



École Polytechnique de l'Université de Tours

64, Avenue Jean Portalis
37200 TOURS, France
Tél. +33 (0)2 47 36 14 14

www.polytech.univ-tours.fr

Département Informatique industrielle

Cahier de spécifications			
Projet :	Système de stabilisation motorisé pour caméra		
Réalisé par :	Sébastien LAPORTE	Encadré par :	Ameur SOUKHAL
Début du projet :	22/09/2015	Fin du projet :	10/02/2015
Historique des modifications			
Version	Date	Note de version	
1.0	14/11/2015	Version de dépôt initial	

TABLE DES MATIERES

1. INTRODUCTION	5
2. CONTEXTE DE LA REALISATION	5
2.1. CONTEXTE	5
2.2. OBJECTIFS	5
2.3. EXISTANT	5
2.4. BASES METHODOLOGIQUES	5
3. DESCRIPTION GENERALE	5
3.1. ENVIRONNEMENT DU PROJET.....	5
3.2. CARACTERISTIQUES DES UTILISATEURS	6
3.3. FONCTIONNALITES ET STRUCTURE GENERALE DU SYSTEME.....	6
<i>Cas d'utilisation</i>	<i>6</i>
<i>Schéma du système</i>	<i>6</i>
3.4. CONTRAINTES DE DEVELOPPEMENT.....	7
4. DESCRIPTION DES INTERFACES EXTERNES	8
4.1. INTERFACES MATERIEL / LOGICIEL	8
4.2. INTERFACES HOMME / MACHINE	8
4.3. INTERFACES LOGICIEL / LOGICIEL.....	9
5. ARCHITECTURE GENERALE DU SYSTEME	10
6. DESCRIPTION DES FONCTIONNALITES	11
6.1. FONCTION D'ACQUISITION DES DONNEES	11
<i>Identification de la fonction.....</i>	<i>11</i>
<i>Description de la fonction</i>	<i>11</i>
6.2. FONCTION DE TRAITEMENT DES DONNEES	11
<i>Identification de la fonction.....</i>	<i>11</i>
<i>Description de la fonction</i>	<i>11</i>

6.3.	FONCTION DE CORRECTION DE L'ORIENTATION.....	12
	<i>Identification de la fonction.....</i>	<i>12</i>
	<i>Description de la fonction.....</i>	<i>12</i>
6.4.	FONCTION D'ASSERVISSEMENT EN POSITION	12
	<i>Identification de la fonction.....</i>	<i>12</i>
	<i>Description de la fonction.....</i>	<i>12</i>
6.5.	FONCTION DE COMMANDE DE LA MOTORISATION	13
	<i>Identification de la fonction.....</i>	<i>13</i>
	<i>Description de la fonction.....</i>	<i>13</i>
7.	CONDITIONS DE FONCTIONNEMENT	14
7.1.	PERFORMANCES.....	14
7.2.	CAPACITE	14
7.3.	MODES DE FONCTIONNEMENT.....	14
7.4.	CONFORMITE AUX STANDARDS	14
8.	DECOUPAGE DU PROJET EN TACHES.....	15
8.1.	ÉTUDE BIBLIOGRAPHIQUE.....	15
8.2.	ÉTUDE DE FAISABILITE ET SPECIFICATIONS.....	15
8.3.	ANALYSE ET MODELISATION.....	15
8.4.	DEVELOPPEMENT	16
8.4.1.	REALISATION DE LA FONCTION « CALIBRATION DU CAPTEUR »	16
8.4.2.	REALISATION DE LA FONCTION « ACQUISITION DES DONNEES »	16
8.4.3.	REALISATION DE LA FONCTION « TRAITEMENT DES DONNEES SUR 2 AXES »	16
8.4.4.	REALISATION DE LA FONCTION « TRAITEMENT DES DONNEES SUR 3 AXES »	17
8.4.5.	REALISATION DE LA FONCTION « ASSERVISSEMENT EN POSITION ».....	17
8.4.6.	REALISATION DE LA FONCTION « CORRECTION DE L'ORIENTATION ».....	17
8.4.7.	REALISATION DE LA FONCTION « COMMANDE DES MOTEURS »	17
8.4.8.	INTEGRATION DES FONCTIONS POUR UN PROTOTYPE INITIAL SUR 2 AXES	17

8.4.9.	INTEGRATION DES FONCTIONS POUR UN PROTOTYPE INITIAL SUR 3 AXES	18
8.4.10.	DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « ACQUISITION DES DONNEES »	18
8.4.11.	DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « CALIBRATION DU CAPTEUR ».....	18
8.4.12.	DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « TRAITEMENT DES DONNEES ».....	18
8.4.13.	DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « ASSERVISSEMENT »	19
8.4.14.	DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « COMMANDE DES MOTEURS »	19
8.4.15.	DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « CORRECTION DE L'ORIENTATION »	19
8.4.16.	INTEGRATION DU CODE NATIF POUR LE PROTOTYPE FINAL.....	19
8.5.	CONCEPTION ELECTRONIQUE	19
8.6.	CONCEPTION MECANIQUE.....	20
8.7.	REALISATION	20
8.7.1.	REALISATION DU MONTAGE ELECTRONIQUE SUR PROTOBOARD.....	20
8.7.2.	REALISATION D'UN BANC D'ESSAIS MOTEUR.....	20
8.7.3.	REALISATION D'UN PROTOTYPE INITIAL.....	20
8.7.4.	REALISATION MECANIQUE DU PROTOTYPE FINAL	20
8.7.5.	INTEGRATION DE L'ELECTRONIQUE AU PROTOTYPE FINAL	21
8.8.	TESTS	21
8.9.	DOCUMENTATION	21
9.	LISTE DES TACHES.....	22
10.	PLANNING.....	23
	BIBLIOGRAPHIE	24
	TABLE DES ILLUSTRATIONS.....	24
	GLOSSAIRE	25

1. INTRODUCTION

Ce document a pour objectif de définir les besoins, l'environnement et les objectifs pour la réalisation du projet de fin d'étude à Polytech Tours. Ce document décrit également le découpage des tâches effectué avec un planning prévisionnel et identifie les différents livrables permettant de valider les étapes au cours du projet. Le porteur du projet qui agit également en tant que MOE est Sébastien Laporte, étudiant en cinquième année de cycle d'ingénieur en informatique industrielle. L'encadrant du projet représentant la MOA est Monsieur Ameur Soukhal, enseignant chercheur au CNRS.

2. CONTEXTE DE LA REALISATION

2.1. CONTEXTE

Depuis quelques années, le marché des caméras d'action est en constante évolution. C'est la marque GoPro qui a démocratisé ce type de matériel et qui s'est imposée comme leader dans le domaine. Grâce à cette gamme de caméra, il est possible de filmer ses activités sportives, sur tous les terrains avec un minimum d'encombrement. Cependant, malgré la miniaturisation et la qualité du capteur, il est souvent difficile de filmer de manière stable lorsqu'on conduit une moto ou que l'on descend une piste de ski. C'est donc dans ce contexte que la MOA propose ce projet afin de trouver une solution à ce besoin.

2.2. OBJECTIFS

L'objectif de ce projet est de concevoir et réaliser un système de stabilisation pour caméra d'action. Fixée sur le système, la caméra sera stabilisée de façon mécanique pour permettre à l'utilisateur de filmer tout en conservant une image stable. Cette stabilisation sera assurée par des effecteurs électriques. Sont utilisation doit être simple et intuitive.

