	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 1 / 18

Identification du document

Titre du document : Spécification d'Exigences Fonctionnelles


Référence : v0.4


Version : v1.0

Date : 20 mars 2012


Nom du fichier : SEF.docx

Identification des intervenants dans le projet


	Nom : Philippe GAUSSIER Nom de la société : ETIS Téléphone : 01 30 73 66 10 E-mail : gaussier@ensea.fr	Adresse : Université de Cergy-Pontoise ETIS - UMR 8051 2 avenue Adolphe Chauvin 95302 Cergy Pontoise Cedex
Contact 1 Nom : DESSALLE Prénom : Bruce Téléphone : 06 84 13 02 19 E-mail : bdessalle@cergy.itin.fr	Contact 2 Nom : Pellisson Prénom : Damien Téléphone : 06 89 88 29 14 E-mail : dpelliss@gmail.com	Contact 3 Nom : Noirot Prénom : Julien Téléphone : 06 84 00 35 13 E-mail : noirotj@gmail.com
Contact 4 Nom : Nesty Prénom : Olivier Téléphone : 06 26 19 38 89 E-mail : nesty.olivier@gmail.com	Contact 5 Nom : Kaiser Prénom : Florian Téléphone : 06 83 70 51 74 E-mail : mail.de.florian.kaiser@gmail.com	

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 2 / 18


Diffusion		
Société	Destinataires	Nb
ETIS	Philippe GAUSSIER	1
ITIN	Bruce DESSALLE	1
Université Cergy Pontoise	Damien PELLISSON	1
Université Cergy Pontoise	Julien NOIRIT	1
Université Cergy Pontoise	Olivier NESTY	1
Université Cergy Pontoise	Florian KAISER	1

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 3 / 18

Rédaction/Chaine de validation			
	Nom	Qualité	Date
Rédigé par :	Damien PELLISSON	Responsable logiciel	14/03/2012
Approuvé par :	Bruce DESSALLE	Chef de projet	20/03/12
Approuvé par :	Florian KAISER	Responsable Qualité	
Approuvé par :	Philippe GAUSSIÉ	Client	


	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 4 / 18

Evolutions du Document			
Version	Date	§ modifiés	Auteur
v0.1	02/12/2011	Premier Jet	Damien PELLISSON
v0.2	18/01/2012	Modification document trie des titres.	Bruce DESSALLE
v0.3	11/03/2012	Précisions sur les cas d'utilisation	Damien PELLISSON
v0.4	14/03/2012	Adaptation suite aux commentaires de Carina ROELS	Damien PELLISSON
V1.0	20/03/2012	Validation	Bruce DESSALLE

	<p>Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012</p>	<p>Réf. : v0.4 Version : v1.0 Date : 20 mars 2012</p>
	Projet ITIN / Université de Cergy-Pontoise	Page : 5 / 18

Sommaire

1	Présentation du client	6
1.1	Activité du client	6
1.2	Besoin exprimé par le client	6
2	Description du projet.....	7
2.1	Description fonctionnelle	7
2.2	Fonctionnement	7
2.2.1	Fonctionnement manuel	7
2.2.2	Fonctionnement autonome	7
2.3	Cas d'utilisation	8
2.3.1	Cas d'utilisation « Piloter avec la télécommande »	8
2.3.2	Cas d'utilisation « Donner des ordres de déplacements simples »	9
2.3.3	Cas d'utilisation « Piloter avec un Joystick ».....	9
2.3.4	Cas d'utilisation « Récupéré les informations de vol »	9
2.3.5	Cas d'utilisation « Donner des ordres de déplacement évolués ».....	10
2.3.6	Cas d'utilisation « Atterrir »	10
2.3.7	Cas d'utilisation « Décoller ».....	10
2.3.8	Cas d'utilisation « Stabiliser l'hélicoptère »	11
2.3.9	Cas d'utilisation « Rejoindre un point GPS ».....	11
2.3.10	Cas d'utilisation « Suivi de cible »	11
2.3.11	Cas d'utilisation « Enregistre un plan de vol »	12
2.3.12	Cas d'utilisation « Programmer un vol autonome »	12
2.4	Environnement de développement.....	12
3	Logiciel embarqué	13
3.1	Langage et architecture	13
3.2	Fonctionnement	14
3.2.1	Utilisation des actions.....	14
3.2.2	Utilisation du mode assisté.....	14
3.2.3	Utilisation d'une mission	15
4	Architecture client (IHM).....	16
4.1	Langage.....	16
4.2	Fonctionnement	16
5	Contraintes	17
5.1	Contraintes d'environnement	17
5.1.1	Poids à transporter	17
5.1.2	Météo	17
5.1.3	Sécurité	17
5.2	Contraintes en temps	17
5.2.1	Temps de traitement d'images	17
5.2.2	Temps de communication	17
5.3	Contrainte d'énergie.....	18

