

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique



MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Réseaux et Multimédias

THEME

Navigation précise basée sur des marqueurs visuels 2D
(applications de navigation des drones)

Présenté par :

MOHAMED BELAIAF

SELLAMI SEYFEDDIN

Soutenu publiquement le : 03/07/2022

Devant le jury composé de:

Président : Belhadj Foudil

Examineur : Belazzoug Mouhoub

Encadreur : Mostefai Messoud

2021/2022

Dédicace

Dédicace

Je dédie ce travail à tous les membres de ma famille et à tous les amis qui nous ont aidés moralement

À ma grand-mère

À mes amis... Adam, Choayb et Mohamad je vous remercie pour tous les moments doux et affreux que nous avons passés ensemble. Vous avez fait passer ces cinq années si vite. Vous étiez plus que des frères pour moi.

Sellami Seyfeddin

Dédicace

C'est avec des sentiments de joie et de fierté que j'ai achevé ce travail de patience et de longue haleine, un mémoire qui vient couronner mes cinq années d'étude universitaire.

Je profite cette occasion qui marquera à jamais mon cursus universitaire pour dédier ce mémoire à tous ceux qui m'ont apporté aide et soutien en particulier

Mes très chers parents Abderrahmane et Noura

A mes très chers sœurs Chahrazed, Imane et khawla

A mes meilleurs amis

Belaifa Mohamed

Remerciement

Remerciement

Tout d'abord, nous remercions Dieu de nous avoir donné le

Pouvoir d'apprendre et de faire ce travail.

Nous remercions notre encadreur M. Mostefai Messoud

Pour son aide, sa disponibilité et les précieux conseils qu'il nous a donnés.

A cette occasion, nous voudrions remercier tous les professeurs qui ont contribué à notre formation et à la qualité de l'éducation qu'ils nous ont donnée.

Nous n'oublions pas nos parents pour leur soutien et leur

Patience, nos familles et amis qui nous ont soutenus et

Encouragés, ainsi que toutes les personnes qui nous ont aidés de près ou de loin.

Merci à tous.

Résumé

Résumé

Ces dernières années, la demande d'utilisation de drones a augmenté, ce qui a entraîné de nombreux problèmes, notamment le problème de l'atterrissage, et de nombreux chercheurs ont développé des protocoles et des applications pour l'atterrissage de véhicules aériens autonomes (UAV). Cependant, la plupart des propositions sont basées sur des équipements coûteux ou ne répondent pas aux exigences de haute précision de certaines applications UAV, dans ce travail, une solution d'atterrissage de haute précision basée sur l'utilisation de balises ArUco est proposée. Parmi les solutions proposées, des drones équipés de caméras bon marché peuvent détecter les dimensions de l'ArUco 4 x 4 cm, depuis des hauteurs allant jusqu'à 20 mètres. Une fois le signe détecté, l'avion modifie son comportement pour atterrir exactement à l'endroit exact. L'application a été évaluée et validée à l'aide de la plate-forme de simulation Gazebo, et les vols de drones simulés montrent une précision d'atterrissage, ce qui représente une amélioration significative de la précision d'atterrissage grâce au système le plus fréquemment utilisé (GPS) où la précision d'atterrissage s'écarte de la distance cible entre 1 et 3 mètres.

Mot-clé : GPS, Aruco, Gazebo.

Abstract

In recent years, the demand for the use of drones has increased, which has led to many problems, including the problem of landing, and many researchers have developed protocols and applications for the landing of autonomous aerial vehicles (UAVs). However, most of the proposals are based on expensive equipment or do not meet the high precision requirements of some UAV applications, in this work, a high precision landing solution based on the use of ArUco beacons is proposed. Among the proposed solutions, drones equipped with inexpensive cameras can detect the dimensions of the Aruco 4 x 4 cm, from heights of up to 20 meters. Once the sign is detected, the plane changes its behavior to land exactly at the exact spot. The application has been evaluated and validated using the Gazebo simulation platform, and the simulated drone flights show landing accuracy, which represents a significant improvement in landing accuracy Compared to the system more frequently used (GPS) where the landing accuracy deviates from the target distance between 1 and 3 meters.

Keyword: GPS, Aruco, Gazebo.

ملخص:

في السنوات الأخيرة، ازداد الطلب على استخدام الطائرات بدون طيار، مما أدى إلى العديد من المشاكل، بما في ذلك مشكلة الهبوط، وقد طور الباحثون العديد من بروتوكولات الهبوط والتطبيقات. ومع ذلك، فإن معظم العروض تعتمد على معدات باهظة الثمن أو لا تتوافق مع الدقة العالية، في هذا العمل اقترحنا حلاً للهبوط عالي الدقة على علامة ARUCO حتى 20 مترًا بمجرد اكتشاف العلامة، تغيير الطائرة سلوكها للهبوط بالضبط في الموقع المحدد، يتم التحقق من صحة ذلك من خلال استخدام منصة محاكاة Gazebo. يمثل هذا تحسنًا كبيرًا في دقة الهبوط على نظام GPS الأكثر استخدامًا، حيث تتحرف دقة الأخير عن مسافة الهدف بين 1 و3 أمتار.

الكلمات المفتاحية: ARUCO , GAZEBO , GPS

Table des matières

Table des matières

Liste des figures	15
Introduction Générale	18
Plan du rapport	19
Chapitre 01 : les États de l'art et les marqueurs	21
1-Introduction	21
2- État de l'Art	21
2-1- Catégorie I :les plates-formes fixes:	21
2-2-Catégorie II - Plateformes mobiles :	22
3-Drones : leurs différentes applications possibles	23
3-1-Les types des drones	23
4-Marqueurs	25
4-1-La différence entre QR code et Aruco :	27
5-ArUco marqueur	28
6-Open CV :	30
6-1-La détection d'ArUco à l'aide d'Open CV	30
7-Conclusion	32
Chapitre 02 : Environnement de simulation adopté	34
1-Introduction	34
2-La simulation	34
3-Environnement de simulation Ardupilot / Gazebo	35
4-Gazebo	35
4-1-World Files	36
4-2-Fichiers types	37
4-3-Suppléments	37
5-ArduPilot :	38
5-1-Équipement :	Erreur ! Signet non défini.
5-2-Firmware :	39
5-3-Software :	39

6-DroneKit-python :	40
7-MAVlink (micro air vehicle link) :	40
. 8-MAVProxy :	41
9-SITL (software in the loop) :	41
10-Drone Raspberry pi	42
10-1-La Connection de la Raspberry Pi avec Pixhawk :	Erreur ! Signet non défini.
12-Conclusion	42
Chapitre 03 : le guide et les étapes de simulation.....	44
1-Introduction	44
2-Conditions préalables	44
2-1-Installation Git :	44
2-2-Installation Python 3.7 :	44
2-3-Installation OpenCV :	44
3-Instalation gazebo :	44
4-Plugin installation	44
5-Installation Mission Planer :	Erreur ! Signet non défini.
6-Exécution des programmes	45
7-Python script :	Erreur ! Signet non défini.
Conclusion.....	47
Conclusion générale	49
Les références	51
Annexe :	Erreur ! Signet non défini.
1-Le script de la détection des marqueurs ArUco avec OpenCV et Python :	Erreur ! Signet non défini.

