

# Stabilisation du drone quadrirotor



**Tuteur :** Yves Brière

CLAVEYROLLAS Antoine  
CORBISIER Jean  
FITTE Sébastien  
RIGAUD Florian

Le 09/06/2008



## Sommaire

Présentation du projet.....	4
Introduction .....	4
Présentation du principe de vol d'un quadrirotor .....	4
Présentation rapide du drone lors de la prise en main.....	6
I. Mise en œuvre du décollage.....	10
1° Optimisation de la puissance délivrée aux moteurs.....	10
2° Optimisation de la taille des hélices .....	12
3° Optimisation de la masse du quadrirotor .....	15
II. Optimisation du code implémentée dans la carte MicroPilot.....	17
1° Synthèse sur l'asservissement initial du quadrirotor .....	17
2° Retouche nécessaire au code original .....	22
III. Création d'une nouvelle structure optimisée.....	28
1° Pourquoi une nouvelle structure ? .....	28
2° Conception de la nouvelle structure .....	28
3° Avantages et inconvénients de cette nouvelle structure .....	34
IV. Amélioration de l'asservissement .....	36
1° Modèle de notre asservissement .....	36
2° Réglage des coefficients de notre correcteur .....	38
3° Une autre solution : nouvelle modélisation de notre système .....	39
V. Rétrospective du projet.....	41
1° Atteinte des objectifs fixés dans l'avant-projet .....	41
2° Proposition de solutions pour améliorer le quadrirotor .....	43
VI. ANNEXES .....	45
Annexe 1 : photos de l'ancienne et de la nouvelle structure .....	45
Annexe 2 : les bancs d'essais .....	47
Annexe 3 : Matériel électronique et électrotechnique utilisé .....	50
Annexe 4 : Matériel informatique utilisé .....	55
Annexe 5 : Modèle Simulink de l'asservissement .....	60
Annexe 6 : Code implémenté dans la carte MicroPilot.....	68
Annexe 7 Procédure de démarrage du quadrirotor.....	78
Annexe 8 Recommandations de sécurité.....	80
VII. Bibliographie .....	81

# Présentation du projet

## Introduction

Traditionnellement, quelques élèves ISAE cursus ENSICA réalisent pour leur Projet d'Initiative Personnelle des micros drones, comme par exemple un drone basé sur le Twinstar ou le quadrirotor.

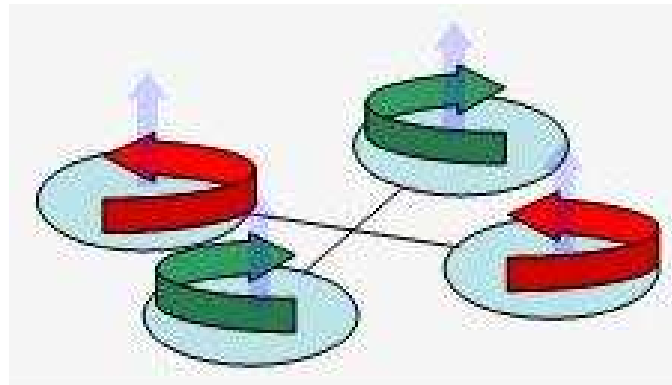
En 2005-2006, Alexis FRENOT, Anthony GOSMANN et Romaric GUILLERM ont eu l'idée de développer un drone quadrirotor. Ils ont donc fabriqué une structure cubique d'un drone à quatre rotors dans laquelle ils ont intégré des moteurs et les équipements nécessaires à leur fonctionnement. Pour stabiliser ce drone, ils ont choisi d'acheter une carte MicroPilot dotée de nombreuses options. En effet, grâce à cette carte implémentée avec un code d'asservissement adéquat, il semble possible de permettre au drone de réaliser des vols autonomes. A la fin de leur PIP, le travail de prise en main du logiciel et la stabilisation du drone étaient loin d'être terminés.

En 2006-2007, Adeline SERRECOURT et Julien CHAPUIS ont décidé de reprendre le projet. Compte tenu de la difficulté de mise en route du drone, ils ont inventé un protocole de mise en marche du drone (6 interrupteurs à actionner dans le bon ordre). De plus, ils ont implémenté un code de stabilisation dans la carte, et ont déterminé de façon empirique les coefficients de leur correcteur PID sur leur banc d'essai, afin de stabiliser les axes de roulis et de tangage. Cependant, aucun essai en vol n'a été effectué, la puissance disponible étant insuffisante pour que le drone puisse se sustenter.

Nous avons choisi cette année de prendre le relais, afin de faire enfin décoller le drone. D'après l'étude de dimensionnement réalisée par le premier groupe d'élève, il ne devrait pas y avoir de problème de puissance pour le drone. En effet, les moteurs sont capables de sustenter 700 grammes chacun et le drone ne pèse que 1.8 kg, batteries comprises. Nous en avons déduit qu'il devait y avoir un problème dans la chaîne de puissance. Après avoir résolu ce problème, nous aurions dû pouvoir faire des essais en vol et valider le modèle de nos prédécesseurs. Cependant, on a pu se rendre compte que le drone n'était pas stabilisé en vol. Il a donc fallu reprendre les essais depuis le début. De même, nous avons décidé de modifier le code de stabilisation car nous avons remarqué des comportements anormaux du drone. De plus, nous avons décidé de fabriquer une nouvelle structure plus légère afin d'avoir plus de marge de puissance. Notre objectif est de stabiliser ce drone quadrirotor grâce à un code générique fonctionnant sur nos deux structures.

## Présentation du principe de vol d'un quadrirotor

Le fonctionnement d'un quadrirotor est assez particulier. En effet, comme son nom l'indique, c'est un drone muni de quatre rotors. Cela implique que pour être stable en lacet, il faut que deux moteurs opposés tournent dans un sens et les deux autres dans l'autre sens, car une hélice en mouvement crée un couple dans le sens inverse de sa rotation.



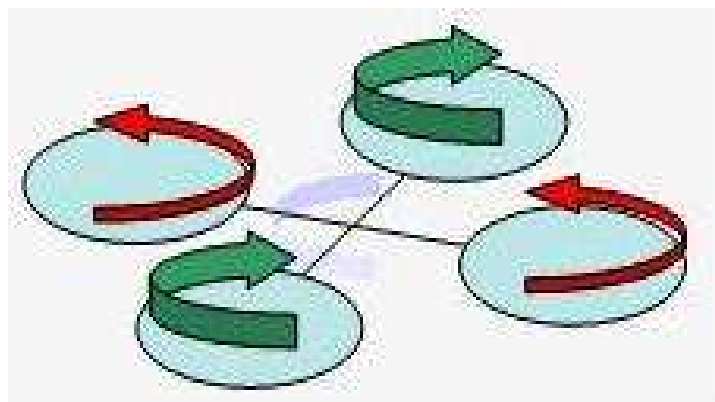
On distingue sur ce type de drone quatre mouvements possibles :

Les gaz

Cette phase représente le mouvement de montée/descente du drone. Les 4 moteurs tournent à la même vitesse durant cette phase.

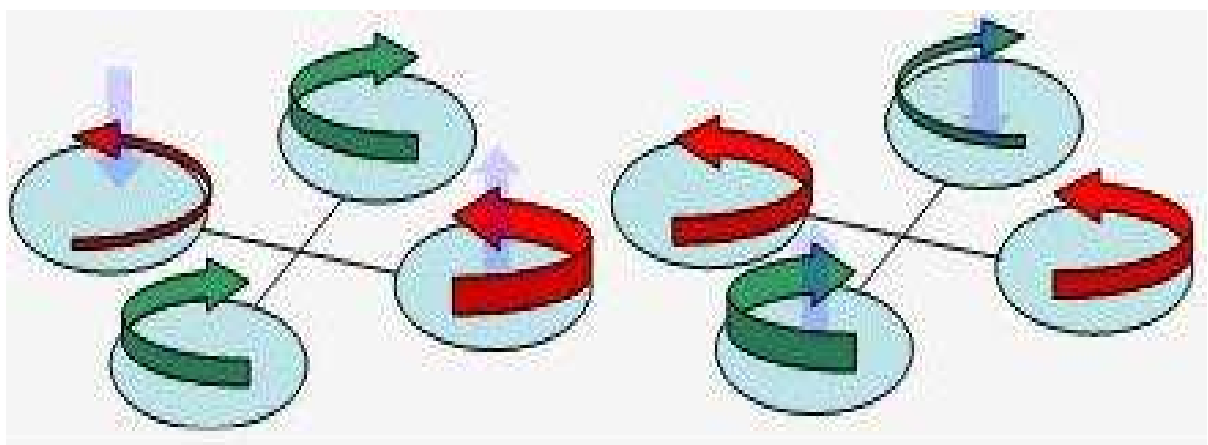
Le lacet

Il permet de faire tourner le quadrirotor sur lui même. Ce mouvement est obtenu en accélérant 2 moteurs opposés et en ralentissant les 2 autres.



Le roulis et le tangage

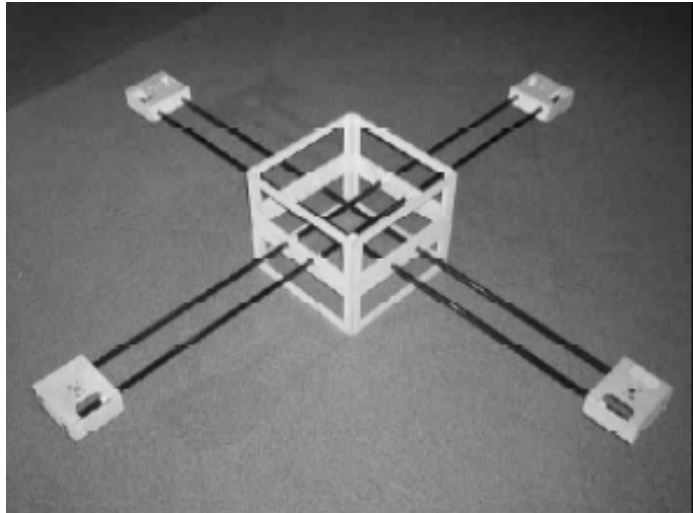
Ce sont des mouvements assez similaires. Il s'agit de pencher le drone sur un axe. Pour cela il faut accélérer un moteur et ralentir le moteur opposé de la même valeur.



Le quadrirotor est un drone instable. Il faut donc inclure à la structure une partie électronique permettant de rectifier les mouvements de lacet, roulis et tangage. Trois gyroscopes sont donc utilisés pour détecter les variations angulaires et une carte doit être implémentée pour transmettre aux moteurs les corrections adéquates, afin de stabiliser le drone.

### Présentation rapide du drone lors de la prise en main

La structure :



Cette structure est constituée d'une partie centrale permettant d'abriter la carte électronique ainsi que les batteries, et de tiges de carbones liant la partie centrale aux supports moteurs.

Cette conception a l'avantage d'être très rigide. En effet, la rigidité en flexion offerte par les barres en carbone est très bonne et la torsion est quasiment inexistante grâce au collage. Cependant cette structure est plutôt encombrante et lourde : 400g.

### Radiocommande

La radiocommande utilisée est une radio Futaba (référence : 9CAP/9CHP) de neuf voies. Comme le drone quadrirotor n'a que quatre types de mouvements possibles, quatre voies sont utilisées pour diriger le drone.

Le récepteur :

Nous avons besoin d'un récepteur pour transmettre les ordres de la radiocommande à la carte Micropilot.



## Les moteurs et les hélices :

Le principe de vol d'un drone quadrirotor impose d'avoir des hélices de pas opposés. Les pales ont été associées a des moteurs de type brushless (référence : brushless typhon-micro 6-23), qui ont l'avantage d'être petits, légers et puissants. L'un des inconvénients de ces moteurs est leur mise en œuvre : ils doivent obligatoirement être associés a des régulateurs particuliers, appelés contrôleurs, qui nécessitent une programmation et une temporisation bien précise. Ce modèle de moteur, de puissance 90 W, peut être alimenté sous tension variant de 7,2 V a 14,4 V. Il pèse 42g et, à la base, convient a un engin de 250 a 700 grammes.



## Les contrôleurs

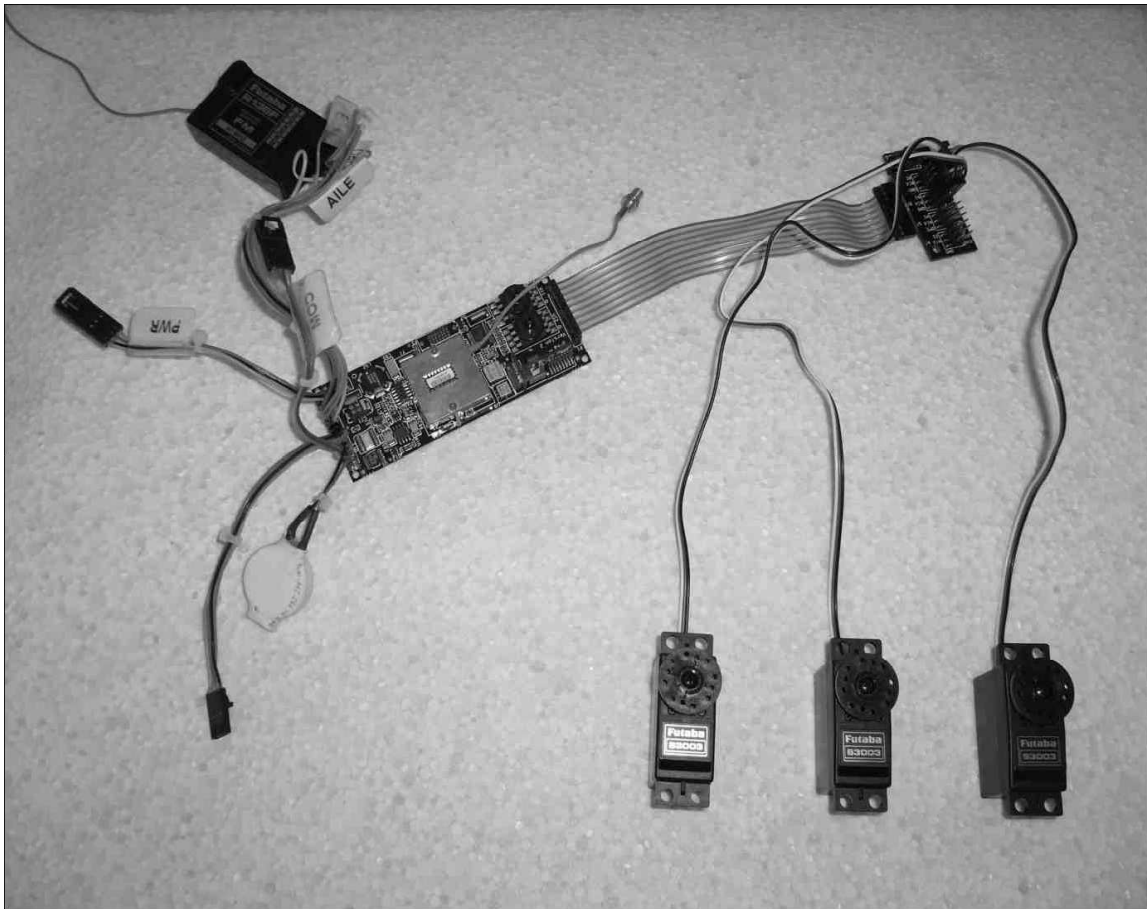
Le variateur est l'interface entre la commande de signal et le circuit de puissance du moteur. Il est relié à un accumulateur (ici de type Li-Po) qui fournira la puissance nécessaire au moteur. Les variateurs sont des éléments qui chauffent beaucoup. Par conséquent nous avons décidé de les placer sous le flux des hélices.





## La carte MicroPilot

Il a été décidé que le drone quadrirotor sera stabilisé à l'aide d'une carte Micropilot MP 2028g. Il s'agit d'une carte conçue pour le contrôle de modèles réduits d'avions. Elle est composée d'un microprocesseur programmable, de systèmes de mesure (gyroscopes, prises de pression, GPS), et d'entrées sorties utiles à la commande du drone (prises pour récepteur de télécommande, carte fille de sorties vers des servocommandes). Cette carte peut évoluer par l'adjonction de cartes filles qui lui donnent des capacités de gestion d'un flux vidéo, ou de communication vers un PC via un radio modem. On peut aussi lui ajouter d'autres cartes de sorties pour d'autres servocommandes.



On peut communiquer avec la carte via le port COM relié par un câble série au PC. On peut ainsi récupérer différentes sorties de la carte ou implémenter celle-ci. En effet, il est possible de créer un nouveau code pour la carte plus adapté à un drone quadrirotor.

Cependant cette solution n'est pas aussi pratique que la programmation d'un microcontrôleur. En effet, nous ne pouvons accéder qu'à une partie du code, et Micropilot conservera toujours un noyau non modifiable qui pourra perturber le bon fonctionnement du drone.

Les accumulateurs :

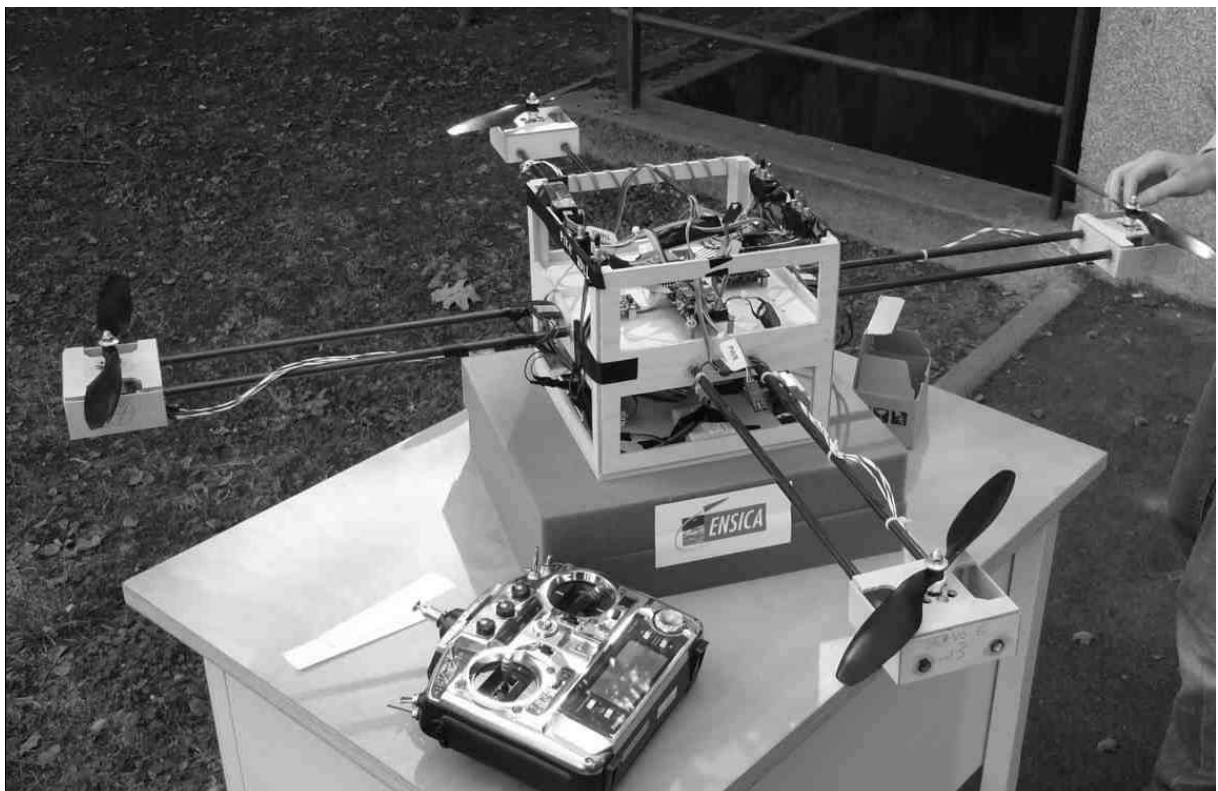




L'accumulateur 4.8V sert à alimenter la carte Micropilot. Les 2 accumulateurs Li-Po servent à alimenter les moteurs.

Cependant, dans la notice de la carte Micropilot, il est spécifié que l'on peut alimenter la carte par du 7.4V. On peut donc se passer de l'accumulateur 4.8V.

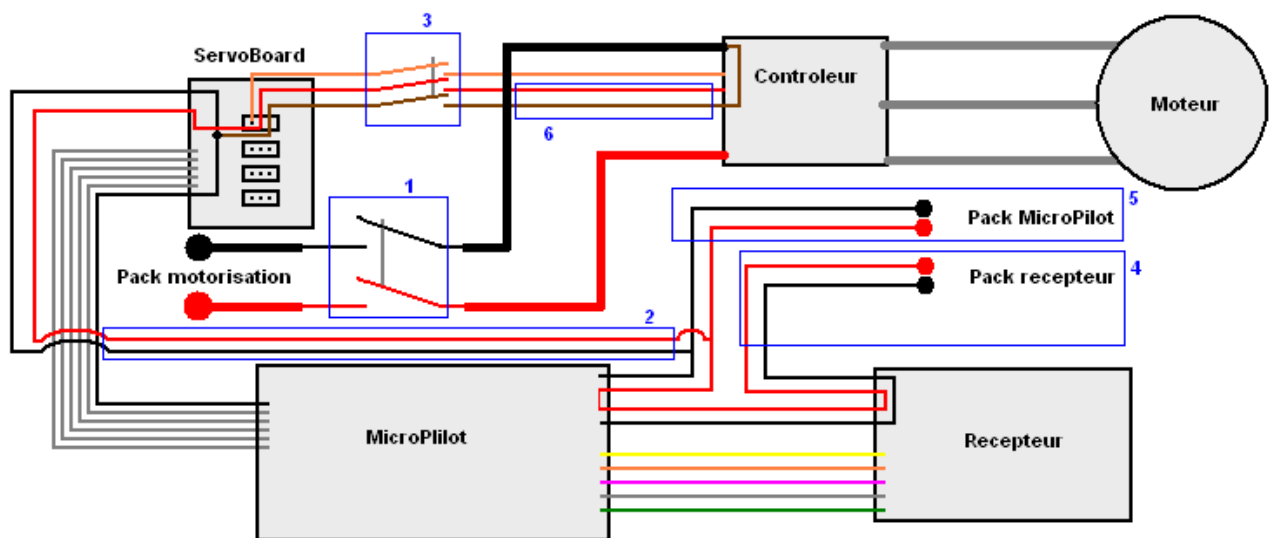
La structure pèse au final, batteries comprises, 1.6kg. Malgré les études de dimensionnement montrant que le drone pouvait voler à ce poids, le drone ne décolle pas. Il manque de puissance pour se sustenter.



# I. Mise en œuvre du décollage

## 1° Optimisation de la puissance délivrée aux moteurs

Le groupe précédent avait des problèmes lors de l'allumage des différents éléments. Il fallait suivre un ordre précis et cette étape nécessitait l'utilisation de six interrupteurs, or parmi ces interrupteurs deux coupaient l'alimentation des quatre moteurs. De plus les variateurs étaient reliés à la partie centrale par cinq câbles. Ce qui donne le schéma suivant.



Comme on peut le remarquer sur ce schéma de nombreux câbles sont redondants et les batteries sont reliées au même potentiel. Nous avons par conséquent décidé d'assainir ce montage par soucis d'efficacité et de sécurité. Les modifications que nous y avons apportées sont encadrées de bleu, nous allons les détailler dans l'ordre de leur application.

Dans un premier temps nous avons remarqué que les interrupteurs placés sur la ligne de puissance étaient câblés avec des fils de section bien moindre que le reste de la partie de puissance. Après différents tests il s'est avéré que ces diminutions de section autour de l'interrupteur entraînaient des chutes de tensions de plusieurs volts au niveau des variateurs. Nous avons par conséquent retiré ces interrupteurs puis recâblé la partie puissance avec des fils de section appropriée.

Nous avons ensuite désiré conformément à ce qui se fait de plus en plus dans l'industrie, rapprocher le plus possible les contrôleurs des actionneurs (ici les moteurs). Ceci pour deux raisons principales. La première est que les trois câbles alimentant les moteurs depuis les variateurs sont alimentés par un courant alternatif de fréquence supérieure à la fréquence de rotation des moteurs (de l'ordre de 10 kHz). Ainsi ces longs câbles alimentés en basse fréquence risquaient de perturber la carte microPilot qui est selon le constructeur très

sensible aux perturbations magnétiques. La seconde raison était de limiter l'échauffement des variateurs en les plaçant dans le flux de l'hélice.