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 6 / 18

1 Présentation du client

1.1 Activité du client

Philippe GAUSSIER est responsable du laboratoire ETIS. Spécialiste en science de l'information et de la communication, il travaille sur la modélisation des mécanismes cognitifs impliqués dans la perception visuelle (vision pré-attentive et attentionnelle) et sur la modélisation de structures telles que l'hippocampe (pour des problèmes d'intégration d'informations spatio-temporelles).

Ces modèles servent de base pour imaginer des architectures de contrôle permettant à des robots mobiles d'apprendre à survivre dans un environnement à priori inconnu.

Ses robots utilisent la vision comme source principale d'information.


Les tâches étudiées sont : l'apprentissage de conditionnements visio-moteurs, la reconnaissance de lieux (retour au nid), la construction et l'utilisation de carte cognitive et enfin l'apprentissage de la discrimination d'objets et/ou de lieux.

1.2 Besoin exprimé par le client

L'objectif du client est d'avoir un drone, ici un hélicoptère, qui doit se déplacer de façon totalement autonome. Les applications possibles du drone peuvent être :

- Exploration de zone pour la recherche de personne
- Surveillance d'une zone et reconnaissance des lieux et objets
- Vol en autonomie si perte du signal radio avec retour au lieu de départ

Le projet est évidemment complexe et présente beaucoup de possibilité. Le client serait très satisfait d'avoir une plateforme de développement opérationnelle et un système en boucle fermée sur l'hélicoptère qui lui permettrait des déplacements simples en toute autonomie.

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 7 / 18

2 Description du projet

2.1 Description fonctionnelle

Le but du projet iFlyBot 2012 est de faire voler un drone en autonomie, sans intervention humaine.

Un vol autonome ne peut pas durer plus de 15 minutes à cause de la capacité de la batterie. De plus, la qualité du vol n'est assurée que s'il n'y a pas d'obstacle sur le chemin du drone, s'il capte un signal GPS correct et s'il n'y a pas de vent.

Sans signal GPS, il est cependant possible d'effectuer un vol assisté, c'est-à-dire de le piloter avec plus de facilité qu'avec la télécommande de modélisme standard. En autres, si le pilote ne touche plus aux commandes, le drone se stabilise sur sa position.

Pour atteindre ce but, une intelligence est embarquée sur le drone et un ordinateur au sol permet au pilote de lancer un vol autonome ou de prendre la main avec un joystick (pour les vols assistés).

Par précaution, le pilote peut reprendre le contrôle de l'appareil à tout moment grâce à la télécommande de modélisme.

2.2 Fonctionnement

2.2.1 Fonctionnement manuel

Au démarrage, le pilotage est en mode manuel, c'est-à-dire que le drone est contrôlé par la télécommande Graupner MC22s.

Depuis cette télécommande, il est possible de passer en mode assistance par ordinateur permettant de le contrôler grâce à un joystick. Quand aucune commande n'est envoyée, l'hélicoptère reste en vol stationnaire. Ce mode fonctionne grâce à une connexion ZigBee. Si cette connexion est perdue, alors le système bascule automatiquement en manuel.


Il est possible de revenir en mode manuel à tout moment en utilisant la télécommande.

Voir l'Utilisation du mode assis.

