurs de modélisation de capteur stéréoscopique.

Bibliographie

BIBLIOGRAPHIE

- [1] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for UAVs: Current developments and trends," *Journal of Intelligent & Robotic Systems*, vol. 87, pp. 141-168, 2017.
- [2] A. Boucheloukh, F. Boudjema, N. Abdelkrim, F. Demim, and F. Yacef, "UAV navigation based on adaptive fuzzy backstepping controller using visual odometry," *International Journal of Modelling and Simulation*, pp. 1-22, 2021.
- [3] A. ALLAM, M. TADJINE, E. KOBZILI, and A. BOUCHELOUKH, "3D Robots Formation cooperative Control Using EFDs," *3rd IEEE International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pp. 1 - 6, 2018.
- [4] K. Schmid, T. Tomic, F. Ruess, H. Hirschmuller, and M. Suppa, "Stereo vision based indoor/outdoor navigation for flying robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3955–3962, 2013.
- [5] J.-J. Lai, C.-C. Chang, and F.-L. Lian, "Adaptive object tracking of quadrotor in presence of disturbance," in *2019 12th Asian Control Conference (ASCC)*, 2019, pp. 667-672.
- [6] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy, "Computer vision based general object following for gps-denied multirotor unmanned vehicles," in *2014 American Control Conference*, 2014, pp. 1886-1891.
- [7] M. Nieuwenhuisen, D. Droschel, M. Beul, and S. Behnke, "Obstacle detection and navigation planning for autonomous micro aerial vehicles," in *2014 international conference on unmanned aircraft systems (ICUAS)*, 2014, pp. 1040-1047.
- [8] H. D. Escobar-Alvarez, N. Johnson, T. Hebble, K. Klingebiel, S. A. P. Quintero, J. Regenstein, and N. A. Browning, "R-ADVANCE: Rapid Adaptive Prediction for Vision-based Autonomous Navigation, Control, and Evasion," *Journal of Field Robotics*, vol. 35, pp. 91-100, 2018.
- [9] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit, "The Navigation and Control technology inside the AR.Drone micro UAV," *IFAC Proceedings Volumes*, vol. 44, pp. 1477-1484, 2011/01/01/ 2011.
- [10] K. Máthé and L. Buşoniu, "Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection," *Sensors*, vol. 15, pp. 14887-14916, 2015
- [11] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, pp. 65–100, 2010.
- [12] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles," *IEEE Control systems magazine*, vol. 33, pp. 61-75, 2013.
- [13] T. P. Nascimento and M. Saska, "Position and attitude control of multi-rotor aerial vehicles: A survey," *Annual Reviews in Control*, vol. 48, pp. 129-146, 2019/01/01/ 2019.
- [14] F. Wang, W. Chen, H. Dai, J. Li, and J. Jia, "Backstepping control of a quadrotor unmanned aerial vehicle based on multi-rate sampling," *Science China Information Sciences*, vol. 62, p. 19203, 2019.
- [15] Y. Jung-Hua and H. Wen-Chun, "Adaptive backstepping control for electrically driven unmanned helicopter," *Control Eng Practice*, vol. 17, pp. 903–913, 2009.
- [16] A. Boucheloukh, F. Boudjema, A. Nemra, F. Demim, and R. Louali, "Stereo Visual Odometry for Mobile Robot," in *2018 3rd International Conference on Pattern*

- Analysis and Intelligent Systems (PAIS)*, 2018, pp. 1-7.
- [17] A. Nemra and N. Aouf, "Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering," *IEEE Sensors Journal*, vol. 10, pp. 789 - 798, 2010
- [18] F. Demim, A. Boucheloukh, A. Nemra, K. Louadj, M. Hamerlain, A. Bazoula, and Z. Mehal, "A new adaptive smooth variable structure filter SLAM algorithm for unmanned vehicle," in *In IEEE 6th International Conf on Systems and Control (ICSC)*, 2017, pp. 6-13.