2.3. EXISTANT

Certains projets ont vu le jour avec la prolifération des drones pourvus de caméras sur nacelles stabilisées. Le choix qui a été pris dans le cadre de ce projet est de considérer que le projet n'a pas d'existant ni matériel, ni logiciel.

2.4. BASES METHODOLOGIQUES

Souhaitant proposer une première version du système le plus tôt possible, nous avons choisi d'utiliser la méthode SCRUM, qui fait parti des méthodes agiles. Le découpage des tâches est détaillé dans la partie 8 de ce document.

3. DESCRIPTION GENERALE

3.1. ENVIRONNEMENT DU PROJET

Ce PFE s'inscrit dans un travail de recherche et développement effectué au sein de Polytech Tours afin de trouver des solutions au problème étudié. Le système à développer est totalement indépendant de tout autre projet. Il contiendra donc l'ensemble des fonctions matérielles et logicielles.

3.2. CARACTERISTIQUES DES UTILISATEURS

L'utilisateur doit pouvoir utiliser le produit sans aucune connaissance sur le plan informatique ou technique. Son utilisation doit être intuitive pour pouvoir être utilisée par des étudiants en école de cinéma ou des professionnels.

3.3. FONCTIONNALITES ET STRUCTURE GENERALE DU SYSTEME

CAS D'UTILISATION

Du point de vue de l'utilisateur, le système a pour unique fonctionnalité la stabilisation d'une caméra. Pour assurer cette fonction de stabilisation, les fonctions internes au système sont détaillées dans la partie « Description des fonctionnalités ».

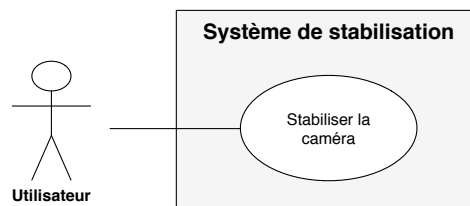
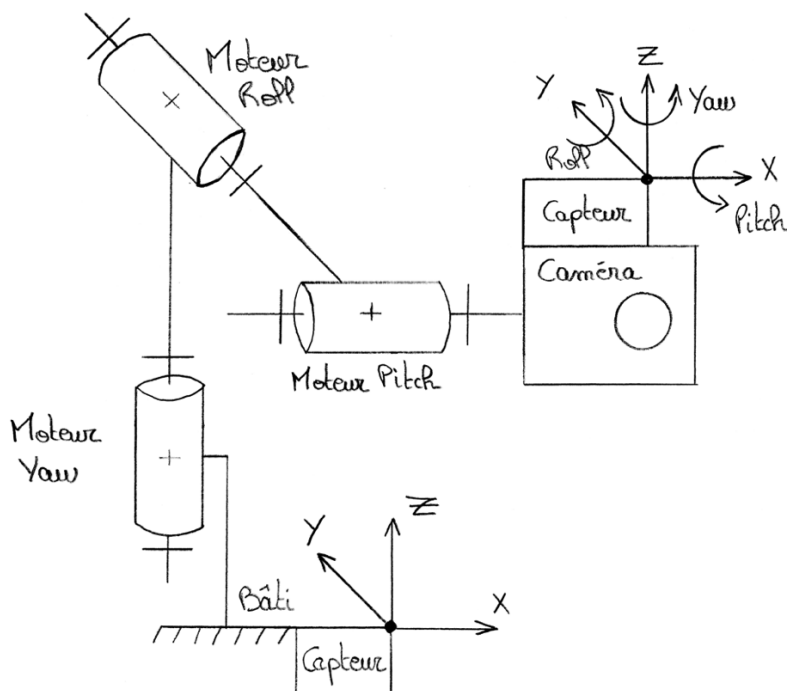


Figure 1 : Cas d'utilisation

SCHEMA DU SYSTEME

Le système est composé d'une structure mécanique motorisée sur trois axes, de capteurs d'orientation et d'un système embarqué, le tout fonctionnant sur batterie. Le schéma ci-après présente la structure générale et définit le repère 3D.



Le système comporte deux capteurs : Le capteur principal est fixé sur la caméra, il permet d'effectuer la stabilisation.

Le second capteur permet d'obtenir l'orientation du bâti, et par conséquent, l'orientation souhaitée par l'utilisateur.

Figure 2 : Schéma du système

3.4. CONTRAINTES DE DEVELOPPEMENT

- **Langage de programmation** : Langage C
- **Traitement des données** : Une étude préliminaire a permis d'identifier le type de capteur à utiliser pour acquérir l'orientation dans l'espace : Un *capteur IMU 9DOF*. Les données issues d'un tel capteur devront être filtrées et traitées afin d'obtenir l'orientation de la caméra sur les 3 axes : X, Y et Z. Il sera donc nécessaire de choisir et d'implémenter un algorithme pour obtenir les informations d'orientation qui sont un prérequis à la stabilisation du système. Ce travail de recherche sera effectué avec la tâche « Étude bibliographique ».
- **Besoin important en ressources de calcul** : Les algorithmes de traitement des données issues d'un capteur 9 axes (gyroscope, accéléromètre, magnétomètre) impliquent une forte demande en calculs. Le microcontrôleur devra être choisi en fonction de cette contrainte.
- **Faible poids et encombrement** : Le système doit être utilisable par un utilisateur en le portant à une seule main. Son poids devra donc être contenu pour qu'il puisse être utilisé sans fatiguer son porteur. De plus, l'encombrement du système doit être réduit pour qu'il puisse être fixé sur une moto ou autre engin motorisé. Ces deux contraintes seront à prendre en compte lors de la conception mécanique.
- **Alimentation autonome** : L'autonomie est un point important de ce système qui doit être utilisable pendant plus d'une heure sans changer de batterie. Cette contrainte devra être prise en compte lors du dimensionnement de la partie alimentation.
- **Prix** : Le coût de l'ensemble des composants doit être limité pour que le projet soit viable. On trouve aujourd'hui des systèmes de stabilisation pour ce type de caméra aux alentours des 300€. Le coût de l'ensemble du système devra s'approcher au mieux de cette somme.

4. DESCRIPTION DES INTERFACES EXTERNES

4.1. INTERFACES MATERIEL / LOGICIEL

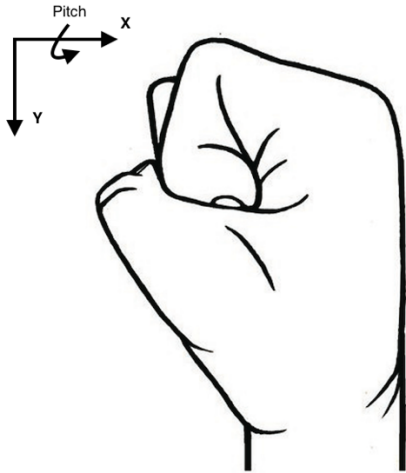
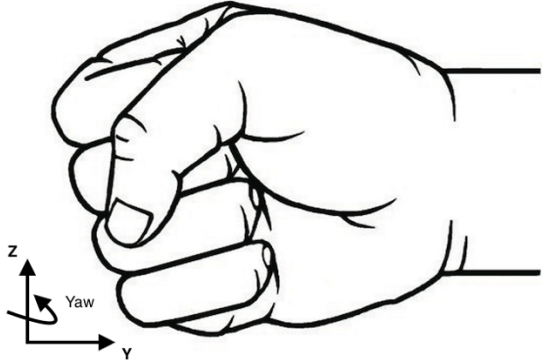
- **Bus I2C** : Le capteur IMU utilise une liaison I2C pour communiquer avec le microcontrôleur. Sur cette liaison transitent les données de l'accéléromètre, du gyroscope et du magnétomètre.
- **Moteurs brushless** : La commande de ce type de moteur se fait par 3 signaux PWM.