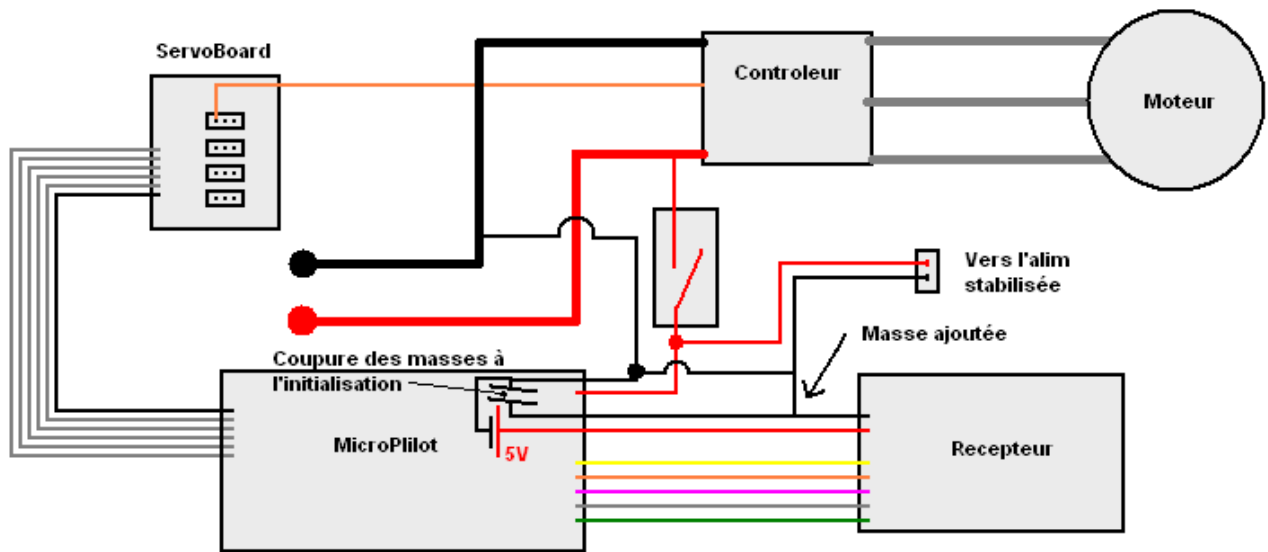
La notice technique des variateurs précisent que ces derniers n'ont pas besoin d'être alimenté par la partie commande au contraire un système à base de régulateur de tension fait qu'ils délivrent une tension stabilisée à 5V sur le fil positif de la ligne de commande. Par conséquent l'alimentation de la servoBoard est inutile ((2) sur le schéma), au contraire elle alimente le pack MicroPilot en 5V et par conséquent le recharge ! Nous avons donc retiré ces câbles superflus afin de gagner de la masse.

La présence des trois autres interrupteurs ((3) sur le schéma) était imposé par des problèmes de démarrage. En effet les variateurs s'initialisaient trop tôt au lancement de la carte. Cette dernière n'étant pas encore pleinement fonctionnelle, elle n'envoyait pas de signal de commande aux variateurs. Ils se mettaient alors en mode sécurisé c'est à dire qu'ils envoyaient des bips d'erreur et refusaient le démarrage. Cependant si ces variateurs s'initialisaient trop tôt c'est bien parce qu'il était alimenté trop tôt. Il aurait fallu qu'il ne démarre qu'au branchement des packs de propulsion. Or le fait d'alimenter la servoBoard les faisait s'initialiser. La modification précédente a donc suffi à supprimer les problèmes de démarrage et donc à rendre obsolètes les interrupteurs (3) simplifiant au passage la procédure d'allumage.

En regardant de plus près la notice de la carte microPilot nous nous sommes aperçu que cette dernière était aussi équipée d'un système de régulation de tension délivrant du 5V stabilisé au récepteur. Le pack de réception c'est ainsi avéré inutile.

Les modifications suivantes ont été appliquées sur la seconde structure, profitant du fait que toute l'électronique était démontée.

Il était prévu à la conception de la structure de ne plus utiliser que les packs de propulsion pour alimenter tout le drone. En effet la carte microPilot est dimensionnée pour accepter de 4V à 25V nous pouvions donc utiliser les 7,4V de ces packs. Le second point intéressant est que la carte microPilot ne consomme que 150mA, négligeable devant les 30A des moteurs. Le gain de masse dû à la perte du pack microPilot ((5) sur le schéma) aurait permis de voler avec un troisième pack de propulsion. Le pack microPilot a donc été retiré mais pas remplacé par un pack de propulsion car notre troisième pack de propulsion a grillé avant. On s'est alors heurté au problème de l'initialisation de la carte. En effet ne démarrer que la carte sans alimenter les contrôleurs est impossible avec une seule source sans utiliser d'interrupteur sur le circuit de puissance. Or nous ne voulions pas mettre d'interrupteurs sur le circuit de puissance car ils auraient été lourds et encombrants. Ainsi nous avons établi une séquence d'allumage basée sur l'utilisation d'une alimentation extérieure stabilisée. Dès lors, du fait de l'alimentation commune aux contrôleurs et à la carte MicroP il a été nécessaire de placer un interrupteur capable d'isoler les deux circuits le temps de l'initialisation de la carte. Cet interrupteur ne fait que couper la continuité variateur/carte MicroPilot il ne permet pas d'isoler l'alimentation extérieure de la carte microPilot. Ainsi on démarre la carte sur alimentation extérieure en prenant soin d'ouvrir l'interrupteur. Une fois la carte initialisée on branche les packs de propulsion et les variateurs s'initialisent. On peut alors fermer l'interrupteur. Les packs de propulsion prennent alors le relais pour l'alimentation de la carte et l'alimentation extérieure doit être retirée. Il est important de préciser que cet interrupteur ne coupe pas les masses, qui restent continues afin d'assurer une initialisation correcte des variateurs.



Enfin nous nous sommes aperçus que les masses de puissance et commande des variateurs étaient commune. Comme de plus nous n'utilisons pas les 5V stabilisés fournis par ces derniers les câbles plus et moins des variateurs ((6) sur le schéma) étaient inutiles et ont été retirés. Cependant suite à ça nous avons rencontré des problèmes de démarrage. En effet après de nombreux tests nous nous sommes aperçus que les masses de la carte microPilot n'était plus continues après son initialisation. Par conséquent la masse de la carte microPilot n'était plus commune à celle des packs de propulsion et donc du contrôleur. Cet effet était auparavant passé inaperçu car les masse de commande variateurs forçaient une continuité des masses. Il à donc suffit d'ajouter un fil de masse reliant la masse du récepteur à la masse des packs. De cette façon nous assurons de nouveau la continuité des masses et le démarrage s'effectue correctement avec beaucoup moins de câble.

## 2° Optimisation de la taille des hélices

Après l'optimisation du câblage, il fallait savoir s'il n'y avait pas de limites au niveau de la chaîne de propulsion et en particulier au niveau des hélices. En effet, les groupes précédents n'avaient fait leurs essais qu'avec un seul type d'hélices. Il était donc indispensable de tester différentes hélices. La contrainte du quadrirotor est qu'il faut deux types d'hélices, comme cela a été expliqué précédemment. Nous n'avons donc pas pu tester tous les types d'hélices que nous désirions, car peu de modèles sont fabriqués en contrarotatifs. Nous avons choisi de tester des hélices 9x6 ; 10x6 ; 10x7 et 11x6 en plus des hélices 10x4.5 que nous avions (petit rappel pour une hélice 11x6, cela signifie que l'hélice a un diamètre de 11 pouces et un pas de 6 cela signifie que l'hélice avance de 6 pouces pour un tour).

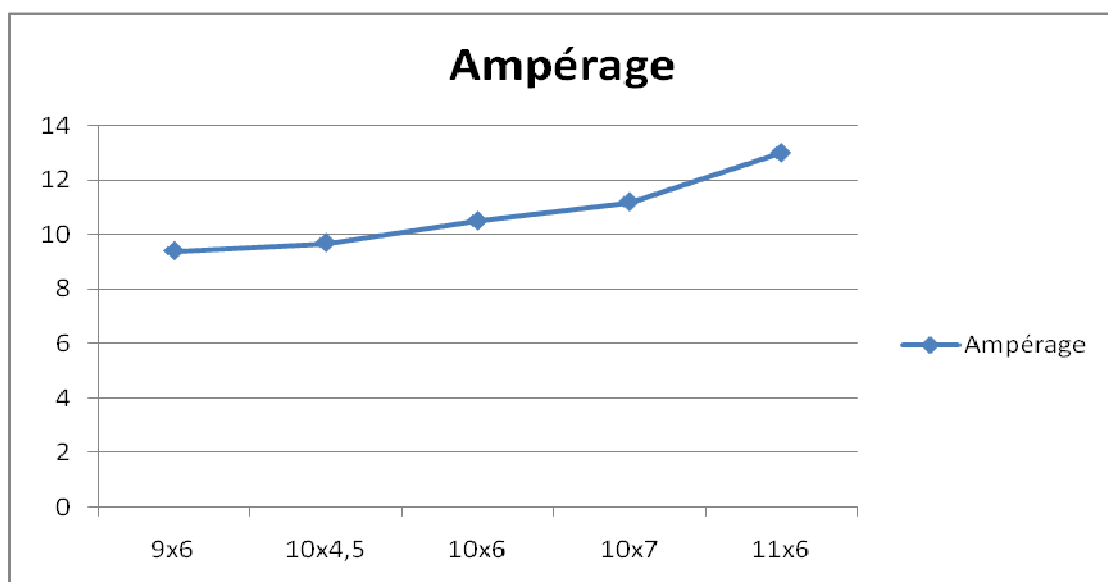
Le principe des tests était le suivant : il fallait déterminer le courant au bon fonctionnement du moteur en fonction de l'hélice et également la poussée fournie par moteur. Pour avoir le plus de précision possible, nous avons décidé d'effectuer ces deux tests sous alimentation stabilisée. Cela nous permettant de ne pas être limité par la puissance des batteries Li-Po. Nous avons effectué ces deux tests simultanément. Nous avons posé le drone

sur une balance, nous avons ensuite lesté le drone pour être sûr qu'il ne puisse pas décoller. Afin de mesurer le courant, nous avons ajouté un ampèremètre en série et pour mesurer la poussée, nous avons mesuré la différence de poids (entre le moment où le drone est au repos et le moment où il est plein gaz). Le fait de lester le drone nous affranchissait des problèmes de lacet car nous ne faisons fonctionner que deux moteurs. Grâce à ces mesures, nous pouvions ensuite déduire la puissance par moteur car la tension d'alimentation était constante et valait 7,4 Volts. Nous en avons déduit le « rendement » des hélices en faisant le ratio (ceci n'est pas un vrai rendement mais nous l'appellerons ainsi pour montrer l'évolution du rapport poussée/puissance). Les résultats sont présentés ci-dessous :

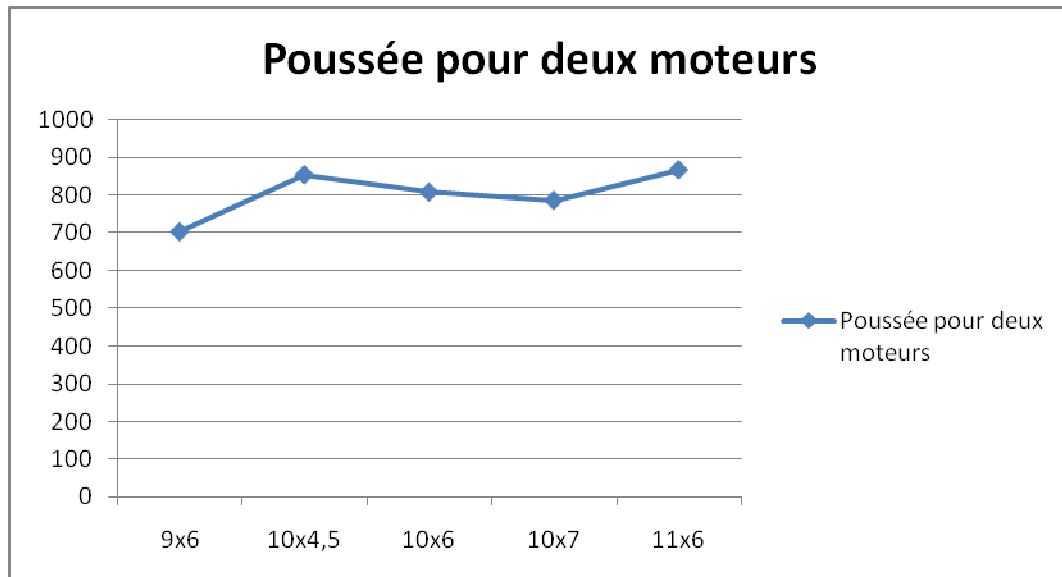
Type d'hélice	Ampérage	Poussée pour deux moteurs	puissance	rendement
9x6	9,4	702	69,56	5,04600345
10x4,5	9,7	853	71,78	5,941766509
10x6	10,5	807	77,7	5,193050193
10x7	11,2	785	82,88	4,735762548
11x6	13	866	96,2	4,501039501

**Tableau 1 Tableau récapitulatif des performances des hélices**

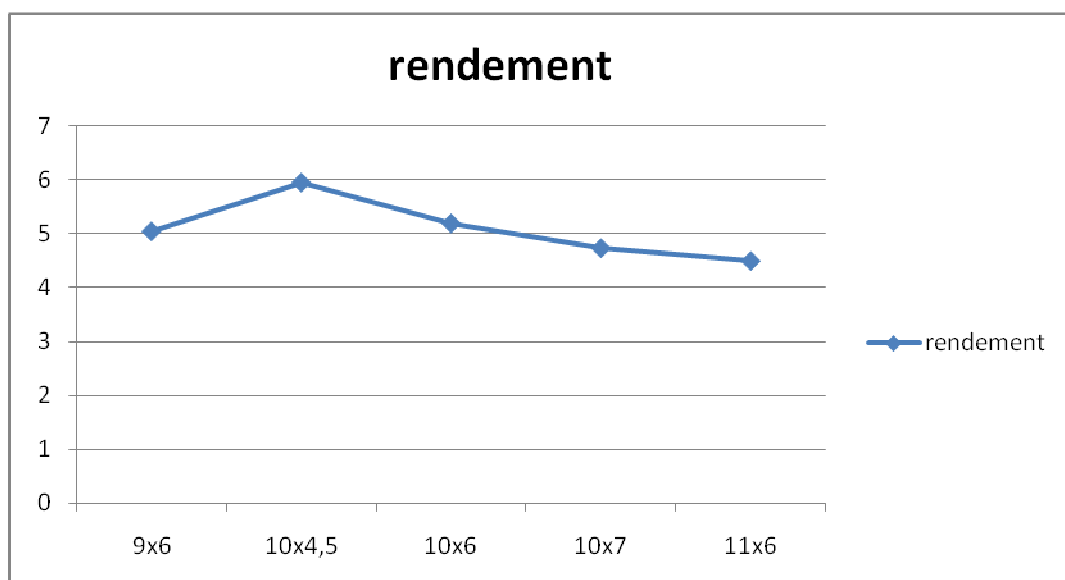
La poussée est donnée en grammes, la puissance en Watts. Il faut cependant poser une réserve quand à la valeur de l'ampérage de l'hélice 11x6. En effet, l'ampèremètre est limité à 10 Ampères, donc pour des valeurs inférieures à 12A on peut les estimer correctes mais au delà de cette valeur, nous ne sommes pas du tout sûrs de la véracité de la valeur : cette valeur peut en fait être bien supérieure mais cela nous suffit car de toutes façons avec les batteries que nous possédons-nous ne pouvons délivrer autant. Les performances sont décrites ci-dessous sous forme de graphiques :



**Tableau 2 Graphique représentant la consommation de courant en Ampères selon le type de l'hélice**



**Tableau 3** Graphique représentant la poussée fournie (en grammes) par 2 moteurs selon le type d'hélice



**Tableau 4** Graphique représentant le rendement pour les différentes hélices

Nous pouvons donc conclure que d'après les graphiques précédents, les hélices 10x4.5 sont les plus appropriées. Ceci peut s'expliquer par le fait que les autres hélices sont des hélices optimisées pour des moteurs thermiques alors que les 10x4.5 sont optimisées pour des moteurs électriques. Or les moteurs sur le drone sont des moteurs électroniques. Les hélices les plus adaptées au drone sont donc les hélices 10x4.5.

### 3° Optimisation de la masse du quadrirotor

Malgré les gains de puissance assurés par la refonte de la partie puissance le drone ne se sustentait toujours pas suffisamment nettement. Il nous a donc fallu jouer sur la masse de ce dernier. Les études de dimensionnement faites par les groupes précédents assuraient une poussée totale de 2,8 kg ainsi la masse totale n'a pas été un point dimensionnant dans les réalisations de la structure et du câblage. C'est pourquoi nous avons pu gagner sur ces deux points près de 400g soit 25% de la masse initiale de 1,6 kg. Les différentes opérations sont détaillées ci-dessous et un tableau récapitule leur apport respectif.

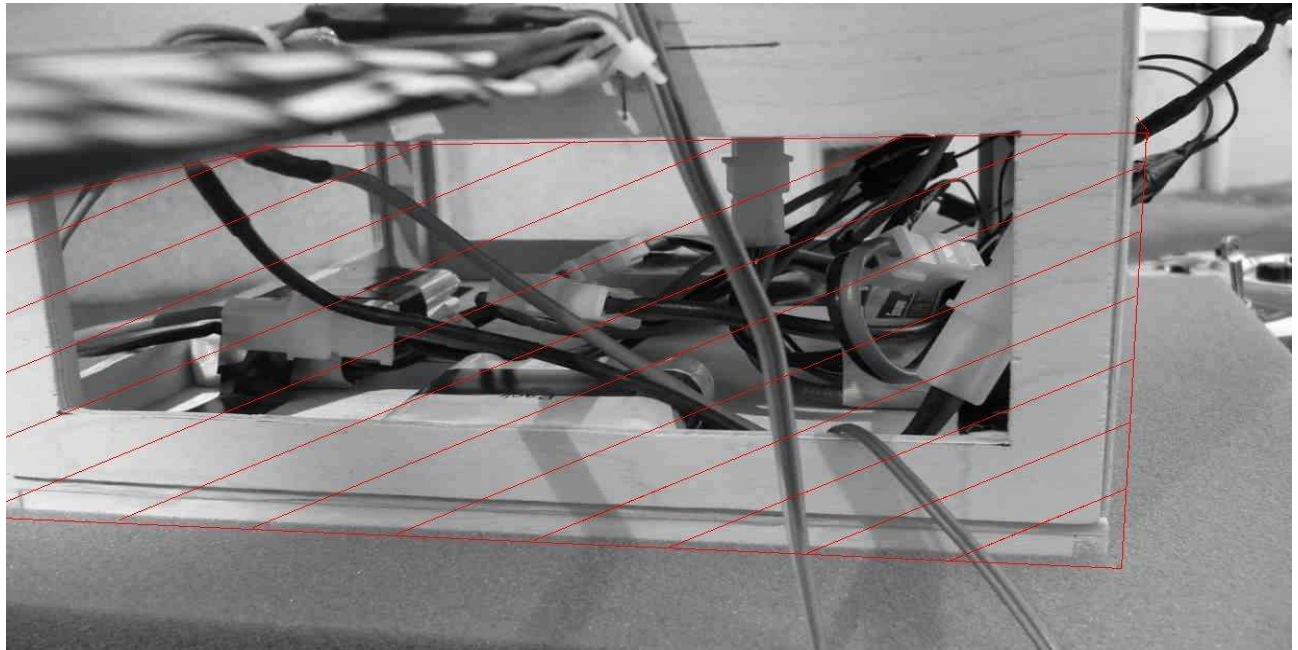
Avant tout c'est la refonte de la partie câblage détaillée plus haut qui nous a donné les meilleurs gains. Avec essentiellement le retrait d'une batterie et de cinq interrupteurs. Nous avons aussi travaillé à réduire au maximum les câbles car la masse de ceux-ci est non négligeable. Il est aussi à noter que pour la seconde structure nous avons réussi à retirer le second pack de 4,8V.



*Six des huit interrupteurs sur leur support*

Nous avons aussi travaillé à un allègement de la structure. Nous avons tout d'abord coupé deux entailles dans les flans du drone afin d'y glisser les packs de propulsion, le pack MicroPilot trouvait quant à lui sa place sous la platine centrale. Cette opération a permis de retirer une partie du bas du drone.





*La zone rayée a été totalement retirée*

Le retrait des interrupteurs a aussi permis de retirer une partie de la partie supérieure. Dès lors la structure était réduite à son strict minimum et il était impossible de gagner encore du poids sans réaliser de très grosses modifications structurales. Nous ne les avons pas appliquées car nous savions que nous ferions sans doute une nouvelle structure.

Masse initiale	1640
Déplacement des variateurs	40
retrait du bloc de 5 interrupteurs	80
retrait des supports interrupteurs	30
retrait pack de réception	120
retrait de la partie inférieure	70

*Tableau récapitulatif des gains*

Comme on peut le remarquer chacune de ces modifications ne nous a pas fait gagner beaucoup de poids. Cependant on arrive sur un total de près de 400g soit un quart de la masse initiale du drone. Ce gain, ainsi que le gain de puissance dû à l'assainissement de la partie électrique nous a permis de faire décoller le drone et donc de nous apercevoir que celui-ci n'était en fait pas du tout stable en vol, contrairement aux résultats en vol. Nous avons donc commencé à nous attaquer à la partie asservissement.

## II. Optimisation du code implémentée dans la carte MicroPilot

### 1° Synthèse sur l'asservissement initial du quadrirotor

Pour une bonne lisibilité du code implémentée, les variables utilisées sont assez explicites avec un jeu de suffixes. Les radicaux utilisés sont :

- ⇒ yaw : cela détermine une position en lacet.
- ⇒ roll : cela détermine une position en roulis.
- ⇒ pitch : cela détermine une position en tangage.
- ⇒ thrust : cela correspond à une valeur de poussée.

Pour une meilleure compréhension des différentes variables, voila une explication des différents suffixes que l'on peut rencontrer dans le code :

- ⇒ t correspond à la valeur de la télécommande : tyaw, troll, tpitch, tthrust.
- ⇒ c correspond à la valeur commande une fois mise en forme (variant de -32767 à 32767) : cyaw, croll, cpitch, cthrust.
- ⇒ m correspond à une valeur mesurée par la carte MicroPilot.
- ⇒ p correspond à une vitesse calculée ou mesurée : pyaw, proll, ppitch.
- ⇒ pp correspond à une accélération calculée par différenciation : ppyaw, pproll, pppitch.
- ⇒ i correspond à une intégration calculée : iyaw, iroll, ipitch.
- ⇒ zero correspond à une valeur initiale.
- ⇒ mix correspond à une valeur qui va permettre aux moteurs d'adapter leurs vitesses de rotation pour permettre un déplacement selon un certain angle du quadrirotor : yawmix, pitchmix, rollmix.
- ⇒ trim correspond à une valeur qui va être ajoutée à la poussée des moteurs pour compenser les écarts relatifs des différents moteurs.
- ⇒ Intermédiaires de calcul : c1 et c2 et d (pour le calcul du rate limiter), p1 et p2 (pour le calcul de l'accélération).
- ⇒ La commande MPVAR(variable) signifie que la variable est lue à partir de la carte MicroPilot.