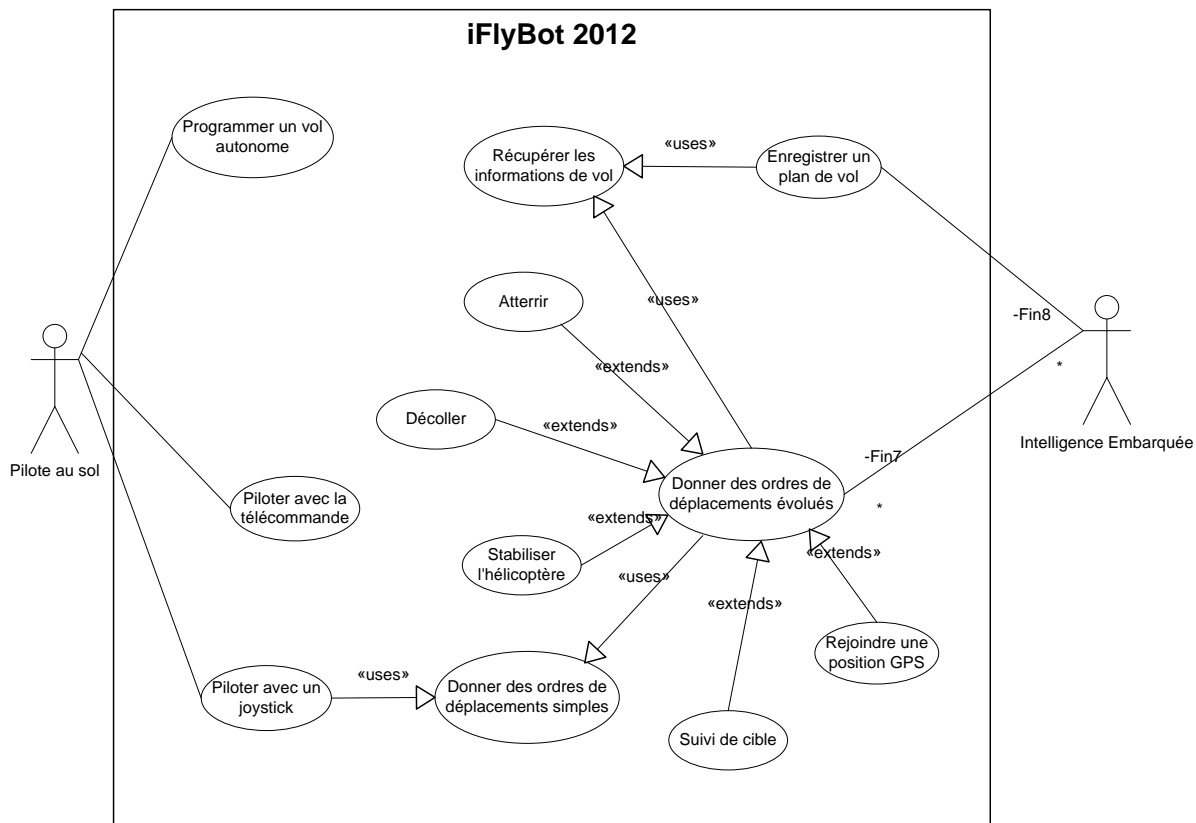
2.2.2 Fonctionnement autonome

Le pilote peut également lancer le mode autonome avec une mission préenregistrée. Selon la mission, le drone peut décoller, atterrir, suivre une cible ou rejoindre une position GPS en plus des contrôles basiques. Durant le vol, tous les mouvements calculés peuvent être enregistrés dans un fichier de log.

Voir l'Utilisation d'une mission.

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 8 / 18

2.3 Cas d'utilisation



Pré-conditions générales :

- Le drone a suffisamment de batterie
- Le drone est démarré

2.3.1 Cas d'utilisation « Piloter avec la télécommande »

Acteur : Le pilote au sol par l'intermédiaire d'une télécommande de modéliste


Pré-condition :

- La communication entre le drone et la télécommande fonctionne

Post-condition :

- Aucune

Action : Active le mode manuel. Le drone exécute les mouvements envoyés par la télécommande (utilisation standard du drone)

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 9 / 18

2.3.2 Cas d'utilisation « Donner des ordres de déplacements simples »

Acteur : L'intelligence embarquée

Pré-condition :

- Les servomoteurs sont correctement configurés et communiquent avec l'intelligence embarquée
- Le mode manuel est désactivé

Post-condition :

- L'envoi d'ordres s'arrête si le mode manuel est activé

Action : La position des servomoteurs est modifiée afin de modifier la trajectoire du drone