4.2. INTERFACES HOMME / MACHINE

Le fonctionnement du système ne requiert pas d'interface homme / machine. Seul un interrupteur de mise sous tension permet la mise en route du système. Cependant, l'orientation de la caméra se fait en fonction du comportement de l'utilisateur.

Le système étant stabilisé, celui-ci tend à compenser l'orientation de la caméra pour la conserver dans une certaine position définie. Cependant, l'utilisateur a la possibilité de modifier l'orientation de la caméra en agissant sur l'orientation du système. En effet, par nature, la stabilisation compense les mouvements intempestifs de la caméra. Si l'utilisateur adopte un mouvement qui tend à changer l'orientation de la caméra, le système va alors s'adapter et corriger sa position de stabilisation.

L'utilisateur a la possibilité de modifier l'orientation de la caméra sur les axes Pitch et Yaw. On ne souhaite pas modifier l'axe Roll car celui-ci est perpendiculaire au sol, ce qui produirait un angle de vidéo indésirable. Le schéma ci-après présente l'interaction de l'utilisateur avec le système.

Correction sur X : Pitch	Correction sur Z : Yaw
 <p>Figure 3 : Correction sur l'axe X par l'utilisateur</p>	 <p>Figure 4 : Correction sur l'axe Z par l'utilisateur</p>
<p>Cette correction permet à l'utilisateur d'incliner la caméra de bas en haut et de haut en bas. L'utilisateur est limité à plus ou moins 35 degrés pour continuer à assurer une stabilisation (qui nécessite une amplitude de mouvement).</p>	<p>Cette correction permet à l'utilisateur de faire pivoter la caméra de gauche à droite et de droite à gauche. Il n'y a aucune limite de mouvement sur cet axe.</p>

4.3. INTERFACES LOGICIEL / LOGICIEL

Le système a pour vocation d'être entièrement autonome. Celui-ci ne s'interface donc avec aucun logiciel. En effet, l'ensemble des réglages sont effectués dès la conception du système. Celui-ci étant conçu pour une seule caméra, il n'est pas nécessaire de corriger des paramètres au cours la vie du système.

5. ARCHITECTURE GENERALE DU SYSTEME

Le système est composé des 3 sous-ensembles suivants : Captage, Traitement et partie opérative.

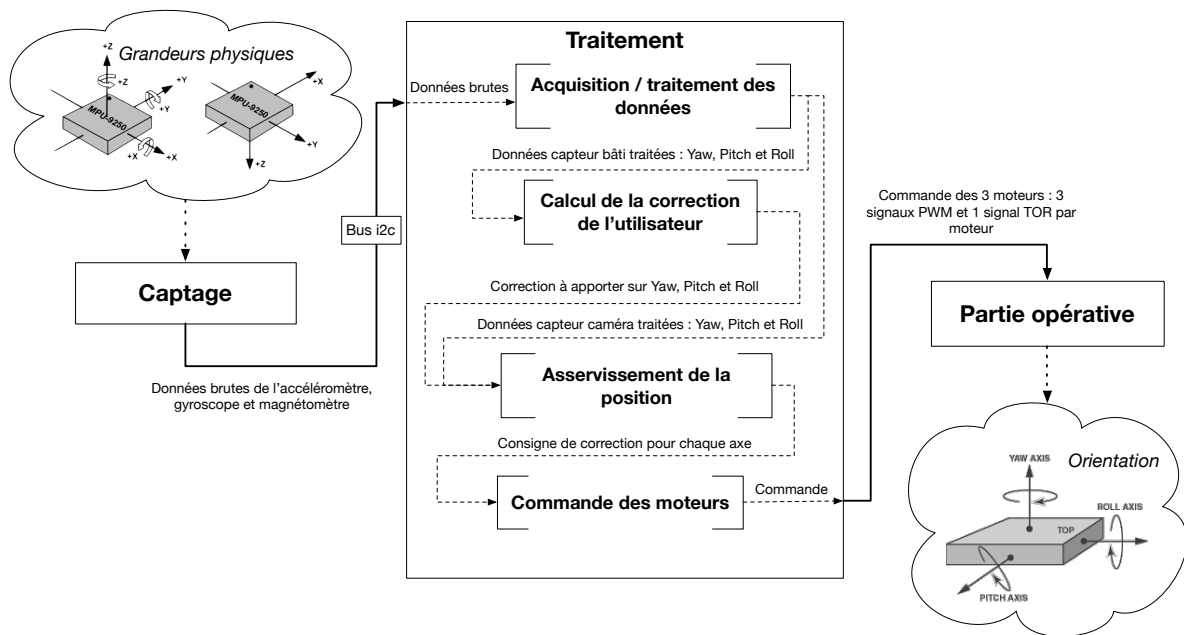


Figure 5 : Architecture du système

- **Captage** : Sous-ensemble permettant de fournir au système les données brutes nécessaires à la modélisation de l'orientation de la caméra et du système dans un espace 3D.
- **Traitement** : Sous-ensemble permettant de réaliser plusieurs fonctions : Traitement des données, calcul de la correction de l'utilisateur, asservissement en position et commande des moteurs.
 - **Fonction « Acquisition / traitement des données »** : Permet d'obtenir l'orientation exacte de la caméra et du bâti à partir des données brutes envoyées par le sous-ensemble « Captage ».
 - **Fonction « Calcul de la correction de l'utilisateur »** : Cette fonction permet de calculer la correction à apporter pour modifier l'orientation de la caméra en fonction de celle du bâti. C'est grâce à cette fonction que l'utilisateur a la possibilité d'agir sur l'angle Yaw et Pitch de la caméra (Présenté dans la partie 4.2).
 - **Fonction « Asservissement de la position »** : Permet de calculer la correction à appliquer sur le sous-ensemble « Partie opérative » pour corriger l'orientation de la caméra en fonction des données produites par la fonction « Traitement des données ».
 - **Fonction « Commande des moteurs »** : Permet d'effectuer la commande des moteurs en fonction de la consigne calculée par la fonction « Asservissement de la position ».
- **Partie opérative** : Sous-ensemble constitué d'une structure mécanique et de moteurs permettant d'orienter la caméra dans un espace 3D.

6. DESCRIPTION DES FONCTIONNALITES

6.1. FONCTION D'ACQUISITION DES DONNEES

IDENTIFICATION DE LA FONCTION

- **Présentation** : Cette fonction permet la récupération des informations issues du capteur MPU-9250. Ce capteur intègre un accéléromètre, un gyroscope ainsi qu'un magnétomètre qui ont leurs données accessibles depuis un bus i2c. L'acquisition des données consiste à dialoguer avec le capteur en i2c pour obtenir les informations.
- **Priorité** : Primordiale

DESCRIPTION DE LA FONCTION

- **Entrée** : Cette fonction nécessite en entrée l'adresse du capteur sur le bus i2c ainsi que l'ensemble des registres nécessaires à son utilisation. L'ensemble de ces valeurs est extrait de la datasheet et disponible en annexe de ce document.
- **Sortie** : Cette fonction doit retourner les données de l'accéléromètre, du gyroscope et du magnétomètre sur les 3 axes : respectivement ax, ay, az, gx, gy, gz, mx, my, mz.