- [19] F. Demim, A. Nemra, K. Louadj, A. Boucheloukh, E. Kobzili, A. Allam, M. Hamerlain, and A. Bazoula, "Visual SVSF-SLAM Algorithm Based on Adaptive Boundary Layer Width," Cham, 2019, pp. 97-112.
- [20] John William Ransom Taylor and K. Munson, "Jane's pocket book of remotely piloted vehicles: Robot aircraft today."
- [21] K. P. WERRELL, "A. M. Low FLIGHT, 3 October 1952 page 436 "The First Guided Missile," 1985
- [22] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*: Cambridge university press, 2006.
- [23] G. Cai, J. Dias, and L. Seneviratne, "A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends," *Unmanned Systems*, vol. 2, pp. 175-199, 2014.
- [24] S. Bouabdallah and R. Siegwart, "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2259-2264, 2005.
- [25] A. T. Tran, N. Sakamoto, M. Sato, and K. Muraoka, "Control augmentation system design for quad-tilt-wing unmanned aerial vehicle via robust output regulation method," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, pp. 357-369, 2017.
- [26] G.FRAPPIER, "Système inertiels de navigation pour robots mobiles " *Séminaire, EC2, Paris*, 1990.
- [27] A. COURCELLE, "Localisation d'un robot mobile : Application à l'aide à la mobilité des personnes handicapées moteur," *Doctorat de l'université de METZ, France*, Janvier 2000.
- [28] S.NOGUEIRA, "Localisation de Mobiles Par Construction de Modeles 3d En Utilisant la Stéréovision," doctorat, l'université de Belfort-Montbéliard, décembre 2009.
- [29] G. Mohamed, M. Samir, and B. Mohamed, "Implémentation d'estimateurs pour la mise en œuvre d'un AHRS pour Quadrotor," *Conférence de génie électrique CGE 2011, École Militaire Polytechnique (EMP)*, 2011.
- [30] A. O. DJEKOUNE, "Localisation Et Guidage Du Robot Mobile Atrv2 Dans Un Environnement Naturel," doctorat, Université des Sciences et de la Technologie HOUARI BOUMEDIENE, décembre 2010.
- [31] LABUHADROUS, "Système embarqué temps réel de localisation et de modélisation 3D par fusion multi capteur," *Thèse de doctorat, Ecole des Mines de Paris*, 14 Janvier 2005.
- [32] D. FILLIAT, "Robotique Mobile," École Nationale Supérieure de Techniques Avancées ParisTech, 5 octobre 2011.
- [33] R.Drouilly, "SLAM Visuel 3D pour robot mobile autonome," Master de Sciences Université de Strasbourg, Septembre 2011.
- [34] O. Araar and N. Aouf, "Visual servoing of a quadrotor uav for autonomous power lines inspection," in *22nd Mediterranean Conference on Control and Automation*, 2014, pp. 1418-1424.
- [35] H. Jabbari, G. Oriolo, and H. Bolandi, "An adaptive scheme for image-based visual servoing of an underactuated UAV," *International Journal of Robotics and Automation*,

- vol. 29, pp. 92-104, 2014.
- [36] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised* vol. 118: Springer, 2017.
- [37] V. Ghadiok, J. Goldin, and W. Ren, "On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor," *Autonomous Robots*, vol. 33, pp. 41-68, 2012.
- [38] B. Zhao, Y. Tang, C. Wu, and W. Du, "Vision-based tracking control of quadrotor with backstepping sliding mode control," *IEEE Access*, vol. 6, pp. 72439-72448, 2018.
- [39] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "FASTER: Fast and Safe Trajectory Planner for Flights in Unknown Environments," *arXiv preprint arXiv:2001.04420*, 2020.
- [40] C.DROCOURT, "Localisation et Modélisation de l'Environnement d'un Robot Mobile par Coopération de deux Capteurs Omnidirectionnels " doctorat, université de technologie de compiègne, février 2002.