Liste des figures

Liste des figures

Figure 1. : plateforme-d-atterrissage-lourd-pour-drones-95cm	22
Figure 2. exemple d'un drone militaire	24
Figure 3. : exemple d'un drone de loisir	24
Figure 4. exemple d'un drone professionnel et développement	25
Figure 5. Certaines versions de code QR, ainsi qu'une simple représentation de la structure d'un QR spécifique. La détection de position permet de déterminer la rotation de l'étiquette lorsqu'elle est détectée et le reste encode les données.....	26
Figure 6. Exemple d'un aruco marqueur.....	27
Figure 7. Exemple d'un QR code	27
Figure 8. Des exemples de différents marqueur.....	28
Figure9 Trois tailles différentes d'étiquettes ArUco. Le cadre noir et la grille binaire qui composent ce type de panneau sont également indiqués.....	28
Figure10. Des exemples d'un aruco marqueur.....	29
Figure 11. : La détection d'aruco maker à travers le web Cam.....	31
Figure 12 : le schéma de la simulation.....	35
Figure13. Exemple d'une world dans le simulateur Gazebo.....	36
Figure 14. La world utilise dans notre projet	36
Figure 15. Un modèle pour le drone Iris dans la simulateur.....	37
Figure 16. Liste des objets dans la Firmware.....	39
Figure 17. L'interface graphique de la version software	39
Figure 18. l'exécution de la fichier world « runway ».....	49

Figure 19. : comment lancer SITl	50
Figure 20. Les commande de console pour contrôler le drone.....	50
Figure 21. exécution de code python pour l’atterrissage.....	51
Figure 23 : exécution de code python pour l’atterrissage.....	51

Introduction Générale

Introduction Générale

Récemment, on s'intéresse de plus en plus aux UAV. Leurs applications vont de la surveillance, de l'inspection et de la surveillance à l'agriculture de précision et au ramassage/livraison de colis. Les entreprises de livraison se sont montrées intéressées à utiliser le drone pour livrer des colis, ce qui va révolutionner le mouvement des marchandises dans les villes, avec des entreprises comme Amazon, Wing et Uber promettant des opérations à grande échelle dans un proche avenir. Au lieu de cela, la technologie s'est concentrée principalement sur de petits essais, fournissant des vaccins et du sang à des endroits éloignés. Aux États-Unis, la livraison par drone a généralement été limitée aux petites villes, où l'utilisation des terres est moins encombrée et complexe, et ces entreprises ont fait face à des problèmes majeurs, notamment quelques incidents, des signalements de manquements présumés à la sécurité et un taux de roulement élevé. Notamment l'atterrissage de drones.

L'atterrissage des drones est la dernière et la plus importante étape de la navigation [1]. Selon les statistiques, le nombre d'accidents liés aux procédures d'atterrissage des drones représente 80% des cas dangereux [2]. Par conséquent, des recherches approfondies sont menées sur l'amélioration des techniques d'atterrissage.

Compte tenu des questions ci-dessus, le but de ce travail est de développer un nouveau système d'atterrissage basé sur la vision capable d'atterrir le drone dans un endroit très spécifique avec une grande précision. C'est pourquoi nous avons proposé une solution qui combine l'utilisation de la caméra avec les balises ArUco et par manque de moyens nous avons dû utiliser le logiciel simulé (Gazebo). Où, Le drone trouve la marqueuse. Une fois la marqueuse détectée, le drone doit ajuster sa position pour se déplacer vers le centre du pointeur et commencer à descendre, en effectuant des ajustements supplémentaires dynamiquement.

Les propositions antérieures s'appuyaient fortement sur le GPS et les capteurs de navigation à inertie (INS) comme méthodes de positionnement initial [3]. Cependant, les données d'altitude fournies par le GPS sont souvent inexactes et devraient être compensées par des capteurs de courte portée tels que les capteurs de pression atmosphérique ou les altimètres radar [4]. Malgré cette compensation, ces méthodes sont encore inexactes, en particulier sur les aéronefs horizontaux, ce qui fait que les sites d'atterrissage se trouvent

souvent à moins de 1 à 3 mètres de l'emplacement prévu. De plus, le GPS ne peut pas être utilisé à l'intérieur. Pour ces raisons, les systèmes GPS et INS sont principalement utilisés lors de longs trajets à l'étranger où l'exactitude n'est pas requise [3].

Plan du rapport

Le reste du travail est organisé comme suit : Dans le chapitre suivant, certains travaux pertinents sur les stratégies d'atterrissage de drones et la fraîcheur générale seront présentés sur les marqueurs et en particulier le marqueur aruco. Le chapitre 2 présente la simulation utilisée dans le projet, ainsi que quelques détails techniques sur le drone utilisé pour publier notre proposition. Le chapitre 3 présente la méthode d'installation des logiciels usagés et de préparation de l'environnement de travail en simulation

Chapitre 01 : les États de l'art et les marqueurs

Chapitre 01 : les États de l'art et les marqueurs

1-Introduction

Dans ce chapitre, nous aborderons les types d'atterrissages des drones et définirons les marqueurs.

Alors que l'utilisation des applications des drones a augmenté, Amazon envisage de lancer une flotte de drones de livraison capables de livrer des produits aux clients, mais le problème est de savoir comment atterrir sur des plates-formes fixes ou mobiles en toute sécurité, car un mauvais atterrissage entraîne la destruction et crash de ces véhicules, qui est ce que cela coûte trop cher.

Dans ce chapitre, nous parlerons des types d'atterrissages des drones, et nous en verrons quelques exemples de :

✓ Plates-formes fixes.

✓ Plateformes mobiles.

Aussi dans ce chapitre, nous parlerons également des marqueurs et des domaines de leur utilisation tels que la médecine, l'électronique, l'aéronautique, etc. Et distinct de la marque aruco que nous utilisons dans notre projet.

2- État de l'Art

La problématique de l'atterrissage des drones a été abordée dans de nombreux travaux, par exemple, des transporteurs (Amazon, Uber...), l'atterrissage du drone peut être classé sur différents types de plateformes d'atterrissage selon la référence [5], toutes les plateformes d'atterrissage appartiennent à une de ces deux Les deux catégories :

2-1- Catégorie I :les plates-formes fixes:

Ce type de plate-forme est plus facile à poser, parce que la cible reste constante. On peut citer :

Nowak et al. [6] suggèrent un système dans lequel un drone peut atterrir aussi bien la nuit que le jour. L'idée est simple mais puissante : un phare est placé sur le sol. La lumière émise par le phare est ensuite captée par une caméra (sans filtre infrarouge) et le drone se déplace de sorte que le phare soit au centre de l'image. Une fois le centre localisé, la hauteur

est estimée en fonction de la zone d'image exploitée par le guide et la hauteur du drone est réduite jusqu'à ce qu'il atterrisse en toute sécurité.

Srights, Patrono et al. [7] Fournir une solution pour l'atterrissage de drones sur une cible d'atterrissage artificielle. Leur objectif est similaire aux plates-formes d'aide traditionnelles, mais avec des rapports de dimension spécifiques, afin qu'ils puissent être détectés sur de longues distances. Les caractéristiques géométriques des marques en forme de H approuvées sont utilisées pour estimer la position avec une grande précision, atteignant une valeur d'erreur moyenne de seulement 0,0137 mètres en position et 1,04 degrés en direction.



Figure 1 : plateforme-d-atterrissage-lourd-pour-drones-95cm

2-2-Catégorie II - Plateformes mobiles :

Les plateformes mobiles sont définies par la capacité de se déplacer avec deux degrés (impulsion, effet). Dans ce cas, le drone doit d'abord suivre la plate-forme et ensuite atterrir sur elle. Parmi celles-ci on a :

Chen et al. [8] ont réussi à faire atterrir un vrai drone sur un objet en mouvement à 1 m/s (catégorie II). Une caméra a été utilisée pour suivre l'emplacement de la plate-forme d'atterrissage et le capteur LIDAR a fourni des informations détaillées sur la hauteur. Cette recherche a fourni un moyen puissant de suivre et d'atterrir sur un objet en mouvement.

Cependant, l'utilisation du capteur LIDAR inhibe la solution, car il a tendance à être très coûteux lorsqu'il s'étend à un grand nombre de drones.

Baca et Al. [9] Ils ont été en mesure de surveiller une voiture en mouvement à 15 km/h et de prévoir ses déplacements futurs. Pour ce faire, ils ont équipé le drone de capteurs et d'un ordinateur, qui détecte le véhicule à l'aide d'une caméra et prédit le mouvement du futur véhicule à l'aide d'un modèle de mouvement non linéaire. Tout en suivant le véhicule, le drone atterrit sur sa surface et se stabilise à l'aide de jambes magnétiques.

Notre travail diffère de celui des travaux précédents parce que nous voulons que les drones détectent l'aire d'atterrissage lorsqu'ils sont élevés au-dessus du sol (hauteur > 20 mètres) pour compenser les valeurs d'erreur GPS élevées, tout en utilisant des capteurs bon marché (seule la caméra Raspberry Pi est requise).

3-Drones : leurs différentes applications possibles

- Utilisés dans l'armée depuis plus d'un siècle, les drones font aujourd'hui partie des objets connectés les plus en vogue auprès du public. Le monde professionnel n'est pas en reste lorsqu'il s'agit d'utiliser cet outil aux multiples possibilités.
- Dans un cadre civil, le soi-disant "drone" comprend les véhicules aériens sans pilote. Petits aéronefs pilotés à l'aide d'un smartphone, d'une tablette ou d'une radiocommande, etc.