Figure 6 : Entrées / sorties de la fonction « Acquisition des données »

6.2. FONCTION DE TRAITEMENT DES DONNEES

IDENTIFICATION DE LA FONCTION

- **Présentation** : Les données issues du capteur sont brutes. Afin d'obtenir l'image de l'orientation du capteur, il est nécessaire d'effectuer un traitement. Cette fonction permet donc, grâce à un algorithme d'obtenir ces données.
- **Priorité** : Primordiale

DESCRIPTION DE LA FONCTION

- **Entrée** : Cette fonction prend en entrée les données brutes produites par la fonction « Acquisition des données ». Il y a donc gx, gy, gz pour les données du gyroscope, ax, ay, az pour les données de l'accéléromètre et mx, my, mz pour les données du magnétomètre.
- **Sortie** : Cette fonction permet d'obtenir l'orientation de la caméra au format Yaw, Pitch, Roll.



Figure 7 : Entrées / sorties de la fonction « Traitement des données »

6.3. FONCTION DE CORRECTION DE L'ORIENTATION

IDENTIFICATION DE LA FONCTION

- **Présentation** : Cette fonction permet de prendre en compte les mouvements volontaires de l'utilisateur afin de modifier l'orientation de la caméra.
- **Priorité** : Secondaire

DESCRIPTION DE LA FONCTION

- **Entrée** : Cette fonction prend en entrée les données Yaw, Pitch, Roll produites par la fonction « Traitement des données » issues du capteur du bâti.
- **Sortie** : On obtient une correction pour les angles Yaw et Pitch.



Figure 8 : Entrées / sorties de la fonction « Correction de l'orientation »

6.4. FONCTION D'ASSERVISSEMENT EN POSITION

IDENTIFICATION DE LA FONCTION

- **Présentation** : Cette fonction permet d'effectuer l'asservissement en position. C'est cette fonction qui calcule la consigne à appliquer sur la commande des moteurs en fonction des données d'orientation.
- **Priorité** : Primordiale

DESCRIPTION DE LA FONCTION

- **Entrée** : Cette fonction prend en entrée les données Yaw, Pitch, Roll produites par la fonction « Traitement des données ».
- **Sortie** : On obtient la consigne pour la commande des 3 moteurs correspondant à la correction à appliquer sur les 3 axes.



Figure 9 : Entrées / sorties de la fonction « Asservissement en position »

6.5. FONCTION DE COMMANDE DE LA MOTORISATION

IDENTIFICATION DE LA FONCTION

- **Présentation** : La commande de la motorisation consiste à agir sur la partie électronique qui pilote les moteurs. En fonction d'une consigne, cette fonction permet de commander la mise en route ou l'arrêt du moteur ainsi que de faire varier la vitesse.
- **Priorité** : Primordiale

DESCRIPTION DE LA FONCTION

- **Entrée** : La fonction nécessite en entrée la consigne générée par la fonction « Asservissement en position » sur chaque axe.
- **Sortie** : La fonction génère 9 signaux PWM et 3 signaux TOR : 3 signaux PWM par moteur qui permettent de commander les 3 phases des moteurs brushless ainsi qu'un signal TOR pour activer ou désactiver le moteur.

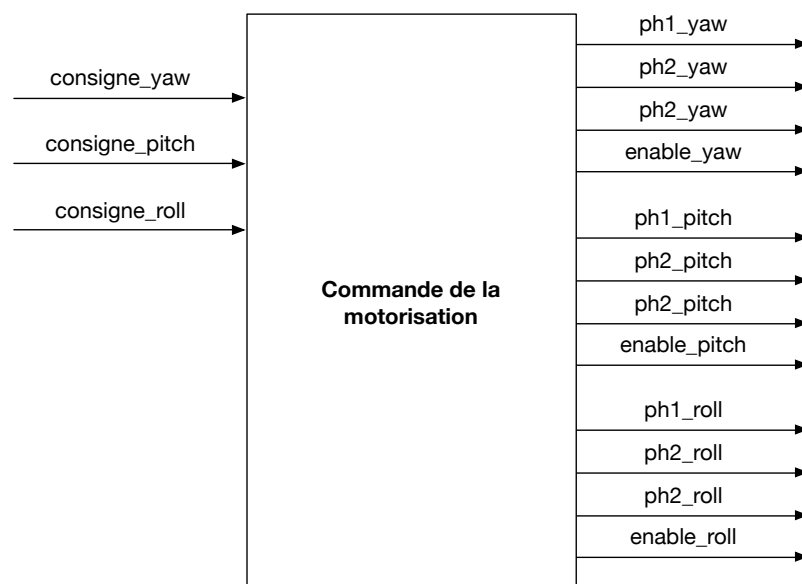


Figure 10 : Entrées / sorties de la fonction « Commande de la motorisation »

7. CONDITIONS DE FONCTIONNEMENT

7.1. PERFORMANCES

Le temps de réponse du système doit être rapide pour permettre une stabilisation avec amortissement des mouvements parasites générés par l'utilisateur.

7.2. CAPACITE

Le système doit être utilisable par un utilisateur en le portant à la main. Il doit être également possible de le fixer sur un support pour être embarqué sur un véhicule. Le poids et l'encombrement doivent être limités au maximum.

7.3. MODES DE FONCTIONNEMENT

A la mise sous tension, le système s'initialise en positionnant la caméra dans le même alignement que le système. Cette initialisation doit être transparente au maximum pour l'utilisateur.

7.4. CONFORMITE AUX STANDARDS

Le système doit pouvoir accueillir un boîtier de fixation type « Frame » conçu pour les caméras GoPro. Ceci doit garantir à l'utilisateur de pouvoir fixer n'importe quelle caméra de la marque GoPro, dans les limites d'encombrement actuellement pratiquées (Modèle 1, 2, 3 et 4).

8. DECOUPAGE DU PROJET EN TACHES

Au vu du travail à réaliser, le projet sera réalisé en plusieurs « Sprints » avec la méthode AGILE. Ainsi, le projet évoluera au fur et à mesure des versions pour arriver jusqu'au prototype final.

Remarque concernant les tâches « Acquisition des données brutes du capteur », « Traitement des données sur 3 axes », « Correction de l'orientation », « Asservissement en position » et « Commande des moteurs » : Le développement sera effectué en 2 phases : Une première version sera produite sur plateforme Arduino afin d'**élaborer l'algorithme** ou les mécanismes à mettre en place. Lorsque celle-ci sera validée, la seconde phase consistera à développer une version en C natif pour le prototype final.

8.1. ÉTUDE BIBLIOGRAPHIQUE

En début de projet, il est nécessaire d'effectuer une étude des différentes technologies existantes. De plus, la partie captage du projet induit des notions physiques qu'il est nécessaire de connaître. Cette étude s'effectue en début de projet mais elle pourra également être présente de façon ponctuelle durant le déroulement du projet.

- **Estimation de la charge de travail** : 1 semaine
- **Livrable** : Archive de l'ensemble des ressources récoltées lors de l'étude.