- [41] A. ZUREIKI, "Fusion de Données Multi-Capteurs pour la Construction Incrémentale du Modèle Tridimensionnel Texturé d'un Environnement Intérieur par un Robot Mobile," Doctorat, Université de Toulouse, septembre 2008.
- [42] Y.RAOUI, "Indexation d'une base de données images: Application à la localisation et la cartographie fondées sur des radio-étiquettes et des amers visuels pour la navigation d'un robot en milieu intérieur," doctorat, université de Toulouse, avril 2011.
- [43] A.NEMRA, "Robust Airdrone 3D Visual Simultaneous Localization And Mapping," PHD, Cranfield Defence and Security, Cranfield University, 2011.
- [44] S. T. Michael Montemerlo, Daphne Koller, Ben Wegbreit, "FastSLAM : A factored solution to the simultaneous localization and mapping problem," *In AAAI National Conference on Artificial Intelligence*, pp. 593–598, 2002.
- [45] T. B. Hugh Durrant-Whyte, "Simultaneous localisation and mapping (SLAM) : Part I the essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 99-110, 2006.
- [46] C. A. I. Tykkala T "A dense structure model for image based stereo slam," *In IEEE International Conference on Robotics and Automation*, pp. 1758 -1763, 2011.
- [47] P. N. Mark Cummins, "FAB-MAP : Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, pp. 647-665, 2008.
- [48] S. G. G. Grisetti, C. Stachniss, P. Pfaff, W. Burgard, , "Efficient estimation of accurate maximum likelihood maps," *in 3d. In IEEE International Conference on Intelligent Robots and Systems*, vol. 16, pp. 3472 -3478, 2007.
- [49] T. P. D.Martinec, "Structure from many perspective images with occlusions," *In European Conference on Computer Vision*, pp. 355-369, 2002.
- [50] P. M. Bill Triggs, Richard Hartley, Andrew Fitzgibbon, "Bundle Adjustment " *A Modern Synthesis*, vol. 1883, 2000.
- [51] J. P. E. L. a. E. I. C. P. Yasutaka Furukawa Google Inc. Seattle, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1362-1376, 2010.
- [52] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, pp. I-652-I-659 Vol.1.
- [53] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 3946-3952.
- [54] A. I. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," in *The*

- International Journal of Robotics Research*, 2010, pp. 245-266.
- [55] M. MEILLAND, "Cartographie RGB-D dense pour la localisation visuelle temps-réel et la navigation autonome," *Thèse Doctorat ParisTech, École nationale supérieure des mines de Paris*, Apr-2012.
- [56] M. S. Chris Harris, "A combined corner and edge detector," *In Proceedings of the 4th Alvey Vision Conference*, pp. 147-151, 1988.
- [57] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91-110, 2004.
- [58] T. T. Herbert Bay, Luc Van Gool, "SURF: Speeded Up Robust Features," presented at the Proceedings of the ninth European Conference on Computer Vision, Mai 2006.
- [59] R. C. B. Martin A. Fischler, "Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography," *ACM Communications*, vol. 24, pp. 381-395, June 1981.
- [60] T. K. Bruce D. Lucas, "An iterative image registration technique with an application to stereo vision," *In International joint conference on Artificial intelligence*, pp. 674-679, 1981.
- [61] S. B. Christopher Mei, Ezio Malis, Patrick Rives, "Constrained multiple planar template tracking for central catadioptric cameras," *In British Machine Vision Conference*, 2006.
- [62] C. T. Jianbo Shi, "Good features to track," *In IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 593-600, June 1994.
- [63] E. M. Guillaume Caron, El Mustapha Mouaddib, "Tracking planes in omnidirectional stereovision," *In IEEE International Conference on Robotics and Automation*, pp. 6306-6311, 2011.