La carte initialise avec la poussée au minimum pour avoir une plage de poussée la plus importante possible. De même toutes les commandes d'angle doivent être au neutre pour une bonne initialisation.

```
//initialisation de la carte, on récupère les valeurs zéro de la télécommande
//ce code n'est exécuté qu'une fois au démarrage de la carte
if(init==1){

    zerothrust = MPVAR(tthrust);
    zeroyaw = MPVAR(tyaw);
    zeroroll = MPVAR(troll);
    zeropitch = MPVAR(tpitch);
    c2pitch = zeropitch;
    c2roll = zeroroll;
    c2yaw = zeroyaw;
    init = 0;
}
MPVAR(offground) = 0;
```

Nous ne pouvons pas utiliser directement les valeurs lues avec la télécommande, il faut les mettre en forme pour avoir des signaux comparables aux valeurs reçues par les fServo.

```
//mise en forme des signaux envoyés par la télécommande en signaux de grandeurs
comparable aux fServo (variant entre -32767 et +32767)

croll = (MPVAR(troll) - zeroroll)*32767/820;
cpitch = (MPVAR(tpitch) - zeropitch)*32767/820;
cyaw = (MPVAR(tyaw) - zeroyaw)*32767/820;
cthrust = -32767+(MPVAR(tthrust)-zerothrust)*32767/820 ; //la position 0 pour
les gazs correspond à la manette en bas, donc à -32767
```

Il n'existe pas de valeur lisible pour l'accélération avec la carte MicroPilot, c'est pourquoi nous devons utiliser une méthode de différenciation des vitesses mesurées par la carte MicroPilot pour calculer l'accélération du système.

```
//calcul de l'accélération le 0.24 est une echelle de temps
p1pitch = p2pitch;
p2pitch = MPVAR(mppitch);
pppitch = (p2pitch - p1pitch)*100/24;

p1roll = p2roll;
p2roll = MPVAR(mproll);
pproll = (p2roll-p1roll)*100/24;
```

Dans une optique de stabilisation du drone, un rate limiter avait été calculé pour limiter la vitesse d'inclinaison du quadrirotor. Une vitesse limite de  $11^\circ/\text{s}$  avait été retenue. Ce rate limiter empêche les changements trop brusques de position du système, cela ralentit un peu le système mais lui assure une meilleure stabilité.

```
//calcul du rate limiter
c1pitch = c2pitch;
c1roll = c2roll;
c1yaw = c2yaw;

c2pitch = cpitch;
c2roll = croll;
c2yaw = cyaw;

dpitch = (c2pitch-c1pitch);
droll = (c2roll-c1roll);
dyaw = (c2yaw-c1yaw);

if(dpitch > 400){ dpitch = 400;} //le rate limiter filtre à partir d'une commande de
plus de  $11^\circ$  par seconde
if(dpitch < -400){ dpitch = -400;}
if(droll > 400){ droll = 400;}
if(droll < -400){ droll = -400;}
if(dyaw > 400){ dyaw = 400;}
if(dyaw < -400){ dyaw = -400;}

cpitch = c1pitch + dpitch;
croll = c1roll + droll;
cyaw = c1yaw + dyaw;
```

L'erreur de position est calculée entre la valeur de la position de la commande une fois mise en forme et la position actuelle mesurée par la carte MicroPilot. Cette erreur sera limitée par une saturation à des angles de  $30^\circ$  pour le roulis et le tangage et de  $45^\circ$  pour le lacet car le drone est considérée non contrôlable en dehors de ces limites.

```
//calcul de l'erreur de position
//on supposera que le drone ne peut pas excéder des angles de 30 degrés et 45 degrés en
lacet

pitch = cpitch-MPVAR(mpitch)*32767/540; // 540 =  $30^\circ$  en radians * 1024
roll = croll-MPVAR(mroll)*32767/540;
yaw = cyaw-MPVAR(myaw)*32767/810; //810= $45^\circ$  en rad *1024
```

On calcule l'effet intégral par sommation et il est limité pour éviter des valeurs aberrantes qui conduiraient à l'instabilité du quadrirotor.

```
//calcul de l'effet integral
ipitch = ipitch + pitch;
iroll = iroll + roll;
iyaw = iyaw + yaw;
if(ipitch > 10000){ipitch = 10000;}
if(ipitch < -10000){ipitch = -10000;}
if(iroll > 10000){iroll = 10000;}
if(iroll < -10000){iroll = -10000;}
if(iyaw > 10000){iyaw = 10000;}
if(iyaw < -10000){iyaw = -10000;}
```

De même que les valeurs commandées par la télécommande, les vitesses et les accélérations ont besoin d'une mise à l'échelle pour être utilisables.

```
//mise à l'échelle des vitesses et accelerations(sur une échelle de -32767 à 32767)
ppitch = MPVAR(mppitch)*32767/2000;
proll = MPVAR(mproll)*32767/2000;
pyaw = MPVAR(mpyaw)*32767/2000;

pppitch = pppitch*32767/2000;
pproll = pproll*32767/2000;
ppyaw = ppyaw*32767/1000;
```

Une fois que la position, la vitesse, l'accélération et l'effet intégral sont calculés il est possible d'écrire un correcteur adapté avec des coefficients à déterminer. On remarque que les coefficients sont des variables qui sont lues par la carte MicroPilot car ces coefficients seront écrits dans la mémoire de celle-ci grâce au logiciel MicroPilot Horizon : ce logiciel sera très utile pour régler le correcteur PIDD<sup>2</sup>.

```
//calcul du correcteur
//le /100 permet d'utiliser des coeffs 100 fois plus grands (plus lisible)
pitchmix = (MPVAR(K1roll)*pitch - MPVAR(K2roll)*ppitch
MPVAR(K3roll)*pppitch + 0*MPVAR(K4roll)*ipitch)/100;

rollmix = (MPVAR(K1roll)*roll - MPVAR(K2roll)*proll -
MPVAR(K3roll)*pproll + 0*MPVAR(K4roll)*iroll)/100;

yawmix = (MPVAR(K1yaw)*yaw - MPVAR(K2yaw)*pyaw -
MPVAR(K3yaw)*ppyaw + 0*MPVAR(K4yaw)*iyaw)/100;
```

On peut alors affecter cette correction au niveau des moteurs. En effet, 70% de la poussée des moteurs servira à la sustentation du système et les 30% restants serviront à corriger et à stabiliser le drone. Il suffit alors de soustraire ou additionner les valeurs correctives calculées précédemment pour avoir un effet correctif sur le quadrirotor : le signe au niveau de chaque moteur est calculée à partir des équations de la mécanique grâce au moment créé par chaque hélice.

```
//répartition de la correction sur chaque moteur et mise en forme
```

```
mot1 = (-pitchmix-yawmix)*0.3;  
mot2 = (+rollmix+yawmix)*0.3;  
mot3 = (+pitchmix-yawmix)*0.3;  
mot4 = (-rollmix+yawmix)*0.3;
```

```
//mise en forme de la poussée, thrust représente 70% max de la poussée du moteur
```

```
thrust = (cthrust+32767)*0.7 - 32767;  
  
rolltrim = 4000 ;  
pitchtrim = 3500;  
if(thrust > -10000){  
    rolltrim = 4200;  
    pitchtrim = 4200;  
}  
if(thrust > 4000){  
    rolltrim = 5000;  
    pitchtrim = 5000;  
}
```

Il est alors possible de mettre en forme la poussée au niveau de chaque moteur qui sera composée de trois termes : un terme de puissance (thrust), un terme de correction (mot) et un terme d'équilibrage (-trim). Tout ceci est aussi contrôlé en saturation pour ne pas dépasser la plage de la télécommande.

```
//Une vérification est faite sur la commande pour éviter de saturer un moteur
if((thrust+mot2) < 32700){
    MPVAR(fServo5) = thrust+mot2+rolltrim;
}
else{MPVAR(fServo5) = 32700;}

if((thrust+mot3) < 32700){
    MPVAR(fServo6) = thrust+mot3+pitchtrim;
}
else{MPVAR(fServo6) = 32700;}

if((thrust+mot4) < 32700){
    MPVAR(fServo7) = thrust+mot4-rolltrim;
}
else{MPVAR(fServo7) = 32700;}

if((thrust+mot1) < 32700){
    MPVAR(fServo8) = thrust+mot1-pitchtrim;
}
else{MPVAR(fServo8) = 32700;}
}
```

Après avoir optimisé la mécanique et l'électronique du quadrirotor, nous nous sommes concentrés sur le code et avons essayé de l'optimiser afin de faire décoller le drone. C'est pourquoi dans une deuxième partie nous allons expliquer quelles modifications nous avons apportées au code original.

## 2° Retouche nécessaire au code original

Nous avons choisi de conserver les notations du groupe précédent par soucis de lisibilité et de compréhension des modifications apportées.

Au niveau de l'initialisation de la carte, rien n'a été modifié. L'établissement des zéros de la télécommande est toujours le même au niveau de la position des manettes au démarrage de la carte MicroPilot.

De même, la mise à l'échelle des signaux envoyés par la télécommande a été conservée.

En ce qui concerne, le calcul de l'accélération, nous avons gardé le même procédé même si les courbes obtenues lors de nos essais nous montrer que nous n'obtenons qu'une faible approximation de la réalité au niveau de l'accélération.

En revanche, en ce qui concerne le rate limiter, même si nous avons choisi de garder le même principe en terme de calcul, nous avons déterminé une nouvelle valeur limite à l'aide d'un simple calcul. Nous avons fixé cette valeur limite à 650 (lim). En effet, si nous nous déplaçons sur la plage de la télécommande de 650 points par itération sachant que la fréquence d'échantillonnage est de 30 Hz, alors nous nous déplaçons de  $650 * 30 = 20\,000$



points par seconde. Or, la plage de la télécommande comporte  $32767 \times 2 + 1 = 65535$  points, par conséquent nous atteindrons notre butée en 3s, ce que nous avons jugé satisfaisant.

```
// Application du limiteur de vitesse (lim = 650)
if((cpitch - c1pitch) > lim){c1pitch = c1pitch + lim;}
else{
    if((cpitch - c1pitch) < -lim){c1pitch = c1pitch - lim;}
    else{c1pitch = cpitch;}
}

if((croll - c1roll) > lim){c1roll = c1roll + lim;}
else{
    if((croll - c1roll) < -lim){c1roll = c1roll - lim;}
    else{c1roll = croll;}
}
```

De plus, nous avons créé un rate limiter au niveau de la poussée. Cela permet au système d'éviter d'avoir des réactions trop violentes lors de perturbations et ainsi créer une instabilité. Par ailleurs, cela permet aussi une sécurité lors des essais, cela permet de réduire l'amplitude de l'instabilité qui peut causer la destruction du matériel. Nous avons utilisé le même calcul que précédemment pour déterminer la limite.

```
//rate limiter sur les gaz
if((thrust1 - thrust)>650){thrust = thrust + 650;}
else{
    if((thrust1 - thrust)<-650){thrust = thrust - 650;}
    else {thrust = thrust1; }
}
```

Pour calculer l'erreur de position, nous avons conservé le même principe. En revanche, au vue de la grande instabilité du système, nous avons décidé d'abaisser les limites en tangage et en roulis du système. Cela permettra de n'avoir que de faibles angles de débattement et une plus grande stabilité. Nous avons choisi de réduire cette limite à  $15^\circ$  qui va correspondre à l'angle fait par les moteurs avec le plot central dans la nouvelle structure que nous avons construite. Par ailleurs, nous avons choisi de ne pas contraindre le lacet ( $180^\circ$ ).

```
//calcul de l'erreur de position
//on supposera que le drone ne peut pas excéder des angles de 15 degrés et 45 degrés en lacet

pitch = c1pitch-MPVAR(mpitch)*32767/270; // 270 = 15° en radians * 1024
roll = c1roll-MPVAR(mroll)*32767/270;
yaw = (cyaw - MPVAR(myaw))*3215/3215; // 3215 = 3.14 * 1024
```

Ensuite, nous avons remarqué que l'effet intégral provoquait une trop grande instabilité car son accroissement était trop rapide, c'est pourquoi nous avons décidé de diviser l'incrément de l'intégral par 10. Nous avons cependant conservé les limites et le processus de fabrication de cet effet intégral.

```
// Calcul de l'effet integral
ipitch = ipitch + pitch/10;
iroll = iroll + roll/10 ;
if(iroll > 10000){iroll = 10000;}
else{
    if(iroll < -10000){iroll = -10000;}
}
if(ipitch > 10000){ipitch = 10000;}
else{
    if(ipitch < -10000){ipitch = -10000;}
}
```

En ce qui concerne la définition des vitesses et des accélérations et de la mise à l'échelle de celles-ci nous avons conservé la méthode précédente. Nous avons envisagé le fait de calculer la vitesse nous-mêmes en utilisant une méthode de différenciation avec les positions issues de la carte MicroPilot. Cela nous aurait permis de gagner en rapidité au niveau du temps de réponse du système et aussi de maîtriser l'erreur commise dans le calcul de la vitesse. Cependant l'erreur commise en utilisant cette méthode de différenciation était trop importante c'est pourquoi nous n'avons préféré la laisser de côté et utiliser la vitesse que nous fournissait la carte MicroPilot.

Dans un souci de stabilité, nous avons mis en place un système de saturation de la vitesse et de l'accélération. En effet cela nous permet d'avoir un système plus lent mais plus stable. Pour fixer la limite de la saturation, nous avons pris un coefficient moyen du gain dérivé (10) et nous avons établi que la saturation devait se produire à partir de 10% de la course de la dérivée soit  $(1/10 \cdot 65\,535 \cdot 100 / 0.3) / 2 = 22\,000 / 2 = 11\,000$ . (Le 100 correspond au coefficient par lequel on a divisé les  $-mix$  et le 0.3 le coefficient par lequel on les multiplie au niveau de la poussée des moteurs.

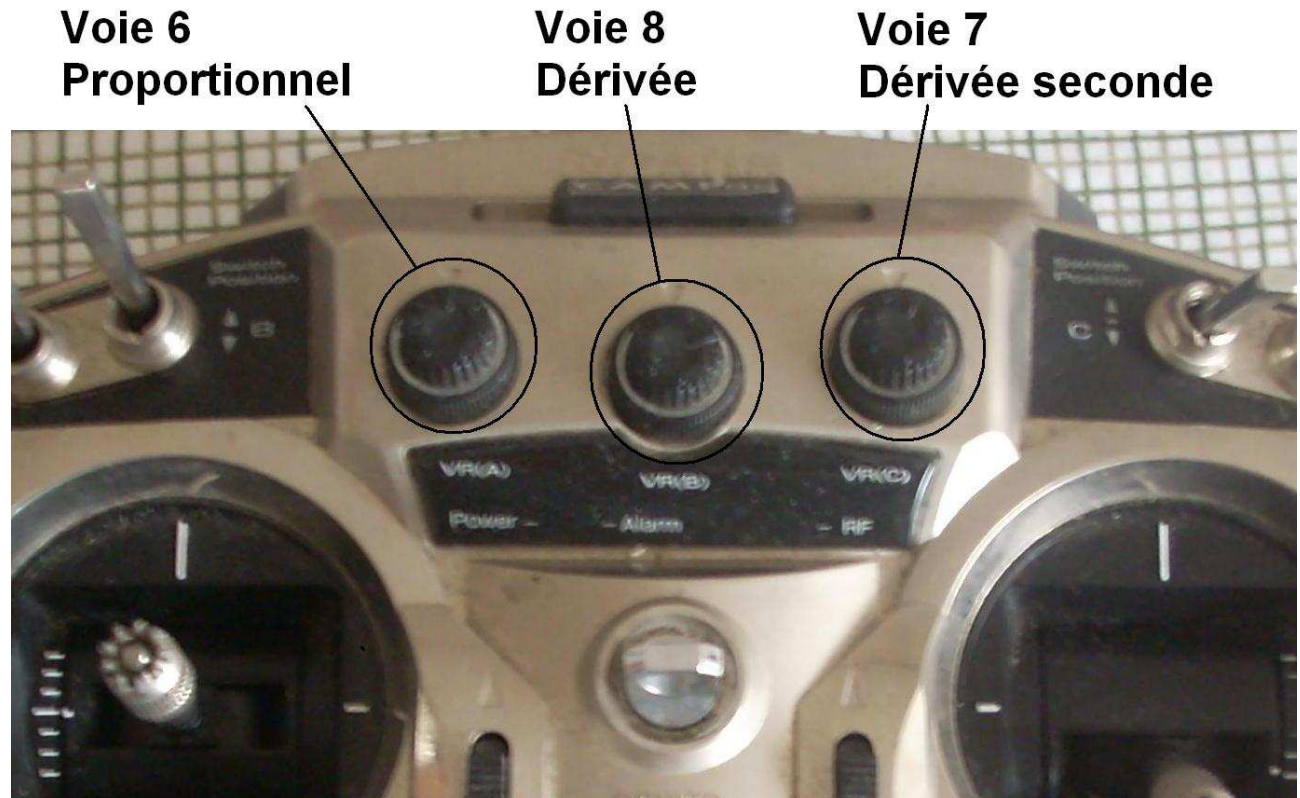
```
//saturation de la vitesse (Kd=10 => 10% de la course pour la dérivée -> saturation à 220000/2=110000)
if(ppitch > 110000){ppitch = 110000;      }
else{
    if(ppitch < -110000){ppitch = -110000;}
}

if(proll > 110000){proll = 110000;}
else{
    if(proll < -110000){proll = -110000;}
}

//saturation de l'accélération (Kd=10 => 10% de la course pour la dérivée -> saturation à 220000/2=110000)
if(pppitch > 110000){pppitch = 110000;}
else{
    if(pppitch < -110000){pppitch = -110000;}
}

if(pproll > 110000){pproll = 110000;}
else{
    if(pproll < -110000){pproll = -110000;}
}
```

Afin de mieux comprendre l'effet de chacun des coefficients du correcteur nous avons décidé de faire des essais sur de nombreuses configurations. Cependant pour que ces essais soient faisables en un temps raisonnable il fallait trouver un moyen rapide de modifier les coefficients. Pour ce faire nous avons utilisé les trois voies de l'émetteur commandées par des potentiomètres.



La carte MicroPilot ne pouvant accepter que quatre entrées, les entrées de roulis, tangage et lacet ont été remplacées par des entrées P, D et D<sup>2</sup>. Ainsi le correcteur était modifiable en temps réel ce qui nous a permis de réaliser un très grand nombre d'essai en un temps réduit.

Nous voulions une plage de réglage de 100. Comme la plage de variation d'une entrée est de -32767 à 32767 nous la ramenons à une variation [0,65000] puis divisons par 650. On s'assure ensuite que ces coefficients soient toujours positifs.

```
kroll = (32700 + croll)/650;
kpitch = (32700 + cpitch)/650;
kyaw = (32700 + cyaw)/650;
```

```
if (kroll<0) {kroll=0;}
if (kpitch<0){kpitch=0;}
if (kyaw<0) {kyaw=0;}
```

Ces coefficients kroll, kpitch et yaw commandés respectivement par les entrées roll, pitch et yaw sont ensuite utilisés dans le calcul de la commande.

Afin de pouvoir récupérer leur valeurs nous les mettons en sorties sur des servos inutilisés, les servos 9,10 et 11. C'est ainsi qu'après avoir réutilisé les coefficients rentrés sous horizon et laissé ce moyen de control, nous nous sommes aperçu que le logiciel appliqué un facteur d'échelle entre les consignes rentrés et les valeurs réellement enregistrées sur la carte. Ces

facteurs pouvaient aller de 0,009 à -26,2 en passant par 0,91 ce qui a par conséquent faussé tout les essais antérieurs.

Pour le correcteur, pour le tangage et le roulis, le même modèle a été utilisé avec la prise en compte des facteurs de forme calculés précédemment. Par ailleurs nous avons aussi diminué le gain du correcteur par 100 par rapport aux autres toujours dans un souci de lisibilité. Cependant au niveau de la correction en lacet, nous avons simplement utilisé un correcteur dérivé car seul les variations de lacet nous intéresse, nous voulons simplement le stabilisé et non le diriger. C'est aussi une des raisons pour lesquelles nous n'utilisons pas le cap et le compas. Nous avons essayé de réduire au minimum les interactions entre les différents angles.

```
//calcul du correcteur
//le /100 permet d'utiliser des coeffs 100 fois plus grands (plus lisible)
pitchmix = (MPVAR(K1roll)*1/0.91*pitch - MPVAR(K2roll)*ppitch -
MPVAR(K3roll)*pppitch + ipitch*MPVAR(K4roll)*(-1/26.2)*1/100 )/100 +
MPVAR(K2yaw) - MPVAR(K3yaw) ;
rollmix = (MPVAR(K1roll)*1/0.91*roll - MPVAR(K2roll)*proll -
MPVAR(K3roll)*pproll + iroll*MPVAR(K4roll)*(-1/26.2)*1/100 )/100 ;
yawmix = MPVAR(K1yaw)*pyaw/100;
```

Au niveau de la répartition de la poussée des moteurs, nous avons enlevé les –trim car les différences de puissance entre chaque moteur sera corrigé par l'asservissement, ce terme n'est pas utile. En revanche, avec la nouvelle structure, les moteurs sont montés à l'envers par conséquent, nous avons du changer les signes des corrections appliquées.

```
//répartition de la correction sur chaque moteur et mise en forme
mot1 = thrust + (-pitchmix + yawmix )*0.3; // les moteurs tournent à l'envers donc
on a changé les yawmix avec leur signe opposé
mot2 = thrust + (+rollmix - yawmix )*0.3;
mot3 = thrust + (+pitchmix + yawmix )*0.3;
mot4 = thrust + (-rollmix - yawmix )*0.3;
```

La mise en place de la poussée n'a pas eu besoin de changer et est restée la même.

Grâce à ces retouches dans le code, le drone a décollé au mois de janvier et nous nous sommes appliqués alors à créer une nouvelle structure plus légère et optimisée et à la stabiliser.

### III. Création d'une nouvelle structure optimisée

#### 1° Pourquoi une nouvelle structure ?

En effet, nous avons plusieurs possibilités : continuer à travailler sur la même structure ou effectuer des modifications de structure. Nous avons remarqué lors d'un essai que cette structure n'assurait plus la sécurité de l'électronique et en particulier de la carte MicroPilot : la carte était sur la plateforme supérieure du drone et donc si le drone se retournait, cela pouvait détériorer la carte. Il nous fallait donc modifier la structure.

Nous aurions pu ajouter un harnais de sécurité à la structure pour assurer la sécurité de la carte MicroPilot mais nous avons préféré construire une nouvelle structure pour plusieurs raisons. La plus importante raison est que la structure était déjà assez lourde et rajouter un harnais n'aurait fait que l'alourdir et nous n'aurions pas été sûrs que le drone ait encore assez de puissance pour décoller. Ensuite, nous voulions remédier à quelques problèmes mineurs tels que les nacelles des moteurs, les logements pour les batteries et également l'encombrement. Tout d'abord au sujet des nacelles, nous voulions réduire les supports moteurs pour éviter que les flux des hélices ne soient réduits. Il fallait donc prévoir des supports moteurs plus petits. Ensuite en ce qui concerne les logements des batteries : notre solution actuelle était de les insérer dans des caches cependant les batteries n'étaient pas calées et donc bougeaient lors des essais et cela était dangereux car si une batterie tombe elle risque de ne plus fonctionner. Et enfin le « problème » de l'encombrement : selon nous, la structure était trop grande, il était possible de réduire la taille tout en gardant les objectifs initiaux à savoir incorporer l'électronique nécessaire afin d'avoir un drone autonome capable d'embarquer un système d'observation.

#### 2° Conception de la nouvelle structure



Figure 1 Modèle Catia de la structure finale

Nous avons donc décidé de faire une nouvelle structure, et nous avons décidé de la faire la plus légère possible. Ceci peut paraître étrange car comme le drone n'est pas stable, il est amené à prendre des chocs et donc on aurait tendance à vouloir le faire le plus robuste possible. Cependant si le drone est léger, son inertie sera faible et donc l'impact au niveau des chocs ne sera pas trop important. Ainsi, nous avons calculé les dimensions des différents éléments électroniques susceptibles de s'abimer lors des chocs (à savoir : la carte MicroPilot, le compas et le récepteur) et avons élaboré une plateforme en étoile pour disposer correctement les éléments.



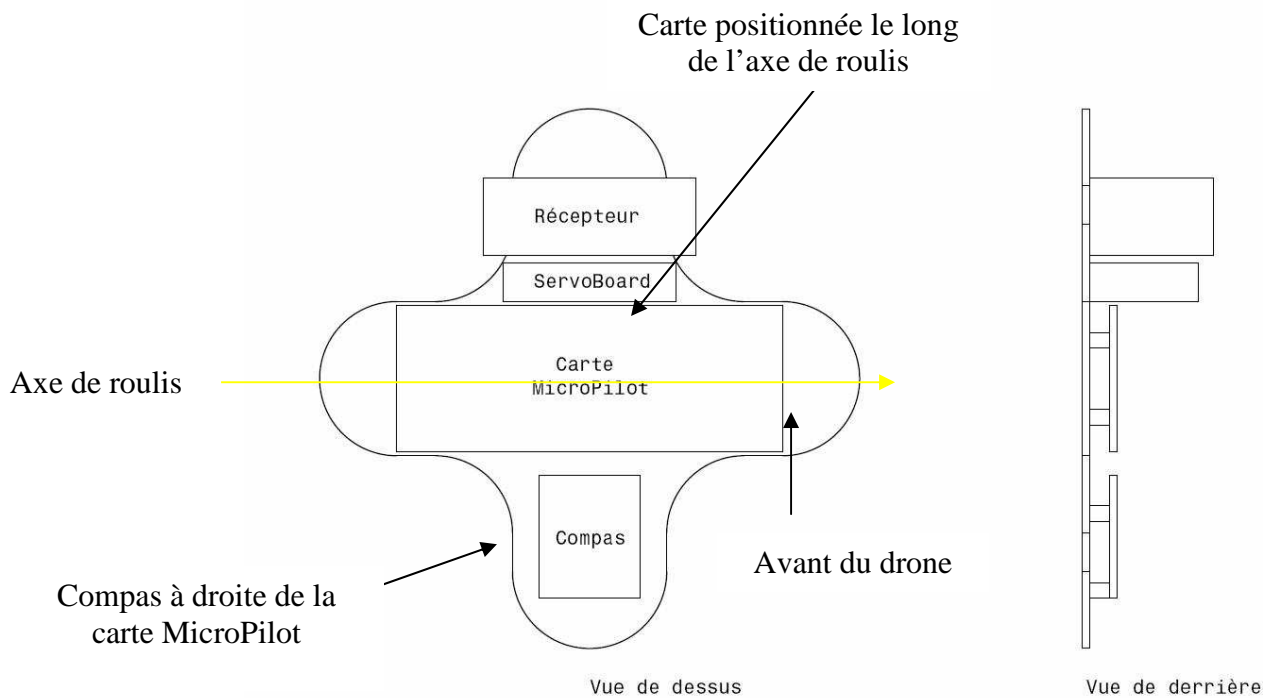
**Figure 2 Plateforme centrale accueillant l'électronique**

Composant\dimensions	Longueur (en cm)	Largeur (en cm)
Carte MicroPilot	10,2	3,7
Récepteur	5,5	3,3
Compas	2,7	2,7
Plaquette « assembleur »	4,8	2,3
Batterie Li-Po	7,7	4
Batterie 4.8V	5,5	4,9

**Figure 3 Tableau des dimensions des composants électroniques (sauf moteurs et variateurs)**

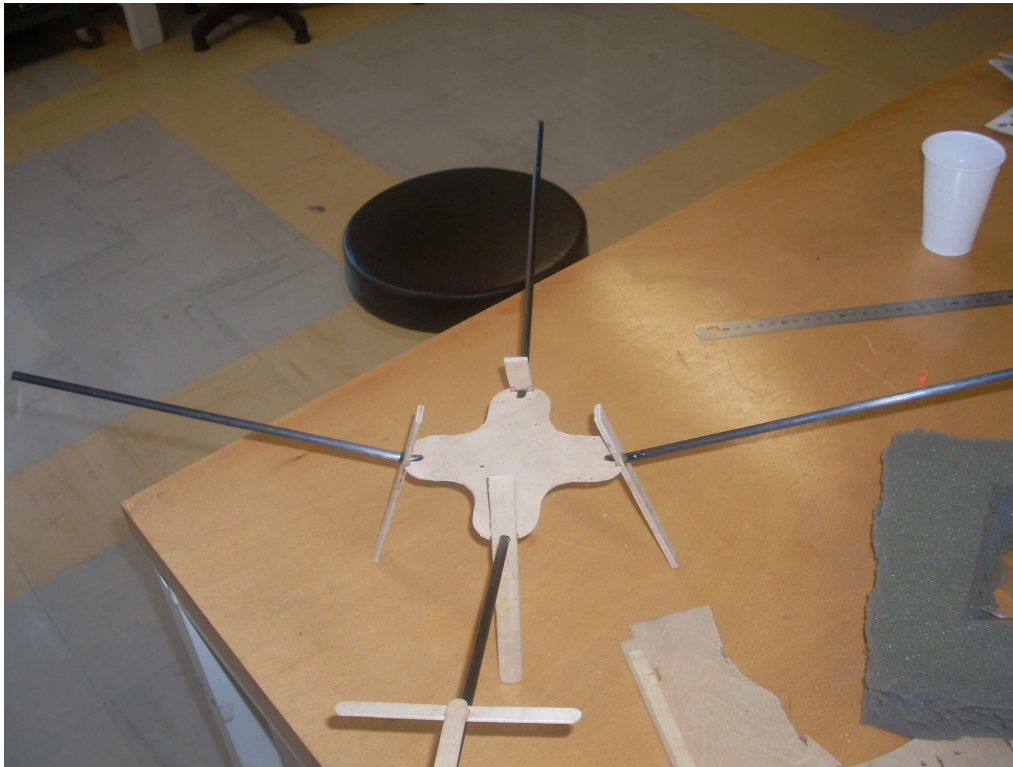
La carte MicroPilot montre l'avant du drone, elle doit être placée de façon symétrique sur l'axe de roulis et le compas doit être disposé à sa droite en regardant devant, le récepteur a ensuite été placé à la gauche de la carte. Le matériau utilisé pour la plateforme était le même que sur l'ancienne version, un contre-plaqué assez rigide et très léger.





