

Auteurs : FRENOT Alexis
GOSSMANN Anthony
GUILLERM Romaric

Tuteur : BRIERE Yves



Rapport PIP

STABILISATION D'UN QUADRIROTOR



2005-2006

Table des matières

1. Introduction	5
2. Principe de vol d'un quadrirotor	6
2.1 <i>Sens de rotation et mouvements possibles</i>	6
2.2 <i>La radiocommande : commandes souhaitées sur les différentes voies</i>	8
3. Conception de la structure	9
3.1 <i>Première réalisation : tout contre-plaqué</i>	9
3.1.1 <i>Présentation de la structure</i>	9
3.1.2 <i>Inconvénients de cette solution</i>	12
3.2 <i>Deuxième réalisation : contre-plaqué, balsa et carbone</i>	12
3.2.1 <i>Présentation détaillée</i>	12
3.2.2 <i>Avantages de cette conception</i>	15
4. Eléments indispensables	16
4.1 <i>Le récepteur, la radiocommande et les quartzs</i>	16
4.2 <i>Les pales et les moteurs</i>	17
4.3 <i>Les contrôleurs</i>	17
4.4 <i>Les accumulateurs</i>	18
5. Etude dynamique du quadrirotor	19
5.1 <i>Hypothèses du modèle</i>	19
5.2 <i>Développement du modèle selon Lagrange</i>	19
5.2.1 <i>Choix des coordonnées</i>	20
5.2.2 <i>Expression de la vitesse</i>	21
5.2.3 <i>Expression de l'énergie cinétique</i>	23
5.2.4 <i>Expression de l'énergie potentielle</i>	24
5.2.5 <i>Equation du roulis</i>	24
5.2.6 <i>Equation du tangage</i>	26
5.2.7 <i>Equation du lacet</i>	28
5.2.8 <i>Expression des forces non conservatives</i>	30
5.2.9 <i>Expression des accélérations angulaires</i>	31
5.3 <i>Effet de l'angle des supports moteur</i>	32
5.4 <i>Calcul des différents paramètres</i>	33
5.4.1 <i>Calcul des moments d'inerties du drone</i>	33
5.4.2 <i>Calcul du coefficient de poussée b</i>	34
5.4.3 <i>Calcul du coefficient de traînée d</i>	34
6. Modèle SimuLink du quadrirotor	36
6.1 <i>Fonction de transfert de l'ensemble moteur-variateur</i>	37
6.2 <i>Choix des correcteurs pour un angle de support moteur nul</i>	38
6.2.1 <i>Influence des effets d'inerties et des effets gyroscopiques par rapport aux effets moteurs</i>	38
6.2.2 <i>Choix du correcteur</i>	42
6.3 <i>Choix des correcteurs pour un angle de support moteur non nul</i>	44
7. Différentes solutions pour la stabilisation	46
7.1 <i>Solution à base d'AOP</i>	46
7.2 <i>Solution mixeurs – gyroscopes</i>	46
7.2.1 <i>Le mixeur</i>	46
7.2.2 <i>Le gyroscope</i>	47
7.2.3 <i>La stabilisation</i>	47
7.3 <i>Solution carte MicroPilot</i>	48
7.3.1 <i>Branchement de la carte MicroPilot et de ses éléments additionnels</i>	48
7.3.2 <i>Le programme de stabilisation</i>	50

8. Tests	58
8.1 Test de la solution mixeurs – gyroscopes	58
8.2 Test de la solution carte MicroPilot	58
8.2.1 Dispositif de test	58
8.2.2 Tests réalisés	59
8.2.2 Résultats des tests	60
8.3 Courbe Masse-Puissance	61
9. Conclusion	62
10. Annexes	63
10.1 Annexe : Schémas électroniques de la solution à base d'AOP	63
10.2 Annexe : Guide d'utilisations de la carte MP2028	66
10.3 Annexe : Programme informatique	71
10.4 Annexe : Schéma SimuLink	80
10.5 Annexe : Budget	83
10.6 Annexe : documents associés aux composants achetés	84
11. Bibliographie	93

1. Introduction

Le quadrirotor est une configuration qui semble intéresser de plus en plus de personnes de nos jours.

Notre PIP, dont le but final est de stabiliser un quadrirotor, a débuté par une étude puis une fabrication de structure d'un modèle d'hélicoptère à quatre rotors. Nous avons ensuite intégré à la structure les moteurs et les éléments indispensables à leur fonctionnement, en veillant à avoir suffisamment de puissance pour faire décoller l'engin. Ensuite, il nous a fallu comprendre le principe de vol d'un tel appareil.

Vous découvrirez alors dans ce rapport trois méthodes pour tenter une stabilisation du quadrirotor, dont deux seulement seront testées. Une seule des méthodes donnera un résultat satisfaisant, qui plus est permettant une évolution future du modèle réduit, dont une qualification possible serait : « drone ». En effet, elle est basée sur l'utilisation d'une carte Micropilot, qui, couplée à un récepteur GPS, permettrait d'effectuer des vols autonomes.

Mais avant d'arriver à utiliser la fonction « vol automatique », il est impératif de rendre le quadrirotor parfaitement stable et manœuvrable. Une étude dynamique de l'hélicoptère permettra, outre le fait d'éclaircir les effets des commandes sur les différentes accélérations de l'appareil, de réaliser un schéma Simulink. Celui-ci pourra alors nous aider dans la recherche de coefficients, pour PID par exemple, permettant la stabilisation.

Vous trouverez en annexe : des plans de construction, des schémas électroniques, un programme informatique de stabilisation, ainsi qu'une explication sur le déroulement du PIP et un récapitulatif du budget mis en jeu. Des documents associés à des composants utilisés sont aussi joints, car ils peuvent être utiles, notamment lors du prolongement du projet par une autre équipe.

2. Principe de vol d'un quadrirotor

2.1 Sens de rotation et mouvements possibles

Un quadrirotor possède, comme son nom l'indique, quatre rotors pour se sustenter. Pour contrer un moment de lacet, il est nécessaire de faire tourner deux hélices dans un sens et les deux autres dans l'autre sens. En effet, lorsque l'on projette les forces aérodynamiques exercées par l'air sur la pale, on s'aperçoit qu'un rotor a toujours tendance à faire tourner le quadrirotor dans le sens inverse de sa rotation.

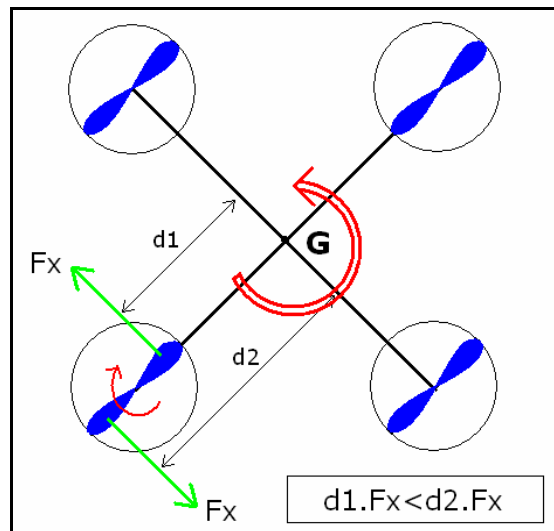


Figure 2.1 : moment de lacet

De plus, pour faciliter la gestion des commandes de tangage et de roulis, on choisit l'avant du quadrirotor au niveau d'un rotor (et non entre deux rotors) et on fait tourner les moteurs qui sont face à face dans le même sens.

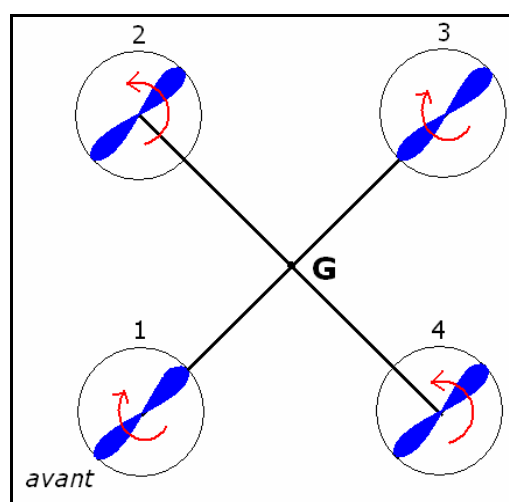


Figure 2.2 : sens de rotation des moteurs

Ainsi, les commandes des quatre moteurs seront utilisées pour modifier l'accélération verticale (commande de puissance) et pour faire tourner le quadrirotor autour de l'axe vertical (commande de lacet).

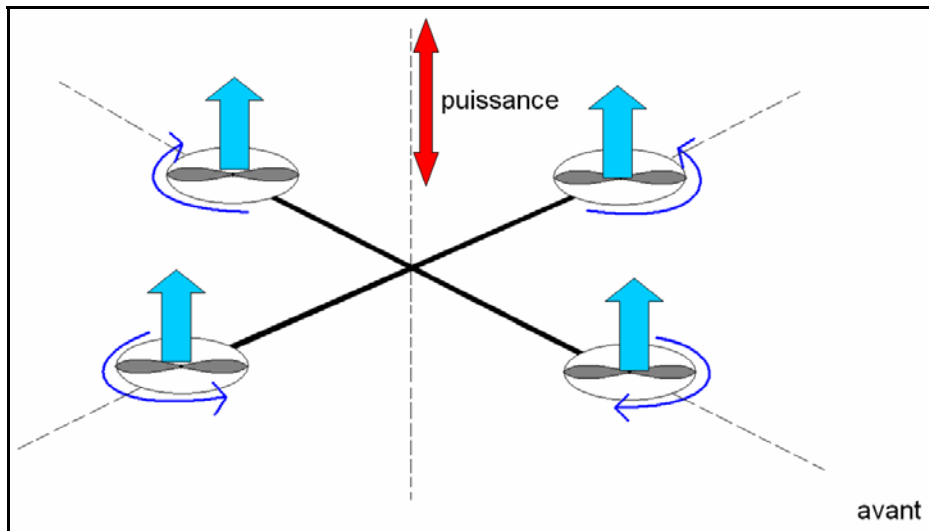


Figure 2.3 : commande de puissance

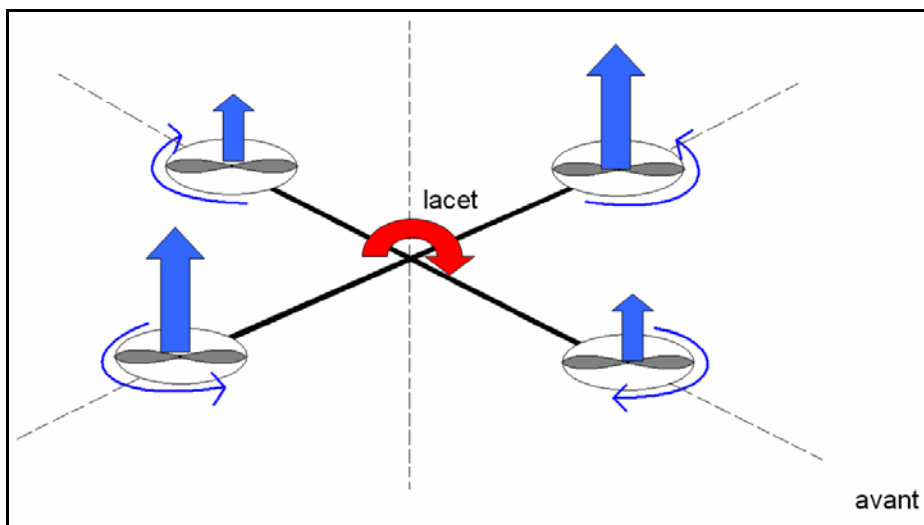


Figure 2.4 : commande de lacet

Seules les commandes de deux moteurs opposés permettent une rotation autour de l'axe longitudinale (commande de roulis) [1]. De la même manière, les commandes des deux autres moteurs permettent une rotation autour de l'axe latéral (commande de tangage).

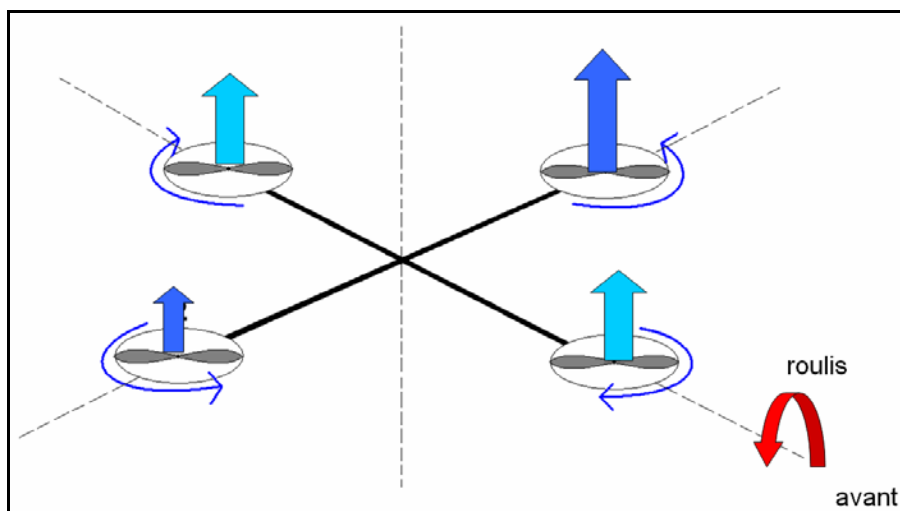


Figure 2.5 : commande de roulis

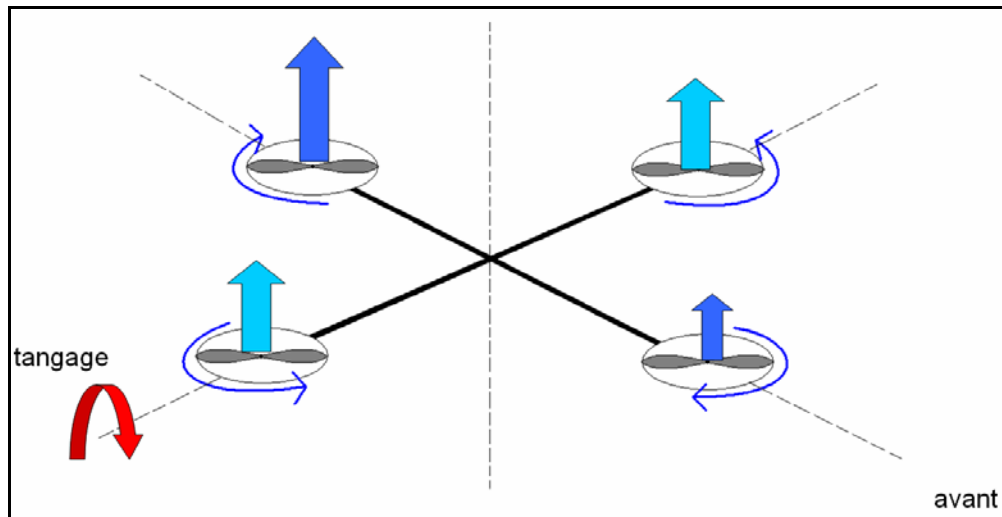


Figure 2.6 : commande de tangage

2.2 La radiocommande : commandes souhaitées sur les différentes voies

La radiocommande utilisée est une radio Futaba (réf. : 9CAP/9CHP) de 9 voies, permettant une transmission PCM ou PPM. Seules 5 voies seront utilisées dans un premier temps. La figure 2.7 représente la radiocommande et fait apparaître les sens de variation des différentes commandes.

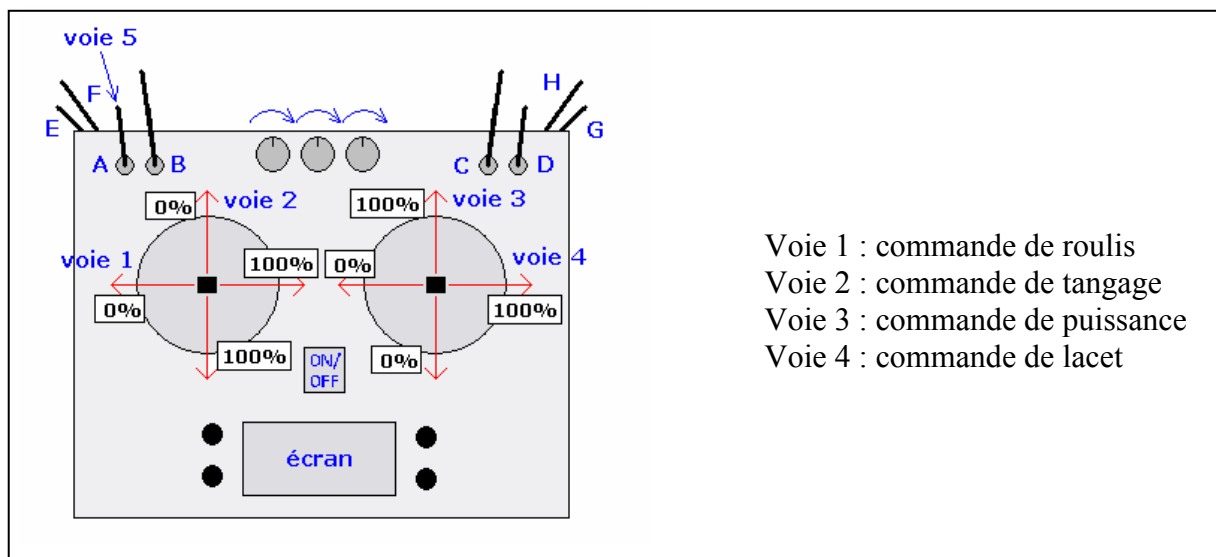


Figure 2.7 : radiocommande

Les voies indiquées correspondent à celles du récepteur Graupner (réf.: R700). En ce qui concerne les commandes de tangage et de roulis, l'utilisation de toute la plage de 0% à 100% peut ne pas être judicieuse. Afin de faciliter le pilotage, on peut améliorer la résolution autour de 50% en programmant les valeurs minimales et maximales avec la radiocommande (par exemple : 25% et 75%).

3. Conception de la structure

3.1 Première réalisation : tout contre-plaqué

3.1.1 Présentation de la structure

La première solution étudiée pour la structure de notre quadrirotor est une réalisation tout en contre-plaqué, d'une part pour sa facilité de fabrication et d'autre part pour la rigidité que l'on peut ainsi obtenir. Nous avons choisi pour cela du contre-plaqué d'ébéniste d'épaisseur 4mm.

La solution choisie est un croisillon simple évidé à intervalles réguliers afin de gagner en masse tout en conservant la rigidité en flexion du drone. Le centre du croisillon a été étudié afin de pouvoir aménager deux espaces pouvant contenir l'avionique et les batteries. Ainsi, nous avons prévu deux compartiments : l'un au niveau des moteurs et l'autre en dessous. Nous avons choisi de fixer tout le système de commande (récepteur + batterie récepteur) et d'aide au pilotage (système gyroscopes+mixers ou carte Micropilot) au même niveau que les moteurs et les batteries des moteurs dans un compartiment inférieur, dans le but d'abaisser légèrement le centre de gravité. En effet, un tel abaissement va permettre au drone d'avoir un meilleur retour à sa position de stabilité horizontale, sans pour autant l'empêcher de se pencher autour de ses axes et de maintenir cette position.



Figure 3.1 : Photographie des compartiments centraux sur la réalisation en bois

Le dessin de cette structure a été préalablement réalisé sous CATIA, et ceci d'une part pour faciliter la fabrication des pièces et surtout de vérifier leur emboîtement et d'autre part pour pouvoir par la suite calculer facilement les moments d'inertie (fonction de CATIA).