- [64] T. Hamel, Mahony .R, Lozano .R and Ostrowski .J, "Dynamic Modeling and Configuration Stabilization for an X4-Flyer," *15th Triennial World Congress, IFAC, Barcelona, Spain*, 2002.
- [65] T. Madani and A. Benallegue, "Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles," *in American Control Conference (ACC)*, pp. 5887-5892, 2007.
- [66] F. Kendoul, "Modélisation et commande de véhicules aériens autonomes, et développement d'un pilote automatique basé sur la vision," Compiègne, 2007.
- [67] J. Wang, "Quadrotor analysis and model free control with comparisons," Université Paris Sud-Paris XI, 2013.
- [68] H. Bouadi, M. Bouchoucha, M. Tadjine "Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor," *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, vol. 1, 2007.
- [69] S. Seghour, M. Bouchoucha, H. Osmani "From Integral backstepping to integral sliding mode attitude stabilization of a quadrotor system: Real time implementation on an embedded control system based on a dsPIC μ C," *in Mechatronics (ICM), 2011 IEEE International Conference on*, pp. 154-161, 2011.
- [70] M. TAIRA, K.-I. TSUTSUI, M. JIANG, K. YARA, and H. SAKATA, "Parietal neurons represent surface orientation from the gradient of binocular disparity," *Journal of Neurophysiology*, vol. 83, pp. 3140-3146, 2000.
- [71] S. T. BARNARD and M. A. FISCHLER, "Computational stereo," *ACM Computing Surveys (CSUR)*, vol. 14, pp. 553-572, 1982.
- [72] J.-Y. Bouguet. (2010). *Camera Calibration Toolbox for Matlab*. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
- [73] G. Bradski and A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library," *O'Reilly Media*, 2008.

- [74] M. Z. Samir, "Asservissement visuel d'un robot mobile sur une cible en mouvement," magister, EMP, 2008.
- [75] W. Burger and S. M. Bir Bhanu, IEEE, "Estimating 3-D Egomotion from Perspective Image Sequences," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 12, 1990.
- [76] D. Scaramuzza and F. Fraundorfer, "Visual Odometry: Part I: The First 30 Years and Fundamentals," *IEEE ROBOTICS & AUTOMATION MAGAZINE*, 8 December 2011.
- [77] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.
- [78] C. Harris and J. Pike, "3d positional integration from image sequences," in *Proc. Alvey Vision Conf*, pp. 87-90, 1988.
- [79] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," *Cambridge University Press*, 2003.
- [80] H. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," *Ph.D. dissertation, Stanford Univ., Stanford, CA*, 1980.
- [81] C. Olson, L. Matthies, M. Schoppers, and M. W. Maimone, "Robust stereo ego-motion for long distance navigation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 453-458, 2000.
- [82] L. Matthies and S. Shafer, "Error modeling in stereo navigation," *IEEE J. Robot. Automat.*, vol. vol. 3, pp. 239-248, 1987.
- [83] L. Matthies, "Dynamic stereo vision," *Ph.D. dissertation, CarnegieMellon Univ., Pittsburgh, PA*, 1989.
- [84] C. Olson, L. Matthies, M. Schoppers, and M. Maimone, "Rover navigation using stereo ego-motion," *Robot. Autonom. Syst.*, vol. vol. 43, pp. 215-229, , 2003.
- [85] M. Warren and B. Upcroft, "High altitude stereo visual odometry," *Robotics: Science and Systems IX*, pp. 1-8, 2013.
- [86] R. Fan, J. Jiao, J. Pan, H. Huang, S. Shen, and M. Liu, "Real-time dense stereo embedded in a uav for road inspection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0-0.
- [87] A. Milella and R. Siegwart, "Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point," in *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on*, 2006, pp. 21-21.
- [88] P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, pp. 239-256, 1992.
- [89] A. I. Comport, E. Malis, and P. Rives, "Accurate Quadrifocal Tracking for Robust 3D Visual Odometry," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 40-45.