**Figure 4 Disposition nécessaire des éléments au bon fonctionnement de la carte**

La plateforme terminée, le problème de la disposition des bras s'est posé. En effet, il faut que les moteurs soient disposés de manière symétrique, à la même distance du centre du quadri-rotor afin de ne pas avoir de dissymétries de poussée. Plusieurs possibilités s'offraient à nous : conserver les bras à plat comme c'était le cas sur la version précédente ou incliner les bras. Nous avons choisi la dernière solution car la première était trop compliquée à réaliser avec le type de plateforme que nous avons envisagé et en considérant le fait que nous voulions conserver des bras en carbone par soucis de légèreté. De plus, avec les bras inclinés, nous augmentons l'effet dièdre qui est un effet stabilisant. En privilégiant l'effet dièdre, nous avons voulu stabiliser d'avantage le drone tout en sachant que cela serait défavorable en termes de manœuvrabilité. Ensuite, le choix de l'angle du dièdre a été plus ou moins contraint par la facilité de conception. Puisque nous avons décidé de faire une plateforme la plus minime possible, nous n'avons pas beaucoup de place pour les bras. Nous avons donc du faire un compromis : nous avons agrandi légèrement la plateforme et avons établi un angle de quinze degrés. Pour assurer le maintien des bras, qui sont fortement sollicités par les moteurs, nous avons décidé que les bras devraient passer dans les pieds du drone pour assurer une bonne rigidité. Les bras étaient fixés au centre du drone sur un petit plot central.



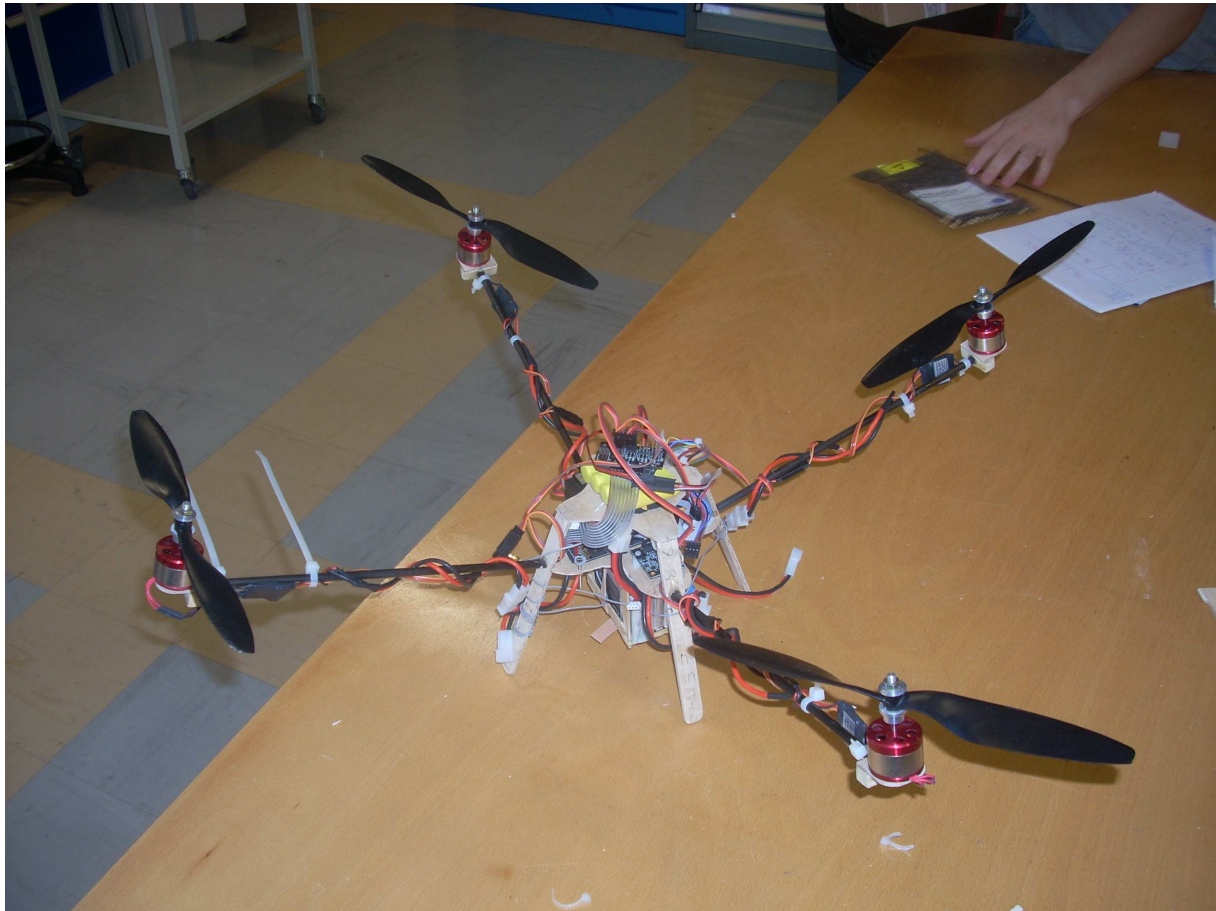
**Figure 5 Bras et pieds collés à la plateforme centrale**

Une fois les bras disposés, il fallait fixer les moteurs. Nous avons voulu faire les supports moteurs les plus petits possibles pour éviter qu'ils ne limitent les performances des hélices. Les supports moteurs viennent s'emboîter dans la tige de carbone.

Il ne restait plus que les pieds à élaborer. Nous avons déjà bien contraints leur conception puisqu'il fallait que les bras en carbone passent au travers. Le matériau le plus approprié était le contre-plaqué rigide (compromis rigidité et poids). Il ne restait plus que leurs longueurs à évaluer. Il fallait que les pieds soient assez longs pour que la distance entre la plateforme supérieure (cf paragraphe suivant) et le sol, sachant que nous avons décidé de mettre les batteries Li-Po sous le drone, en dessous du point de fixation des tiges de carbone. Puis nous avons décidé dans un premier temps d'ajouter les batteries Li-Po sous la structure, on, avait fabriqué une « cage » où venaient se logeaient les batteries.

Enfin, il ne faut pas oublier que l'objectif premier du drone était de concevoir un engin volant capable d'embarquer un système d'observation. Afin de satisfaire cette demande, nous avons conçu une plateforme supérieure qui d'une part protège l'électronique lors des phases de tests pour la stabilisation et d'autre part reste un emplacement libre assez grand qui peut accueillir un système d'observation. Au début cette plateforme accueillait la batterie de 4,8Volts.

Une fois toutes les pièces réalisées, nous les avons assemblées avec de la colle EPOXY. Dans un premier temps nous avons assemblé les deux plateformes, les tiges et les pieds ensemble.



**Figure 6 Première version de la structure**

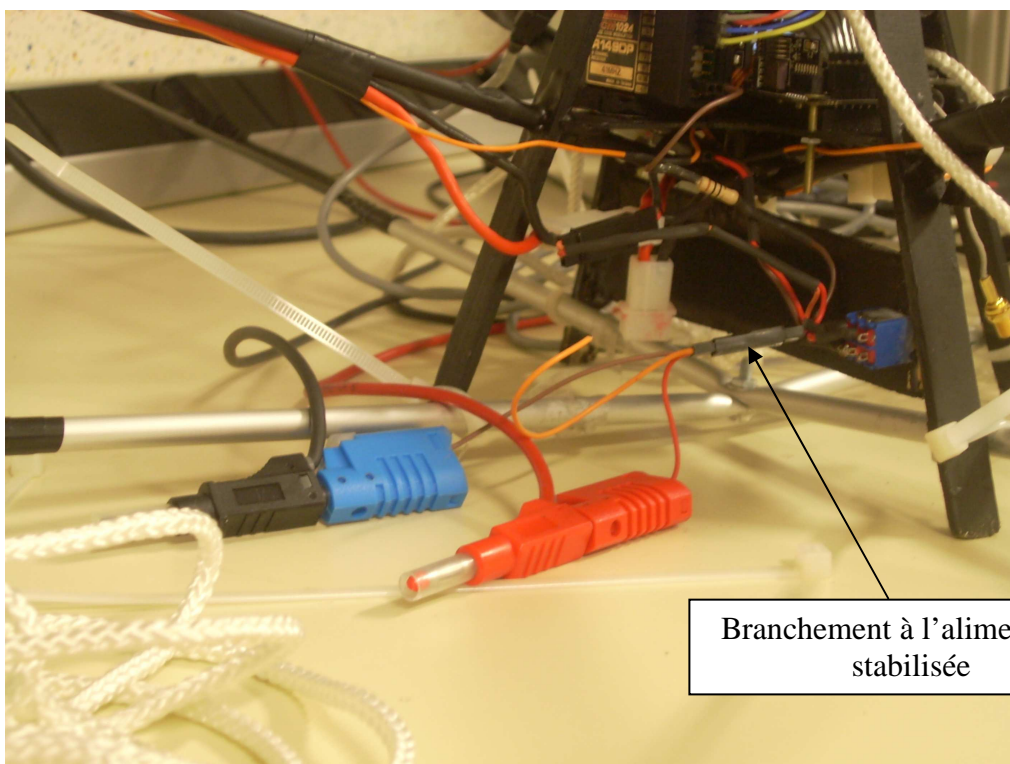
La photo ci-dessus nous montre la première version du drone. Quelques remarques nous ont été faites : l'esthétique, le côté pratique. En premier c'est en effet l'esthétique qui nous choque puisqu'il y a des câbles dans tous les sens, ensuite le contre-plaqué ne rendait pas bien à l'œil. Enfin, pour le côté pratique, cela concerne les batteries, la « cage » inférieure n'était pas optimisée et cela rendait difficile la mise en marche du drone. Puis la batterie de 4,8 Volts située sur la plaque supérieure ne contribuait ni à l'esthétique, ni la simplicité du drone, ni même à la stabilité du drone. Et enfin cela concernait également l'antenne que nous devons enrouler autour de la structure. Néanmoins c'est avec cette structure que nous avons décidé de poursuivre nos essais de tests de stabilisation.

Cependant, lors d'un test, un bras a heurté le mur et le carbone a perdu sa rigidité en torsion mais pas en flexion. Son remplacement était pourtant indispensable. Nous en avons profité pour améliorer notre structure en prenant en compte les remarques. Nous avons tout d'abord ajouté une plaque entre les pieds avant et arrière (le long de l'axe de roulis pour un souci de stabilité statique) pour ajouter les batteries Li-po. Nous avons ensuite décidé d'alimenter la carte MicroPilot à l'aide des batteries Li-Po, cela nous a permis de supprimer la batterie de 4,8 Volts. Afin de permettre l'initialisation de la carte MicroPilot, nous avons ajouté un interrupteur afin d'initialiser la carte sans risque à l'aide d'une alimentation stabilisée. Nous avons percé la plateforme supérieure et avons collé un tube qui permet à l'antenne de passer et de s'enrouler le long de ce tube. Nous avons finalement modifié les supports moteurs car les précédents ne résistaient pas aux efforts, nous les avons refait en plexiglas. Finalement, nous avons peint le drone pour le rendre plus joli et avons enroulé de scotch noir les câbles autour des bras.





Figure 7 Structure peinte juste avant l'intégration de l'interrupteur



Branchement à l'alimentation stabilisée

Figure 8 Interrupteur ajouté à la structure qui permet l'initialisation sans risque de la carte MicroPilot

### 3° Avantages et inconvénients de cette nouvelle structure

Cette structure étant plus petite que la précédente nous avons au début un surplus de câble comme on peut le voir sur la figure 6. Ainsi en coupant ces câbles au plus juste nous avons gagné pour 60g de câble, passant la structure de 850g à 790g. Cependant les finitions que sont la peinture, le scotch et le support d'antenne nous ont amené à une masse finale de 816g en ordre de vol (c'est à dire avec électronique et batteries). Il faut comparer cela aux 1600g de la structure initiale que nous n'avions pas pu descendre sous les 1200g. Ainsi l'un des buts principaux de conception a bien été atteint et la faible masse de cette structure en devient le principal avantage.

Notre second objectif était de créer une structure qui assurerait d'elle même la sécurité des éléments sensibles et ainsi de ne pas avoir à ajouter de protection. Or comme on peut le voir sur les photos précédentes tous les éléments électroniques sont situés entre les deux plaques qui assurent le maintien des pieds. Il est ainsi impossible d'atteindre un de ces éléments sur un choc. De plus le dièdre appliqué aux bras surélève les nacelles assurant par conséquent une meilleure protection des moteurs et des hélices. La plaque supérieure offre par ailleurs un support pour les futurs systèmes embarqués.

Cependant la véritable fonction de ce dièdre est tout autre. Il permet de descendre le centre de gravité du drone de 8cm en dessous des moteurs. Et par conséquent augmente la stabilité du drone. Les essais ont par ailleurs démontré que ce gain de stabilité n'a que peu entamé la manœuvrabilité qui reste plus que suffisante. Cette structure est ainsi plus stable que la précédente.

La présence de pieds a aussi permis l'adaptation d'un dispositif anti-renversement pour les tests. Celui-ci est constitué de carbone et d'aluminium. C'est une croix que l'on fixe sur les pieds du drone. Cette croix fait approximativement la même envergure que le drone et présente elle aussi un dièdre. On s'assure ainsi qu'elle touchera le sol que sur de grande incidence et ne perturbe pas le comportement du drone aux petits angles. Nous avons aussi ajouté des balles de tennis de table aux extrémités afin de s'assurer que les tiges glissent sur le goudron sans se planter.



***Drone équipé de son système anti-renversement***

Cependant, malgré toutes ces précautions il est arrivé que le drone se renverse et parfois très violemment. Nous avons ainsi constaté la solidité du drone. Cette structure est donc très robuste du fait sans doute de son faible poids et donc de sa faible inertie. Néanmoins, il nous est arrivé une fois de rompre un tube de carbone. Ce dernier n'a pas réellement cassé mais avait perdu toute résistance en torsion. C'est alors qu'est apparu un autre avantage de cette structure que nous n'avions pas soupçonné : il est extrêmement simple de changer un élément défectueux.

Enfin le dernier bon point de cette structure était prévu à la conception. Il s'agit de la faible taille des nacelles. Celles-ci font exactement la même taille que les moteurs. Ainsi la seule perturbation dans le flux des hélices est due uniquement aux tubes de carbones qui n'ont qu'un diamètre de 6mm. Nous avons ainsi pu gagner quelques points de poussée.

Certains des atouts de cette structure ont cependant une contrepartie désavantageuse. Par exemple le fait que toute l'électronique soit condensée en un lieu exigu et inaccessible en complique fortement l'accès. Nous avons cependant rarement eu besoin de modifier quelque chose à ce niveau là. Cet accès difficile est donc très peu handicapant.

Le seul vrai problème est plutôt lié aux phases d'allumage. En effet, nous avons fait en sorte que l'initialisation de la carte MicroPilot se fasse sous alimentation stabilisée. Cependant, cela nécessite que les batteries Li-Po ne soient pas branchées. Une fois la carte initialisée, nous devons brancher les batteries Li-Po et actionner l'interrupteur afin d'assurer la continuité d'alimentation de la carte MicroPilot. Mais lorsque quelqu'un branche les batteries, il faut qu'une autre personne tienne le drone car en cas de problème dans la carte les moteurs peuvent démarrer. Il faut néanmoins noter que la complexité de l'allumage est essentiellement due au fait que nous ne voulions pas placer des interrupteurs de puissance lourds et encombrants et non un effet de la structure.

## IV. Amélioration de l'asservissement

### 1° Modèle de notre asservissement

Un modèle Simulink avait été amorcé au départ du projet, nous avons résumé l'établissement de ce modèle à partir des équations de la mécanique dans l'annexe 5a. Nous avons ensuite travaillé ce modèle pour qu'il soit plus représentatif de la réalité.

Tout d'abord, le modèle de départ prévoyait la mise en place d'un PID dans la boucle avec des retours des états de position de la part des capteurs. Mais, en réalité, nous utilisons la vitesse de tangage et de roulis issue des capteurs pour notre correcteur. Par conséquent, pour le tangage et le roulis, le choix de notre correcteur est une action proportionnelle et intégrale sur l'écart en position et une action proportionnelle et dérivée sur l'écart en vitesse, ce qui correspond à un correcteur PIDD<sup>2</sup> globalement. De plus, nous avons pris le même correcteur pour le tangage et le roulis car les effets d'inertie sont globalement les mêmes sur ces deux axes. Pour le lacet en revanche, seule la position retournée par les capteurs est utilisée, par conséquent un simple gain sur l'écart de position peut permettre une bonne stabilité en lacet.

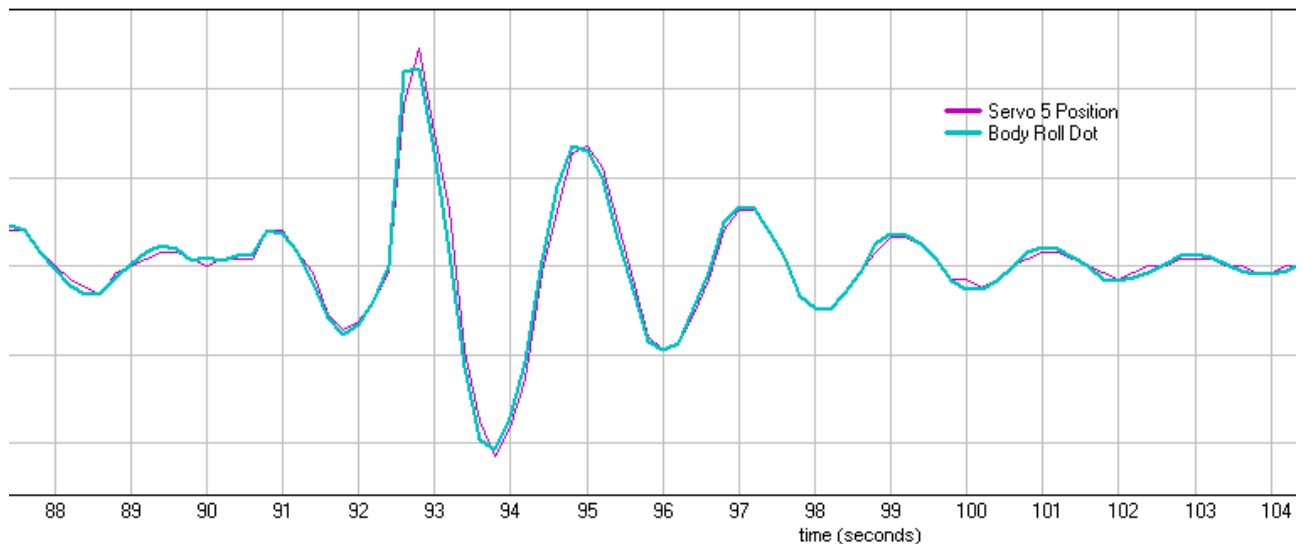
Puis, nous avons intégré des rate limiter à la sortie des moteurs car nous les avons ajouté dans le code comme expliqué dans le paragraphe II.

De même, nous avons mesuré le retard qu'il existe entre une commande et l'action effective de cette commande : nous avons mesuré un retard de 0.4s. Nous avons donc ajouté un retard dans la boucle d'asservissement ce qui déstabilise énormément notre modèle, c'est pourquoi nous avons choisi de créer un retard de seulement 0.1s à notre modèle.

La carte microPilot est équipée d'un système de log. Il est impossible de faire des tracés en temps réel mais par contre on peut « revivre » un vol. On a alors accès aux valeurs de nombreux paramètres. Cependant ces valeurs ne sont notées que toutes les 0,2s la précision n'est donc pas fameuse. De plus pour lire une variable autre que celle prévue il faut la transformer en une sortie servo. Or ces sorties sont échantillonnées sur 120 valeurs entre 32676 et -32676. Il est donc important de s'assurer d'utiliser toute la plage de variation en mettant des facteurs d'échelle assurant que le minimum de la valeur que l'on veut lire se retrouve à -32676 et le maximum à 32676. Cette précaution assure des mesures les plus précises possibles de part l'utilisation des 120 pas disponibles.

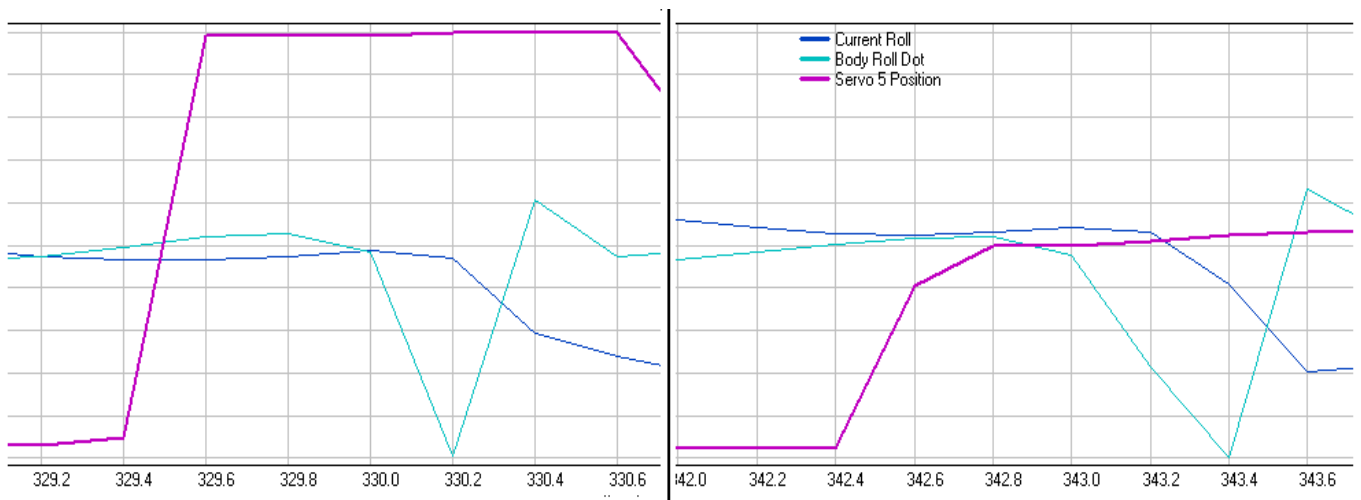
L'utilisation de ces logs nous a permis d'en savoir plus sur le comportement de la carte. Nous avons par exemple pu nous faire une idée du calcul des variables d'attitude et de vitesse de rotation en comparant la vitesse donnée par les gyroscopes à celle obtenue par une dérivation discrète des variables d'attitude.





Sur ce graphique, on retrouve en bleue la sortie en vitesse de roulis et en mauve la vitesse obtenue par dérivation discrète de l'angle de roulis. On remarque que les courbes se superposent parfaitement. Ainsi l'angle de roulis est calculé par intégration des valeurs des gyroscopes. On voit aussi que les deux signaux sont propres et non bruités.