3-1-Les types des drones

3-1-1-Les drones militaires

Les missions principales des drones militaires sont les opérations de renseignement et de lutte contre le terrorisme. Ils ont en fait peu de choses en commun avec leurs homologues de drones civils.



Figure 2 : exemple d'un drone militaire

3-1-2-Les drones de loisir et de compétition

Le marché des drones de loisir est en plein essor, et ça ne s'arrêtera pas. Avec des tarifs allant de 20 euros à plusieurs milliers d'euros, ces drones conviennent à tous les budgets et à tous les niveaux, du débutant à l'expert. Ils sont utilisés à l'intérieur et à l'extérieur.



Figure 3 : exemple d'un drone de loisir

3-1-3-Les drones à usage professionnel et développement :

En matière de drones à usage professionnel, les drones s'adaptent aux besoins et aux envies des entreprises. Pour ce faire, leur utilisation ne cesse de s'étendre au-delà de la photographie aérienne dans le spectre visible.



Figure 4 : exemple d'un drone professionnel et développement

4-Marqueurs

Marqueurs est un élément trouvé dans le champ de visibilité du système d'imagerie qui permet de mesurer quelque chose dans le monde réel à travers l'image obtenue. Les marqueurs sont utilisés dans de nombreux domaines, y compris la métrologie comme outil de mesure, et peuvent également être utilisés dans les systèmes de radiothérapie et de radio chirurgie. Ils sont considérés comme l'un des points forts de la tumeur pour faciliter les cibles correctes de traitement comme dans la figure 5, ils peuvent être utilisés pour améliorer la fabrication de circuits imprimés (PCB) Comme dans la figure 6, il peut être utilisé dans la biologie cellulaire, le suivi social des insectes, imagerie médicale, étude géographique, physique, ou en réalité augmentée, qui est le domaine de ce projet.

Chaque domaine a son propre marqueur. Marqueur utilisée doit remplir les conditions suivantes pour son utilisation :

- L'étiquette doit être détectable et reconnaissable de façon unique à l'aide de la caméra RVB-D.
- L'indicateur doit être fixé sur l'échelle de façon à pouvoir être déterminé à différentes distances.
- Le repère doit être fixé dans la rotation afin qu'il puisse être déterminé lorsque n'importe quelle rotation l'a été.

Comme ce n'est pas le but de ce projet de développer des marques spécifiques pour l'extraction des caractéristiques, il est raisonnable de vérifier les marques disponibles utilisées dans la réalité augmentée et de choisir la marque qui convient le mieux au projet.

[10] et [11] donnent un aperçu des marqueurs utilisés dans la réalité augmentée. Si les notes de niveau sont prises en compte, il est logique de prendre en compte d'abord les codes à barres 2D en raison de leur large utilisation. Il s'agit d'un type de code à barres bidimensionnel (QR code) [12]. Les codes QR comportent également d'autres sous-types, notamment le code QR micro et le code QR standard. Le code QR standard a une forme en forme de boîte, la quantité de données qui peuvent y être stockées dépend de la quantité de caractères ou de symboles, la version et le niveau de correction d'erreur. Sa taille minimale est de 21 21 unités et la limite supérieure est capable de stocker jusqu'à 152 bits, et un exemple apparaît à la figure 7. En outre, les codes QR peuvent être lus dans n'importe quelle direction si des modèles de position sont utilisés. Ceux-ci sont placés dans trois piliers de codes QR. Cependant, les codes QR sont dédiés au stockage de l'information, ce qui entraîne des problèmes lors de la lecture à grande distance. Les codes d'information comme les codes QR n'offrent donc aucun avantage dans notre projet.

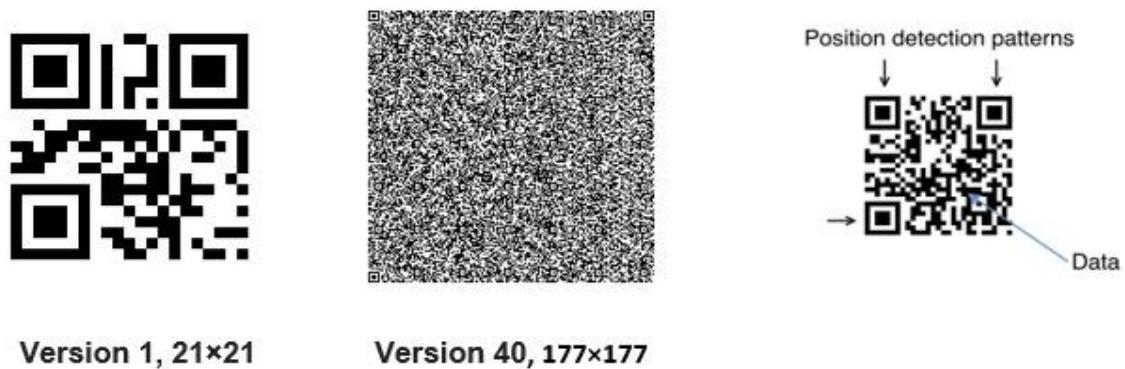


Figure 5 : Certaines versions de code QR, ainsi qu'une simple représentation de la structure d'un QR spécifique. La détection de position permet de déterminer la rotation de l'étiquette lorsqu'elle est détectée et le reste encode les données.

4-1-La différence entre QR code et Aruco :

L'aruco pour les non-initiés, cela peut ressembler à un code QR avec un nom différent, mais ce n'est pas

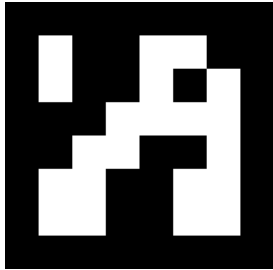


Figure 6 : exemple d'un aruco marquer



Figure 7 : exemple d'un QR code

De l'image ci-dessus, il est clair qu'il y a une différence entre l'information encodée dans un code QR et un marqueur Aruco.

Aruco Marqueur aide les caméras à comprendre l'angle, la hauteur, la profondeur et d'autres paramètres et trouve ses cas d'utilisation dans une excellente vision par ordinateur et des tâches de réalité augmentée. D'autre part, les codes QR codent des données textuelles et peuvent être utilisés à diverses fins, telles que : B. Pour stocker des mots de passe, des identifiants d'accès, des sites Web, des réservations de billets, des noms d'utilisateur, etc.

Les marqueurs sont couramment utilisés dans les applications de réalité augmentée (RA), où les exigences qui viennent d'être mentionnées doivent également être respectées. Une méthode à envisager est l'utilisation de marques circulaires telles que celles contenues dans [13] et [14], et ce type de marqueur fournit généralement un seul point. Alternativement, on peut utiliser des marques 2D basées sur carré, et leurs caractéristiques principales sont qu'ils peuvent fournir quatre points de correspondance, un pour chaque angle. Un système commun utilisant cette approche est ce qu'on appelle ARToolkit [15] qui se compose d'un cadre noir et d'une image interne qui est ensuite stockée dans une base de données. Les méthodes ultérieures basées sur le même principe sont ARTag [10], ARToolkit Plus [16] et ArUco entre autres.

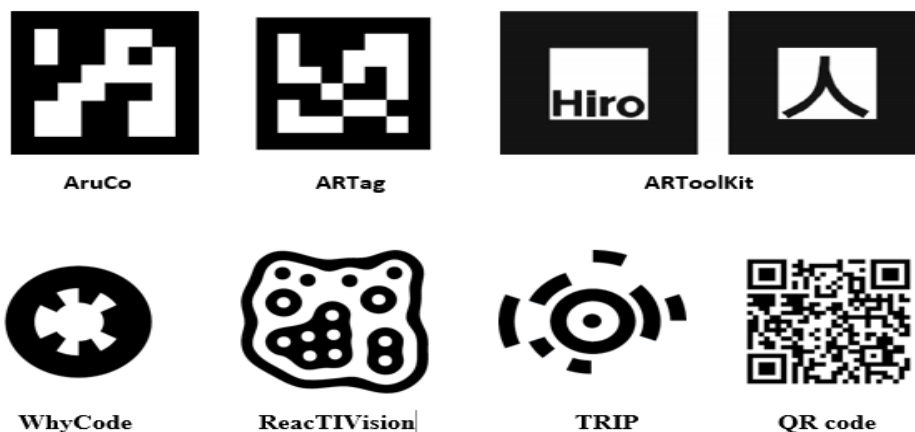


Figure 8 : des exemples de différents marqueurs

Il semble raisonnable de choisir des balises basées sur des boîtes, et les codes QR sont ignorés en raison de la difficulté de les lire à distance et en perspective. Parmi les autres marqueurs, les sélecteurs ArUco sont préférés en raison de la disponibilité de la bibliothèque Open CV.