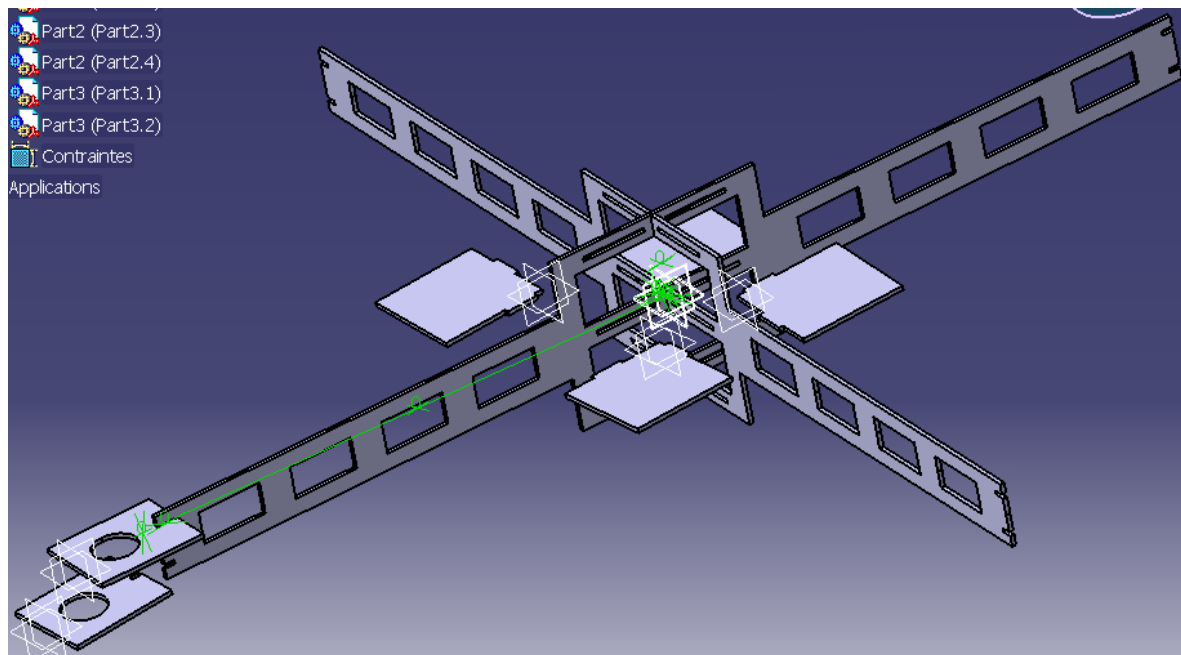


Figure 3.2 : Conception CATIA de la structure tout en bois simple

Cette première conception bien que montrant une très grande résistance en flexion, ce qui était le but principal afin de résister à la force des moteurs perpendiculaire à la structure, a cependant révélée une faiblesse en torsion. Cette faiblesse peut être gênante lorsque le drone se penche et il a donc fallut réfléchir à un système permettant en quelques sortes de raidir la structure.

La première solution à laquelle nous avons réfléchi consiste à doubler la structure initiale, ce qui permettrait au drone une fois assemblé avec ses supports moteurs de résister très efficacement à la torsion.

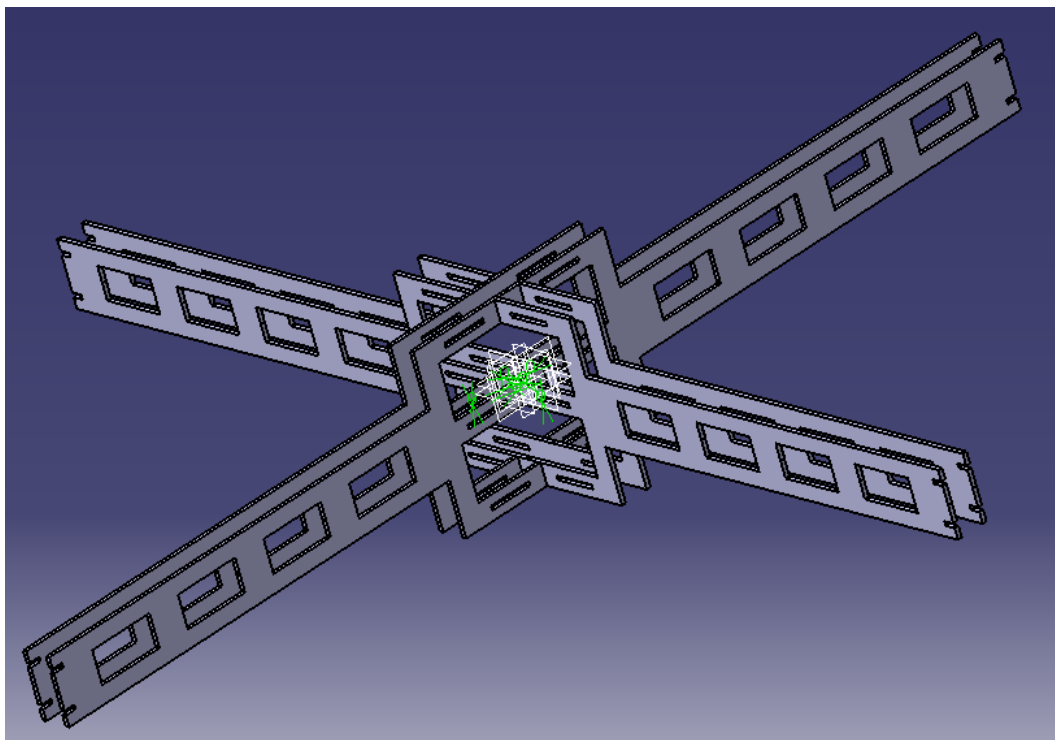


Figure 3.3 : Conception CATIA de la structure tout en bois double

Cependant, nous avons abandonné cette solution car lors de la découpe de la première pièce de la structure, nous nous sommes rendu compte que le poids du drone serait trop important par rapport au gain en torsion que nous obtiendrions.

Ainsi, il a fallu trouver une autre solution afin de rigidifier le drone en torsion sans pour autant l'alourdir de manière trop importante. Nous nous sommes donc penchés sur une solution reprenant la conception en bois simple, agrémentée de raidisseurs placés entre les bras du drone.

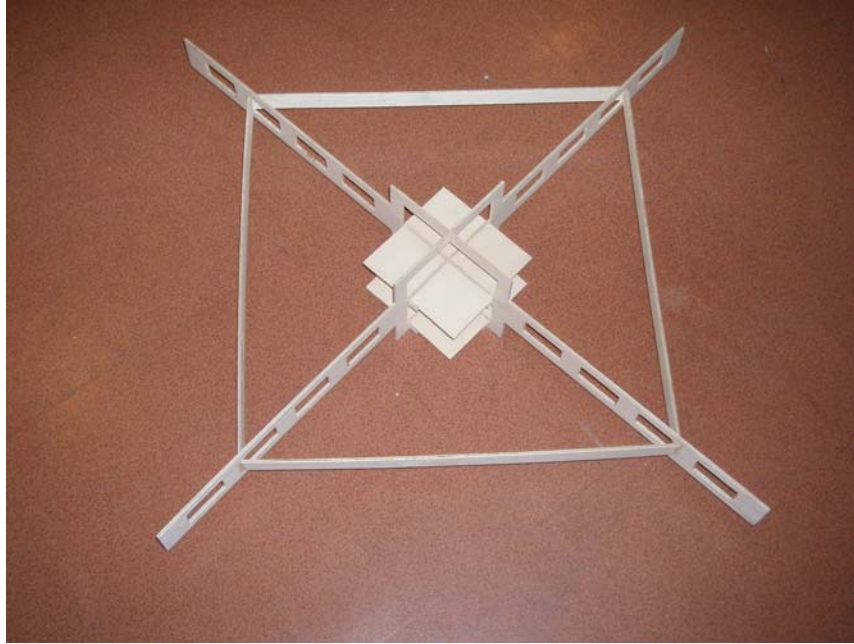


Figure 3.4 : Structure tout en bois avec raidisseurs

Le drone ainsi réalisé est résistant en flexion comme en torsion tout en ayant limité le surpoids. La structure pèse au total 590g.

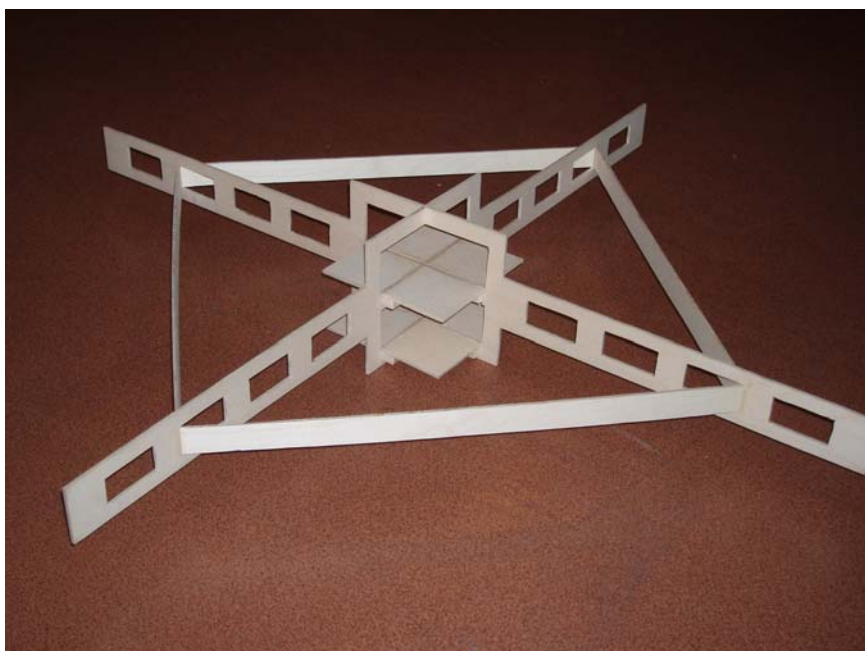


Figure 3.5 : Photographie de la structure finale tout en bois

3.1.2 Inconvénients de cette solution

L'inconvénient majeur de cette structure est son poids, car même s'il est possible de dimensionner des moteurs et des hélices permettant de soulever la structure, ses équipements moteur et avionique, ainsi que les batteries et l'éventuelle charge utile, le drone posséderait trop d'inertie ce qui serait gênant pour la stabilisation. On peut également penser que le dimensionnement évoqué précédemment serait à la limite de la puissance des moteurs (moteurs quasi pleine puissance pour faire décoller le drone), ce qui ne serait pas très bon, non seulement pour leur usure, mais aussi pour des soucis de débattements. En effet, d'après le principe de commande du quadrirotor, l'un des moteurs doit ralentir tandis que l'autre doit accélérer, et cette dernière chose n'est possible, avec un débattement convenable, uniquement si les moteurs sont dimensionnés pour sustenter le drone aux environs de la mi-puissance.

On peut également émettre comme critique sur cette structure, que malgré le fait que la torsion soit évitée aux niveau des raidisseurs, il reste environ 10 cm entre les moteurs et les raidisseurs qui peuvent subir cette torsion.

Enfin, cette structure manque de flexibilité dans le sens où tous l'électronique et les batteries doivent être fixés et obligent donc non seulement à avoir une configuration unique du drone, mais également de déplacer toute la structure pour chaque test, même ceux ne concernant que l'avionique.

3.2 Deuxième réalisation : contre-plaqué, balsa et carbone

3.2.1 Présentation détaillée

Afin de gagner en poids vis-à-vis de la version tout en bois, nous avons décidé d'utiliser des barres de carbone qui seraient fixées entre elles par des caissons en contre-plaqué.

Ainsi la partie centrale a été réalisée en contre-plaqué de 4 mm d'épaisseur et se présente sous la forme d'un cube très largement évidé afin de gagner un maximum de masse. Cette partie centrale comporte, comme pour le drone tout en bois, deux compartiments qui auront la même utilité que ceux décrits précédemment. Toutes les liaisons ont été réalisées par collage en utilisant des renforts de balsa pour rigidifier le tout.

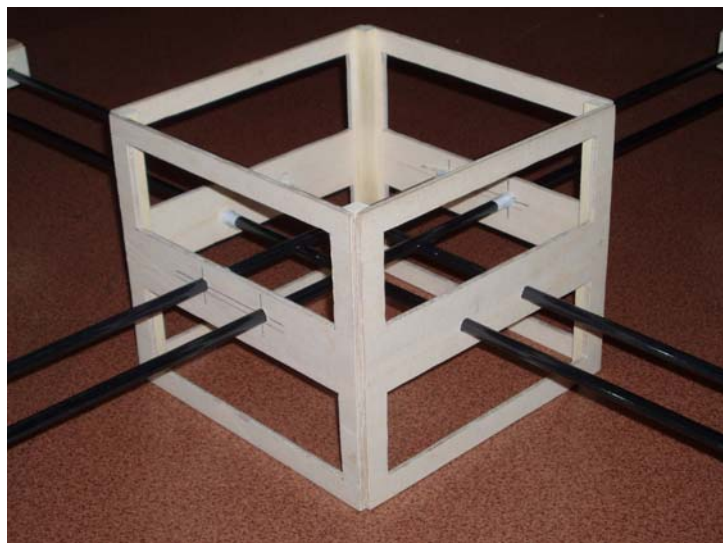


Figure 3.6 : Photographie de la structure centrale du drone bois / carbone

Nous avons utilisé 4 tubes de carbone de 1 m de longueur et de diamètre 8mm. Ces tubes ont été placés de manière à ce que deux d'entre eux soient au dessus des deux autres, le décalage de 8mm étant rattrapé au niveau des supports moteurs. Les tubes sont liés à la structure centrale à l'aide de colliers en plastique, afin d'éviter toute translation, et collés à celle-ci à l'aide de colle EPOXY afin d'éviter toute rotation dans leurs logements.

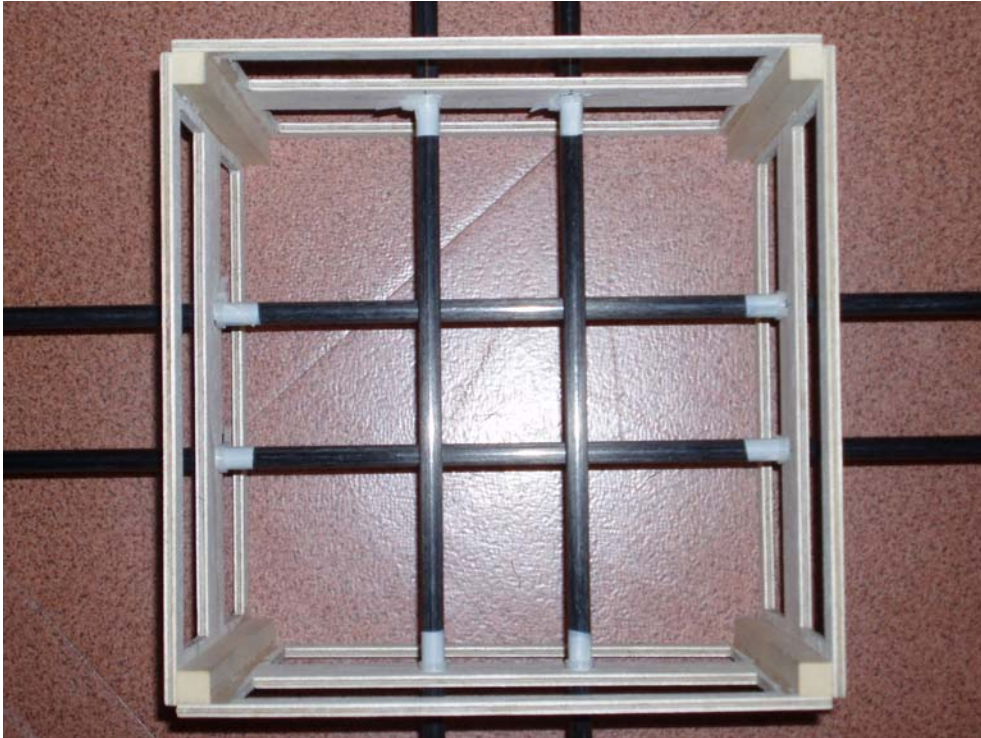


Figure 3.7 : Photographie de la liaison structure centrale / barres de carbone

Les supports moteurs sont quant à eux assez spécifiques. En effet, deux d'entre eux doivent tout d'abord se lier aux barres de carbone 8 mm plus haut que les deux autres ; ceci a été calculé et dessiné lors de la conception CATIA de ce drone.

De plus, afin d'augmenter la tendance du drone à revenir dans sa position d'équilibre de manière naturelle nous avons opté pour la solution d'un angle d'inclinaison des rotors (et donc des moteurs) vers le centre du drone. Ainsi, la partie supérieure des supports moteurs est inclinée de 2° par rapport à l'horizontal du drone.



Figure 3.8 : Photographie d'un support moteur

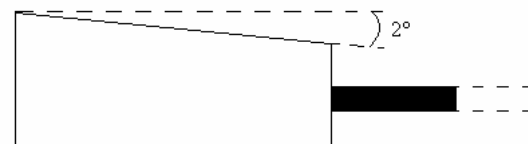


Figure 3.9 : Schéma de l'angle support moteur

Les liaisons entre les supports moteurs et les barres de carbone sont identiques à celles avec la structure centrale, à savoir : des colliers plastiques pour éviter la translation et de la

colle EPOXY afin d'éviter la rotation des barres dans leurs logements. En effet, nous nous sommes rendu compte que sans la colle, il y a de la torsion entre les supports moteurs et le support central, ce qui est à éviter au maximum.

Nous avons de plus dû évider la partie supérieure des supports moteurs afin de permettre d'une part au flux d'air provenant des hélices de ne pas "heurter" la plaque de bois ce qui créerait une force vers le bas et donc une perte d'efficacité de l'ensemble moteur+hélice. D'autre part, il a fallu penser à percer des ouvertures pour l'aération des moteurs afin que ces derniers ne chauffent pas.

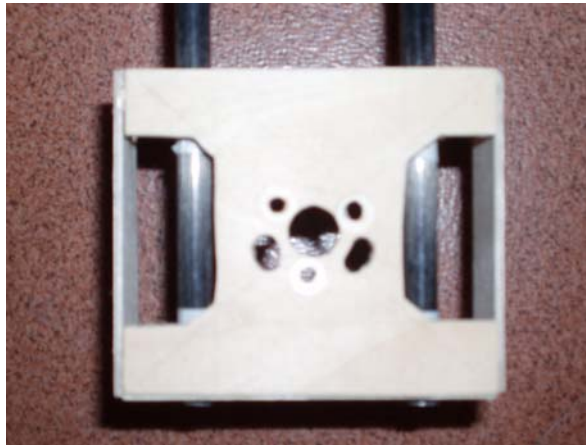


Figure 3.10 : Photographie des ouvertures de la partie supérieure d'un support moteur

Les supports pour l'avionique et les batteries ont été réalisés en contre-plaqué de 1,5 mm et en balsa pour des gains de poids. Ils peuvent s'emboîter dans la structure cubique centrale. Le support supérieur en contre-plaqué+balsa se repose sur les barres de carbone tandis que le support inférieur en balsa est collé en dessous du drone.