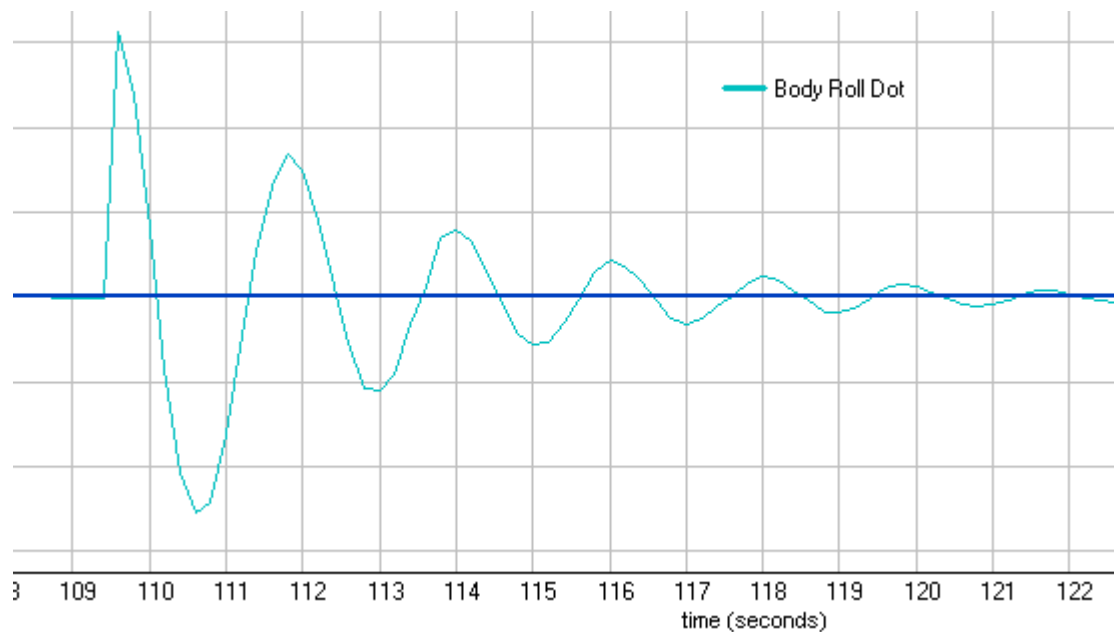
Nous avons aussi pu tester la vitesse de réaction de la partie puissance en mesurant le délai entre l'émission d'une commande de puissance et son effet sur le drone. Pour ce faire on n'alimente au banc qu'un moteur. On envoie alors un échelon au gaz et on mesure le temps entre l'envoi par la carte de la consigne moteur et l'apparition d'une vitesse angulaire. On utilise la vitesse angulaire plutôt que l'angle pour s'affranchir des retards dus à l'intégration.



On remarque sur la réponse obtenue qu'il s'écoule près 0,4 s entre la consigne et sa conséquence sur la vitesse de rotation. Et il se passe 0,2 seconde de plus pour que ceci est un impact sur l'angle. Nos prédécesseurs avaient trouvé des temps de réponse de moins de 0,2s. Et nous n'avions pas contrôlé ces résultats. Or les deux échelons ci-dessus donnent des retards de 0,4s et nous avons trouvé des cas de retard de 0,6s. A la vue de ces nouveaux résultats il paraît important de changer le système de carte et de variateurs. Des propositions sont formulées à la fin de la partie 5.

On ne peut malheureusement pas mesurer les délais entre une commande à l'émetteur et sa sortie de la carte. Cependant on se rend rapidement compte que bien que les temps de réponse interne à la carte soient raisonnables, il faut presque une seconde pour qu'après une commande des gaz on voit une réponse au niveau des moteurs. Il semble par conséquent que l'instance qui écoute les entrées soit cadencée à un rythme bien moindre que les 30Hz de la carte. Ce retard à la commande rend le drone totalement impilotable. Il est donc impossible lors d'un test d'aider l'asservissement. Le drone doit être autonome tout le temps et dès son premier décollage. Ceci a grandement retardé nos avancées.

La lecture des logs nous a aussi permis de confronter notre modèle au drone réel. En effectuant des mesures de fréquence propre de la structure.



On trouve ainsi une fréquence de 0,5 Hz.

## 2° Réglage des coefficients de notre correcteur

Nous avons dans un premier temps utilisé le banc d'essai qui permet de bloquer l'axe de tangage ou de roulis et ainsi s'affranchir des interactions entre les angles aérodynamiques. Nous avons réussi à stabiliser le quadrirotor selon ces deux axes sur le banc d'essais. Malheureusement lors des essais en vol, nous avons compris que notre banc d'essai retranscrivait très mal la réalité et que par conséquent il ne fallait pas s'affranchir du couplage des angles aérodynamiques. Nous pensons que des frottements trop importants sur le banc d'essais ne permettent pas d'utiliser ce procédé pour régler notre système.

Nous avons alors tenté un autre moyen de régler notre quadrirotor avec un nouveau banc d'essai : nous suspendions le quadrirotor au plafond grâce à une ficelle reliée au centre de gravité du drone. Lorsque nous avons réussi à le stabiliser quelque peu, nous avons repris les essais en vol pour tester la stabilisation ainsi déterminée. Malheureusement, encore une fois notre banc d'essai ne retranscrivait pas assez bien la réalité des essais en vol, nous observions toujours une instabilité en vol. Nous pensons que la tension de la ficelle a faussé nos réglages de stabilisation sur ce nouveau banc d'essai.

Nous sommes alors repartis de la base de réglage que nous avions avec les deux bancs d'essai et nous avons tenté de régler empiriquement le correcteur PID du drone. Nous implémentons des valeurs de gain PIDD<sup>2</sup>, puis nous testons le drone en vol. Après une observation de l'instabilité et la connaissance qualitative de l'action des différents gains, nous changions un coefficient qui permet une meilleure stabilisation. Le résultat final est presque atteint, néanmoins le drone ne peut se sustenter dix secondes. Il ne le fait que deux ou trois secondes si on le lâche à environ un mètre du sol à puissance suffisante. Voici ci-dessus un tableau récapitulatif des valeurs de coefficients de quelques essais significatifs.

Jour	N°essai	Pitch/Roll				Yaw				Commentaires
		P	D	D <sup>2</sup>	I	P	D	D <sup>2</sup>	I	
06/03/2008	1	20	50	7	1	5	15	5	0	très instable
	2	30	50	7	0	5	15	5	0	très instable
	3	25	50	7	0	5	15	5	0	instable
	4	20	40	7	0	5	15	5	0	amélioration mais toujours instable
	5	20	40	7	0	5	20	5	0	pas d'amélioration
modification du code et de la structure										
21/05/2008	1	10	6	5	0	0	5	0	0	trop sensible en lacet
	2	10	6	5	0	0	2	0	0	pas assez sensible en lacet
	3	10	6	5	0	0	3	0	0	oscillations un peu prononcées
	4	8	6	5	0	0	3	0	0	mieux - oscillations très lentes
	5	4	3	1	0	0	3	0	0	trop peu de correction
22/05/2008	1	8	10	5	0	0	3	0	0	un peu trop brusque
	2	8	10	2	0	0	3	0	0	pas grand intérêt
	3	8	10	8	0	0	3	0	0	très instable
	4	12	10	8	0	0	3	0	0	pas bon

### 3° Une autre solution : nouvelle modélisation de notre système

Une autre solution pour modéliser notre système serait de créer un modèle au fur et à mesure. En effet, une solution peut être de dégager un modèle du second ordre pour chaque angle. Avec une entrée échelon, il suffit d'observer la réponse du système et de tenter de se rapprocher de cette réponse avec un second ordre. On peut aussi modéliser les interactions entre chaque angle avec un modèle de ce type :

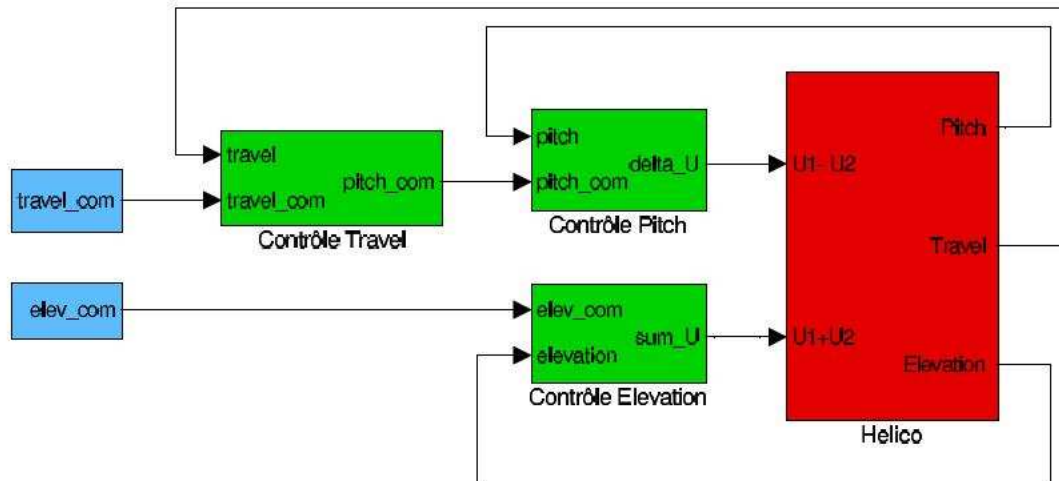


FIG. 3 – Structure de commande à boucles imbriquées

Une fois le modèle constitué et corrigé, on peut répéter l'opération et ainsi améliorer encore la stabilité et la rapidité du système, on peut travailler par échelon.

## V. Rétrospective du projet

### 1° Atteinte des objectifs fixés dans l'avant-projet

#### a/ Rappel de l'avant projet

Objectifs :

- Réussir à faire décoller le drone
  - o Réduction du poids
  - o Amélioration de la chaîne de puissance
  - o Simplification de l'allumage
- Stabiliser le drone : réaliser un vol stabilisé
  - o Amélioration du code
  - o Essai sur les coefficients
  - o Amélioration de la position du centre de gravité
  - o Utilisation du compas
- Réaliser une structure plus efficace
  - o Optimiser la perte au niveau des hélices
  - o Protéger l'électronique de l'humidité et des chocs
- Vérifier que le code est générique : adaptabilité sur d'autres structures similaires

Planning et solutions envisagées :

- Décembre : réussir à faire décoller le drone
  - o Amélioration de la chaîne de puissance du drone : utilisation de câbles plus adaptés et optimisation de la position des variateurs.
  - o Simplification de l'allumage : démarrer simplement le quadrirotor en supprimant les fils doublons et les interrupteurs.
  - o Réduction du poids : suppression de la partie haute de la structure.
  - o Faire un test de décollage sans toucher au code d'asservissement pour décembre.
- Mi – Avril : stabiliser le drone
  - o Amélioration de la position du centre de gravité : équilibrage de la structure en repositionnant les différents composants et en créant deux poches symétriques pour insérer les accumulateurs.
  - o Amélioration du code : épuration du code initial.
  - o Essai sur les coefficients : en se basant sur les coefficients déterminés l'an passé, nous allons utiliser une méthode empirique. Tout d'abord, nous testerons axe par axe (tangage et roulis) pour se rapprocher de la stabilisation, puis nous effectuerons des essais en vol car il y a de forts couplages entre les différents axes.

- Utilisation du compas : activation du compas pour asservir le drone en lacet, tout en faisant attention au couplage entre les axes.
- Faire un test stabilisé pour mi – avril.
- Mi – Avril : réaliser une structure plus efficace
  - Optimiser la perte au niveau des hélices : réduction de la taille des caches moteurs et inclinaison des hélices.
  - Protéger l'électronique de l'humidité et des chocs : création d'une coupole en plastique.
  - Recherche d'un meilleur matériau de base.
  - Construire cette nouvelle structure pour mi – avril.
- Juin : Vérifier que le code est générique
  - Déterminer des coefficients des correcteurs en fonction de la vitesse des moteurs et des angles.
  - Vérifier ce code sur les deux structures et présenter ces coefficients.

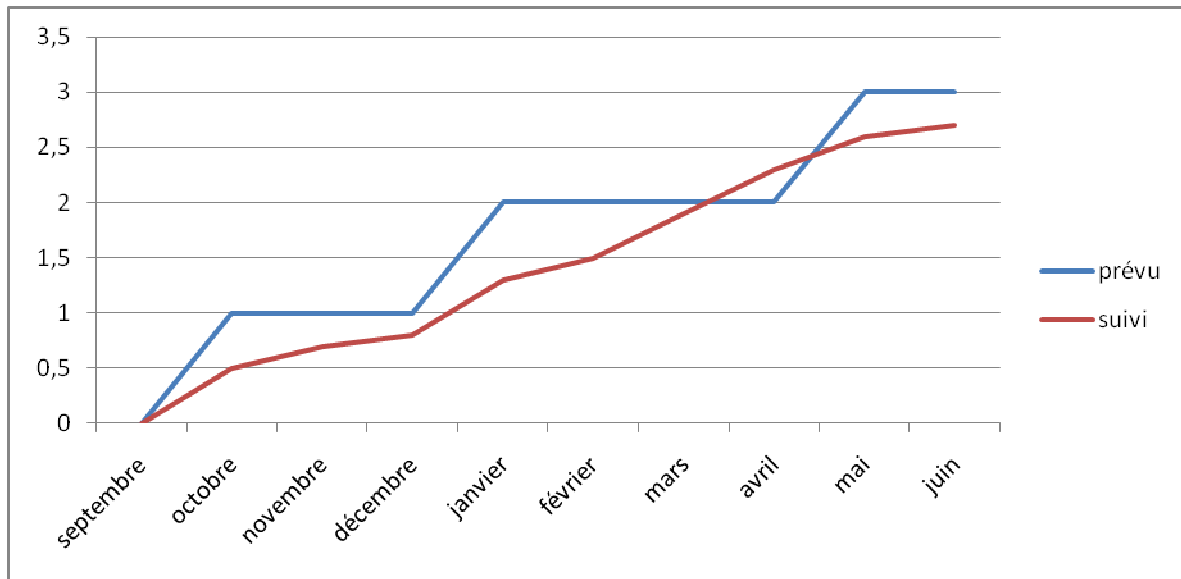
#### b/ Atteinte des objectifs

Nous avons atteint plus lentement que prévu le premier objectif. En effet, même il s'est avéré assez simple de réduire considérablement le poids de la structure en coupant des parties inutiles, nous avons eu du mal à simplifier la partie électronique. En effet, il a fallu supprimer une partie du signal inutile entre la carte MicroPilot et les variateurs (câble marron) pour se passer de tous les interrupteurs de démarrage. Nous avons également refait intégralement les câblages et avons utilisé des câbles plus adaptés à la puissance fournie par les moteurs. Avec ces améliorations, nous avons constatés que le drone avait désormais la puissance nécessaire pour décoller. Cependant, le drone s'est avéré complètement instable.

Il a donc fallu reprendre depuis le début la stabilisation du quadrirotor : vérifications des équations du mouvement du drone, étude du code de stabilisation et recherche des coefficients des correcteurs sur banc d'essai et lors d'essais en vol. Malheureusement, nous ne sommes jamais parvenus à trouver le bon correcteur pour stabiliser le drone. Cela pouvait avoir plusieurs explications possibles. Une première explication porte sur la structure du drone original qui reste, malgré nos améliorations, peu efficace (centre de gravité mal positionné, ...). La deuxième explication porte sur l'opacité de la carte MicroPilot.

Parallèlement, nous avons donc décidé de construire une nouvelle structure plus légère, et mieux finalisée (centre de gravité bien placé, ...). Nous avons pensé cette structure de façon à protéger l'électronique en cas de crash. De plus, nous avons surélevé les moteurs de façon à s'assurer la stabilité du drone. Enfin, nous avons tenu à réaliser un design assez joli. Cependant, il s'est avéré que, sur cette structure également, le pilotage est impossible. En effet, malgré tous nos efforts, le quadrirotor reste instable quelque soit la structure.

Il a donc été impossible d'atteindre tous nos objectifs. En effet, il nous semble difficile de stabiliser un drone quadrirotor à l'aide de la carte MicroPilot. Nous avons réussi à faire décoller le drone et avons presque réussi à le stabiliser mais pas suffisamment pour avoir un vol stabilisé de plus de quatre secondes.



Planning 1

## 2° Proposition de solutions pour améliorer le quadrirotor

Nos différents essais nous ont démontré que l'opacité de la carte était un frein énorme tant sur le plan software que sur le plan hardware avec le problème de continuité des masses expliqués dans la première partie. Il s'est par exemple avéré que lorsque nous rentrions des coefficients ceux-ci étaient multipliés par des constantes parfois négatives! Nous ne savons rien non plus des fréquences des différentes boucles internes. Une seule chose est certaine, la boucle de calcul tourne effectivement à 30Hz, nous l'avons testée. Mais qu'en est-il des autres boucles ? Par exemple la lecture des entrées se fait beaucoup moins souvent et on peut obtenir des retards dans les commandes radio de près d'une seconde. On ne peut rien connaître non plus des fréquences de lecture des capteurs et si elles sont aussi lentes que cette dernière, l'asservissement devient impossible. On ne sait quel est le traitement qui a lieu sur les mesures. Les groupes précédents utilisaient par exemple des valeurs d'attitude moyennées sur 0,2s et qui entraînaient autant de retard. Pour tous ces problèmes la carte MicroPilot n'est pas utilisable pour cette application. Il semble donc beaucoup plus intéressant pour les groupes suivants de penser au développement d'une carte personnelle ou encore d'envisager l'achat d'une carte du commerce dédiée à l'application drone quadrimoteur comme il en existe sur internet (notamment chez MIKROKOPTER).

Créer une nouvelle carte de A à Z est tout à fait possible et on trouve sur les forum (xBird notamment) quelques personnes qui en développaient sur des bases de micro contrôleur. Cependant cela risque d'être très long et fastidieux aussi nous conseillons à nos successeurs d'acquérir une carte du commerce, certaines valent moins de 100 euros. Puis de partir de cette base pour en améliorer les caractéristiques.

L'autre point noir de notre électronique se situe au niveau des variateurs et il serait intéressant de les changer. Cependant dans le monde du modélisme les temps de réponse des contrôleurs de moteurs brushings n'est pas une donnée inintéressante. Aussi aucun constructeur ne fournit ces valeurs. Il faudrait par conséquent se renseigner auprès des personnes qui utilisent des moteurs brushings sur leur drone. Ces personnes sont relativement rares car beaucoup préfèrent la rapidité de réponse des brushing aux rendement des brushings. Une autre solution serait de concevoir un tel contrôleur. Nous avons trouvé quelques schémas



détaillés sur internet et certains électroniciens prétendent, en jouant sur la dynamique du démarrage, obtenir des temps de réponse inférieurs au moteur classique. La encore ceci peut s'avérer difficile pour des débutants. Mener une campagne de tests sur différents types de variateurs peut être intéressante et pas forcément onéreuse. Notre contact au club drone de Thales, nous a affirmé avoir trouvé ses variateurs pour moins de 10 euros et ceux ci sont tout à fait adaptés vu que leur quadrirotor vole parfaitement.

Pour conclure, les prochains groupes n'auront pas à modifier la structure car celle-ci est robuste et fiable. En revanche les deux axes détaillés précédemment semblent très intéressant à développer. Il est très important de trouver une alternative à la carte MicroPilot qui est parfaite pour ce pour quoi elle a été conçue mais absolument inadaptée à notre application. Nous pensons de plus que se lancer à nouveau dans une tentative de détournement de la carte serait vain.

## VI. ANNEXES

### Annexe 1 : photos de l'ancienne et de la nouvelle structure

#### a/ L'ancien modèle



Figure 1 : Support moteur

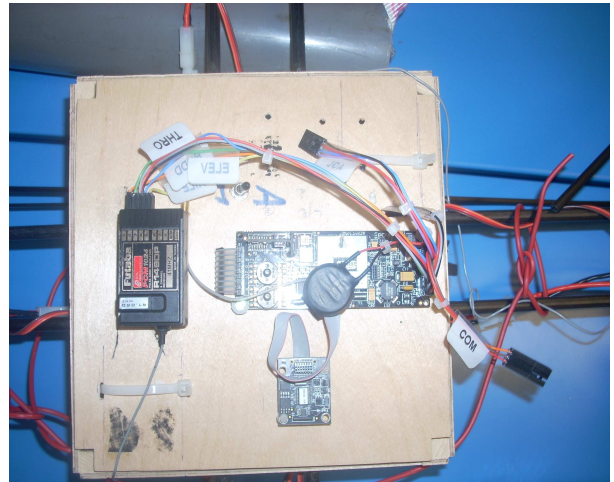


Figure 2 : plateforme électronique

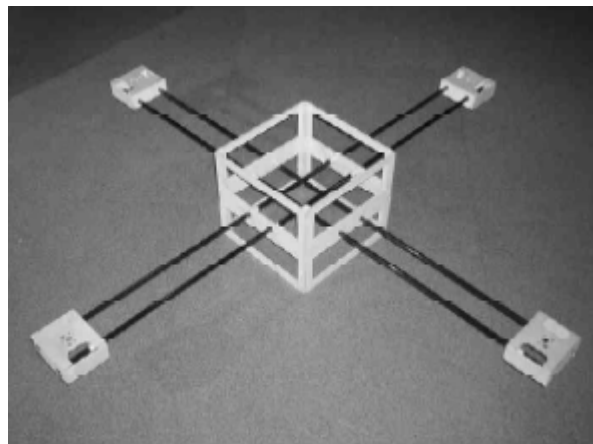


Figure 3 : structure

#### b/ Le nouveau modèle



Figure 4 : structure



Figure 5 : Support moteur

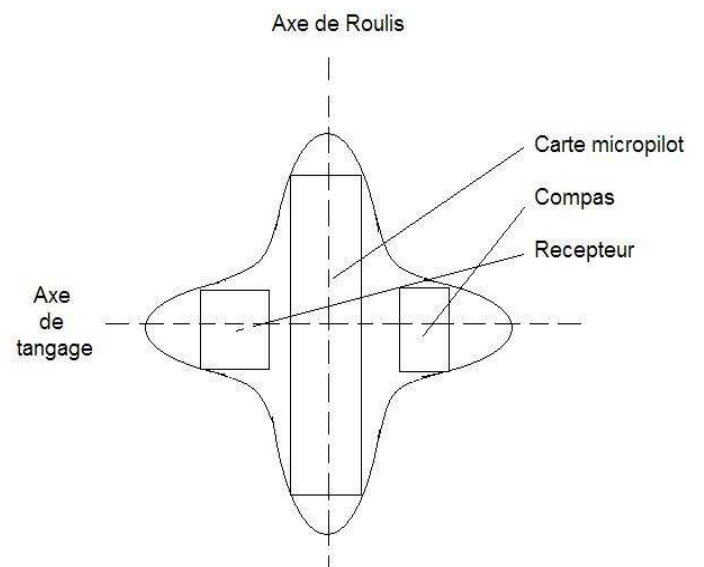


Figure 6 : Plateforme électronique

## Annexe 2 : les bancs d'essais

Effectuer des essais en vol avec un drone dont on n'est pas sûr qu'il soit stable comporte des risques. Le risque majeur est de briser la structure ou les hélices. Nous avons donc décidé de développer différents bancs d'essais pour asservir le drone sur ses différents mouvements. Nous avons utilisé deux types de bancs : un banc permettant d'étudier le mouvement de roulis ou de tangage (ces mouvements sont équivalents sur un quadrirotor) et un banc permettant d'étudier le mouvement de lacet.

Le groupe précédent avait fabriqué une plate forme de test pour ces essais. Cette plateforme est constituée de supports en contre plaqué et sont de forme triangulaire au sommet desquels est disposé un système de mors en aluminium articulés autour d'un axe en acier. Le tout est maintenu par des colliers d'électricien.



Cependant lors de nos essais sur la première structure nous nous sommes vite aperçus que ce banc d'essai ne traduisait pas du tout la réalité du vol. En effet l'amortissement dû au frottement rendait le drone stable au banc alors qu'il ne l'était pas en vol. Cependant au vue de la très grande instabilité du drone en vol il était temps de pendre des mesures. Nous étions donc obligés d'utiliser ce banc bien que fortement imparfait. Nous avons alors essayé d'en améliorer la représentativité. Dans un premier temps nous avons huilé tout les axes qui au vue de la rouille qui s'y était installée, ne l'avait jamais été. Il s'est ensuite avéré que ces derniers n'était pas tout à fait coaxiaux, dès lors quand nous serrions la structure, ceux ci se mettaient en travers et frottaient d'autant plus. Nous avons donc effectué nos mesures en serrant les bras le moins possible. C'est alors que nous avons réalisé que la charge de travail pour la stabilisation serait bien supérieure à ce que nous pensions, en effet le simple fait d'améliorer le banc de test avait rendu le drone instable au banc.

Avec la création de la nouvelle structure ce banc d'essais n'était plus utilisable en l'état. Il nous a cependant semblé être une bonne base, aussi nous avons uniquement modifié le système d'attache. Un système utilisant des mors n'était pas utilisable du fait du « dièdre » des bras. Nous avons donc simplement utilisé une corde tendue qui présentait l'avantage d'être réglable en hauteur. Il était ainsi facile de s'assurer que l'axe de fixation (ici la corde) passait réellement par le centre de gravité du drone. De plus du fait de la surélévation de tous les moteurs due au dièdre nous pouvions faire nos tests avec les quatre moteurs allumés. Ce support était très libre en rotation du fait de la très faible tension de rappel de la corde, il faut ajouter à cela que les tests se faisaient à une puissance assurant la sustentation du drone par conséquent la tension de la corde en test était encore plus faible qu'au repos.



Le deuxième banc d'essais est un banc que nous avons créé afin de tester le mouvement en lacet du drone. Nous avons décidé de suspendre le banc à une corde, afin qu'il ne touche pas le sol, et de lui attacher une masse, afin d'être sûr que le drone n'ait aucun mouvement sur l'axe de roulis ou de tangage. Ainsi, nous avons pu étudier le mouvement de lacet à différents régimes moteurs.

En jouant sur les coefficients des correcteurs, nous avons pu stabiliser le drone en lacet.