8.2. ÉTUDE DE FAISABILITE ET SPECIFICATIONS

Cette tâche est découpée en sous-parties pour se concentrer sur chaque sous-ensemble du système :

- **Partie captage** : Choix et étude du capteur pour l'acquisition des données d'orientation
- **Partie opérative** : Choix des effecteurs électriques et étude de leur commande
- **Partie traitement** : Choix du microcontrôleur et sélection de la plateforme pour le prototypage
- **Partie électronique** : Étude de l'alimentation et choix d'une solution. Choix de la solution matérielle pour la commande des effecteurs électriques choisis.

Remarque : Le choix du matériel fera l'objet d'un document à la remise du rapport de projet.

Cette étude de faisabilité s'appuiera sur l'étude bibliographique précédemment réalisée pour étudier l'ensemble des points nécessaires à la réalisation du projet et mettre en œuvre les ressources nécessaires.

- **Livrable** : Cahier de spécifications. Il doit être validé par la MOA et la MOE.
- **Contrainte temporelle** : Rendu du livrable le 15/11/2015
- **Estimation de la charge de travail** : 4 semaines

8.3. ANALYSE ET MODELISATION

La tâche « Analyse et modélisation » est découpée en 4 sous-tâches :

- Identification des cas d'utilisation
- Réalisation des diagrammes de séquence
- Schématisation de la structure mécanique
- Schémas fonctionnels électroniques

L'objectif de cette tâche est de réaliser le travail nécessaire permettant par la suite le développement, la conception électronique ainsi que la conception mécanique.

- **Livrable** : Document d'analyse et modélisations
- **Contrainte temporelle** : Rendu du livrable le 18/12/2015
- **Estimation de la charge de travail** : 4 semaine

8.4. DEVELOPPEMENT

La partie développement est divisée en deux parties :

- ❖ Dans un premier temps l'objectif est de créer le code permettant de **réaliser les fonctions**. Lors de ces tâches, nous utiliserons l'IDE Arduino avec l'ensemble de ses bibliothèques dans le but de se concentrer sur la **partie algorithmique** des fonctions en faisant abstraction de la cible finale.
- ❖ Après avoir réalisé l'ensemble des fonctions et validé leur fonctionnement, les tâches de réalisation du code natif auront pour objectif de s'affranchir des bibliothèques Arduino dans le but de proposer un **programme entièrement autonome et optimisé** pour la cible.

8.4.1. REALISATION DE LA FONCTION « CALIBRATION DU CAPTEUR »

Les capteurs MPU-9250 intègrent 3 types de capteur qui nécessitent chacun une calibration. Cette fonction doit permettre de réaliser la calibration du capteur pour que les données exploitées par la suite soient correctes. Cette tâche s'appuie donc sur les recherches effectuées lors de la première phase d'étude mais pourra également nécessiter des recherches complémentaires.

- **Estimation de la charge de travail** : 1 semaine
- **Livrable** : Code source du projet Arduino et démonstration de la fiabilité des résultats obtenus à la MOA.

8.4.2. REALISATION DE LA FONCTION « ACQUISITION DES DONNEES »

La réalisation de cette tâche consiste à développer une fonction sur plateforme Arduino permettant de faire la communication avec le capteur MPU-9250 via le bus I2C en récupérant les informations du gyroscope, de l'accéléromètre et du magnétomètre. Lors de la réalisation de cette tâche, nous nous appuierons sur les bibliothèques Arduino pour la communication I2C.

- **Estimation de la charge de travail** : 1 jour
- **Livrable** : Code source du projet Arduino et démonstration de l'acquisition des données brutes du capteur à la MOA.

8.4.3. REALISATION DE LA FONCTION « TRAITEMENT DES DONNEES SUR 2 AXES »

L'objectif de cette tâche est de faire un programme sur plateforme Arduino pour réaliser la fonction de traitement des données sur 2 axes. Celle-ci permet de contribuer à l'élaboration d'une première version du prototype en réalisant la fonction de traitement des données sur les axes : Pitch et Roll. En effet, l'étude préliminaire a mis en avant que l'axe Yaw apporte une complexité au système. Dans un premier temps, il sera nécessaire d'étudier les différentes méthodes et algorithmes pour obtenir la fonction de traitement souhaitée.

- **Estimation de la charge de travail** : 1 semaine
- **Contrainte** : Nécessite que la fonction « Acquisition des données » soit réalisée
- **Livrable** : Code source du projet Arduino et démonstration de l'acquisition des données traitées sur les deux axes Pitch et Roll.

8.4.4. REALISATION DE LA FONCTION « TRAITEMENT DES DONNEES SUR 3 AXES »

La tâche de traitement des données sur 3 axes doit s'appuyer sur le travail réalisé lors de la tâche de traitement des données sur 2 axes. En effet, en fonction des résultats obtenus, il sera peut-être nécessaire d'effectuer à nouveau une étude des méthodes ou algorithmes disponibles pour remplir cette fonction. Cette tâche doit aboutir à un programme sous environnement Arduino permettant d'effectuer le traitement des données sur les axes Pitch, Roll et Yaw.

- **Estimation de la charge de travail** : 2 semaines
- **Contrainte** : Nécessite que la fonction « Acquisition des données » soit réalisée
- **Livrables** : Code source du projet Arduino et démonstration de l'acquisition des données traitées sur les deux axes Pitch, Roll et Yaw.

8.4.5. REALISATION DE LA FONCTION « ASSERVISSEMENT EN POSITION »

Cette tâche a pour but de réaliser la fonction d'asservissement en position en implémentant un algorithme de filtre PID. Le développement de celle-ci sera effectué sur plateforme Arduino.

- **Estimation de la charge de travail** : 1 jour
- **Contraintes** : Nécessite que les fonctions « Acquisition des données » et « Traitement des données » soient réalisées. Il est également nécessaire que le prototype initial soit réalisé.
- **Livrable** : Code source de la fonction d'asservissement.

8.4.6. REALISATION DE LA FONCTION « CORRECTION DE L'ORIENTATION »

Cette tâche consiste à mettre en œuvre un algorithme permettant de corriger l'orientation de la caméra sur Yaw et Pitch en fonction des mouvements volontaires effectués par l'utilisateur.

- **Livrables** : Code source de la fonction et vidéo de démonstration sur prototype de tests.
- **Estimation de la charge de travail** : 3 jours
- **Contrainte** : Il est nécessaire que l'ensemble des fonctions du système soient réalisées pour cette tâche. Cette tâche n'est pas prioritaire.

8.4.7. REALISATION DE LA FONCTION « COMMANDE DES MOTEURS »

La tâche de réalisation de la fonction de commande des moteurs permet de mettre en œuvre la méthode de commande des moteurs brushless sur environnement Arduino.

- **Livrables** : Vidéo de démonstration et code source de la fonction.
- **Estimation de la charge de travail** : 1 jour
- **Contrainte** : Il est nécessaire que le banc d'essais moteur soit réalisé pour valider cette fonction.

8.4.8. INTEGRATION DES FONCTIONS POUR UN PROTOTYPE INITIAL SUR 2 AXES

Cette tâche d'intégration consiste à assembler l'ensemble des fonctions réalisées sur l'environnement Arduino pour permettre le fonctionnement d'un prototype initial sur deux axes : Pitch et Roll. Pour le prototype initial, les moteurs brushless sont remplacés par des servomoteurs afin de simplifier les tests.

- **Livrable** : Code source du projet Arduino et démonstration à la MOA.
- **Estimation de la charge de travail** : 1 jour

- **Contraintes** : Nécessite que les tâches permettant la réalisation des fonctions « Acquisition des données », « Traitement des données sur 2 axes » et « Asservissement en position » soient effectuées. Pour effectuer la validation de cette fonction, il est nécessaire que la tâche « Réalisation du prototype initial » soit terminée.