5-Aruco marqueur

ArUco est un marqueur avec une base carrée composée d'un large cadre noir et d'une matrice interne $n \times n$. n est utilisé pour déterminer la taille de l'étiquette, montrant différentes options de taille. (Figure 11) Le cadre noir facilite sa détection rapide dans l'image et permet de le sélectionner et d'appliquer des techniques de détection et de correction des erreurs. La taille de marqueur détermine la taille de la matrice interne. Par exemple, la taille d'aruco 4x4 se compose de 16 bits.

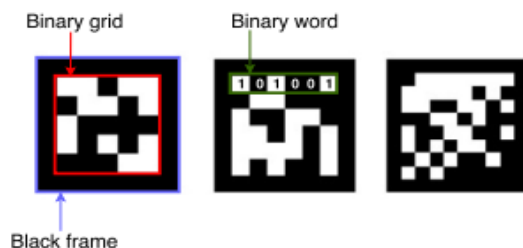


Figure 9 : Trois tailles différentes d'étiquettes Aruco. Le cadre noir et la grille binaire qui composent ce type de panneau sont également indiqués.

Quelques exemples de tags Aruco :

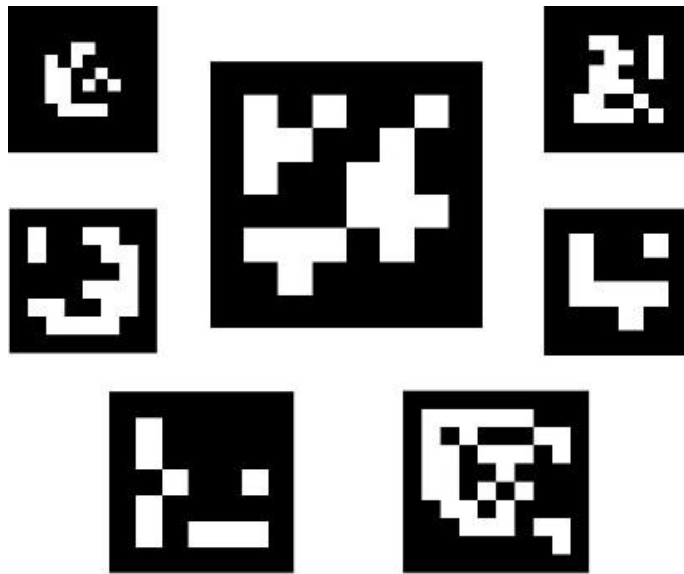


Figure 10 : des exemples d'un aruco marqueur

Il convient de noter qu'une marque peut être trouvée déplacée dans l'environnement, cependant, le processus de détection doit être en mesure de déterminer son angle de rotation initiale, de sorte que chaque angle soit déterminé. Cela se fait également sur la base du codage binaire.

Dictionnaire de marqueurs est l'ensemble de balises qui sont considérés dans une application particulière. C'est juste une liste de symboles binaires pour chacun de leurs marqueurs.

Les principales caractéristiques du dictionnaire sont la taille du dictionnaire, qui est le nombre de marques qui composent le dictionnaire et la taille de la balise (le nombre de bits).

Le module d'Aruco comprend quelques dictionnaires prédéfinis couvrant une gamme de différentes tailles de dictionnaire et de marquages.

On peut penser que la norme d'identification est le nombre obtenu en convertissant le codage binaire en un nombre de base décimal. Cependant, cela n'est pas possible parce que pour les tailles de marques élevées, le nombre de bits est très élevé et la gestion de ces nombres énormes n'est pas pratique. Au lieu de cela, l'ID de marqueur est simplement

l'indicateur de tag dans le dictionnaire auquel il appartient. Par exemple, les cinq premiers marqueurs du dictionnaire ont des ID : 0, 1, 2, 3 et 4.

6-Open CV :

[24]Open CV (Open Source Computer Vision Library) est une bibliothèque de logiciels de vision informatique et d'apprentissage automatique. Open CV a été construit pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception de la machine dans les produits commerciaux. Étant un produit sous licence BSD, Open CV permet aux entreprises d'utiliser et de modifier facilement le code

La bibliothèque dispose de plus de 2500 algorithmes d'optimisation, y compris une gamme complète d'algorithmes classiques et sophistiqués de vision par ordinateur et d'apprentissage automatique. Ces algorithmes peuvent être utilisés pour détecter et identifier des visages, identifier des objets, classer le comportement humain dans la vidéo, suivre le mouvement de la caméra, suivre des objets en mouvement, extraire des modèles 3D d'objets, lier des images pour créer des images précises de scènes entières, trouver des images similaires à partir d'images bases de données, suivre les mouvements oculaires, expérimenter des paysages et créer une superposition de signaux de réalité augmentée.

6-1-La détection d'ArUco à l'aide d'Open CV

Lorsque l'image contient des balises ArUco, le processus de détection devrait résister les balises découvertes. Chaque marque découverte comprend :

La marque cadrée à l'intérieur de l'image (dans son ordre d'origine).

Son identifiant

Le processus de détection des marques comporte deux étapes principales :

1-Détection des marqueurs candidats. Dans cette étape, l'image est analysée pour trouver des formes carrées filtrées pour être des marqueurs. On commence par effectuer un seuillage

adaptatif pour le tranchage de marque, puis les lignes sont extraites de l'image souillée et celles qui ne sont pas convexes ou carrées sont éliminées. Une filtration supplémentaire est également appliquée (suppression des lignes très petites ou très grandes, suppression des lignes très proches les unes des autres, etc.).

2-Après avoir détecté les marqueurs candidats, il est nécessaire de déterminer s'ils sont effectivement des marqueurs valides en analysant leur codage interne. Cette étape commence par l'extraction de parties de chaque marque. Pour ce faire, la conversion en perspective est appliquée d'abord pour obtenir la marque dans sa forme connue. Ensuite, l'image est segmentée en utilisant la méthode d'Otsu [17] pour séparer les pièces blanches et noires. L'image est ensuite divisée en différentes cellules en fonction de la taille de la marque et de la taille de la limite. Par la suite, le nombre de pixels noirs ou blancs dans chaque cellule est calculé pour déterminer si le bit est blanc ou noir. Enfin, les bits sont analysés pour déterminer si la balise appartient au dictionnaire spécifié. Des techniques de correction d'erreurs sont utilisées si nécessaire.

Considérez la photo suivante :

Image avec une variété de balises :

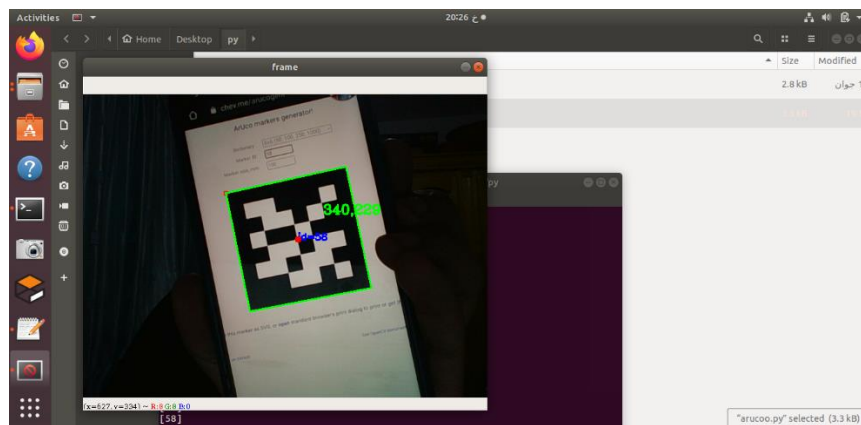


Figure 11 : la détection d'aruco maker à travers le web Cam

7-Conclusion

Dans ce chapitre, nous avons d'abord présenté les aires d'atterrissage des drones les plus importantes et donné quelques expériences réussies à ce sujet.

Après cela, nous avons fourni un aperçu des marqueurs, de leurs types et utilisations dans différents domaines et de leur emplacement dans ce projet.

Il est important de mentionner que les marqueurs, que nous avons brièvement présentés, ne représentent pas le cœur de notre activité. Pour cette raison, nous ne le mentionnerons pas en détail.