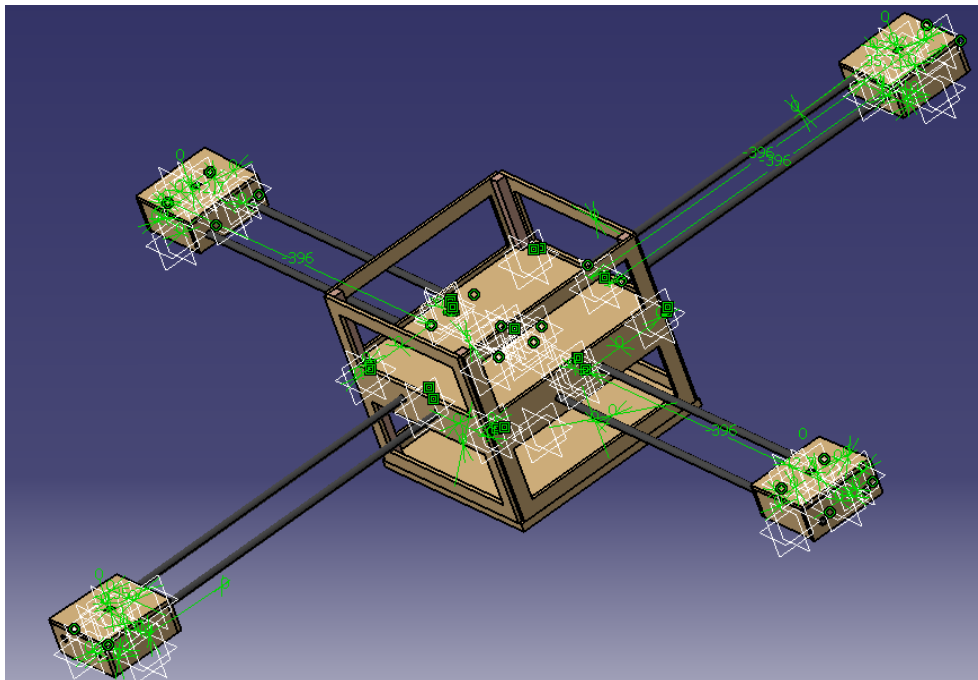


Figure 3.11 : Conception CATIA de la structure en bois + carbone

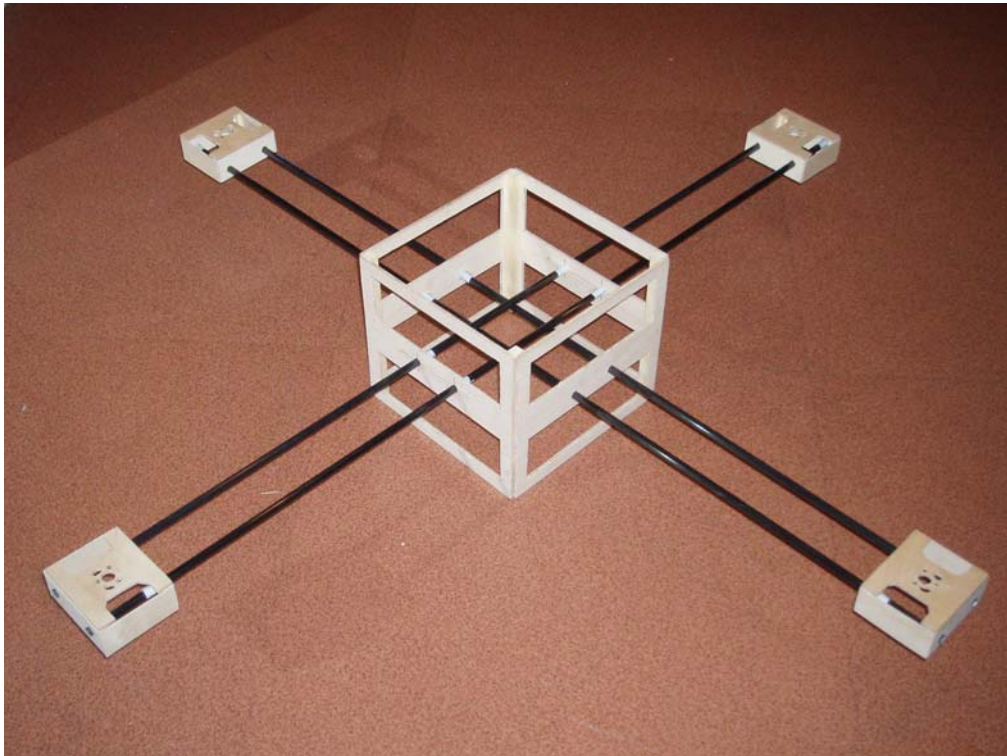


Figure 3.12 : Photographie de la structure en bois + carbone

3.2.2 Avantages de cette conception

Cette conception a l'avantage majeur d'être très rigide tout en étant relativement légère (400g de structure en comptant les supports moteurs ce qui fait plus de 200g de gagné sur la version tout en contre-plaqué dont les 590g ne comprenaient pas les supports moteurs).

La rigidité en flexion offerte par les barres de carbone est très bonne et la torsion est quasiment inexistante grâce au collage.

La structure avec en particulier le support avionique supérieur est de plus très modulaire. En effet, la plaque supérieure en contre-plaqué sur lequel l'avionique peut être attachée est amovible et tient en place dans le drone par simple emboîtement. Ainsi, il est possible de travailler sur la carte MicroPilot ou sur le système gyroscopes/mixeurs en dehors du drone, ce qui constitue un gros avantage. De plus, les deux systèmes de contrôle évoqués précédemment ont pu être montés sur deux plaques différentes ce qui permet de tester alternativement l'une ou l'autre des configurations de manière très simple.

Cette solution structurelle est donc celle qui a été retenue dans le cadre de ce projet.

4. Eléments indispensables [2]

4.1 Le récepteur, la radiocommande et les quartzs

Comme nous l'avons vu au 2.2, nous avons besoin de 5 voies. La radiocommande et le récepteur utilisés, ainsi que leur quartz respectif, permettront alors de faire le lien entre le pilote et le système de commande du quadrirotor.

Il est intéressant de connaître ce que le récepteur transmet comme ordre sur ses différentes voies en fonction des positions des commandes de la radio. Un ordre est en fait un signal PWM avec un rapport cyclique qui varie de 0,05 à 0,1, une tension d'amplitude 5 V et une période d'environ 20ms (voir figure 4.1) [3]. De plus, le récepteur ne transmet pas tous les créneaux en même temps sur ses différentes voies, il s'occupe de chacune des voies alternativement en suivant une horloge. La figure 4.2 présente une visualisation des ordres du récepteur.

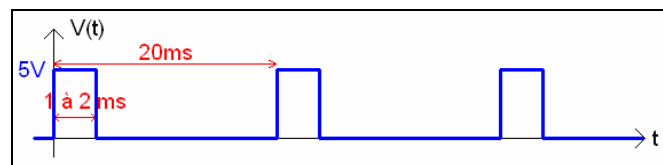


Figure 4.1 : signal PWM

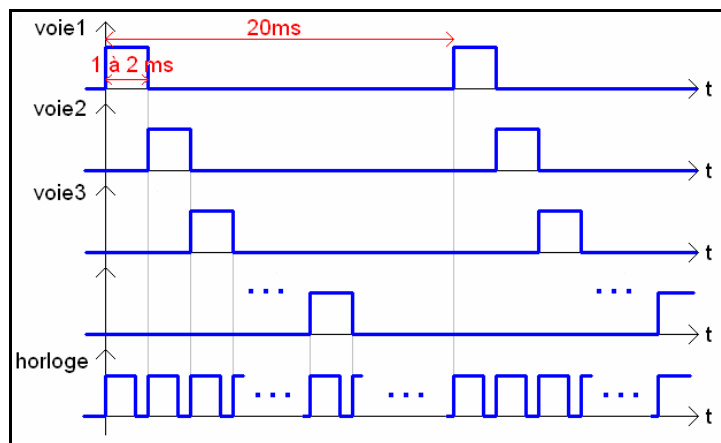


Figure 4.2 : ordres du récepteur



Figure 4.3 : Radiocommande Futaba 9CAP/9CHP



Figure 4.4 : Récepteur Graupner R700

4.2 Les pales et les moteurs

Le principe de vol du quadrirotor impose d'avoir deux pales de pas opposé aux deux autres pales. N'étant pas disponible à l'achat en France, nous avons commandé un lot de dix pales 10x4,5 aux Etats-Unis chez « Todd's Models ». Ces hélices souples sont normalement utilisées sur des avions de voltige qui sont capables de rester immobile à la verticale (tork roll), et conviennent parfaitement au quadrirotor que nous construisons. Nous n'avons pas utilisé des pales d'hélicoptère proprement dites, comme l'on trouve dans le commerce, car celles-ci sont à pas variable et auraient alourdi et compliqué inutilement le système de commande.

Les pales ont ensuite été associées à des moteurs de type brushless (réf. : brushless typhon-micro 6-23), qui ont l'avantage d'être petits, légers et puissants. Le seul inconvénient de ce type de moteur, outre le prix, est sa mise en œuvre. En effet, ils doivent obligatoirement être associés à des régulateurs particuliers, appelés contrôleurs (voir 4.3), qui nécessitent une programmation et obligent une temporisation bien précise. Ce modèle de moteur, de puissance maximale 90W, peut être alimenté sous une tension variant de 7,2V à 14,4V. Il pèse 42g et convient à un engin de 250 à 700 grammes.

Un ensemble pale et moteur peut alors porter au maximum 700g. En multipliant ce nombre par 4, on obtient une masse totale de 2,8kg. Il est évident que le quadrirotor devra peser moins de 2,8kg, pour garder une marge de manœuvre.



Figure 4.5 : Paires d'hélices 10x4.5



Figure 4.6 : Moteur Brushless typhon-micro 6-23

4.3 Les contrôleurs

Un contrôleur fait l'interface entre le signal de commande et le circuit de puissance d'un moteur. Il est relié à un accumulateur qui fournira la puissance nécessaire au moteur. Ainsi, pour notre construction, nous utilisons 4 contrôleurs (réf. : Wena – sinus-12-bec). Ils seront programmés en mode 1 (voir la documentation du contrôleur fourni en annexe), mode par défaut qui permet une sélection automatique du nombre d'éléments de l'accumulateur.

En raison de la chaleur importante dégagée par ces composants, les contrôleurs sont fixés à l'extérieur de la boîte centrale du drone entre les deux barres de carbone.



Figure 4.7 : Contrôleur Wena – sinus-12-bec

4.4 Les accumulateurs

L'accumulateur servant à alimenter le récepteur est un accumulateur classique de 4,8 V et 800mAh.

Les quatre accumulateurs devant fournir la puissance pour les quatre moteurs respectivement sont de type Li-Po à deux éléments. Lors des tests, ils seront remplacés par une alimentation stabilisée puissante.

La figure 4.8 représente le schéma complet du circuit permettant au quadrirotor d'être piloté, avec une boîte encore noire, qui devra contenir le système de commande et de stabilisation.

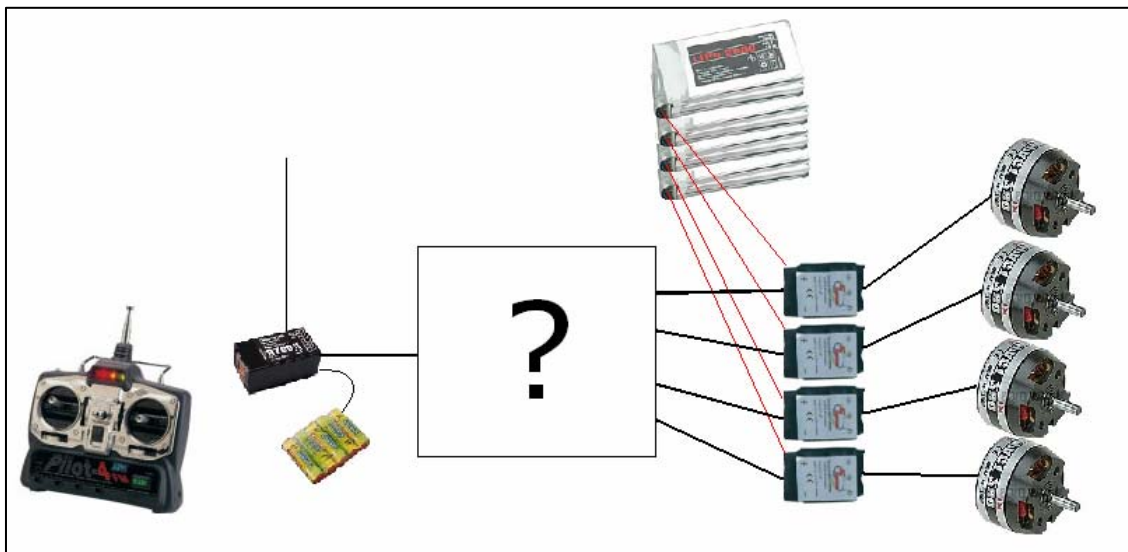


Figure 4.8 : schéma globale du circuit de commande

5. Etude dynamique du quadrirotor [4]

5.1 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 :

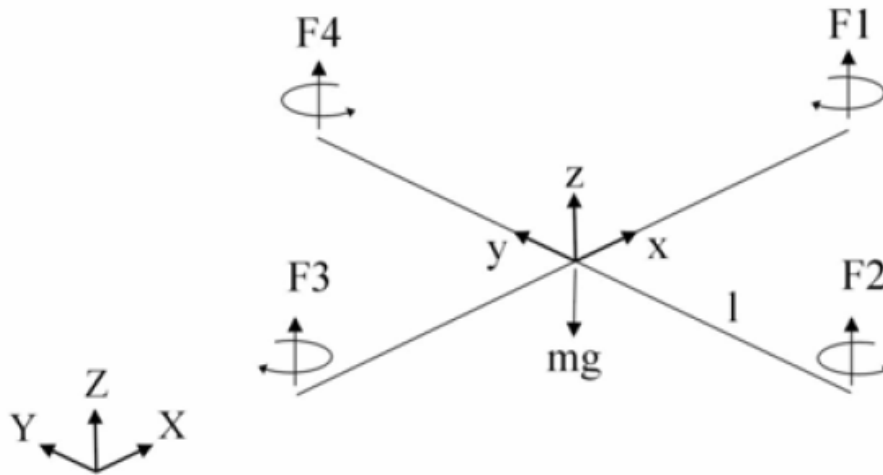


Fig. 5.1 Géométrie du drone

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 2°) 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.

5.2 Développement du modèle selon Lagrange

On choisit de développer le modèle du quadrirotor selon l'approche Lagrangienne [5].

$$\begin{cases} \Gamma_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \\ L = T - V \end{cases}$$

avec q_i : coordonnées généralisées

Γ_i : forces généralisées données par les forces non conservatives

T : énergie cinétique totale

V : énergie potentielle totale

Equations du mouvement

Le Lagrangien est donné par $L = T - V$.

Les équations du mouvement sont données par :

$$\Gamma_\Phi = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\Phi}} \right) - \frac{\partial L}{\partial \Phi} = \tau_x \quad \Gamma_\theta = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \tau_y \quad \Gamma_\psi = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \frac{\partial L}{\partial \psi} = \tau_z$$

5.2.1 Choix des coordonnées

On définit les angles de lacet, tangage et roulis (angles aéronautiques) de la façon suivante :

- rotation de Φ autour de \vec{x} (angle de roulis avec $-\pi \leq \Phi \leq \pi$)
- rotation de θ autour de \vec{y} (angle de tangage avec $-\pi/2 \leq \theta \leq \pi/2$)
- rotation de ψ autour de \vec{z} (angle de lacet avec $-\pi \leq \psi \leq \pi$)

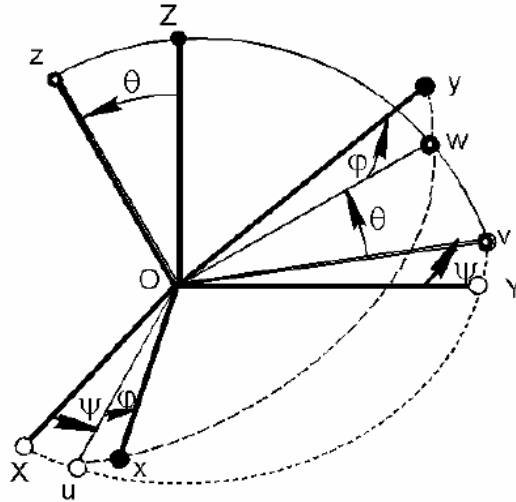


Fig. 5.2 Définition des axes

Les matrices relatives à ces transformations sont donc les matrices suivantes :

$$R(x, \Phi) = \begin{pmatrix} 1 & 0 & 0 \\ \cos(\Phi) & -\sin(\Phi) & 0 \\ \sin(\Phi) & \cos(\Phi) & 0 \end{pmatrix} \quad R(y, \theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad R(z, \psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

En multipliant les trois matrices, on obtient la matrice de rotation totale.

$$R(\Phi, \theta, \psi) = \begin{pmatrix} \cos(\psi) \cos(\theta) & \cos(\psi) \sin(\theta) \sin(\Phi) - \sin(\psi) \cos(\Phi) & \cos(\psi) \sin(\theta) \cos(\Phi) + \sin(\psi) \sin(\Phi) \\ \sin(\psi) \cos(\theta) & \sin(\psi) \sin(\theta) \sin(\Phi) + \cos(\psi) \cos(\Phi) & \sin(\psi) \sin(\theta) \cos(\Phi) - \sin(\psi) \sin(\Phi) \\ -\sin(\theta) & \cos(\theta) \sin(\Phi) & \cos(\theta) \cos(\Phi) \end{pmatrix} \quad (1.1)$$

5.2.2 Expression de la vitesse

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

$$r_{X,Y,Z}(x, y, z) = R(\Phi, \theta, \psi) \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

D'après l'équation (1.1) :

$$r_x(x, y, z) = \cos(\psi) \cos(\theta) x + [\cos(\psi) \sin(\theta) \sin(\Phi) - \sin(\psi) \cos(\Phi)] y + [\cos(\psi) \sin(\theta) \cos(\Phi) + \sin(\psi) \sin(\Phi)] z$$

$$r_y(x, y, z) = \sin(\psi) \cos(\theta) x + [\sin(\psi) \sin(\theta) \sin(\Phi) + \cos(\psi) \cos(\Phi)] y + [\sin(\psi) \sin(\theta) \cos(\Phi) - \sin(\psi) \sin(\Phi)] z$$

$$r_z(x, y, z) = -\sin(\theta) x + \cos(\theta) \sin(\Phi) y + \cos(\theta) \cos(\Phi) z$$

On obtient les vitesses correspondantes par dérivation :

$$\begin{aligned} v_x(x, y, z) = & \left[-\dot{\theta} \sin(\theta) \cos(\psi) - \dot{\psi} \sin(\psi) \cos(\theta) \right] x \\ & + \left[-\dot{\psi} \sin(\psi) \sin(\theta) \sin(\Phi) + \dot{\theta} \cos(\theta) \cos(\psi) \sin(\Phi) + \dot{\Phi} \cos(\Phi) \cos(\psi) \sin(\theta) - \dot{\psi} \cos(\psi) \cos(\Phi) + \dot{\Phi} \sin(\Phi) \sin(\psi) \right] y \\ & + \left[-\dot{\psi} \sin(\psi) \sin(\theta) \cos(\Phi) + \dot{\theta} \cos(\theta) \cos(\psi) \cos(\Phi) - \dot{\Phi} \sin(\Phi) \cos(\psi) \sin(\theta) + \dot{\psi} \cos(\psi) \sin(\Phi) + \dot{\Phi} \cos(\Phi) \sin(\psi) \right] z \end{aligned}$$

$$\begin{aligned} v_y(x, y, z) = & \left[\dot{\psi} \cos(\psi) \cos(\theta) - \dot{\theta} \sin(\theta) \sin(\psi) \right] x \\ & + \left[\dot{\psi} \cos(\psi) \sin(\theta) \sin(\Phi) + \dot{\theta} \cos(\theta) \sin(\psi) \sin(\Phi) + \dot{\Phi} \cos(\Phi) \sin(\psi) \sin(\theta) - \dot{\psi} \sin(\psi) \cos(\Phi) - \dot{\Phi} \sin(\Phi) \cos(\psi) \right] y \\ & + \left[\dot{\psi} \cos(\psi) \sin(\theta) \cos(\Phi) + \dot{\theta} \cos(\theta) \sin(\psi) \cos(\Phi) - \dot{\Phi} \sin(\Phi) \sin(\psi) \sin(\theta) + \dot{\psi} \sin(\psi) \sin(\Phi) - \dot{\Phi} \cos(\Phi) \cos(\psi) \right] z \end{aligned}$$

$$v_z(x, y, z) = -\dot{\theta} \cos(\theta) x + \left[-\dot{\theta} \sin(\theta) \sin(\Phi) + \dot{\Phi} \cos(\Phi) \cos(\theta) \right] y + \left[-\dot{\theta} \sin(\theta) \cos(\Phi) - \dot{\Phi} \sin(\Phi) \cos(\theta) \right] z$$