8.4.9. INTEGRATION DES FONCTIONS POUR UN PROTOTYPE INITIAL SUR 3 AXES

Cette tâche reprend le travail effectué lors de l'intégration des fonctions pour le prototype initial sur 2 axes en remplaçant la fonction « Traitement des données sur 2 axes » par celle permettant des traiter les données sur trois axes afin d'obtenir une stabilisation sur les axes : Pitch, Roll et Yaw. Pour le prototype initial, les moteurs brushless sont remplacés par des servomoteurs afin de simplifier les tests.

- **Livrables** : Code source du projet Arduino et vidéo de démonstration
- **Estimation de la charge de travail** : 1 jour
- **Contraintes** : Nécessite que les tâches « Intégration des fonctions pour un prototype initial sur 2 axes » et « Traitement des données sur 3 axes » soit terminées. Afin de valider cette tâche, il est nécessaire que celle de « Réalisation du prototype initial » soit faite.

8.4.10. DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « ACQUISITION DES DONNEES »

Cette tâche consiste à s'affranchir des bibliothèques Arduino permettant la communication sur un bus i2C. Cette tâche devra donc produire l'ensemble des fonctions permettant la communication avec un équipement i2C esclave afin de récupérer les informations du capteur MPU-9250.

- **Livrable** : Code source de la fonction
- **Estimation de la charge de travail** : 2 jours
- **Contrainte** : Nécessite que la tâche « Réalisation de la fonction Acquisition des données » soit faite.

8.4.11. DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « CALIBRATION DU CAPTEUR »

La fonction de calibration réalisée lors de la tâche « Réalisation de la fonction de calibration » sous environnement Arduino est susceptible de faire appel à des bibliothèques Arduino. Cette tâche a uniquement pour but de supprimer la dépendance à ces bibliothèques en développant les fonctions nécessaires.

- **Livrable** : Code source de la fonction
- **Estimation de la charge de travail** : 1 jour
- **Contrainte** : Nécessite que la tâche « Réalisation de la fonction Acquisition des données » soit faite.

8.4.12. DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « TRAITEMENT DES DONNEES »

La fonction « Traitement des données » étant uniquement algorithmique, le développement de cette fonction en langage C natif consistera uniquement à s'assurer de l'optimisation du code qui aura été précédemment réalisée sur plateforme Arduino.

- **Livrable** : Code source de la fonction
- **Estimation de la charge de travail** : 1 jour
- **Contrainte** : Nécessite que la tâche « Réalisation de la fonction Traitement des données » soit faite.

8.4.13. DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « ASSERVISSEMENT »

La fonction « Asservissement » est également basée sur un algorithme et ne fait pas appel à de fonctions matérielles. Cette tâche consiste donc uniquement à s'assurer de l'efficacité du code produit lors du développement sur environnement Arduino et d'obtenir un code parfaitement natif.

- **Livrable** : Code source de la fonction
- **Estimation de la charge de travail** : 1 jour
- **Contrainte** : Nécessite que la tâche « Réalisation de la fonction Asservissement en position » soit faite.

8.4.14. DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « COMMANDE DES MOTEURS »

Cette tâche nécessite de s'affranchir de nombreuses fonctions proposées par l'IDE Arduino. En effet, la commande des moteurs nécessite une interaction avec le matériel. Cette tâche devra donc mener à un code totalement indépendant de l'environnement Arduino en effectuant la configuration du matériel et la réalisation de la commande en vitesse des moteurs.

- **Livrable** : Code source de la fonction
- **Estimation de la charge de travail** : 2 jours
- **Contrainte** : Nécessite que la tâche « Réalisation de la fonction commande des moteurs » soit faite.

8.4.15. DEVELOPPEMENT DU CODE NATIF POUR LA FONCTION « CORRECTION DE L'ORIENTATION »

La fonction de correction de l'orientation est essentiellement algorithmique. Cette tâche a pour but de s'assurer de la portabilité du code développé durant la tâche « Réalisation de la fonction de correction de l'orientation » et de l'optimiser.

- **Livrable** : Code source de la fonction
- **Estimation de la charge de travail** : 1 jour
- **Contrainte** : Nécessite que l'ensemble des tâches de développement du code natif des fonctions soit effectué. Cette tâche n'est pas prioritaire.

8.4.16. INTEGRATION DU CODE NATIF POUR LE PROTOTYPE FINAL

Cette phase d'intégration a pour but de rassembler l'ensemble des développements effectués en C natif pour obtenir un prototype final fonctionnel.

- **Livrables** : Code source du projet et vidéo de démonstration
- **Estimation de la charge de travail** : 2 jours
- **Contraintes** : Nécessite que les tâches « Développement du code natif pour la fonction Acquisition des données », « Développement du code natif pour la fonction Traitement des données », « Développement du code natif pour la fonction correction de l'orientation », « Développement du code natif pour la fonction Asservissement » et « Développement du code natif pour la fonction Commande des moteurs » soient effectuées. Afin d'effectuer la validation de cette fonction, il est nécessaire que la tâche « Réalisation du prototype final » soit effectuée.

8.5. CONCEPTION ELECTRONIQUE

La conception électronique a pour objectif la réalisation d'une carte électronique sur PCB. Cette tâche comporte une partie d'étude qui a pour objectif la réalisation des schémas de principe et de CAO.

- **Livrables** : Schémas de principe. Archive contenant les fichiers CAO. La carte électronique.
- **Estimation de la charge de travail** : 1 semaine
- **Contraintes** : Nécessite une prise en main du logiciel de CAO.

8.6. CONCEPTION MECANIQUE

La conception mécanique a pour but de modéliser la structure mécanique du prototype final. Cette conception sera effectuée sur le logiciel SolidWorks.

- **Livrables** : Archive contenant les fichiers 3D de chaque pièce et l'assemblage final. Un dossier contenant l'ensemble des mises en plan nécessaires à la réalisation des pièces.
- **Estimation de la charge de travail** : 1 semaine

8.7. REALISATION

Cette tâche se divise en plusieurs parties qui ont pour objectif la réalisation du prototype initial et final.

8.7.1. REALISATION DU MONTAGE ELECTRONIQUE SUR PROTOBOARD

Cette tâche a pour objectif de valider le schéma de principe réalisé lors de la conception électronique. Le montage électronique sur ProtoBoard se limitera aux fonctions : Alimentation, traitement et commande d'un seul moteur.

- **Livrables** : Vidéo de démonstration mettant en œuvre la fonction « Commande des moteurs »
- **Estimation de la charge de travail** : 1 jour
- **Contraintes** : Nécessite que les tâches « Conception électronique » et « Développement du code natif pour la fonction Commande des moteurs » soient terminées.

8.7.2. REALISATION D'UN BANC D'ESSAIS MOTEUR

Le banc d'essai moteur a pour objectif de valider la fonction « Commande des moteurs ». Cette tâche consiste à réaliser une maquette permettant d'effectuer des tests sur un moteur Brushless.

- **Livrable** : La maquette du banc d'essai
- **Estimation de la charge de travail** : 1 jour

8.7.3. REALISATION D'UN PROTOTYPE INITIAL

Le prototype initial consiste à réaliser un premier prototype permettant de valider l'ensemble des fonctions du système. Les moteurs Brushless sont ici remplacés par des servomoteurs. Le microcontrôleur utilisé pour le développement est un Arduino Uno.