Parmi ces marqueurs, nous mettrons en évidence la marque aruco dans notre projet à la place d'autres marques

Chapitre 02 : Environnement de simulation adopté

Chapitre 02 : Environnement de simulation adopté

1-Introduction

Dans le chapitre précédent, nous avons vu les différents types de drones, leurs différentes utilisations, et les types de plateformes d'atterrissage. Notre étude portera sur l'atterrissage du drone à l'aide de la caméra qui identifie la marque aruco, cette dernière représentant le site d'atterrissage.

En raison du manque de drones et des dommages au Raspberry Pi, nous avons dû utiliser un programme d'émulation, et nous avons choisi le simulateur Gazebo / ArduPilot, dont nous parlerons dans ce chapitre.

2-La simulation

La simulation est un outil essentiel pour quiconque s'intéresse à la robotique, car la présence du simulateur accélère le processus de développement et de test des algorithmes sans crainte de problèmes avec les sous-systèmes du robot, par exemple une panne dans l'un des moteurs du robot ou attendant de recharger les batteries utilisées à l'intérieur du robot. Dans ce projet, nous examinerons l'un des simulateurs (Ardupilot / Gazebo).

Plusieurs langages et environnements différents ont été introduits pour prendre en charge la simulation, par exemple [18], où une stratégie de contrôle et de collaboration qui exploite CATA est utilisée, ou [19], où Matlab/Simulink est utilisé pour le contrôle en temps réel des drones. . Ces environnements sont plus simples qu'Ardupilot/Gazebo mais ne permettent pas une représentation graphique des drones.

L'environnement de simulation Ardupilot/Gazebo est un cadre complexe qui comprend de nombreux éléments différents et permet une simulation réaliste des systèmes robotiques.

Ardupilot [20] est un logiciel open source qui permet le contrôle de divers véhicules sans pilote principalement grâce à ses quatre composants différents : Antenna Tracker, APM Rover, ArduPlane et Arducopter. En particulier, Arducopter implémente le contrôle réel du drone, pour simuler le drone, Gazebo a été utilisé [21].

3-Environnement de simulation Ardupilot / Gazebo

L'environnement de simulation et de développement Ardupilot/Gazebo comprend deux composants principaux:

- ✓ Ardupilo [20]
- ✓ Gazebo [22]
- ✓ DroneKit

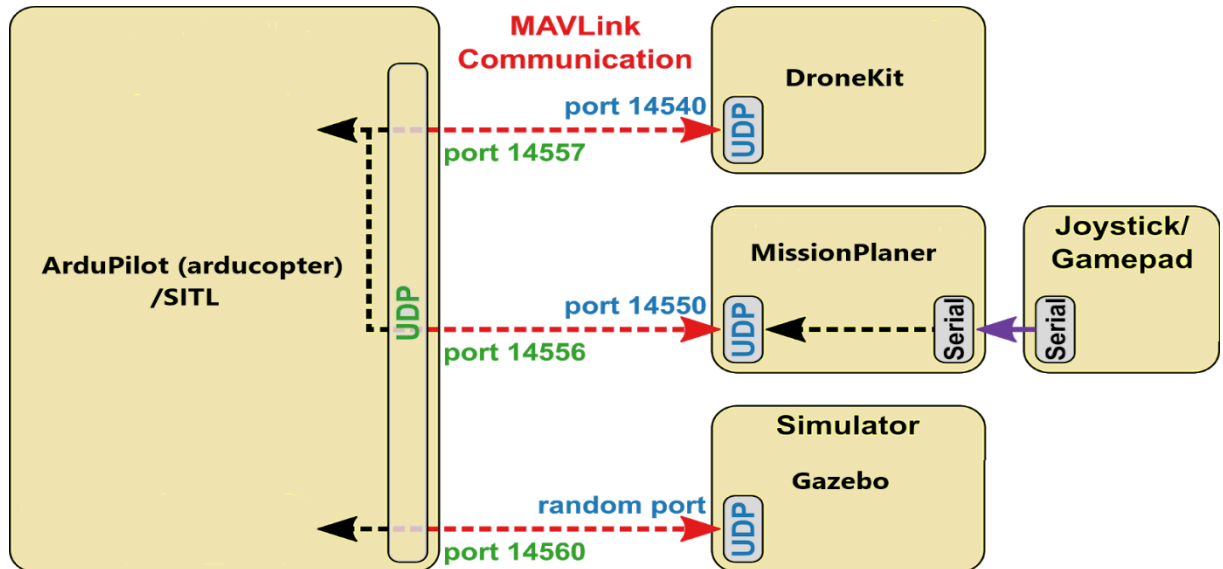


Figure 12 : le schéma de la simulation

4-Gazebo

Gazebo [22] est un programme de simulation 3D. Il s'agit d'un simulateur physique connu pour son utilisation dans la simulation de robots en raison de sa relation avec ROS. L'un de ses avantages est que vous pouvez simuler des capteurs en plus de la physique dynamique et cinétique. Cette simulation est complète avec la dynamique et la physique cinétique.

Le simulateur de gazebo se compose de plusieurs composants. Trois de ces éléments sont décrits dans cette section. Ces composants sont :

- World Files
- Fichiers types
- Suppléments

4-1-World Files

Les fichiers de description globale sont utilisés pour définir l'environnement de simulation. Cela peut inclure des descriptions de l'éclairage, du sol et d'autres éléments fixes, tels que des obstacles de construction et des capteurs. De plus, un fichier global décrit comment ces fichiers modèles sont placés dans la simulation et comment ils interagissent les uns avec les autres.

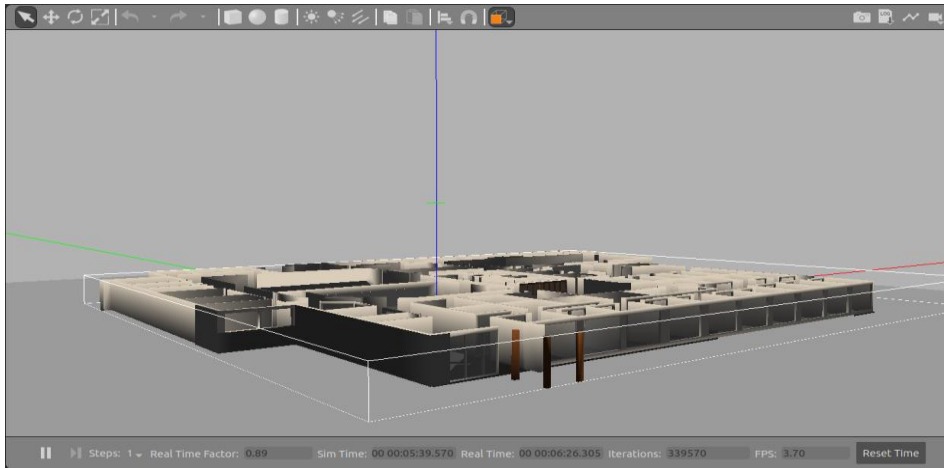


Figure 13 : Exemple d'un world dans la simulateur Gazebo

Un monde inspiré du void.world présenté par ArduPilot SITL est envisagé dans ce projet. Le monde directement utilisé est le fichier vierge world.world composé du sol en asphalte et du soleil. La Terre contient une description de son apparence et de ses caractéristiques d'impact. Les caractéristiques d'impact au sol sont importantes car elles définissent où le drone peut atterrir.

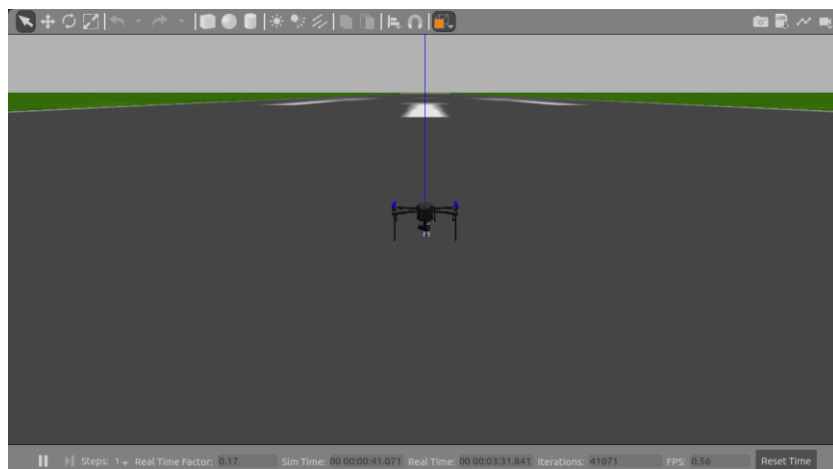


Figure 14 : Le world utilisé dans notre projet

4-2-Fichiers types

Les fichiers de type sont utilisés pour décrire les éléments individuels utilisés dans la simulation.