On peut écrire ces vitesses sous la forme :

$$v_X(x, y, z) = \begin{pmatrix} v_{x_x} & v_{x_y} & v_{x_z} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$v_Y(x, y, z) = \begin{pmatrix} v_{y_x} & v_{y_y} & v_{y_z} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$v_Z(x, y, z) = \begin{pmatrix} v_{z_x} & v_{z_y} & v_{z_z} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Ainsi, le carré de la norme de la vitesse est :

$$v^2(x, y, z) = v_x^2 + v_y^2 + v_z^2$$

$$v^2(x, y, z) = \begin{pmatrix} v_{x_x} & v_{x_y} & v_{x_z} \end{pmatrix} A \begin{pmatrix} v_{x_x} \\ v_{x_y} \\ v_{x_z} \end{pmatrix} + \begin{pmatrix} v_{y_x} & v_{y_y} & v_{y_z} \end{pmatrix} A \begin{pmatrix} v_{y_x} \\ v_{y_y} \\ v_{y_z} \end{pmatrix} + \begin{pmatrix} v_{z_x} & v_{z_y} & v_{z_z} \end{pmatrix} A \begin{pmatrix} v_{z_x} \\ v_{z_y} \\ v_{z_z} \end{pmatrix}$$

$$\text{avec } A = \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix}$$

$$v^2(x, y, z) = (v_{x_x}^2 + v_{x_y}^2 + v_{x_z}^2)x^2 + (v_{y_x}^2 + v_{y_y}^2 + v_{y_z}^2)y^2 + (v_{z_x}^2 + v_{z_y}^2 + v_{z_z}^2)z^2$$

$$+ 2xy(v_{x_x}v_{x_y} + v_{y_x}v_{y_y} + v_{z_x}v_{z_y}) + 2xz(v_{x_x}v_{x_z} + v_{y_x}v_{y_z} + v_{z_x}v_{z_z}) + 2yz(v_{x_y}v_{x_z} + v_{y_y}v_{y_z} + v_{z_y}v_{z_z})$$

$$v^2(x, y, z) = x^2 \left[\dot{\theta}^2 + \dot{\psi}^2 \cos^2(\theta) \right]$$

$$+ y^2 \left[\dot{\psi}^2 (\cos^2(\Phi) + \sin^2(\Phi) \sin^2(\theta)) + \dot{\psi}(-2 \sin(\Phi) \cos(\Phi) \cos(\theta) \dot{\theta} - 2 \dot{\Phi} \sin(\theta)) + \sin^2(\Phi) \dot{\theta}^2 + \dot{\Phi}^2 \right]$$

$$+ z^2 \left[\dot{\psi}^2 (\sin^2(\Phi) + \cos^2(\Phi) \sin^2(\theta)) + \dot{\psi}(2 \sin(\Phi) \cos(\Phi) \cos(\theta) \dot{\theta} - 2 \dot{\Phi} \sin(\theta)) + \cos^2(\Phi) \dot{\theta}^2 + \dot{\Phi}^2 \right]$$

$$+ 2xy \left[\dot{\psi}^2 \sin(\Phi) \sin(\theta) \cos(\theta) + \dot{\psi}(\cos(\Phi) \sin(\theta) \dot{\theta} - \sin(\Phi) \cos(\theta) \dot{\Phi}) - \cos(\Phi) \dot{\Phi} \dot{\theta} \right]$$

$$+ 2xz \left[\dot{\psi}^2 \cos(\Phi) \sin(\theta) \cos(\theta) + \dot{\psi}(-\cos(\Phi) \cos(\theta) \dot{\Phi} - \sin(\Phi) \sin(\theta) \dot{\theta}) + \sin(\Phi) \dot{\Phi} \dot{\theta} \right]$$

$$+ 2yz \left[-\dot{\psi}^2 \sin(\Phi) \cos(\Phi) \cos^2(\theta) + \dot{\psi}(\sin^2(\Phi) \cos(\theta) \dot{\theta} - \cos^2(\Phi) \cos(\theta) \dot{\theta}) + \sin(\Phi) \cos(\Phi) \dot{\theta}^2 \right]$$

On cherche alors à faire apparaître, en factorisant, les moments d'inerties.

$$\begin{aligned}
v^2(x, y, z) = & (y^2 + z^2) \left[\dot{\psi}^2 \sin^2(\theta) - 2 \dot{\Phi} \dot{\psi} \sin(\theta) + \dot{\Phi}^2 \right] \\
& + (x^2 + z^2) \left[\dot{\psi}^2 \sin^2(\Phi) \cos^2(\theta) + 2 \sin(\Phi) \cos(\Phi) \cos(\theta) \dot{\theta} \dot{\psi} + \cos^2(\Phi) \dot{\theta}^2 \right] \\
& + (x^2 + y^2) \left[\dot{\psi}^2 \cos^2(\Phi) \cos^2(\theta) - 2 \sin(\Phi) \cos(\Phi) \cos(\theta) \dot{\theta} \dot{\psi} + \sin^2(\Phi) \dot{\theta}^2 \right] \\
& + 2xy \left[\dot{\psi}^2 \sin(\Phi) \sin(\theta) \cos(\theta) + \dot{\psi} (\cos(\Phi) \sin(\theta) \dot{\theta} - \sin(\Phi) \cos(\theta) \dot{\Phi}) - \cos(\Phi) \dot{\Phi} \dot{\theta} \right] \\
& + 2xz \left[\dot{\psi}^2 \cos(\Phi) \sin(\theta) \cos(\theta) + \dot{\psi} (-\cos(\Phi) \cos(\theta) \dot{\Phi} - \sin(\Phi) \sin(\theta) \dot{\theta}) + \sin(\Phi) \dot{\Phi} \dot{\theta} \right] \\
& + 2yz \left[-\dot{\psi}^2 \sin(\Phi) \cos(\Phi) \cos^2(\theta) + \dot{\psi} (\sin^2(\Phi) \cos(\theta) \dot{\theta} - \cos^2(\Phi) \cos(\theta) \dot{\theta}) + \sin(\Phi) \cos(\Phi) \dot{\theta}^2 \right]
\end{aligned} \tag{2.1}$$

5.2.3 Expression de l'énergie cinétique

$$T = \frac{1}{2} m v^2$$

$$\begin{aligned}
T = & \frac{1}{2} \int (y^2 + z^2) dm \left[\dot{\psi}^2 \sin^2(\theta) - 2 \dot{\Phi} \dot{\psi} \sin(\theta) + \dot{\Phi}^2 \right] \\
& + \frac{1}{2} \int (x^2 + y^2) dm \left[\dot{\psi}^2 \cos^2(\Phi) \cos^2(\theta) - 2 \sin(\Phi) \cos(\Phi) \cos(\theta) \dot{\theta} \dot{\psi} + \sin^2(\Phi) \dot{\theta}^2 \right] \\
& + \frac{1}{2} \int (x^2 + z^2) dm \left[\dot{\psi}^2 \sin^2(\Phi) \cos^2(\theta) + 2 \sin(\Phi) \cos(\Phi) \cos(\theta) \dot{\theta} \dot{\psi} + \cos^2(\Phi) \dot{\theta}^2 \right] \\
& + \int xy dm \left[\dot{\psi}^2 \sin(\Phi) \sin(\theta) \cos(\theta) + \dot{\psi} (\cos(\Phi) \sin(\theta) \dot{\theta} - \sin(\Phi) \cos(\theta) \dot{\Phi}) - \cos(\Phi) \dot{\Phi} \dot{\theta} \right] \\
& + \int xz dm \left[\dot{\psi}^2 \cos(\Phi) \sin(\theta) \cos(\theta) + \dot{\psi} (-\cos(\Phi) \cos(\theta) \dot{\Phi} - \sin(\Phi) \sin(\theta) \dot{\theta}) + \sin(\Phi) \dot{\Phi} \dot{\theta} \right] \\
& + \int yz dm \left[-\dot{\psi}^2 \sin(\Phi) \cos(\Phi) \cos^2(\theta) + \dot{\psi} (\sin^2(\Phi) \cos(\theta) \dot{\theta} - \cos^2(\Phi) \cos(\theta) \dot{\theta}) + \sin(\Phi) \cos(\Phi) \dot{\theta}^2 \right]
\end{aligned} \tag{3.1}$$

Il apparaît alors directement les moments d'inertie qui sont les éléments diagonaux de la matrice d'inertie et les produits d'inertie qui sont les éléments non diagonaux. Cependant, nous avons l'hypothèse que le quadrirotor était parfaitement symétrique. Cela implique que les produits d'inertie sont nuls et donc que la matrice d'inertie du quadrirotor est diagonale.

En simplifiant l'équation (3.1), l'énergie cinétique vaut alors :

$$\boxed{T = \frac{1}{2} I_x (\dot{\Phi} - \dot{\psi} \sin(\theta))^2 + \frac{1}{2} I_y (\dot{\theta} \cos(\Phi) + \dot{\psi} \sin(\Phi) \cos(\theta))^2 + \frac{1}{2} I_z (\dot{\theta} \sin(\Phi) - \dot{\psi} \cos(\Phi) \cos(\theta))^2} \tag{3.2}$$

avec $I_x = \frac{1}{2} \int (y^2 + z^2) dm$, $I_y = \frac{1}{2} \int (x^2 + z^2) dm$, $I_z = \frac{1}{2} \int (x^2 + y^2) dm$

5.2.4 Expression de l'énergie potentielle

$$V = g \int (-\sin(\theta)x + \sin(\Phi) \cos(\theta)y + \cos(\Phi) \cos(\theta)z) dm$$

$$\boxed{V = \int x dm (-g \sin(\theta)) + \int y dm (g \sin(\Phi) \cos(\theta)) + \int z dm (g \cos(\Phi) \cos(\theta))} \quad (4.1)$$

5.2.5 Equation du roulis

$$\begin{cases} \Gamma_{\Phi} = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\Phi}} \right) - \frac{\partial L}{\partial \Phi} = \tau_x \\ L = T - V \end{cases}$$

$$\bullet \quad \frac{\partial L}{\partial \Phi} = \frac{\partial T}{\partial \Phi} - \frac{\partial V}{\partial \Phi}$$

$$\begin{aligned} \frac{\partial T}{\partial \Phi} &= I_y (-\dot{\theta} \sin(\Phi) + \dot{\psi} \cos(\Phi) \cos(\theta)) (\dot{\theta} \cos(\Phi) + \dot{\psi} \sin(\Phi) \cos(\theta)) \\ &\quad + I_z (\dot{\theta} \cos(\Phi) + \dot{\psi} \sin(\Phi) \cos(\theta)) (\dot{\theta} \sin(\Phi) - \dot{\psi} \cos(\Phi) \cos(\theta)) \end{aligned}$$

$$\begin{aligned} \frac{\partial T}{\partial \Phi} &= I_y \left[-\dot{\theta}^2 \cos(\Phi) \sin(\Phi) - \dot{\theta} \dot{\psi} \sin^2(\Phi) \cos(\theta) + \dot{\theta} \dot{\psi} \cos^2(\Phi) \cos(\theta) + \dot{\psi}^2 \cos^2(\theta) \sin(\Phi) \cos(\Phi) \right] \\ &\quad + I_z \left[\dot{\theta}^2 \cos(\Phi) \sin(\Phi) - \dot{\theta} \dot{\psi} \cos^2(\Phi) \cos(\theta) + \dot{\theta} \dot{\psi} \sin^2(\Phi) \cos(\theta) - \dot{\psi}^2 \cos^2(\theta) \sin(\Phi) \cos(\Phi) \right] \end{aligned}$$

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

$$\frac{\partial V}{\partial \Phi} = \int y dm (g \cos(\theta) \cos(\Phi)) + \int z dm (-g \sin(\Phi) \cos(\theta))$$

$$\frac{\partial V}{\partial \Phi} = g \cos(\theta) \left[\cos(\Phi) \int y dm - \sin(\Phi) \int z dm \right]$$

$$\begin{aligned} \frac{\partial L}{\partial \Phi} &= \dot{\psi}^2 \cos^2(\theta) \frac{\sin(2\Phi)}{2} (I_y - I_z) + \dot{\theta}^2 \frac{\sin(2\Phi)}{2} (I_z - I_y) + \dot{\theta} \dot{\psi} \cos(2\Phi) \cos(\theta) (I_y - I_z) \\ &\quad - g \cos(\theta) \left[\cos(\Phi) \int y dm - \sin(\Phi) \int z dm \right] \end{aligned} \quad (5.1)$$

$$\bullet \quad \frac{\partial L}{\partial \dot{\Phi}} = \frac{\partial T}{\partial \dot{\Phi}} - \frac{\partial V}{\partial \dot{\Phi}}$$

$$\frac{\partial V}{\partial \dot{\Phi}} = 0$$

$$\frac{\partial T}{\partial \dot{\Phi}} = I_x \left[\dot{\Phi} - \dot{\psi} \sin(\theta) \right]$$

$$\frac{\partial L}{\partial \dot{\Phi}} = I_x \left[\dot{\Phi} - \dot{\psi} \sin(\theta) \right]$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\Phi}} \right) = I_x \left[\ddot{\Phi} - \ddot{\psi} \sin(\theta) - \dot{\theta} \dot{\psi} \cos(\theta) \right] \quad (5.2)$$

En additionnant (5.1) et (5.2), on obtient :

$$\begin{aligned} \Gamma_{\Phi} = & \ddot{\Phi} I_x \\ & - \ddot{\psi} \sin(\theta) I_x \\ & - \dot{\theta} \dot{\psi} \cos(\theta) \left[\cos(2\Phi)(I_y - I_z) + I_x \right] \\ & - \dot{\psi}^2 \cos^2(\theta) \frac{\sin(2\Phi)}{2} (I_y - I_z) \\ & - \dot{\theta}^2 \frac{\sin(2\Phi)}{2} (I_z - I_y) \\ & + g \cos(\theta) \cos(\Phi) \int y dm \\ & - g \cos(\theta) \sin(\Phi) \int z dm \end{aligned} \quad (5.3)$$

En passant par les vitesses dans le repère solide, on obtient :

$$\frac{\partial L}{\partial \dot{\Phi}} = I_x \left[\dot{\Phi} - \dot{\psi} \sin(\theta) \right] = I_x \omega_x$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\Phi}} \right) = \dot{\omega}_x I_x$$

$$\frac{\partial T}{\partial \Phi} = I_y \omega_y (-\dot{\theta} \sin(\Phi) + \dot{\psi} \cos(\Phi) \cos(\theta)) - I_z \omega_z (\dot{\theta} \cos(\Phi) + \dot{\psi} \sin(\Phi) \cos(\theta))$$

$$\frac{\partial T}{\partial \Phi} = (I_y - I_z) \omega_y \omega_z$$

$$\begin{aligned} \Gamma_{\Phi} = & \dot{\omega}_x I_x + (I_z - I_y) \omega_y \omega_z \\ & + g \cos(\theta) \cos(\Phi) \int y dm \\ & - g \cos(\theta) \sin(\Phi) \int z dm \end{aligned} \quad (5.4)$$

5.2.6 Equation du tangage

$$\begin{cases} \Gamma_\theta = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \tau_y \\ L = T - V \end{cases}$$

$$\bullet \frac{\partial L}{\partial \theta} = \frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta}$$

$$\begin{aligned} \frac{\partial T}{\partial \theta} &= I_x (-\dot{\psi} \cos(\theta)) (\dot{\Phi} - \dot{\psi} \sin(\theta)) + I_y (-\dot{\psi} \sin(\Phi) \sin(\theta)) (\dot{\theta} \cos(\Phi) + \dot{\psi} \sin(\Phi) \cos(\theta)) \\ &\quad + I_z (\dot{\psi} \cos(\Phi) \sin(\theta)) (\dot{\theta} \sin(\Phi) - \dot{\psi} \cos(\Phi) \cos(\theta)) \end{aligned}$$

$$\begin{aligned} \frac{\partial T}{\partial \theta} &= I_x \left[-\dot{\psi} \dot{\Phi} \cos(\theta) + \dot{\psi}^2 \cos(\theta) \sin(\theta) \right] + I_y \left[-\dot{\psi} \dot{\theta} \cos(\Phi) \sin(\Phi) \sin(\theta) - \dot{\psi}^2 \sin^2(\Phi) \cos(\theta) \sin(\theta) \right] \\ &\quad + I_z \left[\dot{\psi} \dot{\theta} \sin(\Phi) \cos(\Phi) \sin(\theta) - \dot{\psi}^2 \cos^2(\Phi) \cos(\theta) \sin(\theta) \right] \end{aligned}$$

$$\frac{\partial T}{\partial \theta} = \dot{\psi}^2 \frac{\sin(2\theta)}{2} [I_x - \cos^2(\Phi) I_z - \sin^2(\Phi) I_y] - \dot{\psi} \dot{\Phi} \cos(\theta) I_x + \dot{\psi} \dot{\theta} \frac{\sin(2\Phi)}{2} \sin(\theta) [I_z - I_y]$$

$$\frac{\partial V}{\partial \theta} = \int x dm (-g \cos(\theta)) + \int y dm (-g \sin(\Phi) \sin(\theta)) + \int z dm (-g \cos(\Phi) \sin(\theta))$$

$$\begin{aligned} \frac{\partial L}{\partial \theta} &= \dot{\psi}^2 \frac{\sin(2\theta)}{2} [I_x - \cos^2(\Phi) I_z - \sin^2(\Phi) I_y] - \dot{\psi} \dot{\Phi} \cos(\theta) I_x + \dot{\psi} \dot{\theta} \frac{\sin(2\Phi)}{2} \sin(\theta) [I_z - I_y] \quad (6.1) \\ &\quad + \int x dm (g \cos(\theta)) + \int y dm (g \sin(\Phi) \sin(\theta)) + \int z dm (g \cos(\Phi) \sin(\theta)) \end{aligned}$$

$$\bullet \frac{\partial L}{\partial \dot{\theta}} = \frac{\partial T}{\partial \dot{\theta}} - \frac{\partial V}{\partial \dot{\theta}}$$

$$\frac{\partial V}{\partial \dot{\theta}} = 0$$

$$\frac{\partial T}{\partial \dot{\theta}} = I_y \cos(\Phi) \left[\dot{\theta} \cos(\Phi) + \dot{\psi} \sin(\Phi) \cos(\theta) \right] + I_z \sin(\Phi) \left[\dot{\theta} \sin(\Phi) - \dot{\psi} \cos(\Phi) \cos(\theta) \right]$$

$$\frac{\partial T}{\partial \dot{\theta}} = I_y \left[\dot{\theta} \cos^2(\Phi) + \dot{\psi} \frac{\sin(2\Phi)}{2} \cos(\theta) \right] + I_z \left[\dot{\theta} \sin^2(\Phi) - \dot{\psi} \frac{\sin(2\Phi)}{2} \cos(\theta) \right]$$

$$\begin{aligned}
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) &= I_y \left[\ddot{\theta} \cos^2(\Phi) - 2 \dot{\theta} \dot{\Phi} \sin(\Phi) \cos(\Phi) + \ddot{\psi} \cos(\theta) \frac{\sin(2\Phi)}{2} - \dot{\psi} \dot{\theta} \sin(\theta) \frac{\sin(2\Phi)}{2} + \dot{\psi} \dot{\Phi} \cos(\theta) \cos(2\Phi) \right] \\
&\quad + I_z \left[\ddot{\theta} \sin^2(\Phi) + 2 \dot{\theta} \dot{\Phi} \sin(\Phi) \cos(\Phi) - \ddot{\psi} \cos(\theta) \frac{\sin(2\Phi)}{2} + \dot{\psi} \dot{\theta} \sin(\theta) \frac{\sin(2\Phi)}{2} - \dot{\psi} \dot{\Phi} \cos(\theta) \cos(2\Phi) \right] \\
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) &= \ddot{\theta} \left[\cos^2(\Phi) I_y + \sin^2(\Phi) I_z \right] + \ddot{\psi} \cos(\theta) \frac{\sin(2\Phi)}{2} [I_y - I_z] \\
&\quad + \dot{\theta} \dot{\Phi} \sin(2\Phi) [I_z - I_y] + \dot{\psi} \dot{\Phi} \cos(\theta) \cos(2\Phi) [I_y - I_z] + \dot{\psi} \dot{\theta} \sin(\theta) \frac{\sin(2\Phi)}{2} [I_z - I_y]
\end{aligned} \tag{6.2}$$