2.3.3 Cas d'utilisation « Piloter avec un Joystick »

Acteur : Le pilote au sol par l'intermédiaire d'un ordinateur de contrôle

Pré-conditions :

- Voir le cas « Donner des ordres de déplacements simples »
- La communication entre le drone et l'ordinateur fonctionne
- Un joystick compatible avec le logiciel est branché sur l'ordinateur

Post-condition :

- Voir le cas « Donner des ordres de déplacements simples »

Action : Active le mode assisté (si le mode n'était pas manuel). L'ordinateur de contrôle envoie des directions à suivre à l'intelligence embarquée. Elles sont ensuite converties en ordres simples. Les demandes envoyées par le joystick sont interprétées par l'ordinateur afin d'assister le pilotage (par exemple, si aucun bouton n'est utilisé, le drone restera en vol stationnaire)

2.3.4 Cas d'utilisation « Récupéré les informations de vol »

Acteurs : Le pilote au sol par l'intermédiaire d'un ordinateur de contrôle ou l'intelligence embarquée


Pré-conditions :

- Les capteurs sont correctement configurés et communiquent avec l'intelligence embarquée
- Quand l'acteur est le pilote, la communication entre le drone et l'ordinateur fonctionne

Post-condition :

- Aucune

Action : Quand l'information d'un capteur est mise à jour, il envoie les nouvelles valeurs à l'intelligence embarquée. Quand l'acteur est le pilote, le drone transmet ces informations à l'ordinateur qui se charge de les afficher à l'écran.

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 10 / 18

2.3.5 Cas d'utilisation « Donner des ordres de déplacement évolués »

Acteur : L'intelligence embarquée

Pré-condition :

- Voir le cas « Donner des ordres de déplacements simples »
- Le mode autonome est activé

Post-condition :

- Voir le cas « Donner des ordres de déplacements simples »
- L'envoi d'ordres s'arrête si le mode autonome est désactivé

Action : Les ordres évolués sont une succession de mouvements simples permettant de réaliser le mouvement souhaité (voir les cas d'utilisation étendant celui-ci). Le mouvement simple suivant est calculé en fonction du déplacement réellement effectué lors des itérations précédentes.

2.3.6 Cas d'utilisation « Atterrir »

Acteur : L'intelligence embarquée

Pré-condition :

- Voir le cas « Donner des ordres de déplacement évolués »
- Le drone vole

Post-condition :

- Voir le cas « Donner des ordres de déplacement évolués »

Action : Extension du cas « Donner des ordres de déplacement évolués ». Permet de faire atterrir le drone sur le sol

2.3.7 Cas d'utilisation « Décoller »

Acteur : L'intelligence embarquée


Pré-condition :

- Voir le cas « Donner des ordres de déplacement évolués »
- Le drone est au sol

Post-condition :

- Voir le cas « Donner des ordres de déplacement évolués »

Action : Extension du cas « Donner des ordres de déplacement évolués ». Permet de faire décoller le drone à une hauteur définie (approximative)

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 11 / 18

2.3.8 Cas d'utilisation « Stabiliser l'hélicoptère »

Acteur : L'intelligence embarquée

Pré-condition :

- Voir le cas « Donner des ordres de déplacement évolués »
- La caméra embarquée communique avec l'intelligence embarquée
- Le drone vole

Post-condition :

- Voir le cas « Donner des ordres de déplacement évolués »

Action : Extension du cas « Donnée des ordres de déplacement évolués ». Permet de stabiliser le drone à sa position actuelle (il peut dériver légèrement avant de se stabiliser)

2.3.9 Cas d'utilisation « Rejoindre un point GPS »

Acteur : L'intelligence embarquée

Pré-condition :

- Voir le cas « Donner des ordres de déplacement évolués »
- Le drone vole

Post-condition :

- Voir le cas « Donner des ordres de déplacement évolués »

Action : Extension du cas « Donnée des ordres de déplacement évolués ». Permet de déplacer le drone vers un point GPS

2.3.10 Cas d'utilisation « Suivi de cible »


Acteur : L'intelligence embarquée

Pré-condition :