L'accessoire SITL propose une maquette du drone Iris. En raison de contraintes de temps, il a été décidé d'utiliser le drone Iris dans le simulateur, qui simule un modèle du drone framboise. Bien que le modèle de drone Iris ne dispose pas de caméra RVB ni de capteur de profondeur, il est possible d'ajouter une caméra RVB-D avec l'accessoire approprié pour simuler la caméra.

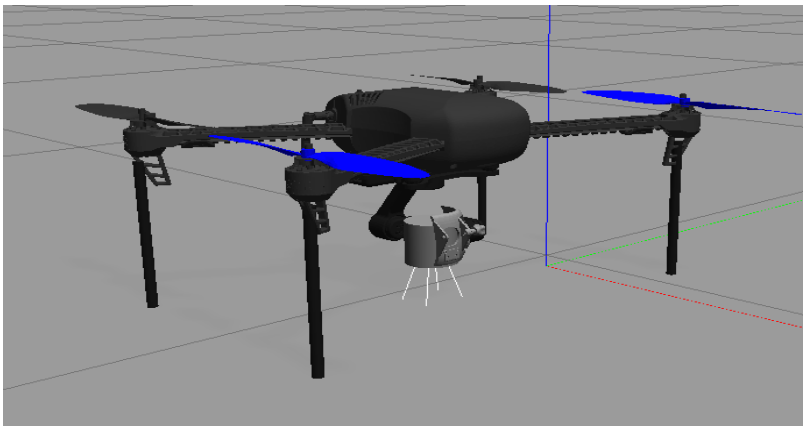


Figure 15 : Un modèle pour le drone Iris dans le simulateur

4-3-Suppléments

Les accessoires sont un mécanisme simple pour interagir avec le simulateur Gazebo. Certains accessoires ont déjà été introduits avec l'installation du gazebo, mais il est également possible d'en développer de nouveaux pour communiquer avec le gazebo d'une certaine manière. Divers accessoires peuvent être classés comme suit :

- Mondialisme
- Modèle
- Capteur

- Système
- visibles
- interface graphique

Ce modèle de drone Iris 3DRR comprend des accessoires tels qu'un support moteur, un support MAVLink, etc. De plus, il était nécessaire d'inclure un accessoire pour la caméra RGB-D. Parmi les accessoires inclus dans l'installation Gazebo se trouve un accessoire qui émule une caméra Raspberry pi, un capteur RGB-D actif similaire à la caméra.

5-ArduPilot :

ArduPilot [20] est une suite logicielle de pilote automatique. ArduPilot permet la création et l'utilisation de systèmes de véhicules autonomes et sans pilote fiables pour le bénéfice de tous. ArduPilot propose une gamme complète d'outils adaptés à presque tous les véhicules et applications. En tant que projet open source, il évolue constamment en fonction des retours rapides d'une large communauté d'utilisateurs. Bien qu'ArduPilot ne fabrique pas le matériel, le micro logiciel ArduPilot fonctionne sur une variété d'appareils différents pour contrôler les véhicules sans pilote de tous types. ArduPilot a une énorme communauté en ligne dédiée à aider les utilisateurs avec des questions, des problèmes et des solutions

Le logiciel Ardupilot propose différents modes de vol, qui peuvent être distingués en modes de vol manuel et automatique, avec un ensemble de paramètres personnalisables. Par exemple, il est possible de contrôler le vol du drone via un mode piloté, où Ardupilot ajuste automatiquement les valeurs de lacet et de roulis, en fonction de la position donnée. De plus, grâce à d'autres modes de vol, le drone peut maintenir l'altitude souhaitée, le drone peut être ramené au point de lancement ou simplement au sol, et bien d'autres options.

Pour communiquer avec le véhicule sans pilote, le protocole MAVLink est exploité [23]. Il permet aux véhicules de communiquer en échangeant des paquets, qui sont représentés à un bas niveau par des chaînes en langage C, où chaque bit a une fonction spécifique dans la communication.

5-1-Firmware :

Le code fonctionne sur la console. Vous choisissez le firmware identique pour votre véhicule et votre mission : hélicoptère, avion, rover, sous-marin ou tracker d'antenne. Option pour vous - Pilote automatique pour toute mission

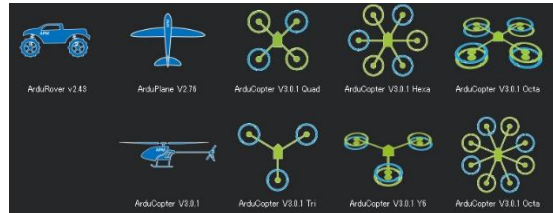


Figure 16 : liste des objets dans la Firmware

5-3-Software :

Le logiciel représente l'interface avec la console. Il est également appelé Ground Control Station (GCS), et le logiciel peut être exécuté sur des ordinateurs personnels ou des appareils mobiles. Le système GCS permet aux utilisateurs de configurer, tester et ajuster le véhicule. Les progiciels avancés permettent une planification, une exploitation et une analyse post-mission indépendantes. Le diagramme de mission est un SCM complet supporté par ArduPilot. Il fournit l'interaction de signal et de clic avec vos appareils, des scripts personnalisés et des simulations.



Figure 17 : l'interface graphique de la version software

6-DroneKit-python :

DroneKit-Python est un projet open source et community-driven projet. On peut trouver tout le code source sur Git hub.

DroneKit-Python est compatible avec les véhicules qui communiquent avec le protocole MAVLink (y compris la plupart des véhicules fabriqués par 3DR). Il fonctionne sous Linux, Mac OS X ou Windows.

DroneKit-Python permet aux développeurs de créer des applications qui s'exécutent sur un ordinateur compagnon embarqué et de communiquer avec le contrôleur de vol ArduPilot à l'aide d'une liaison à faible latence. Les applications embarquées peuvent améliorer considérablement le pilote automatique ArduPilot, en ajoutant une plus grande intelligence au comportement du véhicule et en effectuant des tâches gourmandes en calcul ou sensibles au temps (par exemple, la vision par ordinateur, la planification de trajectoire ou la modélisation 3D).

L'API (Application Programming Interface) communique avec les véhicules via MAVLink. Elle fournit un accès programmatique à la télémétrie, à l'état et aux paramètres d'un véhicule connecté, et permet à la fois la gestion de la mission et le contrôle direct du mouvement et des opérations du véhicule

7-MAVlink (micro air vehicle link) :

MAVlink [23] Est un protocole de communication pour les systèmes sans pilote (par exemple, les drones et les robots).

Ce protocole est utilisé dans les principaux systèmes de pilotage automatique, en particulier ArduPilot et PX4, et fournit des fonctionnalités pour surveiller et contrôler les missions de systèmes sans pilote (Unmanned systems) ainsi que pour les intégrer à Internet.

MAVLink prend en charge les protocoles de communication UDP et TCP entre la station au sol et le drone, UDP est connu pour être un protocole qui ne nécessite aucune communication entre le client et le serveur et n'a pas de mécanisme pour assurer une livraison fiable des messages, mais offre une alternative à un flux de messages rapide et tolérant aux

perdes. D'autre part, TCP est un protocole basé sur la connexion fiable qui offre une meilleure fiabilité de transmission grâce à son mécanisme d'accusé de réception, Mais il peut être sujet à l'encombrement des paquets. Le choix du protocole de transport est laissé à l'utilisateur en fonction des besoins de l'échange de messages entre le système sans pilote et la station au terre.

. 8-MAVProxy :

MAVProxy est un GCS (ground control station) fonctionnel pour les drones, conçu pour être un GCS simple, portable et utilisable pour tout système autonome prenant en charge le protocole MAVLink (comme celui utilisant ArduPilot).

MAVProxy est un logiciel de GROUND STATION "développé" en ligne de commande.

Il peut être étendu via des modules supplémentaires, ou complété par d'autres stations au sol, telles que Mission Planner, QGroundControl, etc., pour fournir une interface utilisateur graphique.