On soustrait (6.1) à (6.2) :

$$\begin{aligned}
\Gamma_{\theta} &= \ddot{\theta} \left[\cos^2(\Phi) I_y + \sin^2(\Phi) I_z \right] \\
&\quad + \ddot{\psi} \cos(\theta) \frac{\sin(2\Phi)}{2} [I_y - I_z] \\
&\quad + \dot{\theta} \dot{\Phi} \sin(2\Phi) [I_z - I_y] \\
&\quad + \dot{\psi} \dot{\Phi} \cos(\theta) \left[\cos(2\Phi) (I_y - I_z) + I_x \right] \\
&\quad - \dot{\psi}^2 \frac{\sin(2\theta)}{2} [I_x - \cos^2(\Phi) I_z - \sin^2(\Phi) I_y] \\
&\quad - \int x dm (g \cos(\theta)) \\
&\quad - \int y dm (g \sin(\Phi) \sin(\theta)) \\
&\quad - \int z dm (g \cos(\Phi) \sin(\theta))
\end{aligned} \tag{6.3}$$

En passant par les vitesses dans le repère solide, on obtient :

$$\begin{aligned}
\frac{\partial L}{\partial \dot{\theta}} &= I_y \cos(\Phi) \omega_y - I_z \sin(\Phi) \omega_z \\
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) &= I_y \left[-\dot{\Phi} \sin(\Phi) \omega_y + \cos(\Phi) \dot{\omega}_y \right] - I_z \left[\dot{\Phi} \cos(\Phi) \omega_z + \sin(\Phi) \dot{\omega}_z \right] \\
\frac{\partial T}{\partial \theta} &= -\dot{\psi} \cos(\theta) \omega_x I_x - \dot{\psi} \sin(\Phi) \sin(\theta) \omega_y I_y - \dot{\psi} \cos(\Phi) \sin(\theta) \omega_z I_z \\
\Gamma_{\theta} &= I_y \cos(\Phi) \dot{\omega}_y - I_z \sin(\Phi) \dot{\omega}_z + \dot{\psi} \cos(\theta) \omega_x I_x - \left[\dot{\Phi} - \dot{\psi} \sin(\theta) \right] \sin(\Phi) \omega_y I_y \\
&\quad - \left[\dot{\Phi} - \dot{\psi} \sin(\theta) \right] \cos(\Phi) \omega_z I_z - \int x dm (g \cos(\theta)) - \int y dm (g \sin(\Phi) \sin(\theta)) - \int z dm (g \cos(\Phi) \sin(\theta))
\end{aligned}$$

$$\Gamma_\theta = I_y \cos(\Phi) \dot{\omega}_y - I_z \sin(\Phi) \dot{\omega}_z + \dot{\psi} \cos(\theta) \omega_x I_x - \sin(\Phi) \omega_x \omega_y I_y - \cos(\Phi) \omega_x \omega_z I_z$$

$$- \int x dm (g \cos(\theta)) - \int y dm (g \sin(\Phi) \sin(\theta)) - \int z dm (g \cos(\Phi) \sin(\theta))$$

Or $\dot{\psi} \cos(\theta) = \sin(\Phi) \omega_y + \cos(\Phi) \omega_z$

$$\boxed{\begin{aligned} \Gamma_\theta = & -\sin(\Phi) \left[I_z \dot{\omega}_z + (I_y - I_x) \omega_x \omega_y \right] \\ & + \cos(\Phi) \left[I_y \dot{\omega}_y + (I_x - I_z) \omega_x \omega_z \right] \\ & - \int x dm (g \cos(\theta)) \\ & - \int y dm (g \sin(\Phi) \sin(\theta)) \\ & - \int z dm (g \cos(\Phi) \sin(\theta)) \end{aligned}} \quad (6.4)$$

5.2.7 Equation du lacet

$$\begin{cases} \Gamma_\psi = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \frac{\partial L}{\partial \psi} = \tau_z .. \\ L = T - V \end{cases}$$

$$\bullet \frac{\partial L}{\partial \psi} = \frac{\partial T}{\partial \psi} - \frac{\partial V}{\partial \psi}$$

$$\left. \begin{aligned} \frac{\partial T}{\partial \psi} &= 0 \\ \frac{\partial V}{\partial \psi} &= 0 \end{aligned} \right\} \Rightarrow \frac{\partial L}{\partial \psi} = 0 \quad (7.1)$$

$$\bullet \frac{\partial L}{\partial \dot{\psi}} = \frac{\partial T}{\partial \dot{\psi}} - \frac{\partial V}{\partial \dot{\psi}}$$

$$\frac{\partial V}{\partial \dot{\psi}} = 0$$

$$\begin{aligned} \frac{\partial T}{\partial \dot{\psi}} = & I_x (-\sin(\theta)) (\dot{\Phi} - \dot{\psi} \sin(\theta)) + I_y (\sin(\Phi) \cos(\theta)) (\dot{\theta} \cos(\Phi) + \dot{\psi} \sin(\Phi) \cos(\theta)) \\ & + I_z (-\cos(\Phi) \cos(\theta)) (\dot{\theta} \sin(\Phi) - \dot{\psi} \cos(\Phi) \cos(\theta)) \end{aligned}$$

$$\begin{aligned}
\frac{\partial T}{\partial \dot{\psi}} &= I_x \left[-\dot{\Phi} \sin(\theta) + \dot{\psi} \sin^2(\theta) \right] + I_y \left[\dot{\theta} \cos(\Phi) \sin(\Phi) \cos(\theta) + \dot{\psi} \sin^2(\Phi) \cos^2(\theta) \right] \\
&\quad + I_z \left[\dot{\psi} \cos^2(\Phi) \cos^2(\theta) - \dot{\theta} \cos(\Phi) \sin(\Phi) \cos(\theta) \right] \\
\frac{\partial T}{\partial \dot{\psi}} &= I_x \left[-\dot{\Phi} \sin(\theta) + \dot{\psi} \sin^2(\theta) \right] + I_y \left[\dot{\theta} \frac{\sin(2\Phi)}{2} \cos(\theta) + \dot{\psi} \sin^2(\Phi) \cos^2(\theta) \right] \\
&\quad + I_z \left[\dot{\psi} \cos^2(\Phi) \cos^2(\theta) - \dot{\theta} \frac{\sin(2\Phi)}{2} \cos(\theta) \right] \\
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) &= I_x \left[\ddot{\psi} \sin^2(\theta) + \dot{\psi} \dot{\theta} \sin(2\theta) - \dot{\Phi} \sin(\theta) - \dot{\Phi} \dot{\theta} \cos(\theta) \right] \\
&\quad + I_y \left[\dot{\theta} \frac{\sin(2\Phi)}{2} \cos(\theta) - \dot{\theta}^2 \frac{\sin(2\Phi)}{2} \sin(\theta) + \dot{\Phi} \dot{\theta} \cos(\theta) \cos(2\Phi) + \dot{\psi} \cos^2(\theta) \sin^2(\Phi) - \dot{\psi} \dot{\theta} \sin(2\theta) \sin^2(\Phi) + \dot{\psi} \dot{\Phi} \sin(2\theta) \cos^2(\Phi) \right] \\
&\quad + I_z \left[\dot{\psi} \cos^2(\Phi) \cos^2(\theta) - \dot{\psi} \dot{\Phi} \sin(2\Phi) \cos^2(\theta) - \dot{\psi} \dot{\theta} \sin(2\theta) \cos^2(\Phi) + \dot{\theta}^2 \frac{\sin(2\Phi)}{2} \sin(\theta) - \dot{\Phi} \dot{\theta} \cos(\theta) \cos(2\Phi) - \dot{\theta} \frac{\sin(2\Phi)}{2} \cos(\theta) \right] \quad (7.2)
\end{aligned}$$

En soustrayant (7.1) à (7.2), on a :

$$\begin{aligned}
\Gamma_{\psi} &= \ddot{\psi} \left[I_x \sin^2(\theta) + \cos^2(\theta) (\sin^2(\Phi) I_y + \cos^2(\Phi) I_z) \right] \\
&\quad - \ddot{\Phi} \sin(\theta) I_x + \dot{\theta}^2 \frac{\sin(2\Phi)}{2} \cos(\theta) [I_y - I_z] \\
&\quad + \dot{\psi} \dot{\theta} \sin(2\theta) [I_x - \sin^2(\Phi) I_y - \cos^2(\Phi) I_z] \\
&\quad - \dot{\Phi} \dot{\theta} \cos(\theta) [I_x + \cos(2\Phi) (I_z - I_y)] \\
&\quad + \dot{\psi} \dot{\Phi} \sin(2\Phi) \cos^2(\theta) [I_y - I_z] \\
&\quad + \dot{\theta}^2 \frac{\sin(2\Phi)}{2} \sin(\theta) [I_z - I_y] \quad (7.3)
\end{aligned}$$

En passant par les angles dans le repère solide on obtient :

$$\begin{aligned}
\frac{\partial L}{\partial \dot{\psi}} &= -I_x \sin(\theta) \omega_x + I_y \sin(\Phi) \cos(\theta) \omega_y + I_z \cos(\Phi) \cos(\theta) \omega_z \\
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) &= -I_x \dot{\theta} \cos(\theta) \omega_x - I_x \sin(\theta) \dot{\omega}_x + I_y \dot{\Phi} \cos(\Phi) \cos(\theta) \omega_y \\
&\quad - I_y \dot{\theta} \sin(\Phi) \sin(\theta) \omega_y + I_y \sin(\Phi) \cos(\theta) \dot{\omega}_y + I_z \cos(\Phi) \cos(\theta) \dot{\omega}_z \\
&\quad - I_z \dot{\Phi} \sin(\Phi) \cos(\theta) \omega_z - I_z \dot{\theta} \cos(\Phi) \sin(\theta) \omega_z
\end{aligned}$$

$$\begin{aligned}
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) &= -I_x \sin(\theta) \dot{\omega}_x + I_y \sin(\Phi) \cos(\theta) \dot{\omega}_y + I_z \cos(\Phi) \cos(\theta) \dot{\omega}_z \\
&\quad -I_x \cos(\theta) (\omega_y \cos(\Phi) - \omega_z \sin(\Phi)) \omega_x + I_y (\cos(\Phi) \cos(\theta) \omega_x + \sin(\theta) \omega_z) \omega_y \\
&\quad -I_z (\sin(\Phi) \cos(\theta) \omega_x + \sin(\theta) \omega_y) \omega_z
\end{aligned}$$

$$\boxed{
\begin{aligned}
\Gamma_{\psi} &= -\sin(\theta) \left[I_x \dot{\omega}_x + (I_z - I_y) \omega_y \omega_z \right] \\
&\quad + \cos(\Phi) \cos(\theta) \left[I_z \dot{\omega}_z + (I_y - I_x) \omega_x \omega_y \right] \\
&\quad + \sin(\Phi) \cos(\theta) \left[I_y \dot{\omega}_y + (I_x - I_z) \omega_x \omega_z \right]
\end{aligned}
} \quad (7.4)$$

5.2.8 Expression des forces non conservatives

Portance

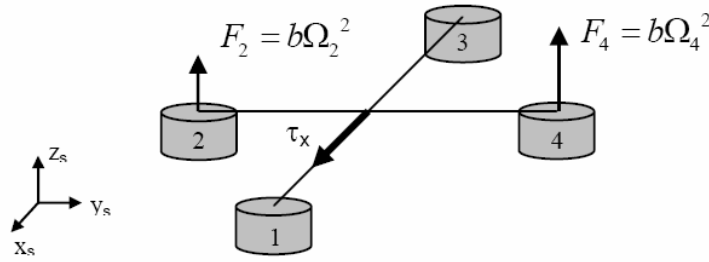


Fig. 5.3 Moment du à la poussée des moteurs

La portance des moteurs crée en direction des axes X et Y, des couples :

$$\tau_x = bl(\Omega_4^2 - \Omega_2^2) \quad \tau_y = bl(\Omega_3^2 - \Omega_1^2)$$

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.

Traînée

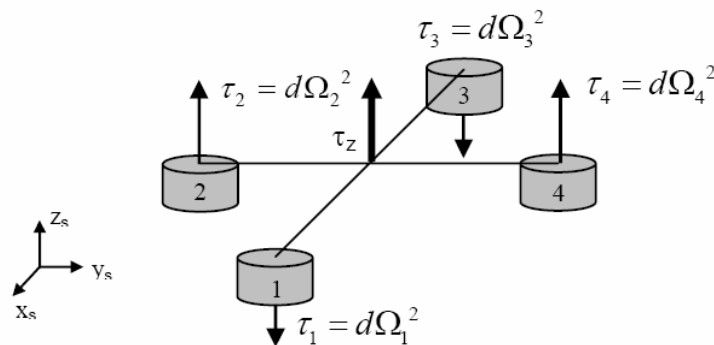


Fig. 5.4 Moment du à la traînée des hélices

La traînée des hélices crée un couple vertical.

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

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

Effet gyroscopique

Les effets gyroscopiques sont dû aux hélices lors d'une rotation autour de l'axe X ou Y du repère du solide.

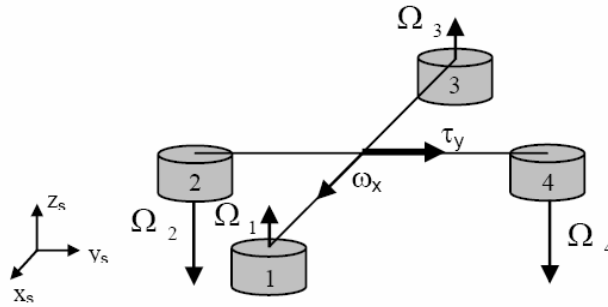


Fig. 5.5 Effet gyroscopique

En effectuant le produit vectoriel, on calcule :

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

En additionnant tous les couples dû aux différents effets, on obtient :

$$\begin{cases} \tau_x = bl(\Omega_4^2 - \Omega_2^2) + I_{rotor} \omega_y (\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4) \\ \tau_y = bl(\Omega_3^2 - \Omega_1^2) + I_{rotor} \omega_x (-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4) \\ \tau_z = d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \end{cases} \quad (8.1)$$

5.2.9 Expression des accélérations angulaires

En considérant l'hypothèse des petits angles les équations du mouvement deviennent :

$$\begin{aligned} \Gamma_\Phi &= I_x \dot{\omega}_x + (I_z - I_y) \omega_y \omega_z \\ \Gamma_\theta &= I_y \dot{\omega}_y + (I_x - I_z) \omega_x \omega_z \\ \Gamma_\psi &= I_z \dot{\omega}_z + (I_y - I_x) \omega_x \omega_y \end{aligned} \quad (9.1)$$

Donc en isolant les accélérations dans les équations (9.1), on obtient :

$$\begin{aligned}
\ddot{\Phi} &= \frac{I_{rotor} \dot{\theta}(\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4)}{I_x} + \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} + \frac{bl(\Omega_4^2 - \Omega_2^2)}{I_x} \\
\ddot{\theta} &= \frac{I_{rotor} \dot{\Phi}(-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4)}{I_y} + \frac{(I_z - I_x)}{I_y} \dot{\Phi} \dot{\psi} + \frac{bl(\Omega_3^2 - \Omega_1^2)}{I_y} \\
\ddot{\psi} &= \frac{(I_x - I_y)}{I_z} \dot{\theta} \dot{\Phi} + \frac{d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)}{I_z}
\end{aligned} \tag{9.2}$$

5.3 Effet de l'angle des supports moteur

Chacun des quatre supports moteurs fait un angle α avec la verticale, vers le centre du drone. Cet angle permet au drone de se stabiliser plus facilement en roulis et en tangage.

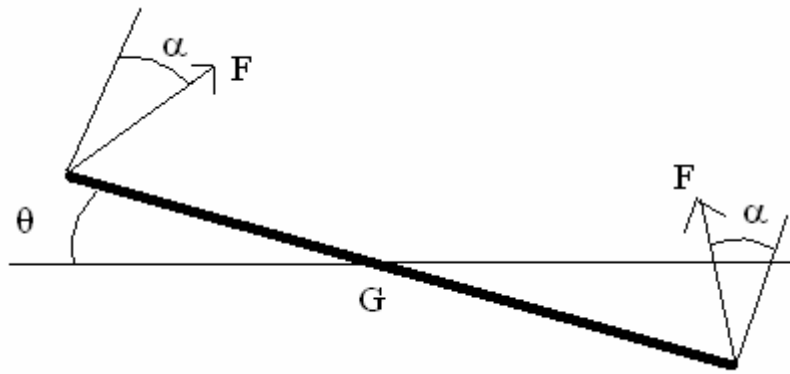


Fig. 5.6 Angle des supports moteur

La portance du moteur gauche s'écrit : $F_{z_g} = F_g \cos(\alpha + \theta)$, et celle du moteur droit : $F_{z_d} = F_d \cos(\alpha - \theta)$.

Le moment engendré par cette différence de portance est alors :

$$\tau = l\Delta F_z = l(F_d \cos(\alpha - \theta) - F_g \cos(\alpha + \theta))$$

Or $F = b\Omega^2$

Les moments selon les axes X et Y deviennent alors :

$$\tau_x = bl(\Omega_4^2 \cos(\alpha - \Phi) - \Omega_2^2 \cos(\alpha + \Phi)) \quad \tau_y = bl(\Omega_3^2 \cos(\alpha - \theta) - \Omega_1^2 \cos(\alpha + \theta))$$

On obtient alors :

$$\begin{aligned}
\ddot{\Phi} &= \frac{I_{rotor} \dot{\theta}(\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4)}{I_x} + \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} + \frac{bl(\Omega_4^2 \cos(\alpha - \Phi) - \Omega_2^2 \cos(\alpha + \Phi))}{I_x} \\
\ddot{\theta} &= \frac{I_{rotor} \dot{\Phi}(-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4)}{I_y} + \frac{(I_z - I_x)}{I_y} \dot{\Phi} \dot{\psi} + \frac{bl(\Omega_3^2 \cos(\alpha - \theta) - \Omega_1^2 \cos(\alpha + \theta))}{I_y} \\
\ddot{\psi} &= \frac{(I_x - I_y)}{I_z} \dot{\theta} \dot{\Phi} + \frac{d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)}{I_z}
\end{aligned} \tag{10.1}$$

On note que plus l'angle d'inclinaison des moteurs sera élevé, plus l'effet stabilisant sera important. Cependant, la portance du drone s'en retrouve diminuée ainsi que sa manoeuvrabilité. Il convient donc de trouver un compromis entre portance, stabilité et manoeuvrabilité.

5.4 Calcul des différents paramètres

5.4.1 Calcul des moments d'inerties du drone

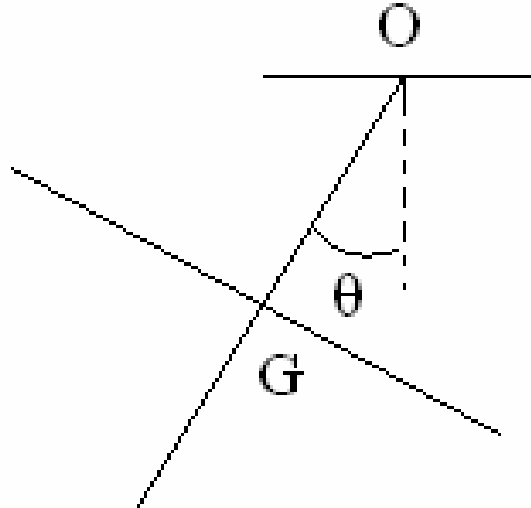


Fig. 5.7 Expérience du « pendule »

En considérant que les frottements sont négligeables, l'énergie mécanique se conserve.

L'énergie potentielle s'écrit : $E_p = mgh = mgl(1 - \cos(\theta))$

L'énergie cinétique s'écrit : $E_c = \frac{1}{2}mv^2 = \frac{1}{2}I_o \dot{\theta}^2$ où I_o est le moment d'inertie en O par rapport à l'axe perpendiculaire au plan.

Or $E_m = E_c + E_p$

$$dE_m = 0 \Leftrightarrow d(E_c + E_p) = 0$$

$$\Leftrightarrow I_o \ddot{\theta} + mgl \sin(\theta) = 0$$

En faisant l'approximation des petits angles, on obtient : $\ddot{\theta} + \frac{mgl}{I_o} \theta = 0$.

Les solutions de cette équation harmonique sont de type $A \cos(\omega t) + B \sin(\omega t)$ avec $\omega^2 = \frac{mgl}{I_o}$.

$$\text{Donc } I_o = \frac{mglT^2}{4\pi^2}.$$

D'après le théorème de Huygens, $I_G = I_o - ml^2$.