- Voir le cas « Donner des ordres de déplacement évolués »
- Le GPS et la caméra embarqués communiquent avec l'intelligence embarquée
- Le drone vole
- Une cible, facilement repérable (luminosité suffisante et couleur « flashy »), est positionnée en-dessous du drone lorsque la demande de suivi est lancée

Post-condition :

- Voir le cas « Donner des ordres de déplacement évolués »
- Le suivi s'arrête quand :
 - o La cible sort du champ de vision du drone
 - o Il est interrompu par une demande d'arrêt

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 12 / 18

Action : Extension du cas « Donnée des ordres de déplacement évolués ». Permet de suivre une cible située en-dessous de lui (un objet au sol par exemple)

2.3.11 Cas d'utilisation « Enregistre un plan de vol »

Acteur : L'intelligence embarquée

Pré-conditions :

- Le GPS embarqué communique avec l'intelligence embarquée
- Le pilote au sol a demandé à l'intelligence embarquée d'enregistrer un plan de vol

Post-condition :

- L'enregistrement s'arrête quand :
 - o Le pilote au sol lui demande par l'intermédiaire de l'ordinateur de contrôle
 - o Le signal GPS est perdu

Action : Le drone enregistre son déplacement (positions GPS) afin de pouvoir le reproduire plus tard.

2.3.12 Cas d'utilisation « Programmer un vol autonome »

Acteur : Le pilote au sol par l'intermédiaire d'un ordinateur de contrôle

Pré-conditions :

- Voir le cas « Donner des ordres de déplacement évolués »
- La communication entre le drone et l'ordinateur fonctionne
- La mission demandée est connue par le drone

Post-condition :

- Voir le cas « Donner des ordres de déplacement évolués »
- Le vol autonome n'est pas interrompu par demande d'arrêt

Action : Le drone exécute les ordres envoyés par l'intelligence embarquée par l'intermédiaire du cas « Donner des ordres de déplacements évolués »

2.4 Environnement de développement

Le développement logiciel se réalisera avec les langages C++ et Java, sous Windows 7 ou Linux avec les logiciels suivants :

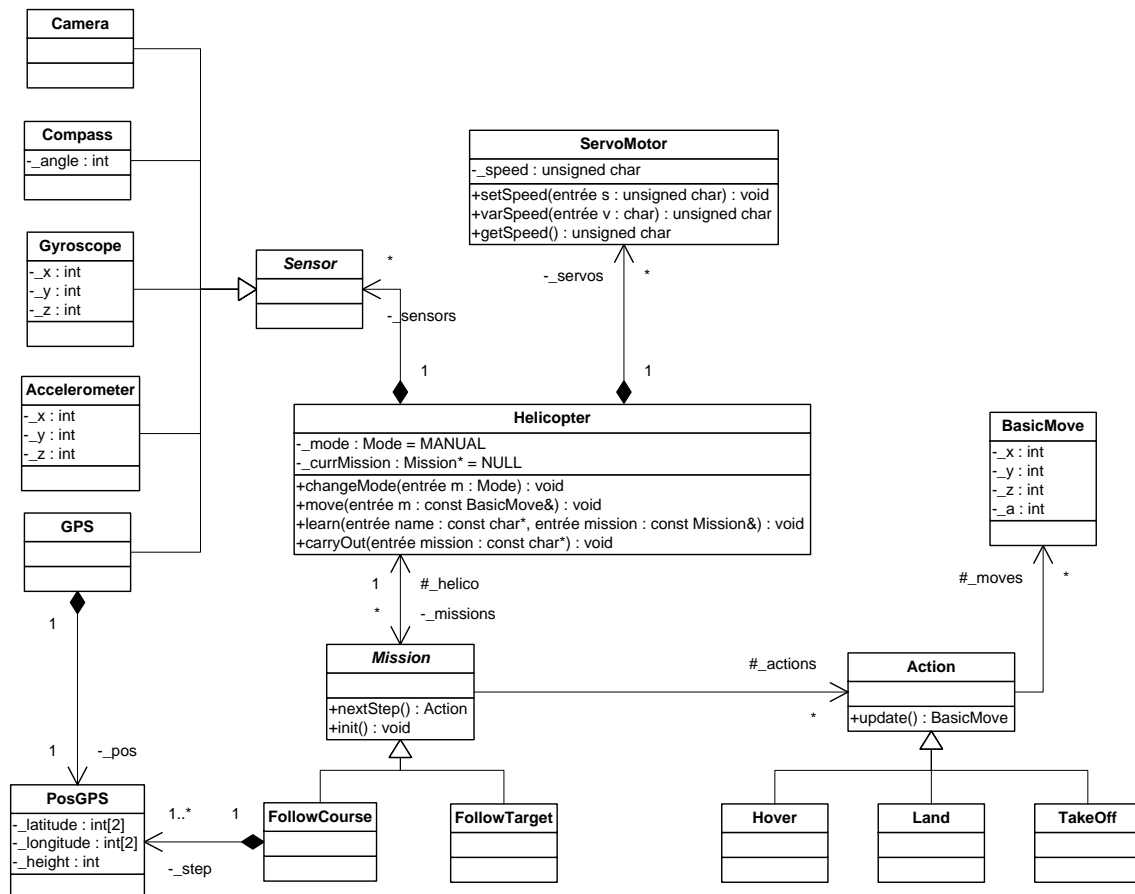
Logiciels	Description
Eclipse	IDE multi-langages
SVN	Logiciel de gestion de versions
GCC	Compilateur pour le langage C++