- **Livrable** : Le prototype initial
- **Estimation de la charge de travail** : 1 jour

8.7.4. REALISATION MECANIQUE DU PROTOTYPE FINAL

La réalisation mécanique inclut la réalisation des pièces et l'assemblage des celles-ci.

- **Livrable** : Le prototype final
- **Estimation de la charge de travail** : 1 jour

- **Contraintes** : Nécessite que la tâche « Conception mécanique » soit terminée.

8.7.5. INTEGRATION DE L'ELECTRONIQUE AU PROTOTYPE FINAL

Cette tâche consiste à souder les composants sur la carte électronique, fixer la carte au prototype final et à réaliser l'ensemble des connexions électriques.

- **Livrables** : La carte électronique complète et fonctionnelle sur le système.
- **Estimation de la charge de travail** : 2 jours
- **Contraintes** : La carte électronique doit être envoyée en production 15 jours avant cette tâche.

8.8. TESTS

Des phases de tests interviennent tout au long du développement dans le but de valider le fonctionnement attendu par des tests unitaires. Des phases de validations sont planifiées pour valider chaque fonction. De plus, pour le développement en code C natif, la phase de test se réalisera en continu.

- **Acquisition des données** : Les tests pour cette fonction consistent à s'assurer de la communication avec le capteur et de la cohérence des données acquises.
- **Calibration du capteur** : Cette fonction permet d'obtenir un capteur calibré qui doit être donc exempt d'erreur au maximum. Le test consistera à vérifier l'exactitude des données acquises après l'application des données issues de la calibration.
- **Traitement des données** : La fonction de traitement permet d'obtenir les angles Yaw, Pitch et Roll. Le test doit permettre de valider l'exactitude des données fournies par cette fonction.
- **Asservissement en position** : Ce test s'effectue sur une maquette et doit vérifier le bon fonctionnement de la fonction, c'est à dire garantir la stabilisation de la caméra.
- **Commande des moteurs** : Les tests sont à effectuer sur un banc d'essai moteur. On doit vérifier la commande tout ou rien et la variation de la vitesse.
- **Correction de l'orientation** : La fonction de correction de l'orientation doit permettre à l'utilisateur de modifier l'orientation de la caméra en agissant sur l'orientation du bâti. Cette procédure de test doit vérifier le bon fonctionnement qui se traduit par la correction de l'orientation tout en conservant une stabilisation du dispositif.

Une phase de test en condition réelle sera effectuée en fin de projet pour valider le bon fonctionnement du prototype final.

- **Estimation de la charge de travail** : 1 jour par fonction à valider, 13 jours pour le code natif
- **Livrable** : Procédures de test mises en place

8.9. DOCUMENTATION

La partie documentation du projet comprend la rédaction de deux types de documents :

- Document permettant la reprise du projet
- Rapport de fin de projet synthétisant l'ensemble du projet dans sa gestion et sa réalisation

En fin de projet, une soutenance sera organisée pour présenter le travail réalisé devant un jury.

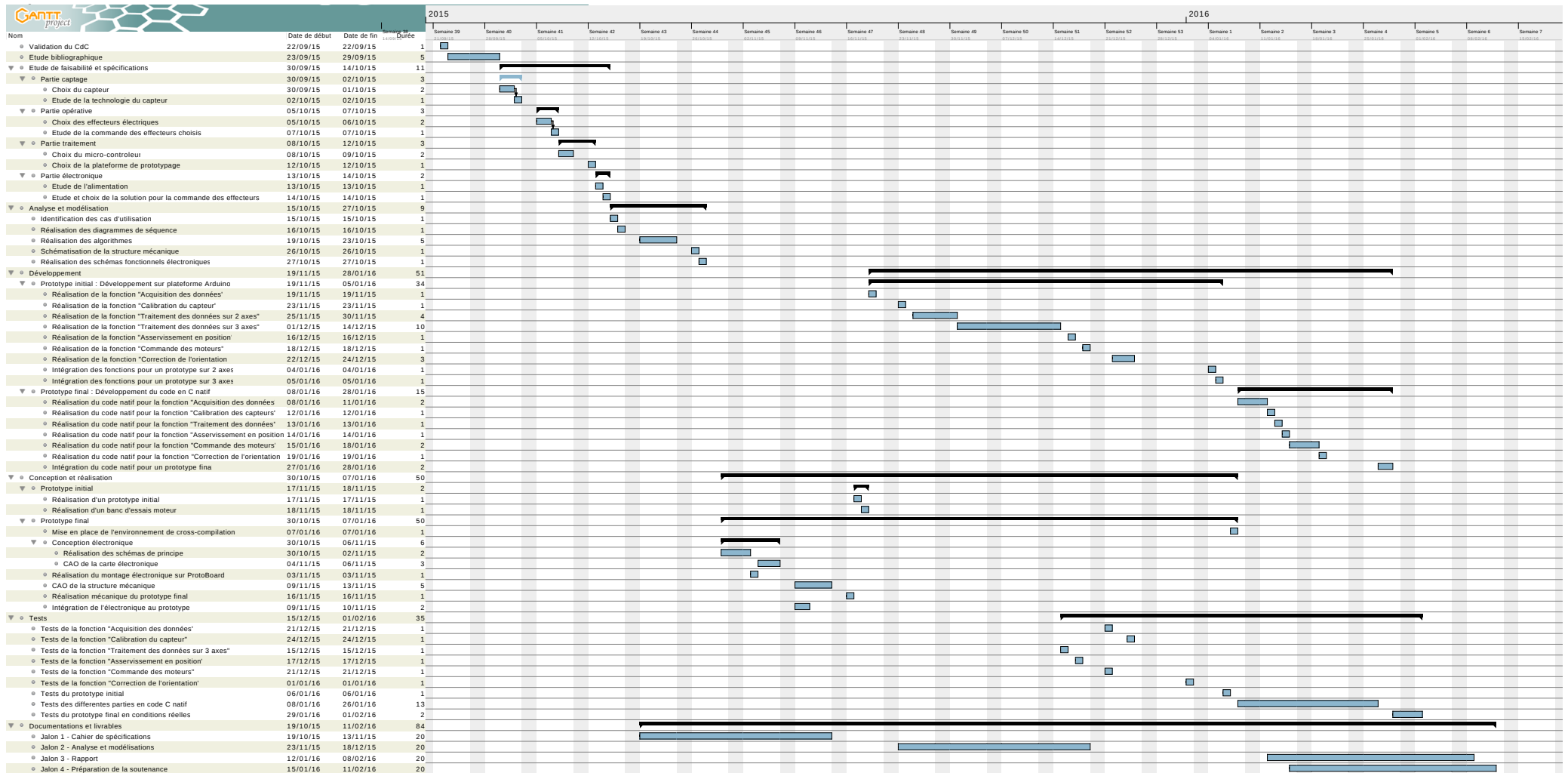
- **Estimation de la charge de travail** : 4 semaines
- **Contraintes temporelles** :
 - **Rendu du document « Reprise du projet »** : 08/02/2016

- **Rendu du document « Rapport » : 08/02/2016**
- **Soutenance de projet : 11-12-13/02/2016**