Finalement,

$$I_G = \frac{mglT^2}{4\pi^2} - ml^2$$

En effectuant l'expérience, on trouve :

$$\begin{aligned} I_x &= 6,22.10^{-2} \text{ kg.m}^2 \\ I_y &= 7,33.10^{-2} \text{ kg.m}^2 \\ I_z &= 9,64.10^{-2} \text{ kg.m}^2 \end{aligned}$$

5.4.2 Calcul du coefficient de poussée b

La force générée par l'ensemble moteur et hélice est de la forme $F_i = b\Omega_i^2$.

A pleine puissance, les moteurs ont une vitesse de rotation de 10800 tr/min et peuvent soulever 630 grammes chacun.

Donc $F_i = 6,18N$ et $\Omega_i = 1131 \text{ rad/s}$.

On obtient alors $b = 4,83.10^{-6} \text{ kg.m.rad}^{-2}$.

5.4.3 Calcul du coefficient de traînée d

Pour calculer le coefficient d de traînée, on effectue l'expérience suivante : on pose le drone sur un support permettant une rotation libre selon l'axe vertical c'est-à-dire le lacet, puis on entre une consigne de puissance connue dans deux des quatre moteurs (ici les moteurs 1 et 3) et on chronomètre le temps mis par le drone pour effectuer un quart de tour.

La relation entre l'angle de lacet et les vitesses des moteurs est la suivante :

$$\ddot{\psi} = \frac{d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)}{I_z}$$

Le drone étant libre seulement selon l'axe de lacet, il ne peut y avoir de roulis ou de tangage,

ce qui implique que le terme $\frac{(I_x - I_y)}{I_z} \dot{\theta} \dot{\Phi}$ soit nul.

On a alors : $\ddot{\psi} = \frac{d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)}{I_z}$.

En intégrant à deux reprises, on obtient la loi de l'angle de lacet (en considérant les conditions initiales nulles, ce qui est le cas si on rentre des échelons comme consignes moteurs).

$$\psi(t) = \frac{d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)}{I_z} \frac{t^2}{2}$$

Lors de l'expérience, on met les moteurs 1 et 3 à mi-puissance, c'est-à-dire 5400 tr/min et les moteurs 2 et 4 à l'arrêt. On effectue un quart de tour.

$$\frac{\pi}{2} = \frac{d\Omega^2 T^2}{I_z}$$

$$d = \frac{\pi I_z}{2\Omega^2 T^2}$$

On obtient alors $d = 2,39.10^{-8} \text{ kg.m}^2.\text{rad}^{-2}$.

6. Modèle SimuLink du quadrirotor

Connaissant les relations entre les poussées des différents moteurs et les accélérations angulaires. On peut établir un modèle sous SimuLink.

Le but est de déterminer le type de correcteurs à mettre en place afin d'assurer une bonne stabilisation du drone sur les trois axes de lacet, tangage et roulis.

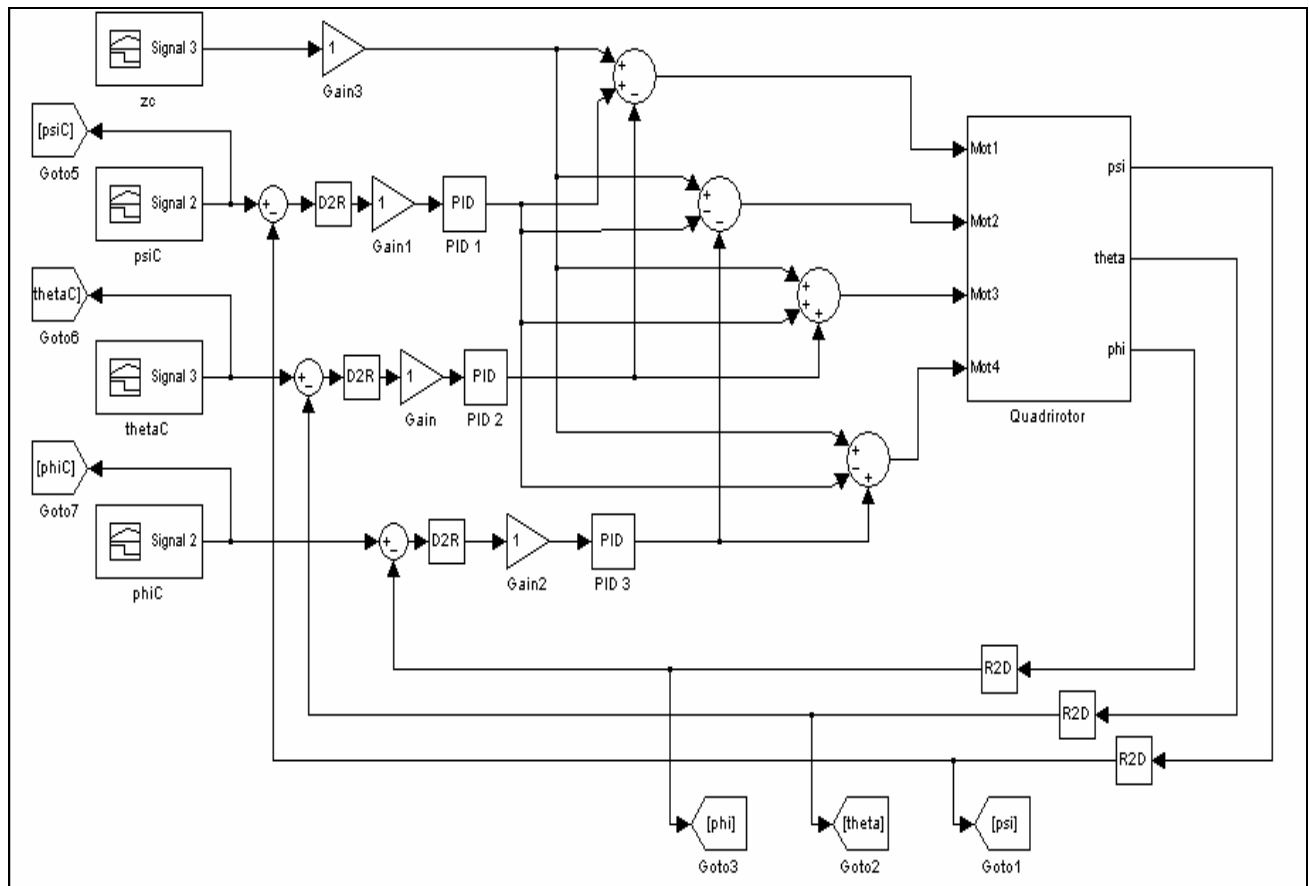


Figure 6.1 Schéma du modèle SimuLink

Le drone est piloté grâce à quatre consignes : les consignes de puissance z_c (qui permet de contrôler l'altitude), de roulis ϕ_iC , de tangage θ_iC et de lacet ψ_iC . Ces consignes sont ensuite envoyées aux différents moteurs. La consigne de puissance est envoyée à tous les moteurs, tout comme le lacet. Le tangage et le roulis sont obtenus en contrôlant seulement deux moteurs (1 et 3 pour le tangage et 2 et 4 pour le roulis).

Le sous-système Quadrirotor modélise les équations mécaniques explicitées au point 5.3.

Les correcteurs implémentés sont des correcteurs Proportionnel-Intégral-Dérivé (PID).

Les différents gains que l'on peut observer permettent de convertir les angles (radians) en tensions (volts) qui sont envoyées aux variateurs. Les valeurs de ces gains nous sont inconnues. Cependant, il est toujours possible d'étudier le type de correcteurs à mettre en place afin d'obtenir une dynamique satisfaisante.

6.1 Fonction de transfert de l'ensemble moteur-variateur

La fonction de transfert d'un moteur électrique est du second ordre et s'écrit :

$$H(p) = \frac{K}{K^2 + (R + Lp)(f + Jp)}$$

K : gain du moteur en volt.s/rad

R : résistance interne du moteur

L : inductance

f : frottements

J : inertie du rotor

Cependant, on peut approximer cette fonction de transfert par un premier ordre. En effet, la résistance d'un moteur est de l'ordre du ohm ou de la dizaine d'ohms alors que l'inductance est de l'ordre du milliHenry. De même, les frottements d'un moteur électrique sont négligeables devant son inertie. On obtient ainsi :

$$H(p) = \frac{K}{K^2 + RJp}$$

$$H(p) = \frac{K}{1 + \tau p}$$

K : gain du moteur en rad/s/volt

τ : constante de temps de l'ensemble moteur+variateur.

Le gain du moteur nous est donné par le constructeur : $K = 1500 \text{ tr / min / volt}$.

On détermine la constante de temps en effectuant un enregistrement sonore de la réponse du moteur à un échelon de tension.

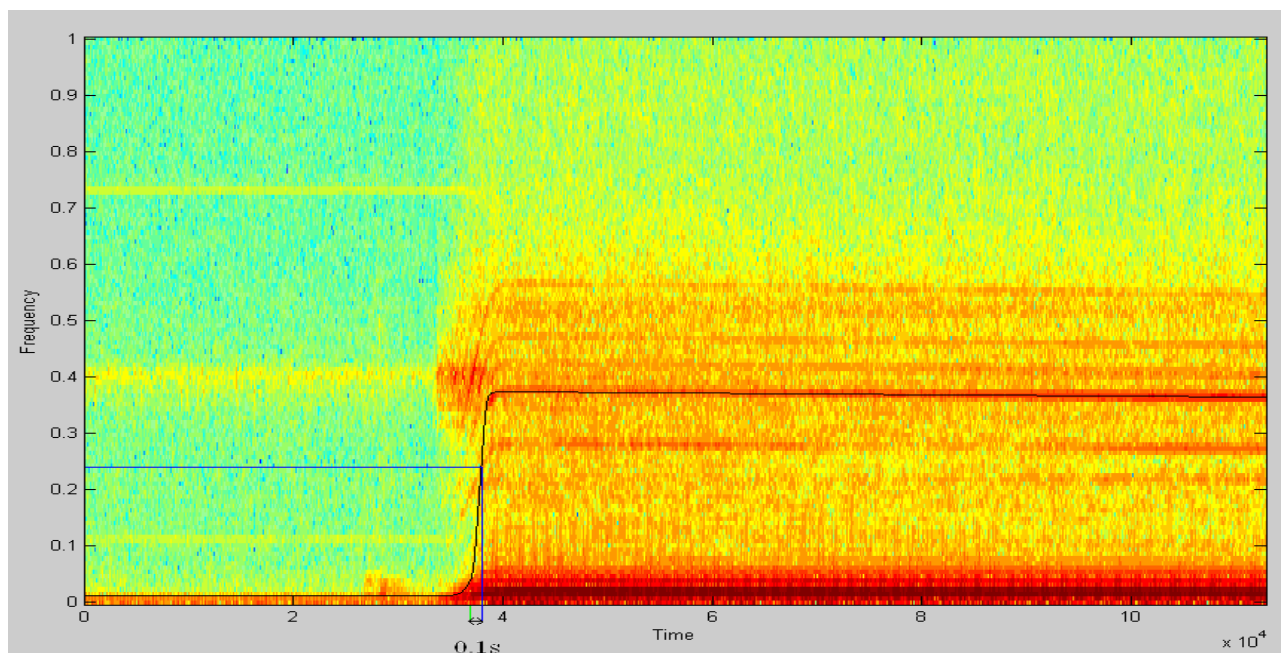


Figure 6.2 Réponse indicielle d'un moteur

On détermine alors que $\tau = 0.1 \text{ s}$.

$$H(p) = \frac{157.08}{1 + 0.1p}$$

6.2 Choix des correcteurs pour un angle de support moteur nul

6.2.1 Influence des effets d'inerties et des effets gyroscopiques par rapport aux effets moteurs

Couplage du système

Si on se réfère aux équations mécaniques, on remarque que si on soumet le drone à une consigne de roulis ou de tangage, cela induit un mouvement de lacet. En effet, on pilote le roulis et le tangage du drone en agissant sur les moteurs de façon symétriques et comme la traînée est une fonction quadratique de la vitesse de rotation du moteur, il en résulte une composante non nulle pour la traînée.

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

$$\text{Commande de tangage : } \begin{cases} \Omega_1 = \omega_0 - a \\ \Omega_3 = \omega_0 + a \end{cases}$$

$$\text{Commande de roulis : } \begin{cases} \Omega_2 = \omega_0 - b \\ \Omega_4 = \omega_0 + b \end{cases}$$

$$\tau_z = d(a^2 - b^2).$$

Les commandes de roulis et de tangage interviennent aussi sur le lacet à travers du terme d'inertie.

De la même manière, le lacet et le roulis (respectivement le tangage) agissent sur le tangage (resp. le roulis) grâce aux effets d'inertie et aux effets gyroscopiques. Ainsi les trois rotations sont clairement couplées.

Echelons non simultanés

On soumet le système à trois échelons non simultanés. On obtient pour des correcteurs de type Proportionnel-Dérivé (P+Td*s), les courbes suivantes :

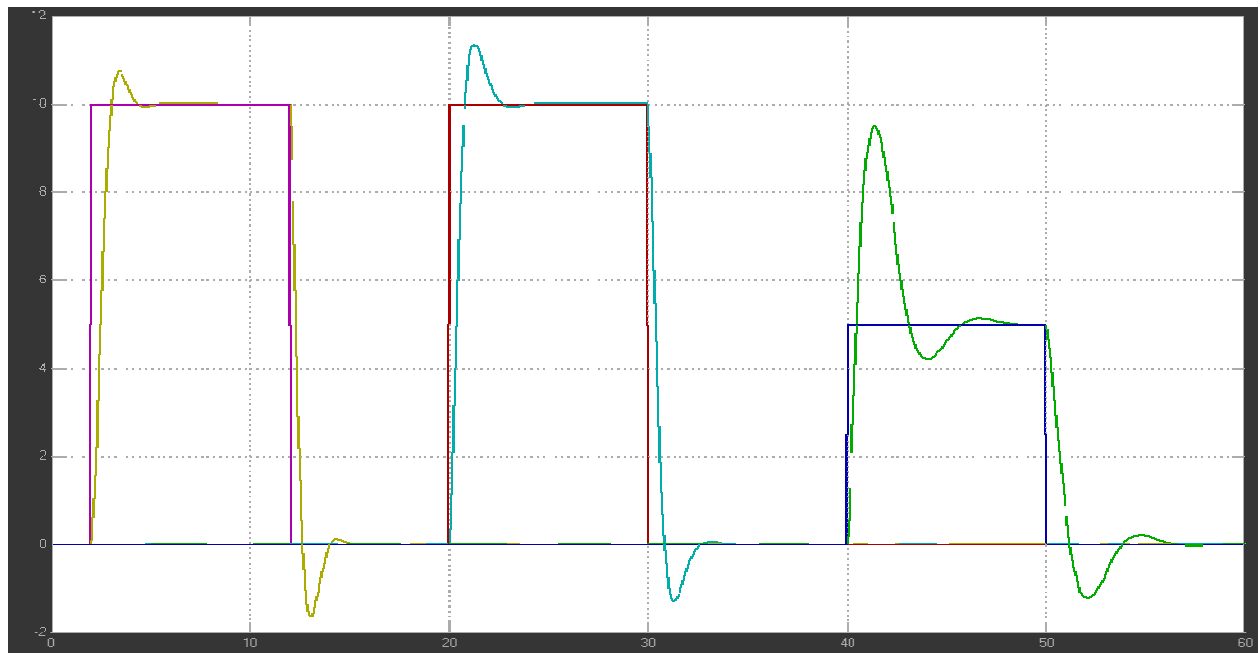
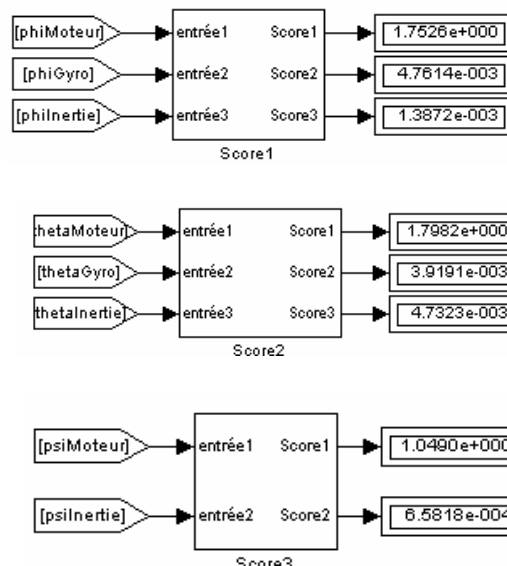


Figure 6.3 Echelons non simultanés
 Lacet : $P=12$, $Td=10$ (bleu, vert), Tangage : $P=0.4$, $Td=0.3$ (rouge, cyan),
 Roulis : $P=0.5$, $Td=0.3$ (violet, jaune)

Les consignes étant non simultanées, les effets d'inertie et les effets gyroscopiques sont donc quasiment nuls. En effet, si on calcule l'intégrale de l'erreur sur la durée de la simulation, on trouve les résultats suivants :



Echelons simultanés

Afin d'étudier l'influence des autres effets que ceux dus aux moteurs, il convient de commander trois échelons simultanés.