- [90] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," *Proc. Int. Conf. on Computer Vision*, vol. 2, pp. 1403-1410, Octobre 2003.
- [91] H. P. MORAVEC, "Toward automatic visual obstacle avoidance," *Dans IJCAI*, vol. 2, pp. 584-584, 1977.
- [92] F. Devernay, "COURS « Détection de points d'intérêts », UFRIMA " *Unité de Formation et de Recherche en Informatique et Mathématiques Appliquées, Université Joseph Fourier*.
- [93] P. Moravech, "Towards automatic visual obstacle avoidance," in *Proceedings of the 5 Int. Joint Conference Artificial Intelligence*, p. 584, 1977.
- [94] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, pp. 147-152, 1988.
- [95] D. G. Lowe, "Distinctive Image Features from Scale-Invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.

- [96] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008/06/01/ 2008.
- [97] E. KOBZILI, C. LARBES, A. ALLAM, F. DEMIM, and A. BOUCHELOUKH, "Geometric Binary Descriptor Based Monocular SLAM " *3rd IEEE International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pp. 1 - 6 2018.
- [98] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," Berlin, Heidelberg, 2010, pp. 778-792.
- [99] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International conference on computer vision*, 2011, pp. 2564-2571.
- [100] E. Karami, S. Prasad, and M. Shehata, "Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images," *arXiv preprint arXiv:1710.02726*, 2017.
- [101] D. G. LOWE, "Object Recognition from Local Scale-Invariant Features," *Dans ICCV*, vol. 2, pp. 1150-1157, 1999.
- [102] K. Arun, Huang. TS, Blostein. SD, "Least squares fitting of two 3-d point sets," vol. 9, 1987.
- [103] Annalisa Milella, Bruno Nardelli, Donato Di Paola, and G. Cicirelli., "Robust Feature Detection and Matching for Vehicle Localization in Uncharted Environments," *IROS 3rd Workshop: Planning, Perception and Navigation for Intelligent Vehicles*, 2009
- [104] H. M. S. M. Elhachemi, "commande décentralisée pour un groupe de robots mobiles," Magister en automatique, EMP, janvier 2012.
- [105] OpenCV, "[OpenCV] Open Source Computer Vision Library (OpenCV)."
- [106] M. ECHERGUI and A. NASRALLAH, "Localisation d'un Robot Mobile par Odomètre Visuel Stéréoscopique," PFE, EMP, Alger, 2010.
- [107] T. Huang and A. Netravalli, "Motion and structure from feature correspondences: A review," *Proc. IEEE*, vol. 82, pp. 252-268, 1994.
- [108] D. Nister, "A minimal solution to the generalised 3-point pose problem," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 560-567, 2004.
- [109] P.Moravech, "Towards automatic visual obstacle avoidance,," in *Proceedings of the 5 Int.Joint Conference Artificial Intelligence*, p. 584.
- [110] Y.Chen and G.Medioni, "Object modeling by registration of multiple range images," presented at the IEEE Conf. on Robotics and Automation, 1991.
- [111] H. M. Kjer and J. Wilm, "Evaluation of surface registration algorithms for pet motion correction," Technical University of Denmark (DTU). 2010.
- [112] S. Umeyama, "Least-Squares Estimation If Transformations Parameters Between Two Points Patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1991.
- [113] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International journal of computer vision*, vol. 13, pp. 119-152, 1994.
- [114] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381-395, 1981.
- [115] P.Besl and N.McKay, "A method for registration of 3-d shapes," *In Trans. PAMI*, vol. 14, 1992.
- [116] S. Umeyama, "Least-Squares Estimation If Transformations Parameters Between Two Points Patterns," presented at the IEEE Transactions on Pattern Analysis and Machine Intelligence, avril 1991.
- [117] K.S.Arun, T.S.Huang, and S.D.Blostein, "Leastsquares fitting of two 3-d point sets," vol. 9, 1987.