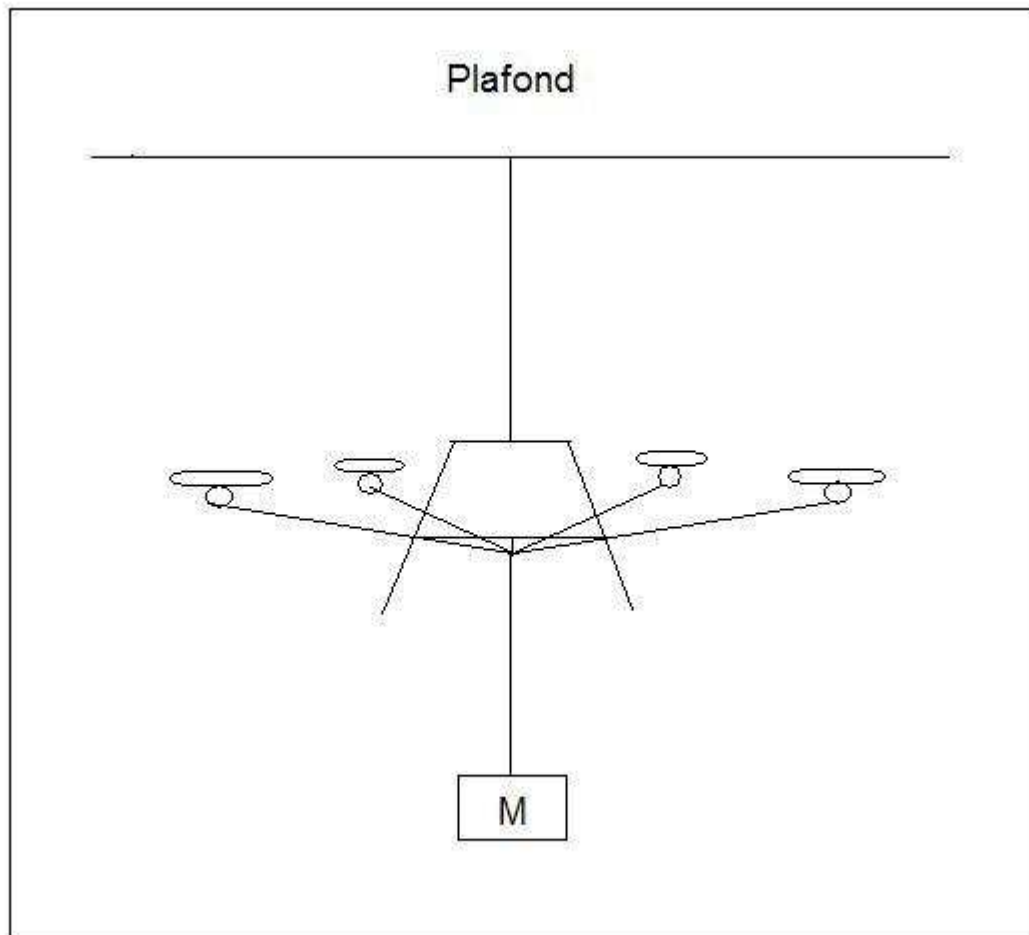


Figure 2 : 2ème banc d'essais

Le problème de ce banc est que la corde fait subir au drone une force vers le haut qui modifie sa dynamique. Cependant, le résultat obtenu lors des essais en vol était satisfaisant au niveau du mouvement en lacet.

### Annexe 3 : Matériel électronique et électrotechnique utilisé

#### a/ Les moteurs

Le principe de vol d'un quadrirotor impose d'avoir deux hélices contrarotatives et deux moteurs qui tournent dans le sens opposé aux deux autres. Les moteurs mis à notre disposition sont de type brushless (référence : brushless typhon-micro 6-23), qui ont l'avantage d'être petits, légers et puissants. Ce modèle de moteur, de puissance 90 W, peut être alimenté sous une tension variant de 7,2 V à 14,4 V. Il pèse 42g et, à la base, convient à un engin de 250 à 700 grammes.

Notre drone pèse 816g, ces moteurs ont donc suffisamment de puissance pour sustenter le quadrirotor.



Figure 1 : moteurs

#### b/ Les variateurs

L'un des inconvénients de nos moteurs est leur mise en œuvre : ils doivent obligatoirement être associés à des régulateurs particuliers, appelés contrôleurs ou variateurs, qui nécessitent une programmation et une temporisation bien précise.

Le variateur est l'interface entre la commande de signal et le circuit de puissance du moteur. Il est relié à un accumulateur (ici de type Li-Po) qui fournira la puissance nécessaire au moteur. Les variateurs sont des éléments qui chauffent beaucoup. Par conséquent nous avons décidé de les placer sous le flux des hélices.



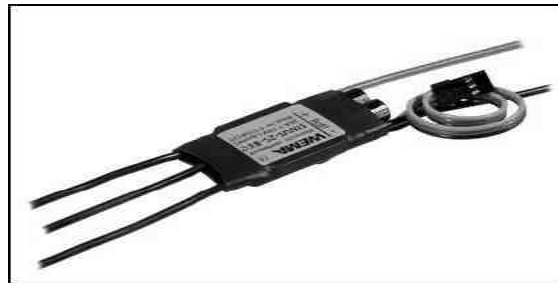


Figure 2 : variateur

### c/ Les hélices

Pour éviter que le drone tourne sur lui même, il est nécessaire d'utiliser deux différents types d'hélices : des hélices classiques et des hélices dites propulsives ou contrarotatives. Nous avons testé de nombreux types d'hélices différents :

- 9x6
- 10x6
- 11x6
- 10x4.5
- 10x7

Les tests de portance ont révélé que les hélices 10x4.5 étaient les plus performantes avec nos moteurs Brushless. De plus, seules les hélices Max-Prod 10x4.5 ont donné des résultats satisfaisants, car elles sont bien adaptées à nos moteurs. Les autres hélices sont plus lourdes, plus rigides et elles entraînent une vibration de la structure.



Figure 3 : hélices

#### d/ Les batteries

Nous avons eu à notre disposition deux types d'accumulateurs : les batteries Li-Po 2000 servant à alimenter les moteurs et une batterie 4.8V servant à alimenter la carte Micropilot.

Cependant, dans la notice de la carte Micropilot, il est spécifié que l'on peut alimenter la carte Micropilot par une tension inférieure à 9V. On a donc décidé de se passer de l'accumulateur 4.8V, ce qui entraîne un gain de masse de plus de 100g. Cela a eu pour conséquence une procédure de démarrage légèrement plus complexe. Cette procédure est décrite dans l'annexe 7.



Figure 4 : Accumulateur 4.8V



Figure 5 : Accumulateur Li-Po 2000

#### e/ La carte Micropilot

Il a été décidé que le drone quadrirotor serait stabilisé à l'aide d'une carte Micropilot MP2028g. Il s'agit d'une carte conçue pour le contrôle de modèles réduits d'avions. Elle est composée d'un microprocesseur programmable, de systèmes de mesure (gyroscopes, prises de pression, GPS), et d'entrées sorties utiles à la commande du drone (prises pour récepteur de télécommande, carte fille de sorties vers des servocommandes). Cette carte peut évoluer par l'adjonction de cartes filles qui lui donnent des capacités de gestion d'un flux vidéo, ou de communication vers un PC via un radio modem. On peut aussi lui ajouter d'autres cartes de sorties pour d'autres servocommandes.

On peut communiquer avec la carte via le port COM relié par un câble série au PC. On peut ainsi récupérer différentes sorties de la carte ou implémenter celle ci. En effet, il est possible de créer un nouveau code de commande plus adapté à un drone quadrirotor.

Cependant cette solution n'est pas aussi pratique que la programmation d'un microcontrôleur. En effet, nous ne pouvons accéder qu'à une partie du code, et Micropilot

conservera toujours un noyau non modifiable qui pourra perturber le bon fonctionnement du drone.

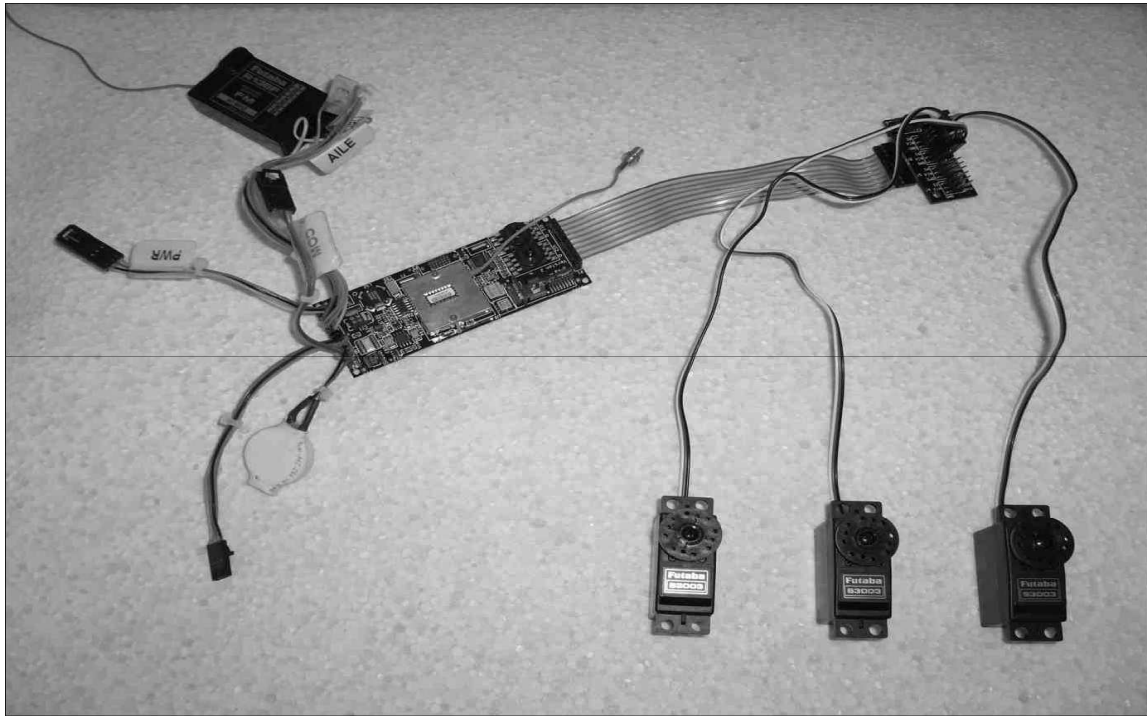


Figure 6 : La carte Micropilot

#### f/ Le récepteur et l'émetteur

La radiocommande utilisée est une radio Futaba (référence : 9CAP/9CHP) de neuf voies. Comme le drone quadrirotor n'a que quatre types de mouvements possibles, quatre voies sont utilisées pour diriger le drone. Pendant les essais, nous avons également utilisé trois voies pour modifier en temps réels les correcteurs et avons laissé la dernière voie libre pour la commande des gaz..

Pour que la carte Micropilot reçoive les ordres de l'émetteur, nous utilisons un récepteur Graupner R700 branché à la carte.



Figure 7 : récepteur



Figure 8 : radiocommande

g/ Le compas

Le compas est un compas magnétique dont l'objectif est de fixer le cap à suivre dans le cas d'un vol autonome.

## Annexe 4 : Matériel informatique utilisé

Nous avons utilisé plusieurs logiciels lors de ce projet :

- Des logiciels permettant de modifier le code de la carte Micropilot et de contrôler ce code. Il s'agit de l'Hyper terminal, Micropilot Horizon et de Log Viewer
- Le logiciel Matlab afin de faire du calcul formel et le logiciel Simulink pour créer un modèle du drone dans le but de l'asservir

### a/ l'Hyper terminal

La carte communique avec l'ordinateur via un port série. Il est possible, via un logiciel tel que HyperTerminal ou RealTerm, de lire et d'écrire des données sur la carte. Ce moyen permettra d'analyser l'état des variables systèmes. Il s'agit des variables décrites dans l'annexe du manuel Micropilot. Pour ce faire, il faut configurer le logiciel pour lire les données du port série à un débit de 9600 bauds. Sur l'ordinateur IBM «microdrone», le port série étant absent, on branchera l'adaptateur sur le port USB du haut. Il s'agit du port COM1 pour lequel RealTerm est déjà configuré (raccourci sur le bureau).

Il suffit ensuite de brancher la carte et d'attendre la fin de son initialisation qui dure environ 20 secondes. Cette phase peut être plus longue si le GPS n'est pas désactivé. On peut exécuter un démarrage de la carte en désactivant le GPS grâce à la commande « ffff ».

La configuration du système se fait via le menu, accessible en tapant « qqqq ». Une fois dans le menu, différentes commandes permettent d'agir sur le système, par exemple :

H	Lister les commandes disponibles
A	Ajuster manuellement un servo. Il faut ensuite taper le numéro du servo, puis taper – ou + pour l'ajuster
X	Lire un paramètre
P	Ecrire un paramètre
L	Lister les paramètres

Les paramètres à lire ou écrire sont appelés par leur numéro, détaillés dans l'annexe du manuel Micropilot.

Voici une liste des numéros utiles :

- 1231 fServo5 : commande du moteur 2
- 1232 fServo6 : commande du moteur 3
- 1233 fServo7 : commande du moteur 4
- 1234 fServo8 : commande du moteur 1
- 1153 Commande de roulis de la télécommande
- 1152 Commande de tangage de la télécommande
- 1154 Commande de lacet de la télécommande
- 1155 Commande de gaz de la télécommande
- 5451 Spécifie si le UserPid 1 est actif (=3 si actif)
- 117 Activer ou désactiver le capteur à ultrasons
- 157 Définit le niveau critique de la batterie de la carte Micropilot

- 155 Définit le niveau critique de la batterie du ServoBoard
- 1057 Mesure de tangage
- 1060 Mesure de lacet
- 1059 Mesure de roulis
- 1403 Mesure de vitesse en tangage
- 1404 Mesure de vitesse en roulis
- 1405 Mesure de vitesse en lacet
- 116 Activer la télémétrie

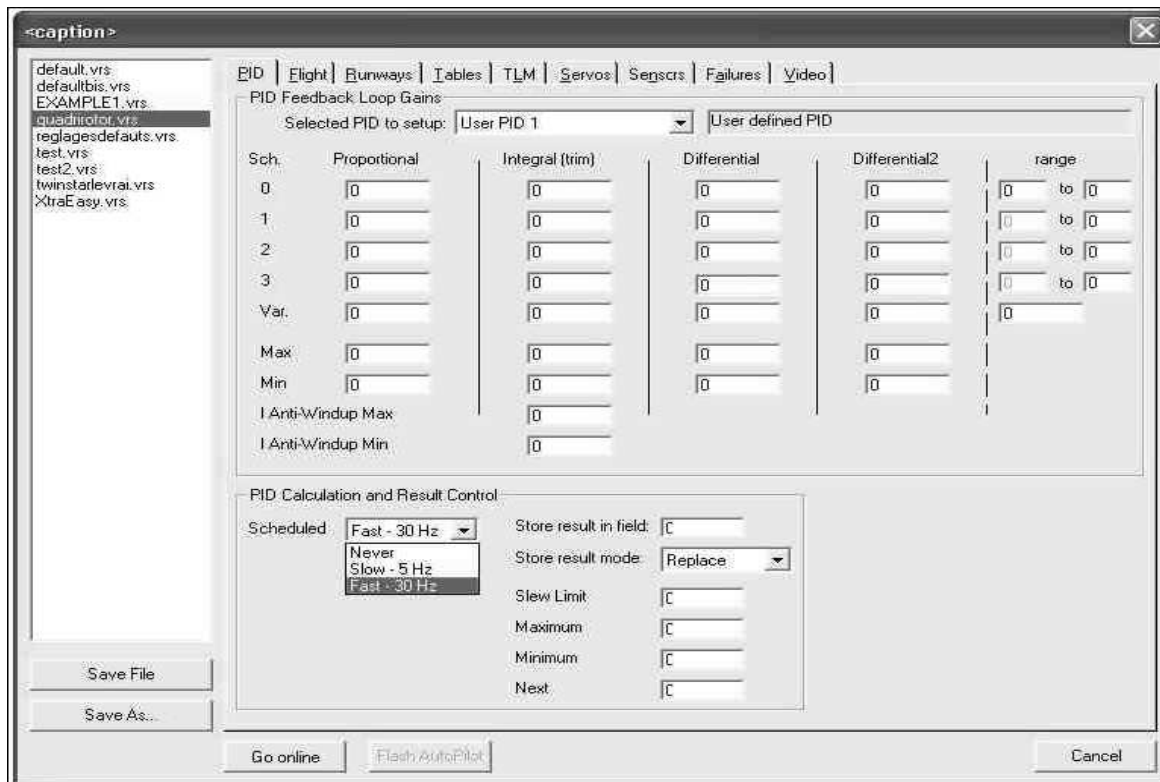
On peut ainsi configurer complètement le drone, voire le contrôler puisque l'on peut faire bouger les servos. On pourrait ainsi imaginer un programme simple en java de configuration du drone, qui enverrait ces données au port série, et donc modifier les coefficients des correcteurs.

Cependant, cette interface possède un défaut majeur : le programme de la carte est stoppé lorsque l'on rentre dans le menu de configuration. Il faut donc pouvoir configurer, puis quitter le menu pour reprendre le fonctionnement du programme. Et c'est là que réside le problème : il n'y a pas de commande pour quitter le menu ! La seule solution est donc d'éteindre la carte et de la rallumer, ce qui est une perte de temps. C'est pourquoi nous utilisons plutôt le logiciel Horizon.

#### b/ Micropilot Horizon

Horizon est un logiciel fournissant une interface agréable de configuration et de contrôle des états de Micropilot. De plus, Horizon permet de configurer le système, à l'instar de l'Hyper terminal. Les données de configuration sont sauvegardées dans des fichiers portant l'extension « .vrs ». La liste des fichiers de configuration est disponible dans la liste déroulante à gauche de l'interface. Un double clic sur quadrirotor.vrs ouvre par exemple la configuration sauvegardée du drone. Les deux flèches à droite de la liste permettent respectivement de charger les données sauvegardées sur le pc vers le drone, ou de sauvegarder les données du drone sur le pc.

Ces fichiers permettent donc d'enregistrer des configurations différentes. Jetons un œil sur le contenu d'un fichier vrs. Lorsqu'on double clique sur quadrirotor.vrs, on obtient cette fenêtre :



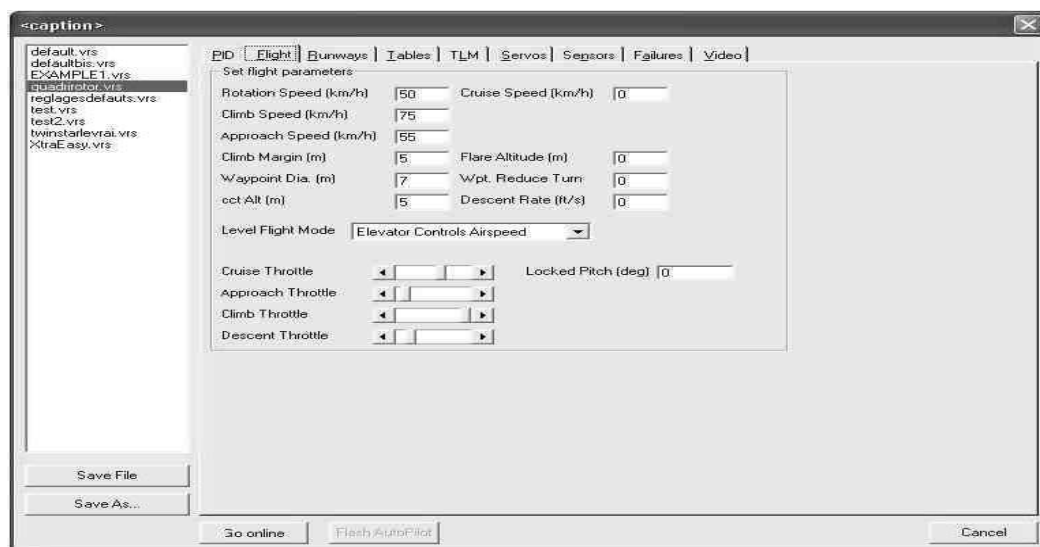
Cet écran regroupe la liste des paramètres à modifier, regroupés en onglets :

- Le premier onglet, « PID » contient notamment les paramètres de configuration des user PID. On peut ainsi définir le User PID 1 à 30Hz (cf. écran). Ainsi le code contenu dans User PID 1 sera exécuté 30 fois par seconde. L'action effectuée est la même que si on mettait à 3 la variable 5451 sous HyperTerminal. C'est la seule chose qui nous intéresse dans cet onglet. On s'assurera également que les autres User PID sont désactivés.
- L'onglet Sensors permettra de désactiver le GPS, la sonde à ultrasons (appelée AGL), et de définir les niveaux critiques des batteries Micropilot et servoboard. Lorsque les batteries alimentant ces éléments auront atteint un voltage inférieur à celui proposé, la carte générera un code d'erreur. On ne pourra alors plus exécuter le code utilisateur dans ces conditions. Lors des tests, on devra donc alimenter les éléments avec une tension supérieure à ces seuils. Il faut noter que les seuils sont exprimés en centivolts (400=4 volts).
- L'onglet Flight est particulièrement intéressant. Nous remarquons en effet qu'il est impossible d'accéder sous Horizon aux variables utilisateurs. Ces variables auraient été bien pratiques pour stocker les coefficients de réglage du correcteur. En revanche, l'onglet Flight regroupe un certain nombre de variables (Rotation Speed, Climb Speed...) qui ne nous sont d'aucune utilité. Nous allons donc les détourner de leur usage premier pour régler les coefficients du correcteur.



Nom sous Horizon	Nom dans le code	Fonction
Rotation Speed	K1roll	Correction proportionnelle roulis et tangage
Climb Speed	K2roll	Correction dérivée roulis et tangage
Approach Speed	K3roll	Correction dérivée 2 fois roulis et tangage
Climb Margin	K1yaw	Correction proportionnelle lacet
Waypoint Dia	K2yaw	Correction dérivée lacet
Cct Alt.	K3yaw	Correction dérivée deux fois lacet
Flare Altitude	K4roll	Correction intégrale roulis et tangage
Descent Rate	K4yaw	Correction intégrale lacet

Pour modifier ces coefficients et ainsi régler la stabilisation, il suffit de brancher le drone, d'ouvrir l'écran quadrirotor.vrs dans Horizon, puis dans l'onglet Flight de cliquer sur « Go Online » (le bouton devient vert si tout se passe bien). De cette manière, les données proposées dans les champs sont celles chargées dans le drone. Il faut alors modifier les champs correspondant aux coefficients correcteurs, puis de cliquer sur « Flash Autopilot » pour les charger dans le drone.



### c/ Log Viewer

LogViewer est un utilitaire qui permet de lire et tracer les données enregistrées au cours d'un vol. Son icône est sur le bureau, et les logs sont enregistrés dans le dossier log, également sur le bureau.

Tout d'abord, il faut s'assurer qu'aucun programme ne monopolise le port série (HyperTerminal, Horizon...). Puis dans LogViewer, faire « File > Read Log File from Autopilot ». Le fichier correspondant au vol précédent est chargé. Dans l'onglet « Log Data », il suffit alors de double cliquer sur les titres des données à afficher (l'indicatif « visible » apparaît alors) pour tracer l'évolution de ce paramètre dans « Log Plots ». On peut alors zoomer, puis imprimer la courbe. Les fichiers log sont sauvegardés dans le répertoire

correspondant en fonction de l'heure et de la date, mais on peut changer leur titre pour les retrouver facilement.

#### d/ Matlab et Simulink

Nous avons utilisé le logiciel Matlab pour faire du calcul formel, afin de déterminer les équations de mouvement du drone.

Puis nous avons utilisé Simulink afin d'éditer un modèle du quadrirotor le plus proche possible de la réalité. Ce modèle avait pour but de nous permettre de tester différents correcteurs et de trouver approximativement les coefficients des correcteurs retenus afin de gagner du temps lors des essais sur banc puis en vol.