9-SITL (software in the loop) :

SITL Simulator nous permet de faire fonctionner un drone sans aucun matériel. Il est construit à partir d'un algorithme Autopilot utilisant un compilateur C++ standard, qui nous donne un fichier exécutable, qui permet de tester le comportement des algorithmes sans aucun matériel.

SITL ne permettra pas d'exécuter ArduPilot directement sur notre PC, sans aucun matériel spécial. Cela profite au fait qu'ArduPilot est un pilote automatique portable qui peut fonctionner sur une très grande variété de plates-formes. Notre ordinateur n'est qu'une autre plate-forme sur laquelle ArduPilot peut être construit et exécuté.

10-Drone Raspberry pi

Raspberry Pi a conquis le monde des drones. Il serait bon de comprendre comment fonctionnent les drones Raspberry Pi. Les drones avancés se composent de deux types de cartes : un ordinateur compagnon et un contrôleur de vol. Un contrôleur de vol (FC) est un appareil à basse altitude qui fait fonctionner du matériel, ce qui permet au drone de se mettre en palier des centaines de fois par seconde. FC doit être très fiable parce que s'il y a un retard de temps le drone va s'écraser.

L'ordinateur compagnon est un peu différent. Essentiellement, l'appareil peut exécuter des scripts avancés et transmettre des commandes aux flight controller (FC).

Dans n'importe quel drone pi framboise, la carte pi framboise agira comme ordinateur compagnon

12-Conclusion

Dans ce chapitre, nous avons vu le drone que nous aurions utilisé dans le projet sans le manque de capacités, et nous avons également donné une explication du simulateur Gazebo et comment il simule le drone et quels sont les moyens nécessaires pour cela et une explication détaillée du ArduPilot et DroneKit.

Nous avons rencontré des bogues dans le programme PX4, nous nous sommes donc tournés vers l'utilisation du programme ArduPilot.

Chapitre 03 : le guide et les étapes de simulation

Chapitre 03 : le guide et les étapes de simulation

1-Introduction

Dans ce chapitre, nous allons expliquer les étapes d'installation nécessaires pour un simulateur de drone. Guide testé pour fonctionner sur Ubuntu 18.04 LTS

2-Conditions préalables

Quelques programmes doivent être installés avant de suivre les guides

2-1-Installation Git :  **git**

2-2-Installation Python 3.7 :



2-3-Installation OpenCV :



3-Instalation gazebo :



4-Plugin installation

Nous utiliserons le plugin khancyr pour la prochaine explication. D'abord clonez-le quelque part dans notre répertoire personnel.

```
git clone https://github.com/khancyr/ardupilot\_gazebo
cd ardupilot_gazebo
mkdir build
cd build

cmake ..
make -j4
sudo make install
```

```
echo 'source /usr/share/gazebo/setup.sh' >> ~/.bashrc
```

6-Exécution des programmes

1- Pour démarrer les simulateurs dans Gazebo, nous ouvrons le terminal et exécutons les commandes suivantes :

```
gazebo --verbose ~/ardupilot_gazebo/worlds/iris_arducopter_runway.world
```

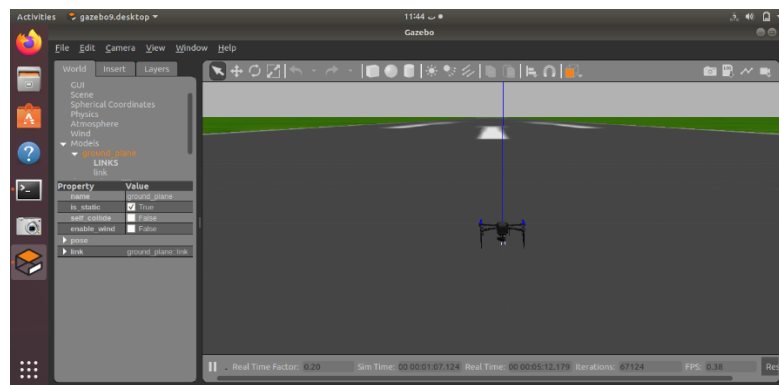


Figure 18 : l'exécution de la fichier world « runway »

2- launch SITL

```
cd ~/ardupilot/ArduCopter/ sim_vehicle.py -v ArduCopter -f gazebo-iris --console
```

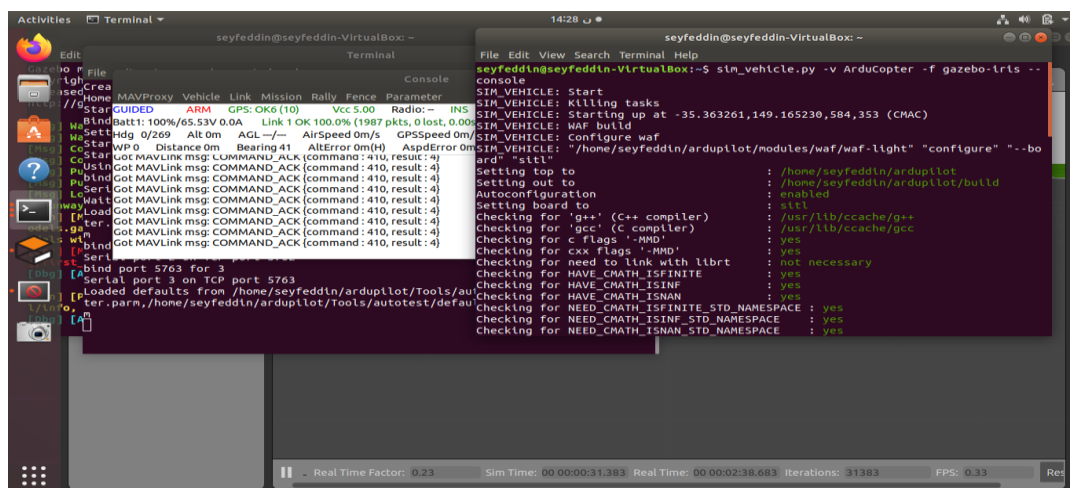


Figure 19 : comment lancer SITL

3-Contrôle des drones par les commandes de console :

Lancer la commande **GUIDE** pour exécuter le code python

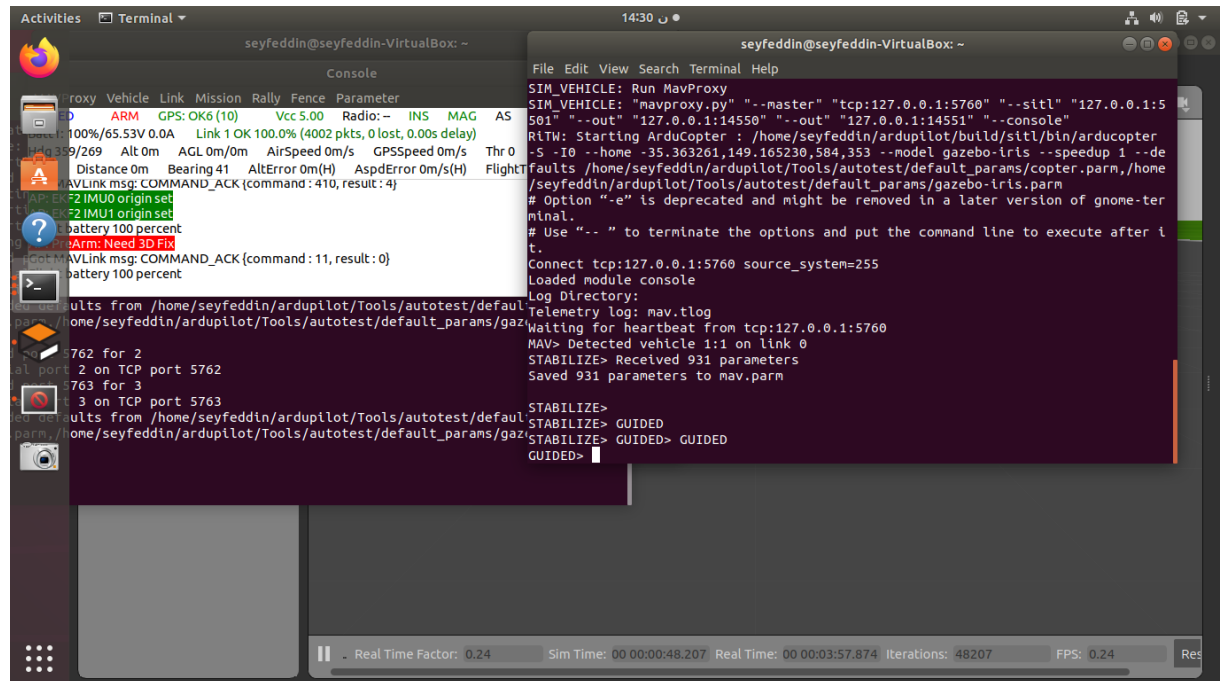


Figure 20 : les commande de console pour contrôler le drone

4 – Exécuter le code python :