3 Logiciel embarqué

3.1 Langage et architecture

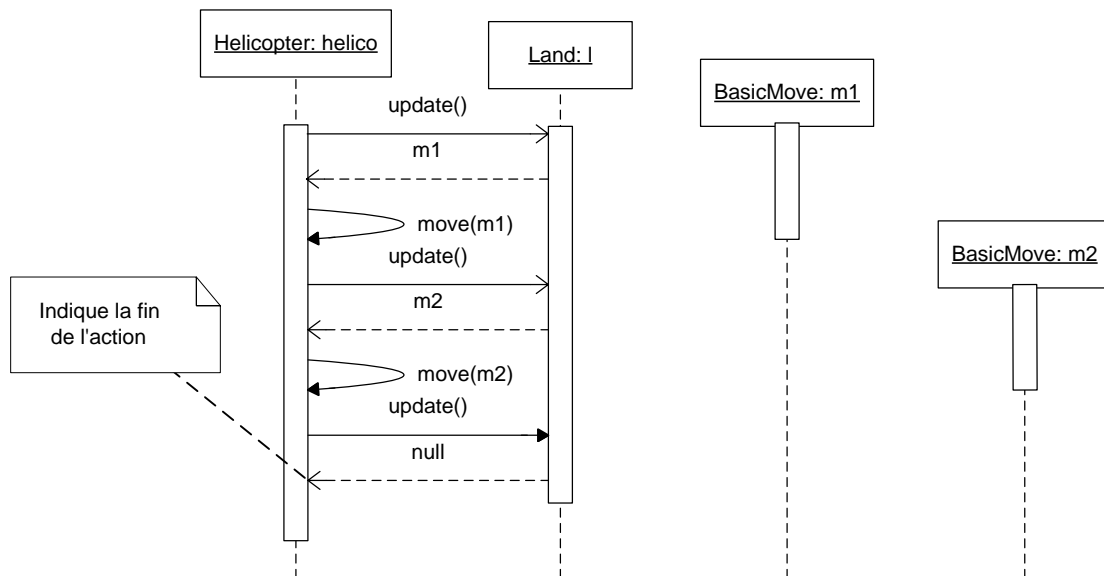
La partie logicielle qui sera intégrée dans le matériel embarqué sera développée en C++. Nous avons choisi ce langage pour deux raisons :