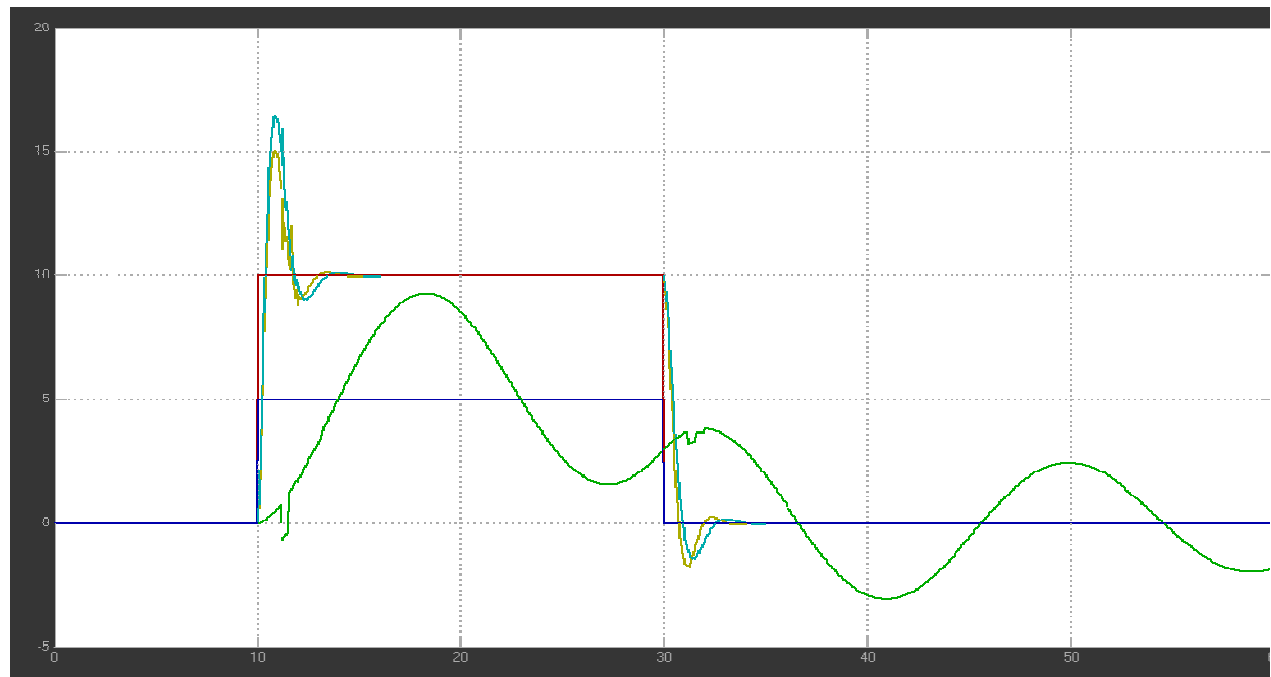


Figure 6.4 Echelons simultanés

Lacet : $P=1$, $Td=0.5$ (bleu, vert), Tangage : $P=0.5$, $Td=0.3$ (rouge, cyan)

Roulis : $P=0.4$, $Td=0.3$ (violet, jaune)

Roulis

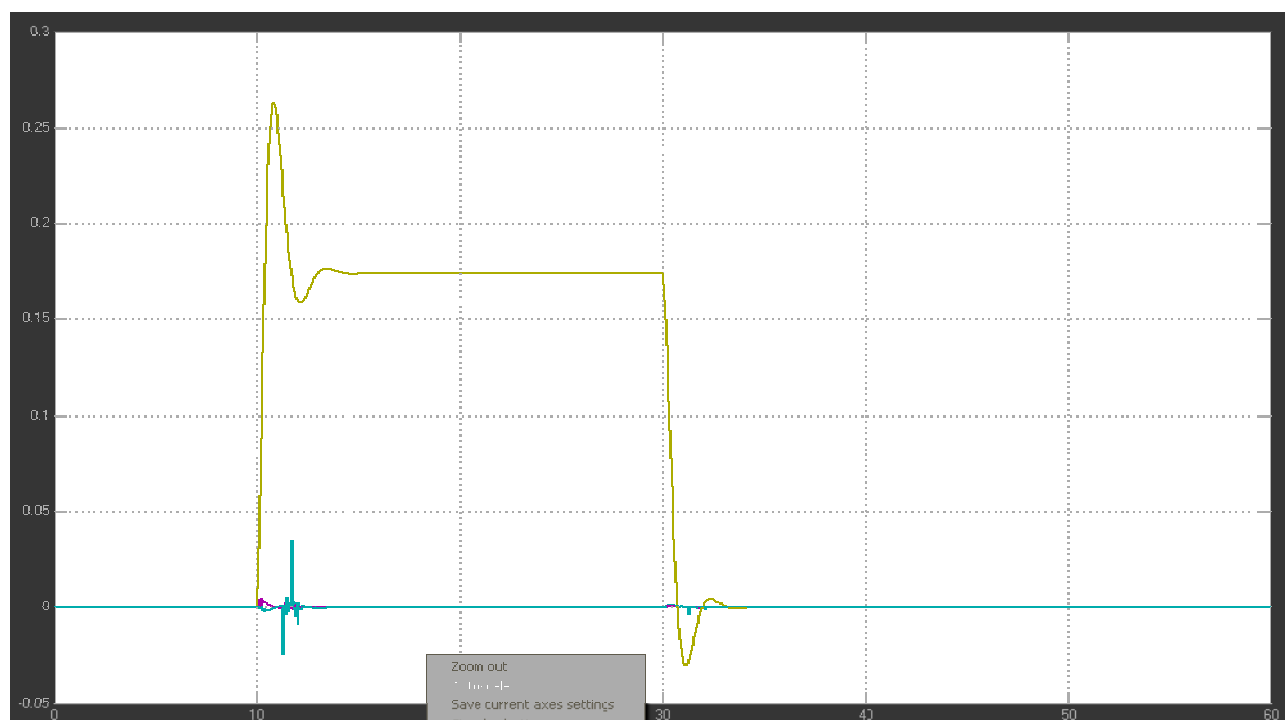
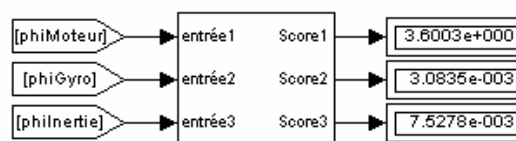


Figure 6.5 Influence des différents effets sur le roulis
(jaune : poussée, violet : effet gyroscopique, cyan : effet d'inertie)

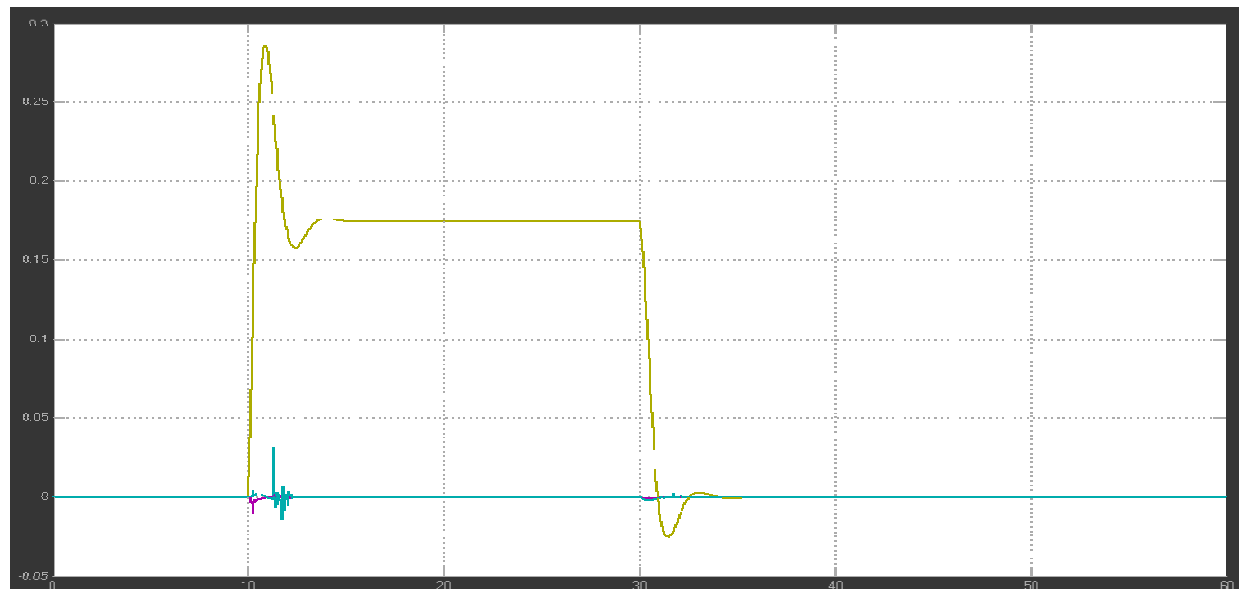
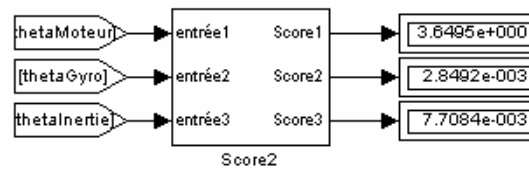
Tangage

Figure 6.6 Influence des différents effets sur le tangage
(jaune : poussée, violet : effet gyroscopique, cyan : effet d'inertie)

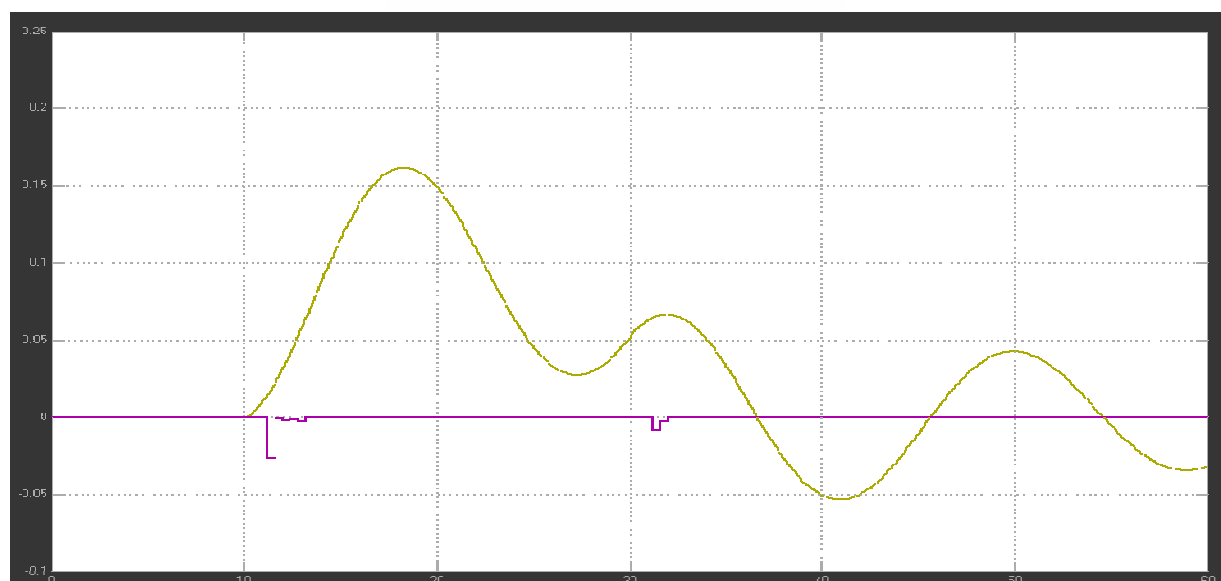
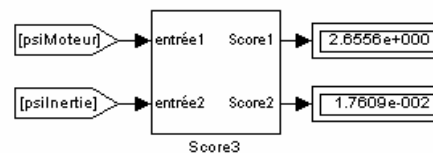
Lacet

Figure 6.7 Influence des différents effets sur le lacet
(jaune : traînée, violet : effet d'inertie)

On remarque que les effets d'inertie et les effets gyroscopiques sont négligeables par rapport à la poussée et à la traînée des moteurs selon les trois axes. Cela a pour effet de découpler le problème : si une consigne de roulis ou de tangage implique un mouvement de lacet, une consigne de lacet n'aura quasiment aucun effet sur les deux autres axes de rotation.

Dans la suite, on s'intéressera uniquement à la poussée et à la traînée des moteurs.

6.2.2 Choix du correcteur

D'après ce qui a été vu dans la partie précédente, les correcteurs PID peuvent être réglés de manière indépendante (sauf pour le lacet). On réglera tout d'abord les correcteurs stabilisant le roulis et le tangage puis celui du lacet, dans le cas de trois échelons simultanés. En effet, c'est dans le cas d'échelons simultanés que le lacet induit par le roulis et le tangage sera le plus important.

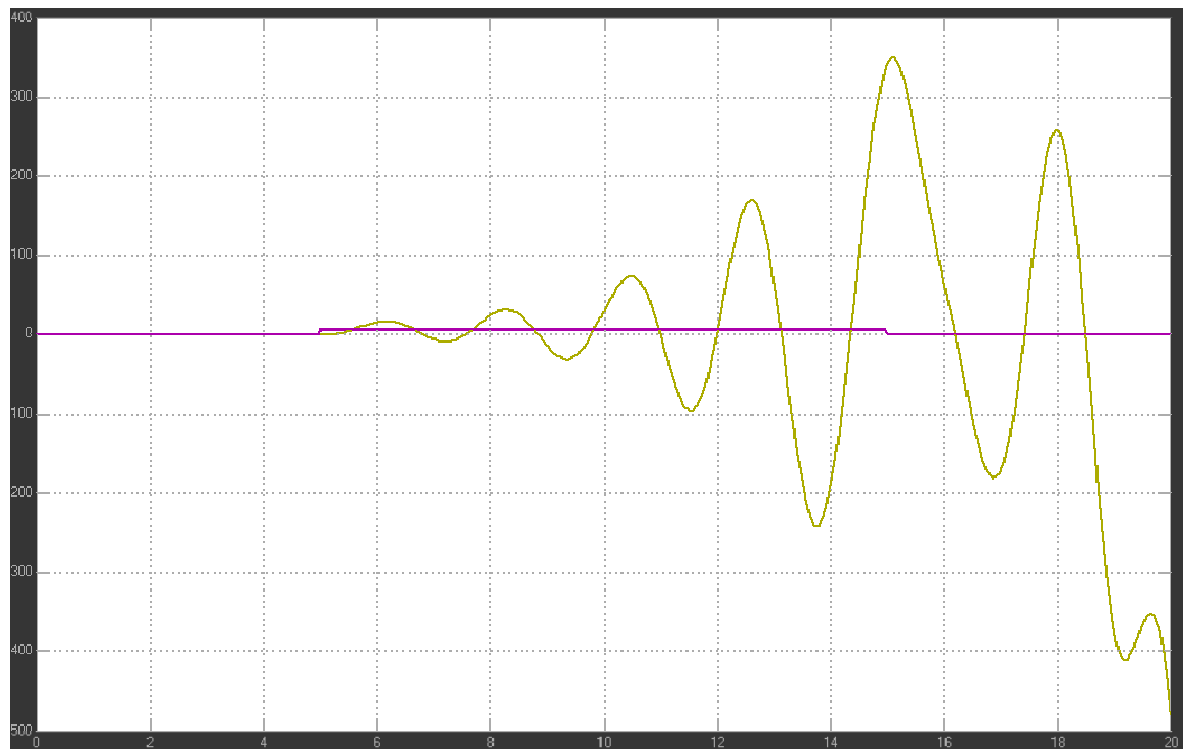


Figure 6.8 Roulis : $P=1$, $T_d=0$

Si on essaye de mettre en place un correcteur proportionnel simple comme dans le cas de la figure 6.9 pour le roulis, on s'aperçoit qu'on a une instabilité pour une consigne en échelon. Il n'est pas possible de stabiliser le système avec de simples correcteurs proportionnels. En effet, le système se corrige trop tardivement. Pour corriger ce problème de retard de correction, il est nécessaire d'introduire un terme dérivé dans le correcteur qui permettra d'augmenter la vitesse de réaction du système.

Avec des correcteurs proportionnel-dérivé sur les trois angles, il est possible de stabiliser le drone (figure 6.9). Le système est naturellement précis, il n'est donc pas utile d'ajouter un terme intégral. Le terme proportionnel permet le temps de réponse du système, sa rapidité mais génère des oscillations que l'on peut corriger avec le terme dérivé.

Nous avons essayé d'obtenir les réponses les plus rapides possibles pour un dépassement maximal de 20%. Avec les correcteurs explicités à la figure 6.9, on obtient des dépassements maximums de 15%, 10% et 17% respectivement pour le tangage, roulis et lacet.

Les temps de réponse à 5% sont de l'ordre d'une seconde pour le tangage et le roulis et de deux secondes pour le lacet.

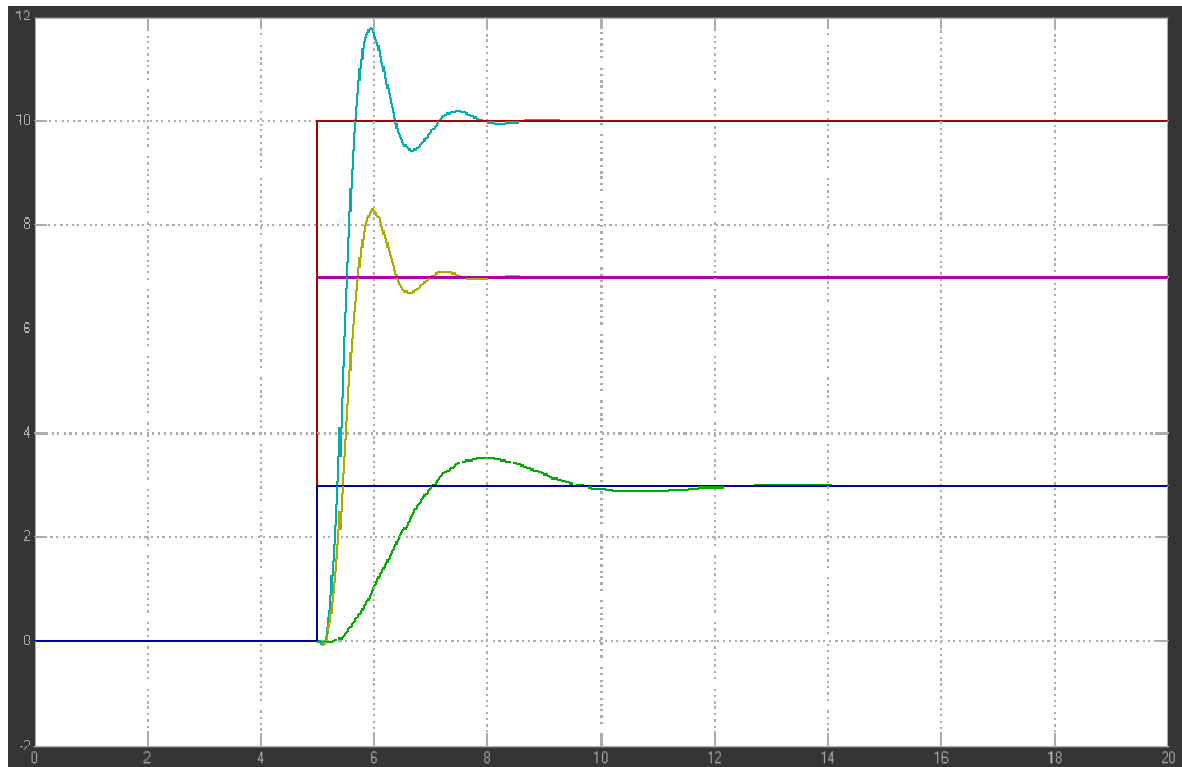


Figure 6.9 Lacet : $P=12$, $Td=10$, Tangage : $P=1.5$, $Td=0.5$, Roulis : $P=1.5$, $Td=0.5$

Les correcteurs fonctionnent toujours si la consigne contient une rampe (figure 6.10).

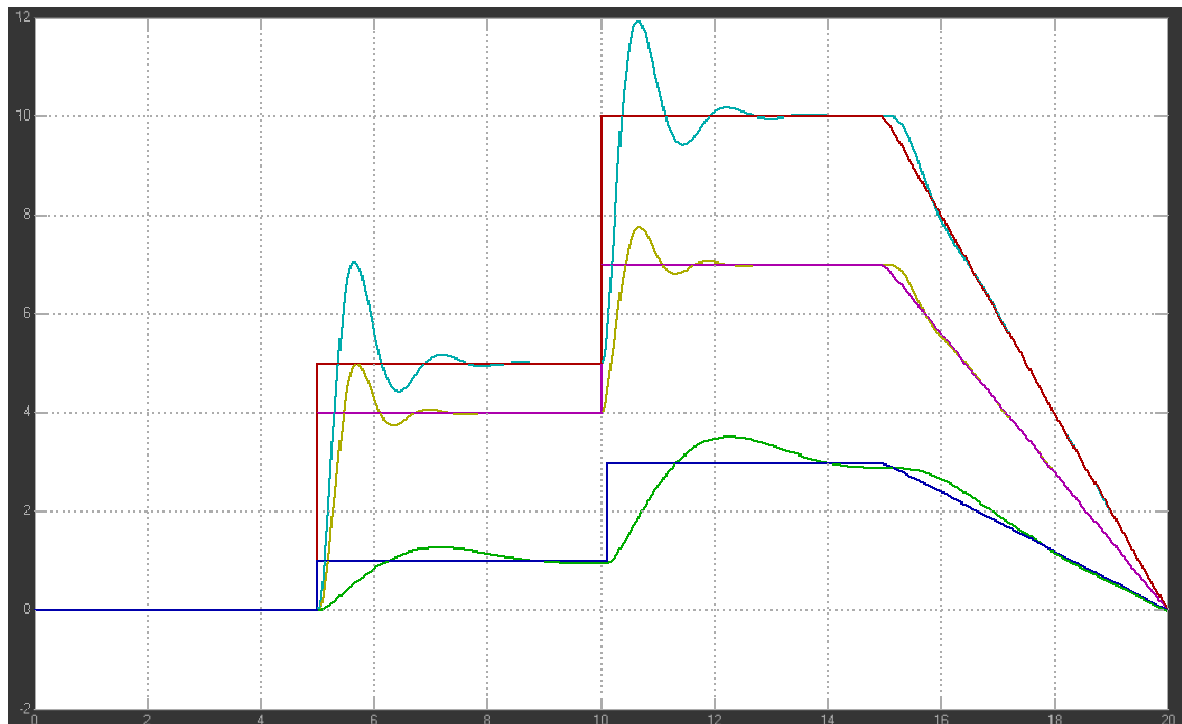


Figure 6.10 Lacet : $P=12$, $Td=10$, Tangage : $P=1.5$, $Td=0.5$, Roulis : $P=1.5$, $Td=0.5$

6.3 Choix des correcteurs pour un angle de support moteur non nul

Chacun des quatre supports moteurs est légèrement incliné vers le centre du drone. Cette inclinaison affecte la précision du système pour le roulis et le tangage comme on peut le constater sur les figures 6.11 et 6.12. L'erreur est d'autant plus importante que l'angle des supports moteurs est élevé. Il est donc nécessaire d'implémenter une intégrale dans le correcteur. On a donc un correcteur PID de la forme $P+Td*s+Ti/s$ (figure 6.13 et 6.14).