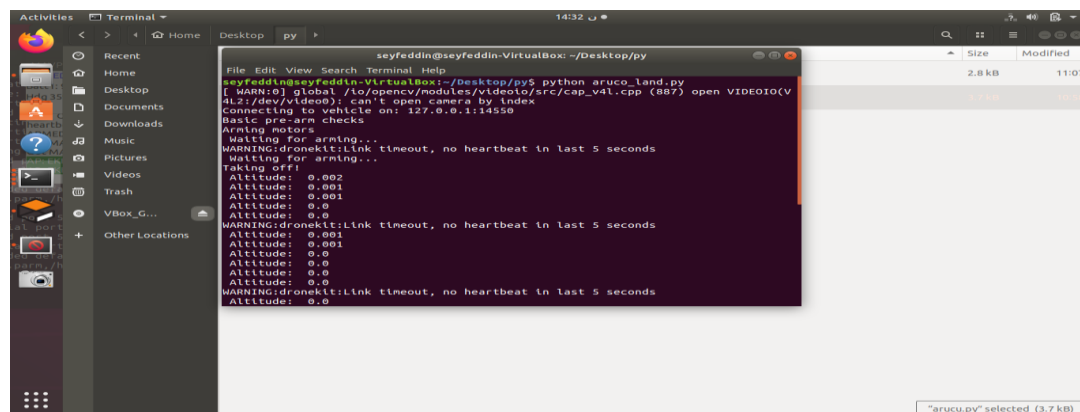


Figure 21: exécution de code python pour l'atterrissage

Conclusion

Dans ce chapitre, nous avons vu une manière détaillée d'expliquer l'installation de tous les programmes et moyens utilisés pour la réussite de ce projet

Nous avons rencontré de nombreuses difficultés dans ubuntu 20.04 Donc, nous avons dû travailler sur ubuntu 18.04 ainsi que des difficultés dans l'utilisation de PX4 donc nous sommes allés à Ardupilot

Conclusion générale

Conclusion générale

L'atterrissage précis des drones reste un problème difficile, car les procédures d'atterrissage basées sur le GPS sont associées à des erreurs allant jusqu'à quelques mètres, même dans des conditions de réception satellite idéales, ce qui entraîne de mauvaises performances dans de nombreux cas. De plus, l'atterrissage assisté par GPS n'est pas une option pour les opérations en intérieur. Pour résoudre ce problème, dans ce travail, une solution d'atterrissage basée sur la vision utilisant un capteur proportionnel tel qu'une caméra Raspberry Pi basée sur des balises ArUco est présentée. Ces marquages permettent au drone de détecter le site d'atterrissage exact depuis une altitude élevée (20 mètres).

Une solution basée sur une caméra aiderait potentiellement à augmenter la sécurité des drones lors de l'atterrissage où ils pourraient perdre leur marque.

Une fois la marque visible par la caméra, le drone a estimé plus précisément sa position avec la marque et s'est déplacé au-dessus de la marque. Après cela, le drone s'est posé dessus progressivement et méticuleusement.

Ce travail s'étend à la simulation Ardupilot/Gazebo avec capacité de simulation UAV.

La solution proposée consiste en un drone existant équipé d'un contrôleur de vol Ardupilot ainsi qu'un ensemble de contrôleurs sur site. Chaque élément de la solution est développé dans un environnement DroneKit.

Les références

Les références

- [1]. Pan, X.; Ma, D.Q.; Jin, L.L.; Jiang, Z.S. Vision-based approach angle and height estimation for UAV landing. In Proceedings of the 1st International Congress on Image and Signal Processing, CISP 2008, Sanya, China, 27–30 May 2008; Volume 3, pp. 801–805.
- [2]. Tang, D.; Li, F.; Shen, N.; Guo, S. UAV attitude and position estimation for vision-based landing. In Proceedings of the 2011 International Conference on Electronic and Mechanical Engineering and Information Technology, EMEIT 2011, Harbin, China, 12–14 August 2011; Volume 9, pp. 4446–4450.
- [3]. Gautam, A.; Sujit, P.B.; Saripalli, S. A survey of autonomous landing techniques for UAVs. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014, Orlando, FL, USA, 27–30 May 2014; pp. 1210–1218
- [4]. PX4 Dev Team. Holybro Pixhawk 4 · PX4 v1.9.0 User Guide. 2019. Available online: https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html (accessed on 23 November 2019).
- [5]. Jin, S. ; Zhang, J.; Shen, L.; Li, T. On-board vision autonomous landing techniques for quadrotor: A survey. In Proceedings of the Chinese Control Conference, Chengdu, China, 27–29 July 2016; pp. 10284–10289
- [6]. Nowak, E. ; Gupta, K.; Najjaran, H. Development of a Plug-and-Play Infrared Landing System for Multirotor Unmanned Aerial Vehicles. In Proceedings of the 2017 14th Conference on Computer and Robot Vision, Edmonton, AB, Canada, 16–19 May 2017; pp. 256–260.
- [7]. Patruno, C.; Nitti, M.; Petitti, A.; Stella, E.; D’Orazio, T. A Vision-Based Approach for Unmanned Aerial Vehicle Landing. *J. Intell. Robot. Syst.* 2019, 95, 645–664.
- [8]. Chen, X.; Phang, S.K.; Shan, M.; Chen, B.M. System integration of a vision-guided UAV for autonomous landing on moving platform. In Proceedings of the IEEE International Conference on Control and Automation, Kathmandu, Nepal, 1–3 June 2016; pp. 761–766
- [9]. Baca, T.; Stepan, P.; Spurny, V.; Hert, D.; Penicka, R.; Saska, M.; Thomas, J.; Loianno, G.; Kumar, V. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *J. Field Robot.* 2019, 36, 874–891.
- [10]. Mark Fiala. Designing highly reliable fiducial markers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1317 – 1324, July 2009.
- [11]. F. J. Madrid-Cuevas S. Garrido-Jurado, R. Muñoz-Salinas and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn.* 47, 6 (June 2014), 2014
- [12]. Quick response (qr) code. <http://www.qrcode.com/en/>.
- [13]. Lightbody, Peter (2017). "An Efficient Visual Fiducial Localisation System" (PDF). *SIGAPP Applied Computing Review*. 17 (3): 28–37. doi:10.1145/3161534.3161537. S2CID 23129425
- [14]. Tomas, Krajník (2014). "A Practical Multirobot Localization System" (PDF). *Journal of Intelligent & Robotic Systems*. 76 (3–4): 539–562. doi:10.1007/s10846-014-0041-x. S2CID 4985852.

- [15]. M. Billinghamurst H. Kato. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, pages 85–, 1999.
- [16]. D. Schmalstieg D. Wagner. Artoolkitplus for pose tracking on mobile devices. Computer Vision Winter Workshop, pages 139 – 146, 2007
- [17] https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
- [18]. Rasmussen, S.J., Chandler, P.R.: Multiuav: A multiple uav simulation for investigation of cooperative control. In: Simulation Conference, 2002. Proceedings of the Winter. vol. 1, pp. 869{877. IEEE (2002)
- [19]. Lu, P., Geng, Q.: Real-time simulation system for uav based on matlab/simulink. In: Computing, Control and Industrial Engineering (CCIE), 2011 IEEE 2nd International Conference on. vol. 1, pp. 399{404. IEEE (2011)
- [20]. ardupilot : <http://ardupilot.org>.
- [21]. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., von Stryk, O.: Comprehensive simulation of quadrotor uavs using ros and gazebo. In: Noda, I., Ando, N., Brugali, D., Kuffner, J.J. (eds.) Simulation, Modeling, and Programming for Autonomous Robots. pp. 400{411. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
- [22]. Gazebo api reference (2022), <https://gazebosim.org>
- [23]. Dronecode-Project: Mavlink developer guide (2018), <https://mavlink.io/en>
- [24]. open cv : <https://opencv.org/about/>