- La programmation orientée objet
- La rapidité d'exécution

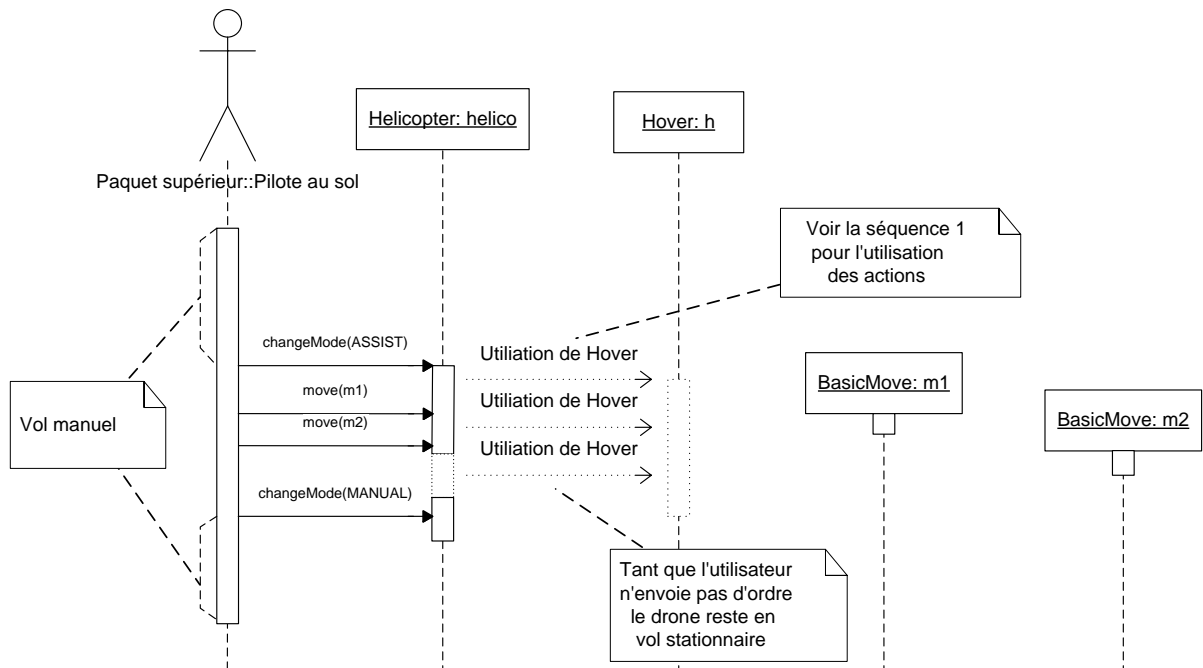


3.2 Fonctionnement

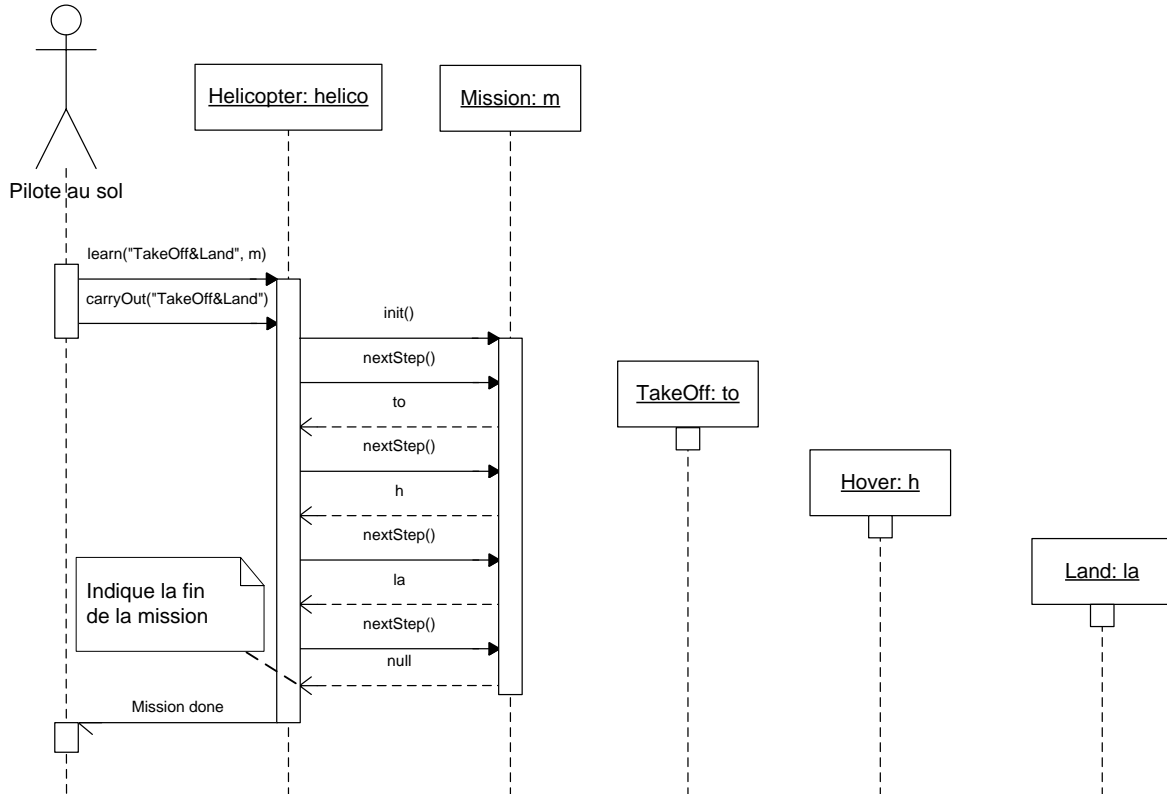
3.2.1 Utilisation des actions




3.2.2 Utilisation du mode assisté



3.2.3 Utilisation d'une mission



	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 16 / 18

4 Architecture client (IHM)

4.1 Langage

La partie logicielle qui sera intégrée au PC permettant de communiquer avec le drone sera développée en Java. Nous avons choisi ce langage pour deux raisons :


- Le fonctionnement multiplateforme
- La simplicité pour créer des interfaces graphiques

4.2 Fonctionnement

L'application permettra à la fois d'envoyer des commandes à l'hélicoptère mais aussi de recevoir et d'afficher des informations provenant de l'hélicoptère.

Cette application devra être claire et simple d'utilisation avec peu de fonctionnalités. L'utilisateur pourra enregistrer des missions ce qui lui permettra d'enrichir les options de pilotage.

Ce référer à la partie 8, résumé des fonctionnalités dans le document des spécifications IHM.

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 17 / 18

5 Contraintes

5.1 Contraintes d'environnement

5.1.1 Poids à transporter

Pour le quadricoptère, se référer au paragraphe 3.2.1 du document « Spécifications d'Architecture Matérielle ».

Pour l'hélicoptère, se référer au paragraphe 3.3.1 du document « Spécifications d'Architecture Matérielle ».

5.1.2 Météo

Le quadricoptère ne peut être utilisé que dans des conditions climatiques où le vent est quasi-inexistant. La plupart des tests que nous aurons à effectuer se feront dans un milieu clôt (ex : gymnase).

5.1.3 Sécurité