## Annexe 5 : Modèle Simulink de l'asservissement

### a/ Equations du mouvement du drone

- Choix des coordonnées  
Nous avons choisi d'utiliser les coordonnées aéronautiques

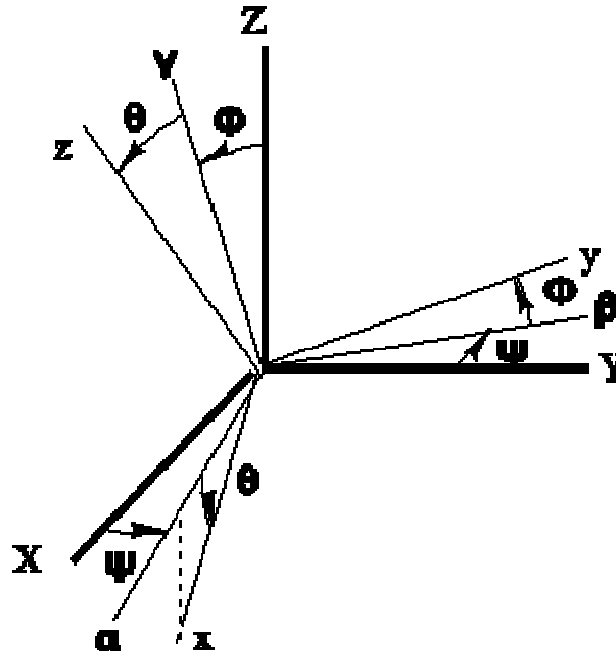
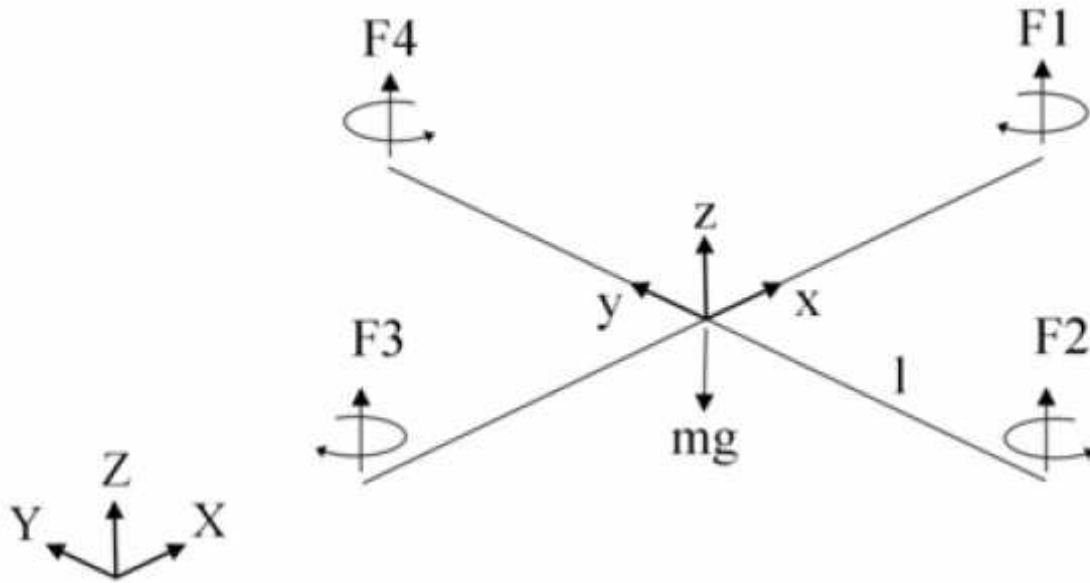


Figure. Axes aéronautiques

- Hypothèses du modèle

Le quadrirotor est constitué de quatre barres de carbone parallèles placées en croix aux extrémités desquelles sont placés les moteurs commandant les quatre rotors d'axe vertical. La masse du drone est de 1 kg pour une envergure de 1 m. La rotation des quatre rotors fournit une force verticale, la portance, et un couple de lacet dû à la traînée. Au centre se trouve tout l'équipement nécessaire au bon fonctionnement du drone : carte micro-pilote, batteries, récepteur, gyrocompas.

On choisit la convention d'axes suivante :



### Géométrie du drone

Ainsi pour contrôler le roulis du drone, il suffit d'agir sur les vitesses de rotations des moteurs 2 et 4. De la même manière, le tangage du drone est contrôlable en agissant sur les vitesses des moteurs 1 et 3. Les moteurs 1 et 3 tournent en sens inverse des moteurs 2 et 4 afin que, globalement, le couple d'anti-rotation généré par la traînée soit nul.

De plus, les axes des moteurs ne sont pas parfaitement verticaux : ils sont légèrement inclinés vers le centre du drone (environ  $15^\circ$ ) afin de garantir un retour à la position d'équilibre en cas de faibles perturbations.

On considère les hypothèses suivantes :

- La structure en carbone est supposée rigide.
- La structure est parfaitement symétrique.
- La portance et la traînée sont proportionnelles au carré de la vitesse de rotation des moteurs.
- On ne prend pas en compte la vitesse de rotation des rotors par rapport au sol.

#### • Calcul de la vitesse

On commence par déterminer la matrice de rotation totale du système dans le repère aéronautique :

$$R(\phi, \theta, \psi) = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \sin \phi \cos \psi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}$$

Soit  $\left[ \vec{X}, \vec{Y}, \vec{Z} \right]$  une base orthonormée constituant un repère fixe. Si le solide subit trois rotations successives selon les angles aéronautiques, on a alors :

$$\mathbf{r}_{x,y,z}(x, y, z) = R(\phi, \theta, \psi) \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Par dérivation, de chaque terme de ce vecteur, on en déduit facilement les trois termes de vecteur vitesse. Au final, la norme au carré de la vitesse s'écrit en prenant en compte les moments d'inertie :

$$\begin{aligned} v^2(x, y, z) = & (y^2 + z^2) \left[ \dot{\psi}^2 \sin^2 \theta - 2 \dot{\psi} \dot{\phi} \sin \theta + \dot{\phi}^2 \right] \\ & + (x^2 + z^2) \left[ \dot{\psi}^2 \sin^2 \phi \cos^2 \theta + 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \cos^2 \phi \dot{\theta}^2 \right] \\ & + (x^2 + y^2) \left[ \dot{\psi}^2 \cos^2 \phi \cos^2 \theta - 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \sin^2 \phi \dot{\theta}^2 \right] \\ & + 2xy \left[ \dot{\psi}^2 \sin \phi \sin \theta \cos \theta + \dot{\psi} (\dot{\theta} \cos \phi \sin \theta - \dot{\phi} \sin \phi \cos \theta) - \dot{\theta} \dot{\phi} \cos \phi \right] \\ & + 2xz \left[ \dot{\psi}^2 \cos \phi \sin \theta \cos \theta + \dot{\psi} (-\dot{\theta} \sin \phi \sin \theta - \dot{\phi} \cos \phi \cos \theta) - \dot{\theta} \dot{\phi} \sin \phi \right] \\ & + 2yz \left[ \dot{\psi}^2 \sin \phi \cos^2 \theta \cos \phi + \dot{\psi} (\dot{\theta} \sin^2 \phi \cos \theta - \dot{\theta} \cos^2 \phi \cos \theta) - \dot{\theta}^2 \sin \phi \cos \phi \right] \end{aligned}$$

- Obtention des équations du mouvement par la méthode de Lagrange

La méthode de Lagrange est basée sur l'énergie cinétique T, qui se calcule facilement à l'aide du carré de la vitesse :  $T = \frac{1}{2} m v^2$ .

On obtient alors l'expression de l'énergie cinétique :

$$T = \frac{1}{2} I_x (\dot{\phi} - \dot{\psi} \sin \theta)^2 + \frac{1}{2} I_y (\dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta)^2 + \frac{1}{2} I_z (\dot{\theta} \sin \phi - \dot{\psi} \cos \phi \cos \theta)^2$$

Il est alors possible de former les trois lagrangiens en utilisant la formule de dérivation :

$$\frac{d}{dt} \left( \frac{d}{d\dot{q}} \right) - \frac{d}{dq} . \text{ Par ailleurs, on fait l'hypothèse de petits angles d'où : } \cos \approx 1 \text{ et } \sin \approx 0 .$$

$$\Gamma_{\phi} = \dot{\omega}_x I_x + (I_z - I_y) \omega_y \omega_z$$

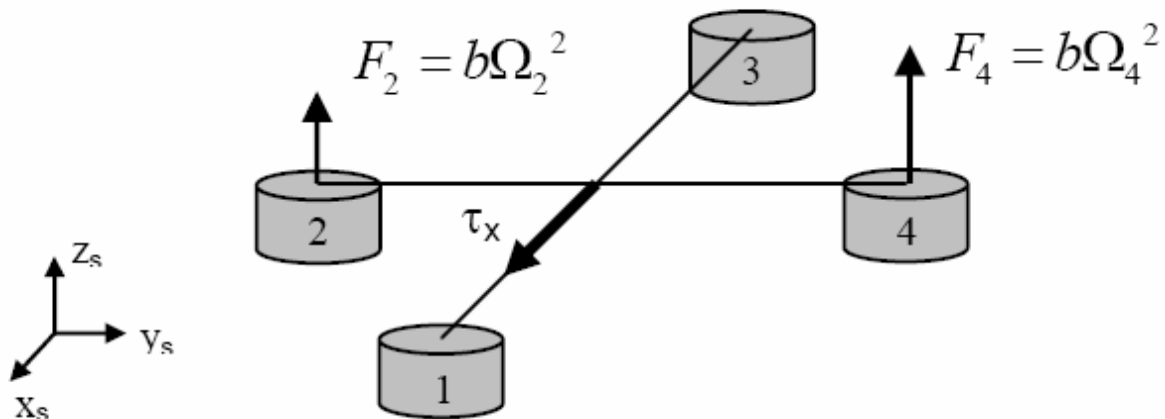
$$\Gamma_{\theta} = I_y \dot{\omega}_y + (I_x - I_z) \omega_x \omega_z$$

$$\Gamma_{\psi} = I_z \dot{\omega}_z + (I_y - I_x) \omega_x \omega_y$$

Il faut maintenant exprimer les forces non conservatives et les égaliser aux lagrangiens trouvés.

⇒ la portance crée des couples en direction des axes X et Y :

⇒

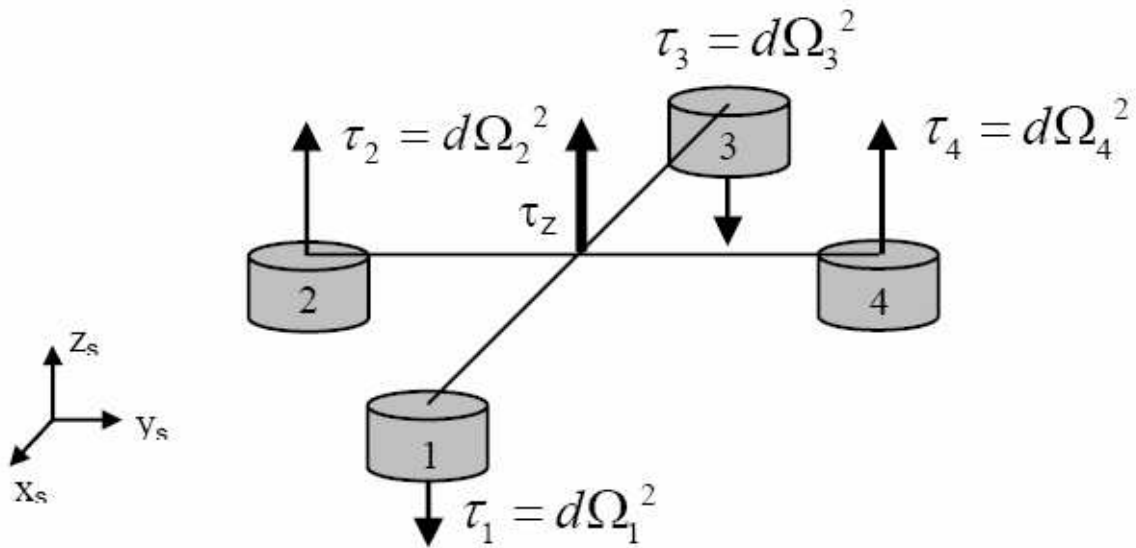


*Moment dû à la poussée des moteurs*

$$\tau_x = bl(\Omega_4^2 - \Omega_2^2) \text{ et } \tau_y = bl(\Omega_3^2 - \Omega_1^2)$$

Où b et l sont respectivement une constante unissant la poussée et la vitesse de rotation d'un moteur, et la demie envergure du drone.

⇒ la traînée des hélices crée un couple vertical :

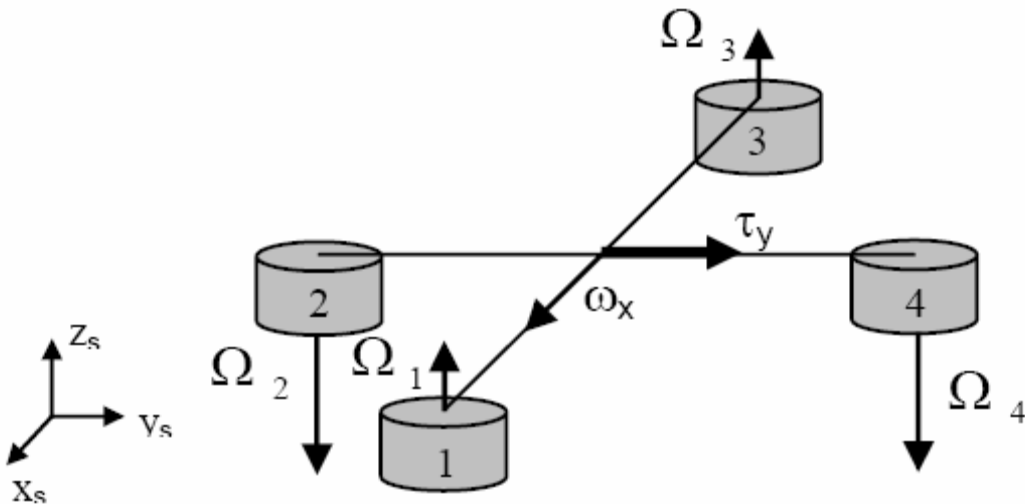


*Moment dû à la traînée des hélices*

$$\tau_z = d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)$$

Où d est une constante reliant la traînée et la vitesse de rotation d'un moteur.

⇒ effets gyroscopiques dus aux hélices lors d'une rotation autour de l'axe X ou Y:



*Effet gyroscopique*

$$\tau_x = I_{rotor} \omega_y (\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4) \text{ et } \tau_y = I_{rotor} \omega_x (-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4)$$

On peut alors obtenir les équations du mouvement par identification :

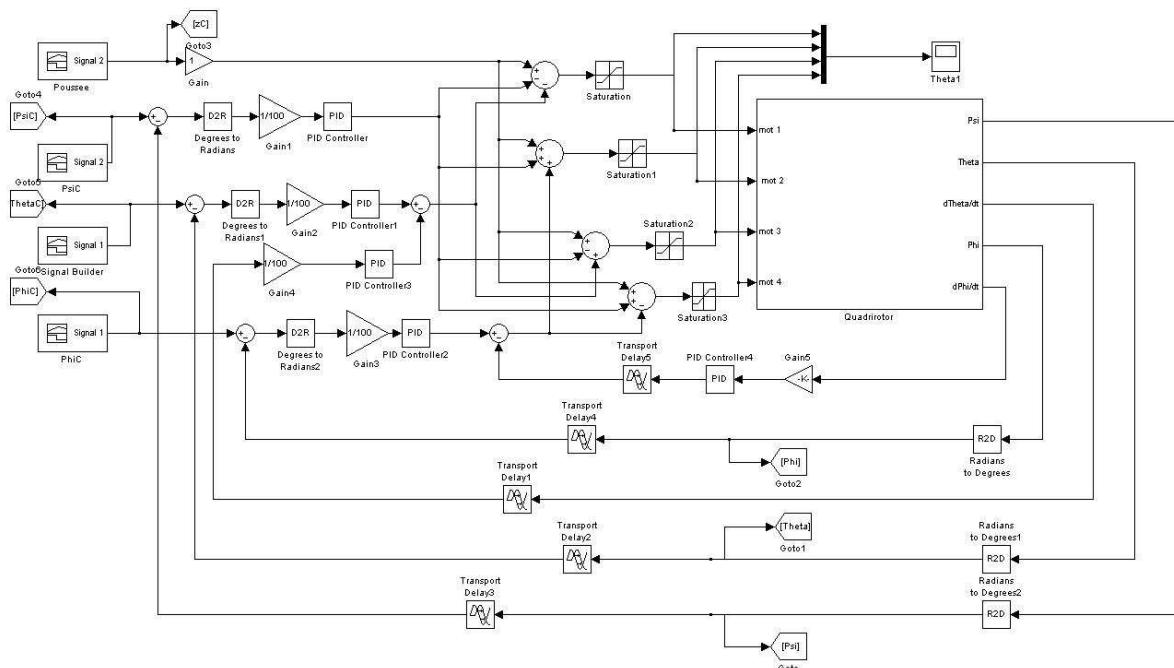


$$\begin{aligned}
 \ddot{\phi} &= \frac{I_{rotor}}{I_x} \dot{\theta} (\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4) + \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} + \frac{bl}{I_x} (\Omega_4^2 - \Omega_2^2) \\
 \ddot{\theta} &= \frac{I_{rotor}}{I_y} \dot{\phi} (-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4) + \frac{I_z - I_x}{I_y} \dot{\phi} \dot{\psi} + \frac{bl}{I_y} (\Omega_3^2 - \Omega_1^2) \\
 \ddot{\psi} &= \frac{I_x - I_y}{I_z} \dot{\theta} \dot{\phi} + \frac{d}{I_z} (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)
 \end{aligned}$$

### b/ Modèle Simulink associé

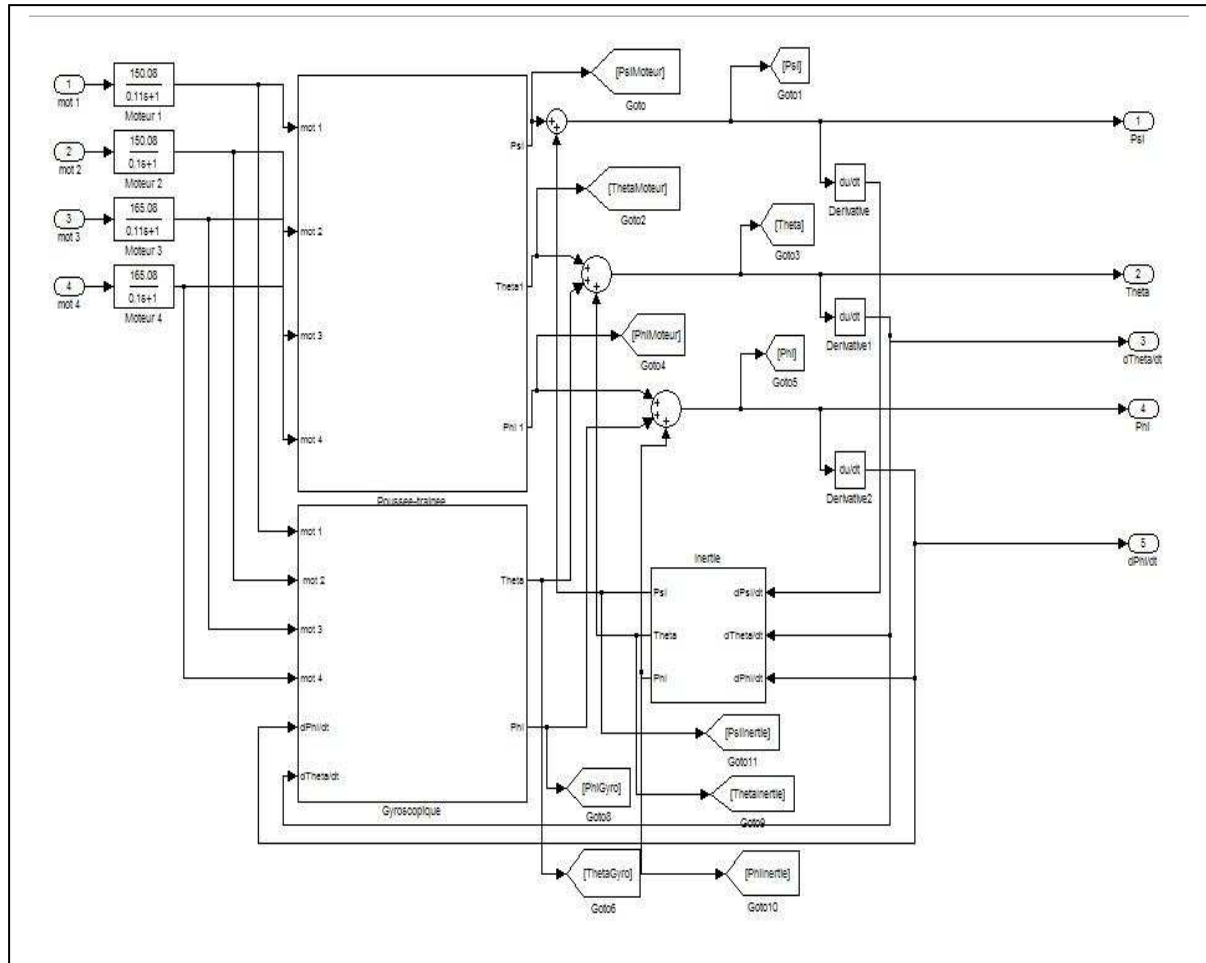
Nous avons obtenu à partir des équations du mouvement un modèle du drone. Nous avons apporté quelques modifications au modèle initial afin qu'il soit plus proche de la réalité :

- Mise en place de rate limiter
- Prise en compte d'un retard de 0.1s dans la boucle d'asservissement
- Correction avec un PIDD<sup>2</sup>



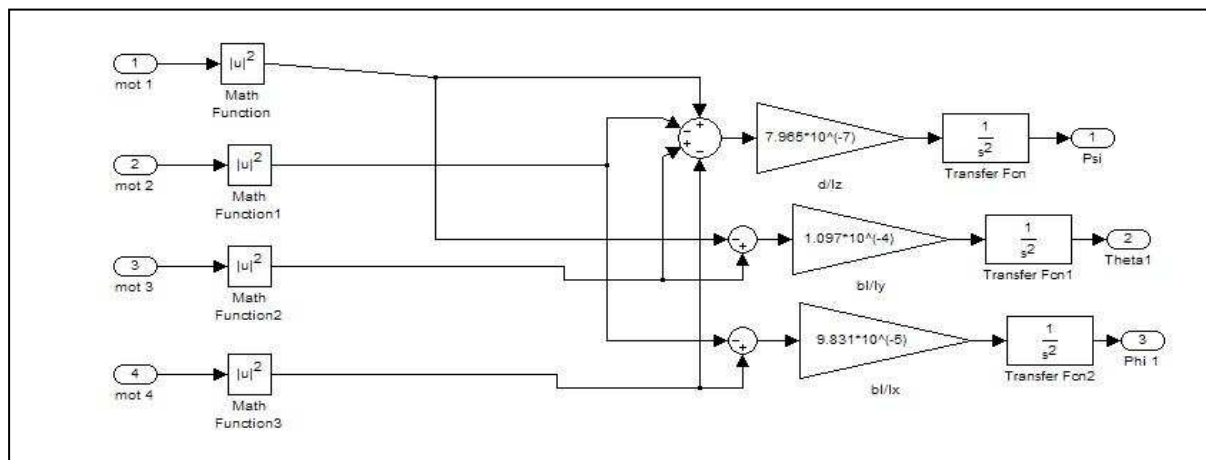
*Modélisation Simulink du système de correction*

L'entrée de notre modèle Simulink est la vitesse de rotation des moteurs avec une fonction de transfert due aux moteurs et qui avait été mesuré par un autre groupe auparavant. A travers la combinaison des 3 effets principaux agissant sur le système, nous obtenons en sortie la position des 3 angles aérodynamiques.

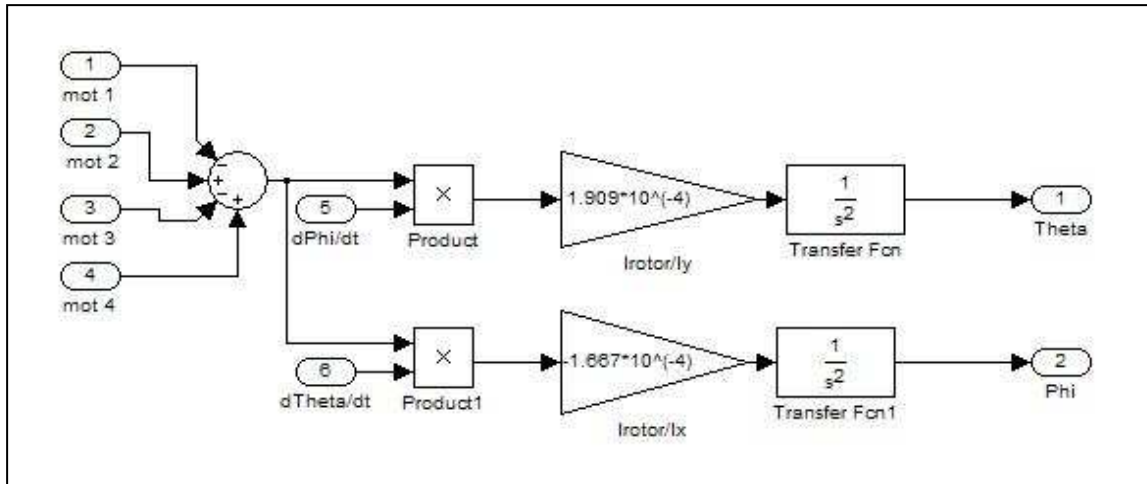


*Modélisation Simulink du quadrirotor*

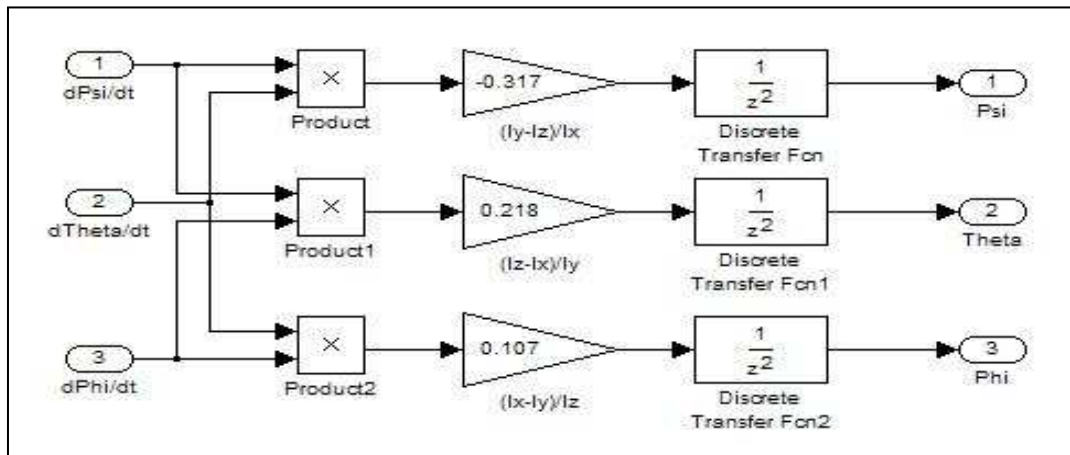
On peut distinguer trois types d'actions sur notre quadrirotor : l'effet gyroscopique, l'effet d'inertie et l'effet de la poussée et de la traînée. Nous avons du linéariser ces effets en considérant des petits angles notamment.