- [118] M. Ballesta, Ó. Reinoso, A. Gil, L. Payá, and M. Juliá, "Evaluation of aligning methods for landmark-based maps in visual SLAM," *Sistemas de Percepción Visual Móvil y Cooperativo como Soporte para la Realización de Tareas con Redes de Robots*, 2007.
- [119] M. DEVY, B. COUDRIN, L. BRETHERS, and J. J. ORTEU, "3D Modeling of objects from a visual sensor moved manually by an operator," *ORASIS - Congrès des jeunes chercheurs en vision par ordinateur*, 2009.
- [120] R. HORAUD and O. MONGA, *vision par ordinateur: outils fondamentaux*, Deuxième ed., 1995.
- [121] A. L. D.W. Eggert, R.B. Fisher, "Estimating 3-D rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, Springer-Verlag, pp. 272–290, 1997.
- [122] A. Milella, B. Nardelli, D. D. Paola, and G. Cicirelli, "Robust Feature Detection and Matching for Vehicle Localization in Uncharted Environments " presented at the 3rd Workshop: Planning, Perception and Navigation for Intelligent Vehicles, 2009.
- [123] Z. ZHANG, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, pp. 119-152, 1994, received june 1992.
- [124] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *Journal of Field Robotics*, vol. 30, pp. 803-831, 2013.
- [125] F. Demim, A. Nemra, A. Boucheloukh, K. Louadj, M. Hamerlain, and A. Bazoula, "Robust SVSF-SLAM Algorithm for Unmanned Vehicle in Dynamic Environment," in *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, 2018, pp. 1-5.
- [126] J. Li and Y. Li, "Dynamic analysis and PID control for a quadrotor," in *2011 IEEE International Conference on Mechatronics and Automation*, 2011, pp. 573-578.
- [127] P. Zanne, G. Morel, and F. Piestan, "Robust vision based 3D trajectory tracking using sliding mode control," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, 2000, pp. 2088-2093.
- [128] S. Raiesdana, "Control of quadrotor trajectory tracking with sliding mode control optimized by neural networks," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 234, pp. 1101-1119, 2020.
- [129] L. D. Minh and C. Ha, "Modeling and control of quadrotor MAV using vision-based measurement," in *International Forum on Strategic Technology 2010*, 2010, pp. 70-75.
- [130] F. Yacef, O. Bouhali, M. Hamerlain, and N. Rizoug, "Observer-based adaptive fuzzy backstepping tracking control of quadrotor unmanned aerial vehicle powered by Li-ion battery," *Journal of Intelligent & Robotic Systems*, vol. 84, pp. 179-197, 2016.
- [131] Oussar Abdelatif, Guiatni Mohamed, Hamerlain Mustapha, and B. Mohamed, "Formation flight of multiple quadrotors in unknown environment," *International Conference on Automatic control, Telecommunications and Signals (ICATS15), University BADJI Mokhtar - Annaba - Algeria - November 16-18*, 2015.
- [132] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," *Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne*, 2007.
- [133] F. Yacef, O. Bouhali, M. Hamerlain, and A. Rezoug, "PSO optimization of Integral Backstepping Controller for Quadrotor attitude stabilization," in *3rd International Conference on Systems and Control*, 2013, pp. 462-466.
- [134] S. Ebbesen, P. Kiwitez, and L. Guzzella, "A Generic Particle Swarm Optimization Matlab Function," presented at the in American Control Conference (ACC), Montreal, Canada, 2012.

- [135] V. Kumar, "Integration of inertial navigation system and global positioning system using Kalman filtering," *Indian Institute of Technology, Bombay Mumbai*, 2004.
- [136] J. Kim and S. Sukkarieh, "Real-time implementation of airborne inertial-SLAM," *Robotics and Autonomous Systems*, vol. 55, pp. 62-71, 2007.
- [137] K. Boudjit and C. Larbes, "Detection and target tracking with a quadrotor using fuzzy logic," in *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, 2016, pp. 127-132.