9. LISTE DES TACHES

Nom	Date de début	Date de fin	Durée
Validation du CdC	22/09/15	22/09/15	1
Etude bibliographique	23/09/15	29/09/15	5
Etude de faisabilité et spécifications	30/09/15	14/10/15	11
Partie captage	30/09/15	02/10/15	3
Choix du capteur	30/09/15	01/10/15	2
Etude de la technologie du capteur	02/10/15	02/10/15	1
Partie opérative	05/10/15	07/10/15	3
Choix des effecteurs électriques	05/10/15	06/10/15	2
Etude de la commande des effecteurs choisis	07/10/15	07/10/15	1
Partie traitement	08/10/15	12/10/15	3
Choix du micro-contrôleur	08/10/15	09/10/15	2
Choix de la plateforme de prototypage	12/10/15	12/10/15	1
Partie électronique	13/10/15	14/10/15	2
Etude de l'alimentation	13/10/15	13/10/15	1
Etude et choix de la solution pour la commande des effecteurs	14/10/15	14/10/15	1
Analyse et modélisation	15/10/15	27/10/15	9
Identification des cas d'utilisation	15/10/15	15/10/15	1
Réalisation des diagrammes de séquence	16/10/15	16/10/15	1
Réalisation des algorithmes	19/10/15	23/10/15	5
Schématisation de la structure mécanique	26/10/15	26/10/15	1
Réalisation des schémas fonctionnels électroniques	27/10/15	27/10/15	1
Développement	19/11/15	28/01/16	51
Prototype initial : Développement sur plateforme Arduino	19/11/15	05/01/16	34
Réalisation de la fonction "Acquisition des données"	19/11/15	19/11/15	1
Réalisation de la fonction "Calibration du capteur"	23/11/15	23/11/15	1
Réalisation de la fonction "Traitement des données sur 2 axes"	25/11/15	30/11/15	4
Réalisation de la fonction "Traitement des données sur 3 axes"	01/12/15	14/12/15	10
Réalisation de la fonction "Asservissement en position"	16/12/15	16/12/15	1
Réalisation de la fonction "Commande des moteurs"	18/12/15	18/12/15	1
Réalisation de la fonction "Correction de l'orientation"	22/12/15	24/12/15	3
Intégration des fonctions pour un prototype sur 2 axes	04/01/16	04/01/16	1
Intégration des fonctions pour un prototype sur 3 axes	05/01/16	05/01/16	1
Prototype final : Développement du code en C natif	08/01/16	28/01/16	15
Réalisation du code natif pour la fonction "Acquisition des données"	08/01/16	11/01/16	2
Réalisation du code natif pour la fonction "Calibration des capteurs"	12/01/16	12/01/16	1
Réalisation du code natif pour la fonction "Traitement des données"	13/01/16	13/01/16	1
Réalisation du code natif pour la fonction "Asservissement en position"	14/01/16	14/01/16	1
Réalisation du code natif pour la fonction "Commande des moteurs"	15/01/16	18/01/16	2
Réalisation du code natif pour la fonction "Correction de l'orientation"	19/01/16	19/01/16	1
Intégration du code natif pour un prototype final	27/01/16	28/01/16	2
Conception et réalisation	30/10/15	07/01/16	50
Prototype initial	17/11/15	18/11/15	2
Réalisation d'un prototype initial	17/11/15	17/11/15	1
Réalisation d'un banc d'essais moteur	18/11/15	18/11/15	1
Prototype final	30/10/15	07/01/16	50
Mise en place de l'environnement de cross-compilation	07/01/16	07/01/16	1
Conception électronique	30/10/15	06/11/15	6
Réalisation des schémas de principe	30/10/15	02/11/15	2
CAO de la carte électronique	04/11/15	06/11/15	3
Réalisation du montage électronique sur ProtoBoard	03/11/15	03/11/15	1
CAO de la structure mécanique	09/11/15	13/11/15	5
Réalisation mécanique du prototype final	16/11/15	16/11/15	1
Intégration de l'électronique au prototype	09/11/15	10/11/15	2
Tests	15/12/15	01/02/16	35
Tests de la fonction "Acquisition des données"	21/12/15	21/12/15	1
Tests de la fonction "Calibration du capteur"	24/12/15	24/12/15	1
Tests de la fonction "Traitement des données sur 3 axes"	15/12/15	15/12/15	1
Tests de la fonction "Asservissement en position"	17/12/15	17/12/15	1
Tests de la fonction "Commande des moteurs"	21/12/15	21/12/15	1
Tests de la fonction "Correction de l'orientation"	01/01/16	01/01/16	1
Tests du prototype initial	06/01/16	06/01/16	1
Tests des différentes parties en code C natif	08/01/16	26/01/16	13
Tests du prototype final en conditions réelles	29/01/16	01/02/16	2
Documentations et livrables	19/10/15	11/02/16	84
Jalon 1 - Cahier de spécifications	19/10/15	13/11/15	20
Jalon 2 - Analyse et modélisations	23/11/15	18/12/15	20
Jalon 3 - Rapport	12/01/16	08/02/16	20
Jalon 4 - Préparation de la soutenance	15/01/16	11/02/16	20

10. PLANNING



BIBLIOGRAPHIE

Informations sur le MPU-9250 : <http://store.invensense.com/ProductDetail/MPU9250-InvenSense-Inc/487537/>

TABLE DES ILLUSTRATIONS

Figure 1 : Cas d'utilisation	6
Figure 2 : Schéma du système	6
Figure 3 : Correction sur l'axe X par l'utilisateur	9
Figure 4 : Correction sur l'axe Z par l'utilisateur.....	9
Figure 5 : Architecture du système	10
Figure 6 : Entrées / sorties de la fonction « Acquisition des données ».....	11
Figure 7 : Entrées / sorties de la fonction « Traitement des données »	12
Figure 8 : Entrées / sorties de la fonction « Correction de l'orientation »	12
Figure 9 : Entrées / sorties de la fonction « Asservissement en position ».....	13
Figure 10 : Entrées / sorties de la fonction « Commande de la motorisation »	13

GLOSSAIRE

Capteur IMU 9DOF	Centrale inertielle sur 9 axes. Est composée d'un accéléromètre, d'un gyroscope et d'un magnétomètre.
Accéléromètre	Capteur permettant de mesurer l'accélération linéaire sur les axes X, Y et Z. L'unité de mesure est le m/s^2 ou peut également être exprimé en « g ».
Gyroscope	Capteur permettant de mesurer la vitesse angulaire. L'unité de mesure est le degré / seconde ou encore le radian / seconde.
Magnétomètre	Également appelé boussole, ce capteur permet de mesurer le champ magnétique de l'environnement. L'unité de mesure est le Tesla ou le Gauss.
Yaw, Pitch, Roll	Permet de décrire l'orientation d'un solide dans l'espace. La notion « Yaw » représente l'orientation sur l'axe Z, « Pitch » représente l'orientation sur X et « Roll » celle sur Y. Cette orientation s'appuie sur les angles d'Euler.
Moteur brushless	Le moteur brushless évoqué dans ce document est un type de moteur particulier utilisé dans le domaine des drones pour la stabilisation d'un appareil photo. Ce moteur brushless a la particularité de pouvoir être commandé comme un servomoteur et bénéficie d'une grande précision en positionnement.
Bus I2C	Un bus i2C est un bus de données utilisé en électronique. C'est un bus série synchrone bidirectionnel half-duplex.
Signal PWM	Un signal PWM ou MLI pour Modulation à Largeur d'Impulsions est utilisé pour générer des signaux continus à partir de circuits à fonctionnement tout ou rien.
PCB	Circuit imprimé en électronique.