*Modélisation Simulink de la poussée et de la traînée*



*Modélisation Simulink des effets gyroscopiques*



*Modélisation Simulink des effets d'inertie*

## Annexe 6 : Code implémenté dans la carte MicroPilot

```
#include "usercode.h"
#include "simdll.h"
#include "mpfields.h"
#include "mptypes.h"
#include "math.h"

#define MPVAR(p2var)    (*p2var)

int (*savedGetVarPointer)( void **result, int id);

int getMPVarPointer( void **result, int id) { return (*savedGetVarPointer)( result, id); }

int mpUserInit( int (*function)( void **result, int id))
{
    savedGetVarPointer = function;
    return USER_RETURN_INIT_OK;
}

/*
 * mpUserEvent - This is the function that is called when the user code is implimented
 */
int mpUserEvent( long *result, int userEventIdentifier)
{
    //-- MP Vars -----

        //variables des moteurs
    static long* fServo5;
    static long* fServo6;
    static long* fServo7;
    static long* fServo8;
    static long* fServo9;
    static long* fServo10;
    static long* fServo11;
    static long* fServo12;
    static long* fServo13;
    static long* fServo14;
    static long* fServo15;
    static long* fServo16;
    //variables d'attitude mesurées
    static long* mpitch;
    static long* myaw;
    static long* mroll;
    //vars d'attitude dérivée (vitesse) mesurées
    static long* mppitch;
    static long* mproll;
    static long* mpyaw;
```

```
//vars d'attitude dérivées (vitesse) mises à l'échelle
static long ppitch;
static long proll;
static long pyaw;
//vars d'attitude dérivée deux fois (accélération)
static long pppitch;
static long pproll;
static long ppyaw;

//variables des données de la télécommande
static long* tpitch;
static long* troll;
static long* tyaw;
static long* tthrust;
//variables des commandes de position (var de la telecommande apres mise en forme)
static long cpitch;
static long croll;
static long cyaw;
static long cthrust;
// variable pid
static long kroll;
static long kpitch;
static long kyaw;
//variables des commandes de correction de position
static long pitch=0;
static long roll=0;
static long yaw=0;
static long thrust;
//variables des commandes moteur
static long pitchmix;
static long rollmix;
static long yawmix;
static long mot1;
static long mot2;
static long mot3;
static long mot4;

//vars utilisées pour la mise en forme des signaux

static long zerothrust;
static long zeroyaw;
static long zeroroll;
static long zeropitch;

//vars utilisées pour le calcul de l'accélération
static long p1pitch=0;
static long p1roll=0;
static long p1yaw=0;
static long p2pitch=0;
static long p2roll=0;
```

```
static long p2yaw=0;

//valeurs utilisées pour le rate limiter
static long c1pitch=0;
static long c2pitch=0;
static long c1roll=0;
//static long c1yaw=0;
static long c2roll=0;
static long c2yaw=0;

//coefficients des correcteurs PID
static long* K1roll;
static long* K2roll;
static long* K3roll;
static long* K1yaw;
static long* K2yaw;
static long* K3yaw;

//variables diverses
static int lim =10;
static int init=1;
static long* offground;
static long* radioon;

//variables integrales
static long* K4roll;
static long* K4yaw;
static long ipitch=0;
static long iroll=0;
static long iyaw=0;

//rajout pour le compas, cap
static long *Hdg;
static long Hdg_ref;

//pour le rate limiter des gaz
static long thrust1;
```

```
//-----
switch( userEventIdentifier)
{
//-----
//
case USER_HARDWARE_INITIALIZED :
```

```

getMPVarPointer( &fServo5, MPFID_FINE_SERVO_BASE + SERVO_FIVE
);
getMPVarPointer( &fServo6, MPFID_FINE_SERVO_BASE + SERVO_SIX
);
getMPVarPointer(      &fServo7,      MPFID_FINE_SERVO_BASE      +
SERVO_SEVEN );
getMPVarPointer(      &fServo8,      MPFID_FINE_SERVO_BASE      +
SERVO_EIGHT );
getMPVarPointer( &fServo9, MPFID_FINE_SERVO_BASE + 8 );
getMPVarPointer( &fServo10, MPFID_FINE_SERVO_BASE + 9 );
getMPVarPointer( &fServo11, MPFID_FINE_SERVO_BASE + 10 );
getMPVarPointer( &fServo12, MPFID_FINE_SERVO_BASE + 11 );
getMPVarPointer( &fServo13, MPFID_FINE_SERVO_BASE + 12 );
getMPVarPointer( &fServo14, MPFID_FINE_SERVO_BASE + 13 );
getMPVarPointer( &fServo15, MPFID_FINE_SERVO_BASE + 14 );
getMPVarPointer( &fServo16, MPFID_FINE_SERVO_BASE + 15 );

getMPVarPointer( &mpitch, 1266 );
getMPVarPointer( &myaw, 1283 );
getMPVarPointer( &mroll, 1267 );
getMPVarPointer( &Hdg, MPFID_CURRENT_HEADING);

getMPVarPointer( &mppitch, 1403 );
getMPVarPointer( &mproll, 1404 );
getMPVarPointer( &mpyaw, 1405 );

getMPVarPointer( &tpitch, MPFID_ELEVATOR_PULSE );
getMPVarPointer( &troll, MPFID_AILERON_PULSE );
getMPVarPointer( &tyaw, MPFID_RUDDER_PULSE );
getMPVarPointer( &tthrust, MPFID_THROTTLE_PULSE );

getMPVarPointer( &K1roll, MPFID_ROTATION_SPEED );
getMPVarPointer( &K2roll, MPFID_CLIMB_SPEED );
getMPVarPointer( &K3roll, MPFID_APPROACH_SPEED );
getMPVarPointer( &K4roll, MPFID_FLARE_ALTITUDE );
getMPVarPointer( &K1yaw, MPFID_CLIMB_MARGIN );
getMPVarPointer( &K2yaw, MPFID_WAYPOINT_DIA );
getMPVarPointer( &K3yaw, MPFID_CCT_ALTITUDE );
getMPVarPointer( &K4yaw, MPFID_DESCENT_RATE );

getMPVarPointer( &offground, MPFID_OFF_GROUND );
getMPVarPointer( &radioon, MPFID_PIC_CIC );

MPVAR(fServo5)=-32767;
MPVAR(fServo6)=-32767;
MPVAR(fServo7)=-32767;
MPVAR(fServo8)=-32767;

```



```

return USER_RETURN_IGNORE;
//-----
// PID loop works with 5/30 Hz
case USER_PID1 :

    //initialisation de la carte, on récupère les valeurs zero de la telecommande
    //ce code n'est executé qu'une fois au démarrage de la carte
    if(init==1){

        zerothrust=MPVAR(tthrust);
        zeroyaw=MPVAR(tyaw);
        zeroroll=MPVAR(troll);
        zeropitch=MPVAR(tpitch);
        c2pitch=zeropitch;
        c2roll=zeroroll;
        c2yaw=zeroyaw;
        init=0;
    }

    MPVAR(offground)=0;

    //mise en forme des signaux envoyés par la telecommande en signaux de
    grandeurs comparable aux fServo (variant entre -32767 et +32767)

    croll = (MPVAR(troll) - zeroroll)*32767/820;
    cpitch = (MPVAR(tpitch) - zeropitch)*32767/820; //on a changé les /1000 et
/600 par /820
    cyaw = (MPVAR(tyaw) - zeroyaw)*32767/820;
    cthrust = -32767+(MPVAR(tthrust)-zerothrust)*32767/820; //la position 0 pour
les gazs correspond à la manette en bas, donc à -32767

    // Application du limiteur de vitesse

    if((cpitch - c1pitch)>lim){
        c1pitch = c1pitch + lim;
    }
    else{
        if((cpitch - c1pitch)<-lim){
            c1pitch = c1pitch - lim;
        }
        else{
            c1pitch = cpitch;
        }
    }

    if((croll - c1roll)>lim){
        c1roll = c1roll + lim;

```

```

}
else{
    if((croll - c1roll)<-lim){
        c1roll = c1roll - lim;
    }
    else{
        c1roll = croll;
    }
}
}

```

moteur

//mise en forme de la poussée, thrust représente 80% max de la poussée du

```
thrust1=(cthrust+32767)*0.8 - 32767;
```

```

//rate limiter sur les gaz
if((thrust1 - thrust)>650){
    thrust = thrust + 650;
}
else{
    if((thrust1 - thrust)<-650){
        thrust = thrust - 650;
    }
    else {
        thrust = thrust1;
    }
}

```

```

kroll = (32700 + croll)/650;
kpitch = (32700 + cpitch)/650;
kyaw = (32700 + cyaw)/650;

```

```

if (kroll<0) {kroll=0;}
if (kpitch<0){kpitch=0;}
if (kyaw<0) {kyaw=0;}

```

```

//calcul de l'accélération le 0.24 est une echelle de temps
p1pitch=p2pitch;
p2pitch=MPVAR(mppitch);
pppitch=(p2pitch - p1pitch)*100/24;

```

```
p1roll=p2roll;
p2roll=MPVAR(mproll);
pproll=(p2roll-p1roll)*100/24;
```

```
p1yaw=p2yaw;
p2yaw=MPVAR(mpyaw);
ppyaw=(p2yaw-p1yaw)*100/24;
```

```
//calcul de l'erreur de position
//on supposera que le drone ne peut pas excéder des angles de 30 degrés et 45
degrés en lacet
```

```
pitch = c1pitch-MPVAR(mpitch)*32767/270; // 540 = 30° en radians * 1024
roll = c1roll-MPVAR(mroll)*32767/270;
yaw = (0 - MPVAR(myaw))*3215/3215; // 3215 = 3.14 * 1024
```

```
// Calcul de l'effet integral
```

```
ipitch = ipitch + pitch/10;
iroll = iroll + roll/10 ;
```

```
if(iroll>10000){
    iroll = 10000;
}
else{
    if(iroll<-10000){
        iroll = -10000;
    }
}
```

```
if(ipitch>10000){
    ipitch = 10000;
}
else{
    if(ipitch<-10000){
        ipitch = -10000;
    }
}
```

```
//mise à l'échelle des vitesses et accelerations(sur une échelle de -32767 à
32767)
```

```
ppitch = MPVAR(mppitch)*32767/2000;
proll = MPVAR(mproll)*32767/2000;
pyaw = MPVAR(mpyaw)*32767/2000;
```

```
pppitch = pppitch*32767/2000;  
pproll = pproll*32767/2000;  
ppyaw = ppyaw*32767/1000;
```

```
//saturation de la vitesse (Kd=10 => 10% de la course pour la dérivée ->  
saturation à 220000/2=110000)
```

```
if(ppitch>110000){  
    ppitch = 110000;  
}  
else{  
    if(ppitch<-110000){  
        ppitch = -110000;  
    }  
}
```

```
if(proll>110000){  
    proll = 110000;  
}  
else{  
    if(proll<-110000){  
        proll = -110000;  
    }  
}
```

```
//saturation de l'accélération (Kd=10 => 10% de la course pour la dérivée ->  
saturation à 220000/2=110000)
```

```
if(pppitch>110000){  
    pppitch = 110000;  
}  
else{  
    if(pppitch<-110000){  
        pppitch = -110000;  
    }  
}
```

```
if(pproll>110000){  
    pproll = 110000;  
}  
else{  
    if(pproll<-110000){  
        pproll = -110000;  
    }  
}
```

```
//calcul du correcteur
```

```
//le /100 permet d'utiliser des coeffs 100 fois plus grands (plus lisible)
```

```
pitchmix = (MPVAR(K1roll)*1/0.91*pitch - MPVAR(K2roll)*ppitch -
MPVAR(K3roll)*pppitch + ipitch*MPVAR(K4roll)*(-1/26.2)*1/100 )/100 +
MPVAR(K2yaw) - MPVAR(K3yaw) ;
```

```
rollmix = (MPVAR(K1roll)*1/0.91*roll - MPVAR(K2roll)*proll -
MPVAR(K3roll)*pproll + iroll*MPVAR(K4roll)*(-1/26.2)*1/100 )/100 ;
```

```
yawmix = MPVAR(K1yaw)*pyaw/100;
```

```
//répartition de la correction sur chaque moteur et mise en forme
```

```
mot1 = thrust + (-pitchmix + yawmix )*0.3; // les moteurs tournent à l'envers
donc on a changé les yawmix avec leur signe opposé
```

```
mot2 = thrust + (+rollmix - yawmix )*0.3;
```

```
mot3 = thrust + (+pitchmix + yawmix )*0.3;
```

```
mot4 = thrust + (-rollmix - yawmix )*0.3;
```

```
//le if qui suit permet une initialisation correcte des moteurs et leur arrêt manête
des gazs en bas
```

```
//en effet les variateurs doivent recevoir un zero (c-à-d -32767) de la carte pour
s'initialiser
```

```
if(thrust1<-29000){
```

```
    MPVAR(fServo5)=-32767;
```

```
    MPVAR(fServo6)=-32767;
```

```
    MPVAR(fServo7)=-32767;
```

```
    MPVAR(fServo8)=-32767;
```

```
}
```

```
else{
```

```
    //Une vérification est faite sur la commande pour éviter de saturer un
```

```
moteur
```

```
    if((mot2)<32700){
```

```
        MPVAR(fServo5) = mot2;
```

```
    }
```

```
    else{ MPVAR(fServo5)=32700;}
```

```
    if((mot3)<32700){
```

```
        MPVAR(fServo6) = mot3;
```

```
    }
```

```
    else{ MPVAR(fServo6)=32700;}
```

```
    if((mot4)<32700){
```

```
        MPVAR(fServo7) = mot4;
```

```
    }
```

```
    else{ MPVAR(fServo7)=32700;}
```

```
        if((mot1)<32700){
            MPVAR(fServo8) = mot1;
        }
        else{ MPVAR(fServo8)=32700;}

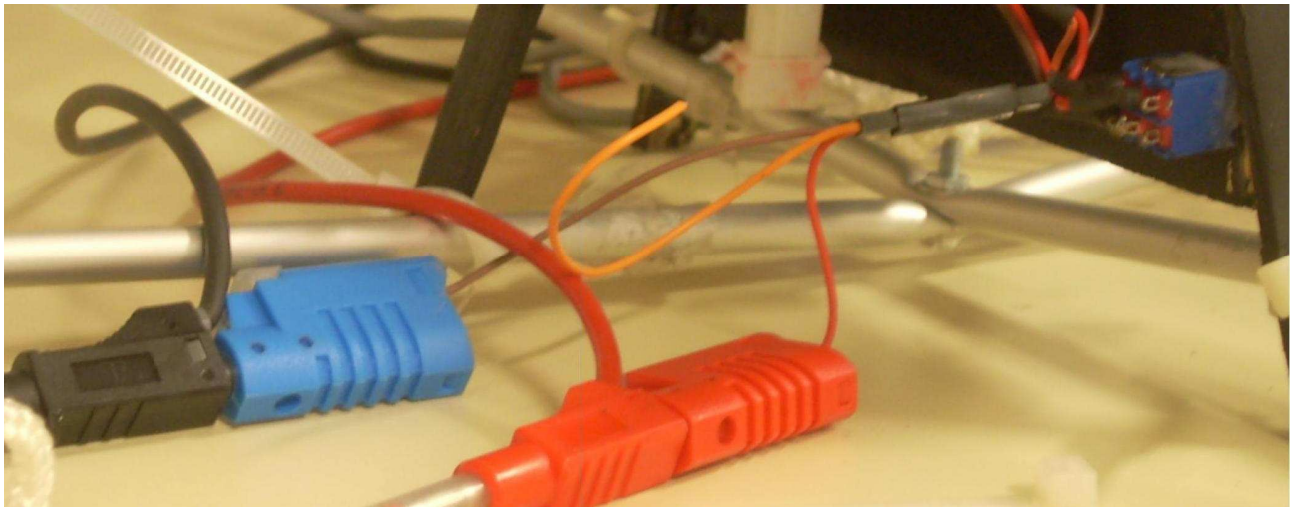
    }

    MPVAR(fServo9)=MPVAR(K1roll)*1/0.91;
    MPVAR(fServo10)=MPVAR(K2roll);
    MPVAR(fServo11)=MPVAR(K3roll);
    MPVAR(fServo12)=MPVAR(K4roll)*(-1/26.2);
    MPVAR(fServo13)=MPVAR(K1yaw);
    MPVAR(fServo14)=iroll;

    return USER_RETURN_REPLACE;
//-----
// PID loop works with 5/30 Hz
case USER_PID2 :
    return USER_RETURN_REPLACE;
//-----
// Event not handled
default:
    return USER_RETURN_UNUSED;
}
}
```

## Annexe 7 Procédure de démarrage du quadrirotor

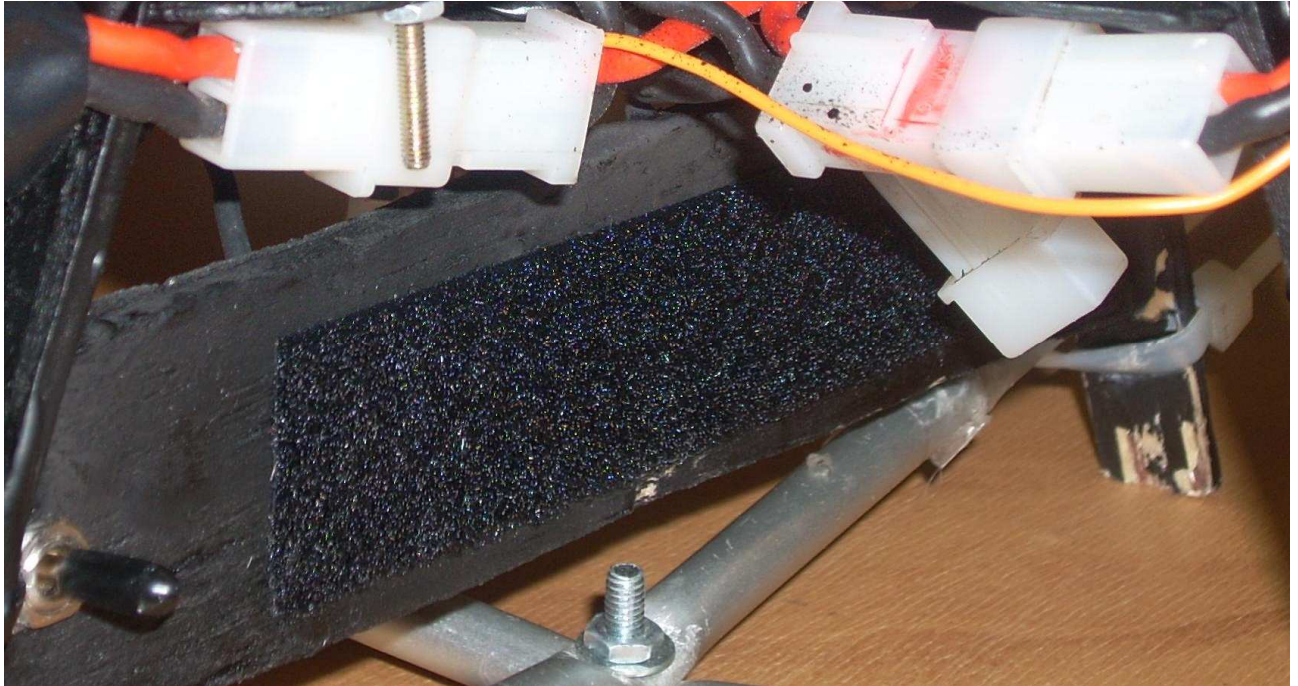
Le démarrage du drone nécessite l'utilisation d'une alimentation stabilisée. Lors du démarrage les packs de propulsion et l'alimentation stabilisée seront pendant un court laps de temps en lien direct. Il est donc très important de régler la source sur une tension proche de la tension des packs (de l'ordre de 7,4V). Nous choisissons en général une tension de 8,4V qui présente l'avantage d'être supérieure à celle des packs, même pleins. Ainsi lors de la liaison pack/alimentation, cette dernière recharge légèrement les packs plutôt que de les décharger. Une fois l'alimentation stabilisée réglée il faut la brancher sur la petite prise sortant de l'interrupteur.



*Interrupteur et prise d'alimentation extérieure.*

Si les moteurs font des bips d'initialisation c'est simplement que l'interrupteur est dans la mauvaise position. Il suffit donc de le basculer. On peut alors s'assurer du démarrage de la carte MicroPilot en contrôlant le débit de l'alimentation extérieure, lors d'une initialisation la carte doit consommer entre 100 et 200mA. Une fois que la carte est complètement initialisée on peut brancher les packs. Cette manœuvre est relativement dangereuse du fait de la proximité des hélices il est donc conseillé d'être aidé d'une personne qui tient le drone au sol. On distingue sur la photo ci-dessous une des prises pour les packs en blanc à droite ainsi que le scratch sur lequel le fixer.





Prise des packs et fixation

Une fois branchés, les variateurs émettent un bip puis deux séries de deux bips pour signaler une initialisation correcte. Si ces derniers émettent une série continue de bips c'est qu'ils ne peuvent s'initialiser : la carte MicroPilot n'a sans doute pas eu le temps de terminer son démarrage. Une fois les variateurs en ordre de fonctionnement, on peut alors basculer l'interrupteur. La carte MicroPilot sera alors alimentée à la fois par la source extérieure et par les packs. Il est donc important de débrancher le plus vite possible la source extérieure. Le drone est alors pleinement autonome et les essais peuvent débuter.

## **Annexe 8 Recommandations de sécurité**

Avant toute chose il ne faut pas oublier que le drone est muni de quatre moteurs puissants il est donc important de toujours prendre des précautions. Il arrive que sans raison la carte plante et que les moteurs accélèrent sans raison de plus en cas de perte de l'émission la carte MicroPilot interprète la sortie récepteur comme un ordre à 80% de la puissance. Nous avons pour cela ajouté un limiteur de vitesse sur la commande des gaz qui permet de ne pas avoir d'accélération soudaine. Cependant le drone reste dangereux et il est conseillé lors de manœuvre sur celui-ci sous tension d'être aidé d'une personne qui le tient.

### **Cas de panne les plus fréquents.**

#### **La carte ne démarre pas :**

Si la carte ne s'initialise pas c'est le plus souvent un problème de tension. Il faut alors s'assurer que celle-ci est alimentée avec au moins 4V. On peut aussi contrôler la continuité du circuit en contrôlant sur alimentation stabilisée que la carte ait une consommation de l'ordre de 140mA. Il peut cependant aussi arriver que le dernier chargement de code ait été un échec, auquel cas la carte ne peut s'initialiser. Il faut alors reprendre la procédure de chargement de code en s'assurant que la tension d'alimentation soit suffisante.

#### **Les quatre variateurs ne démarrent pas :**

Si les quatre variateurs émettent une série continue de bips c'est un problème central. Le plus souvent il s'agit d'une mauvaise initialisation de la carte (se référer au paragraphe précédent). Il est aussi possible que la carte n'ait juste pas eu le temps de démarrer, en effet celle ci attend une stabilisation des gyroscopes qui peut être très longue si l'on bouge le drone. Il peut aussi s'agir d'un problème de masse. Un petit câble de masse équipé d'une résistance part de l'interrupteur et doit être connecté à la masse du récepteur (attention au sens de branchement dans le récepteur).

#### **Un des variateurs ne démarre pas :**

Il peut arriver qu'un seul variateur ne s'initialise pas correctement. Dans ce cas il peut s'agir soit d'un problème de branchement soit d'un faux contact au niveau du variateur lui même. En effet il peut arriver qu'avec les vibrations ou après un choc le fil de signal se dessoude au niveau du contrôleur

## VII. Bibliographie

ALEXIS FRENOT - ANTHONY GOSSMANN - ROMARIC GUILLERM  
« Stabilisation d'un quadrirotor » Rapport de PIP 2005-2006

ADELINE SERRECOURT - JULIEN CHAPUIS  
« Stabilisation d'un quadrirotor » Rapport de PIP 2006-2007

Forum de discussions consacré en grande partie aux drones quadrirotor  
<http://forum.xbird.org/>

JOEL BORDENEUVE-GUIBE  
« Commande des systèmes linéaires » Cours d'automatisme de seconde année à l'ISAE  
cursus ENSICA

FRANCOIS VINCENT  
« Traitement numérique du signal » cours de première année à l'ISAE cursus ENSICA

DOCUMENTATION DIVERSE  
« MP2028g Installation and Opération » Manuel MicroPilot  
Manuels d'utilisation des composants  
[http://sksbir.free.fr/dossier\\_quadrirotor.html](http://sksbir.free.fr/dossier_quadrirotor.html)  
<http://tom.poub.free.fr/blog/index.php>  
<http://www.heli4.com/article368.html>