L'hélicoptère ne peut être piloté, légalement, que par un personnel qualifié. Nos tests ne peuvent donc être faits que par un modéliste : ce qui implique une certaine organisation pour que nos tests soient faits dans le cadre de sessions de pilotage par ce dernier à intervalle régulier.

5.2 Contraintes en temps

5.2.1 Temps de traitement d'images


Nous distinguons deux cas de figures : le vol stationnaire et la suivie de cible.

Le vol stationnaire n'a pas besoin de beaucoup d'images par seconde, nous estimons que la stabilisation du drone nécessite 4 images par seconde.

En revanche, la suivie de cible nécessite un peu plus : par conséquent, pour suivre une cible se déplaçant à 3km/h, nous aurons besoin d'environ 10 images par seconde. Nous devons donc nous plier à cette dernière contrainte pour avoir un suivi de cible correct.

5.2.2 Temps de communication

Pour éviter que l'humain pilotant le drone ait une impression de latence, il faut que l'ordre donné au drone soit réalisé par ce dernier dans un délai n'excédant pas 250ms, comprenant la communication et l'exécution (durée moyenne que l'humain met pour exécuter une action qu'on lui a ordonné).

	Document : Spécification d'Exigences Fonctionnelles Projet : iFlyBot 2012	Réf. : v0.4 Version : v1.0 Date : 20 mars 2012
	Projet ITIN / Université de Cergy-Pontoise	Page : 18 / 18

5.3 Contrainte d'énergie

Sur le quadricoptère, le vol ne peut durer plus de 15 minutes, durée maximale des batteries. En effet, une batterie ayant une plus grande autonomie serait trop lourde.

Sur l'hélicoptère, le vol dure environ 30 à 45 min grâce au réservoir de carburant. Et nous pouvons embarquer une charge plus élevée et utiliser des batteries ayant une meilleure autonomie pour que les cartes embarquées soient alimentées durant la totalité du vol.