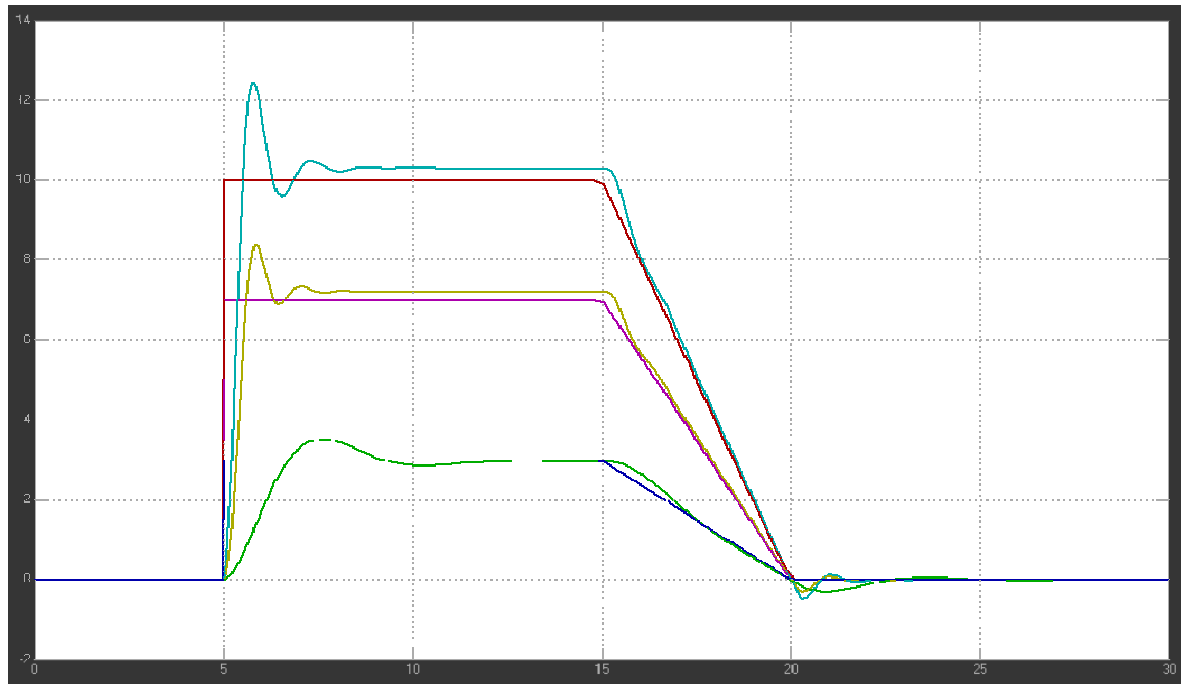


Figure 6.11 Inclinaison des supports moteurs de 2°.

Lacet : $P=12$, $Td=10$, Tangage : $P=1.5$, $Td=0.5$, Roulis : $P=1.5$, $Td=0.5$

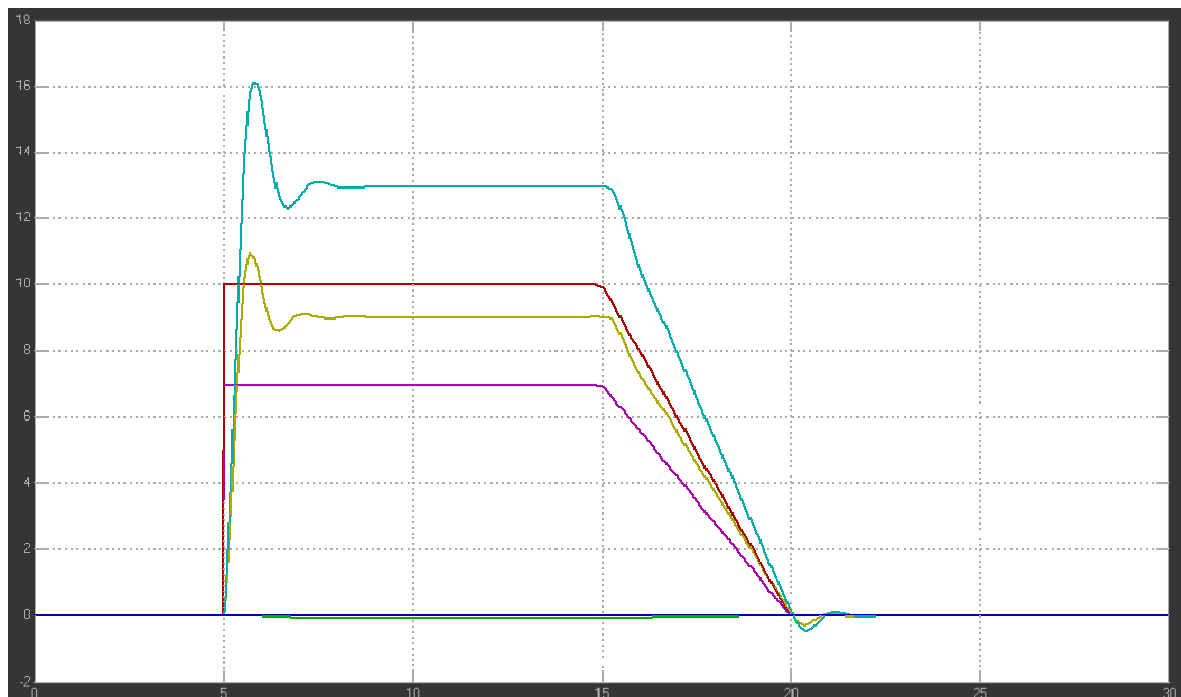


Figure 6.12 Inclinaison des supports moteurs de 15°.

Lacet : $P=12$, $Td=10$, Tangage : $P=1$, $Td=0.5$, Roulis : $P=1$, $Td=0.5$

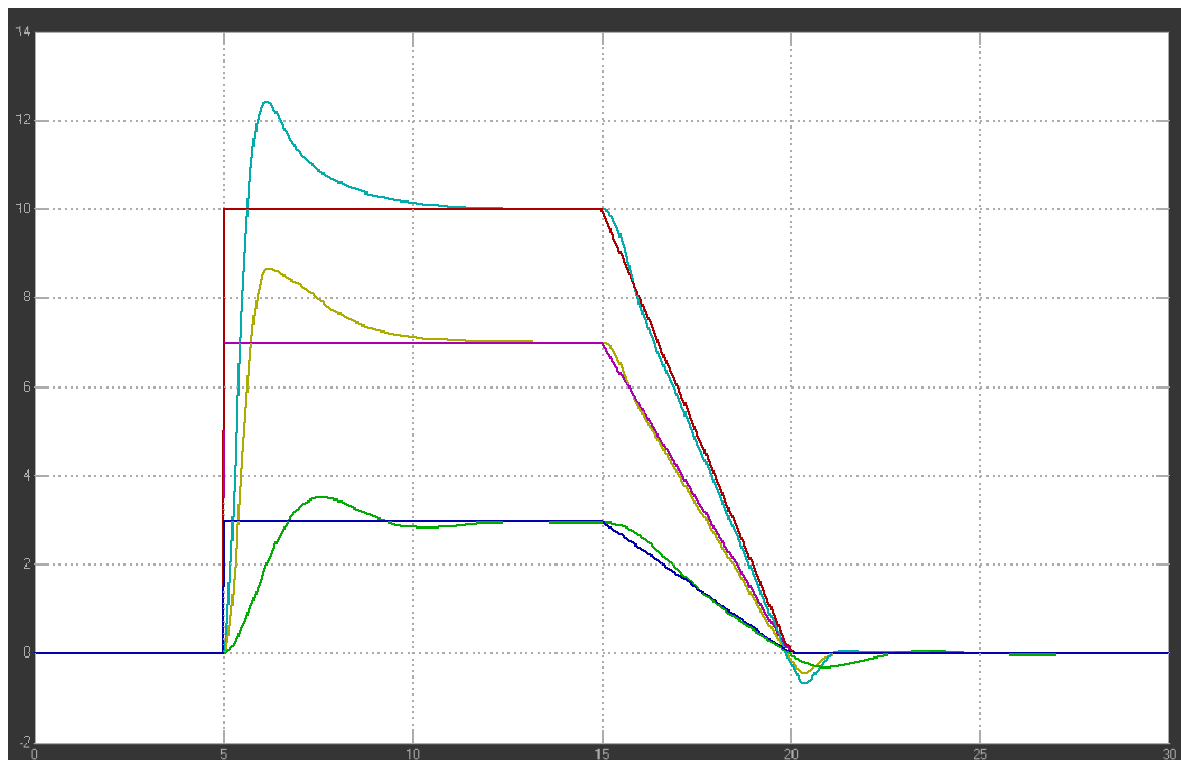


Figure 6.13 Inclinaison des supports moteurs de 2° .

Lacet : $P=12$, $Td=10$, Tangage : $P=1.5$, $Td=0.5$, $Ti=0.5$, Roulis : $P=1.5$, $Td=0.5$, $Ti=0.5$

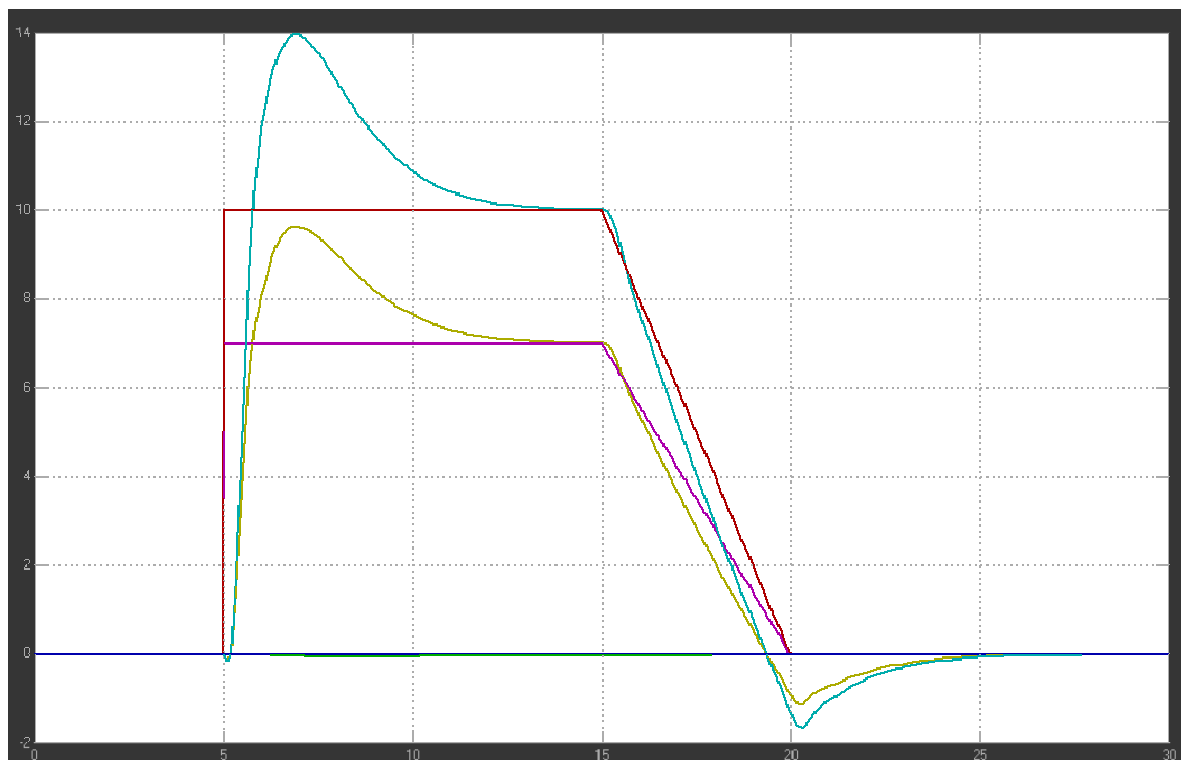


Figure 6.14 Inclinaison des supports moteurs de 15° .

Lacet : $P=12$, $Td=10$, Tangage : $P=1$, $Td=0.5$, $Ti=0.5$, Roulis : $P=1$, $Td=0.5$, $Ti=0.5$

Le terme intégral du correcteur permet de diminuer l'erreur sur un échelon mais il subsiste tout de même une erreur pour une rampe.

7. Différentes solutions pour la stabilisation

7.1 Solution à base d'AOP

Cette solution, basée sur des opérations effectuées à l'aide d'AOP (addition, soustraction, gain, comparateur), nécessite de connaître la valeur moyenne des signaux des différentes voies du récepteur. Mais, sachant que le rapport cyclique du signal PWM varie de 0,05 à 0,1 environ, il est très difficile d'obtenir cette moyenne. Un filtre passe-bas avec une fréquence de coupure très faible serait trop lent, et un filtre rapide fournirait un signal peu stable. En fait, il faudrait utiliser un condensateur de très grande capacité.

Le schéma électronique de cette solution est donné en annexe, mais il n'a pas été réalisé en raison des difficultés précédemment exposées.

7.2 Solution mixeurs – gyroscopes

L'utilisation de simples mixeurs et de gyroscopes pourrait permettre de stabiliser le quadrirotor facilement et à moindre coût [6].

7.2.1 Le mixeur

Le mixeur est un petit circuit électronique couramment utilisé en aéromodélisme, notamment pour des avions à ailes delta, des ailes volantes ou des avions à empennage en V. Il calcule, en quelque sorte, la somme et la différence des deux entrées. La figure 7.1 présente un mixeur (Graupner - Best.-Nr.3020) et la figure 7.2 explique les variations des sorties en fonction des entrées.

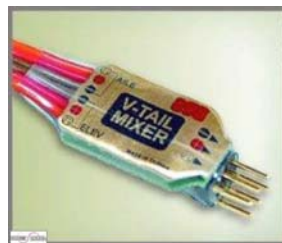
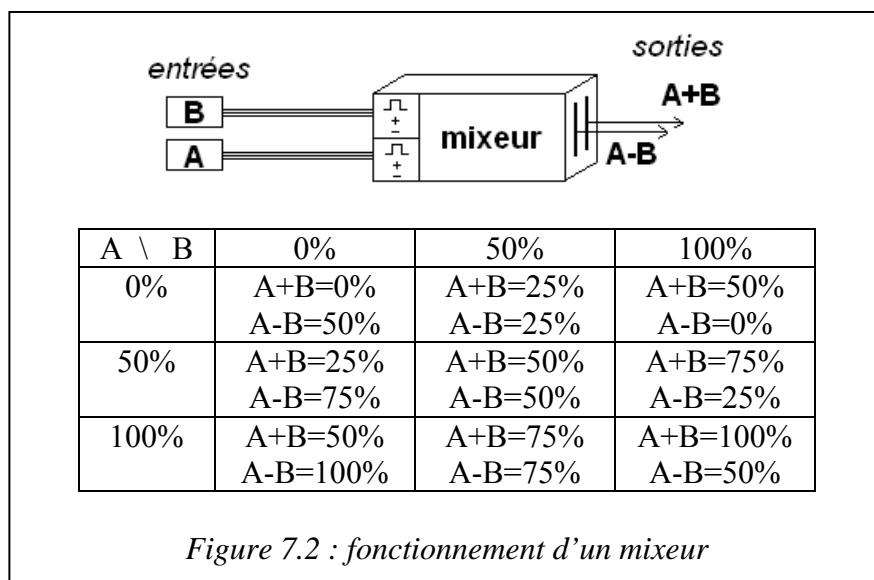


Figure 7.1 : Mixeur



7.2.2 Le gyroscope

Un gyroscope est un élément très largement utilisé en aéromodélisme, particulièrement sur les hélicoptères. Intercalé entre le récepteur et le circuit de commande du quadrirotor, il pourra contrer une variation brusque et non souhaitée de l'angle de roulis, de lacet ou de tangage, suivant la voie à laquelle le gyroscope est associé.

Nous utiliserons un « Piezo Gyro G400 » de Robbe (Figure 7.3) pour l'axe de lacet et des « heading-lock MS-044 » de MS Composit (Figure 7.4) sur les axes de roulis et de tangage. Ces deux types de gyroscopes offrent la possibilité de régler un « gain » et un « delay » de 0% à 100%. Le réglage du gain, pouvant être effectué par la radiocommande en connectant le troisième fil du gyroscope à une voie libre du récepteur, permet de sélectionner entre le mode normal et le mode conservateur de cap.



Figure 7.3 : Piezo Gyro G400



Figure 7.4 : Heading-lock MS-04

7.2.3 La stabilisation

Ainsi, nous souhaitons réaliser le circuit de stabilisation et de commande à l'aide de mixeurs et de gyroscopes. Du chapitre 2, traitant le principe de vol d'un quadrirotor, nous tirons directement les 4 équations correspondant aux commandes des 4 moteurs :

$M1 = P - L + T$	$M2 = P + L - R$
$M4 = P + L + R$	$M3 = P - L - T$
P : puissance ; L : lacet ; T : tangage ; R : roulis	

Le schéma de la figure 7.5 explique les différentes connections réalisées pour la solution mixeurs-gyroscopes. En ce qui concerne le branchement des mixeurs, il est impératif de se référer au paragraphe 7.2.1, car de nombreuses erreurs de signe peuvent être commises. Les gyroscopes peuvent être intercalés à la sortie des voies 1, 2 et 4, qui correspondent respectivement aux commandes de roulis, de tangage et de lacet.

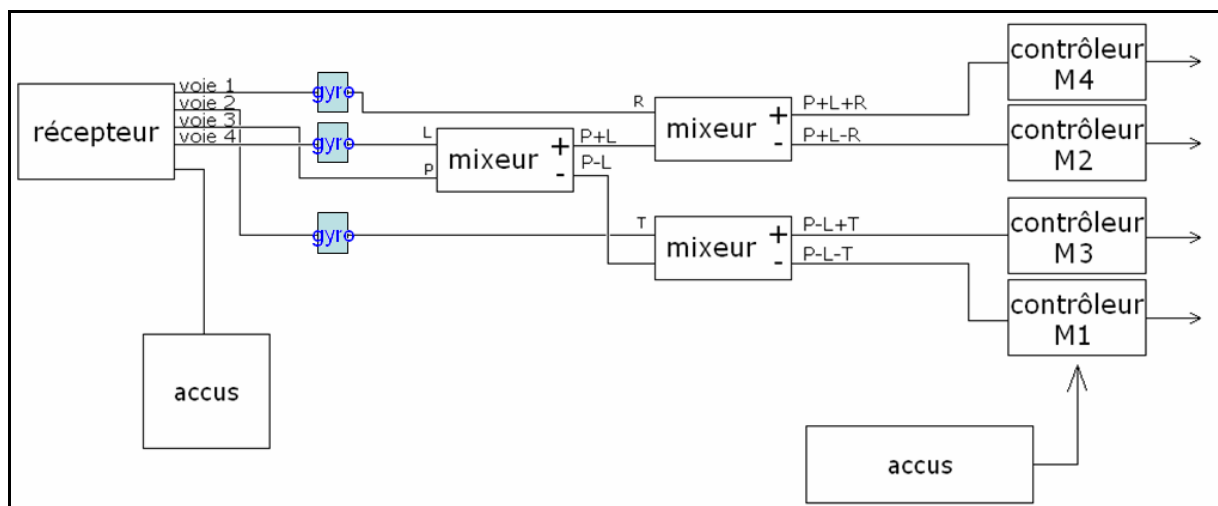


Figure 7.5 : solution mixeurs-gyroscopes

7.3 Solution carte MicroPilot

L'utilisation de la carte MicroPilot permet d'effectuer les mêmes opérations que les mixeurs [7], l'avantage vient de la possibilité d'implémenter des correcteurs (proportionnels, intégrales et dérivés). La Figure 7.6 présente le schéma globale de cette troisième solution pour la commande et la stabilisation du drone.

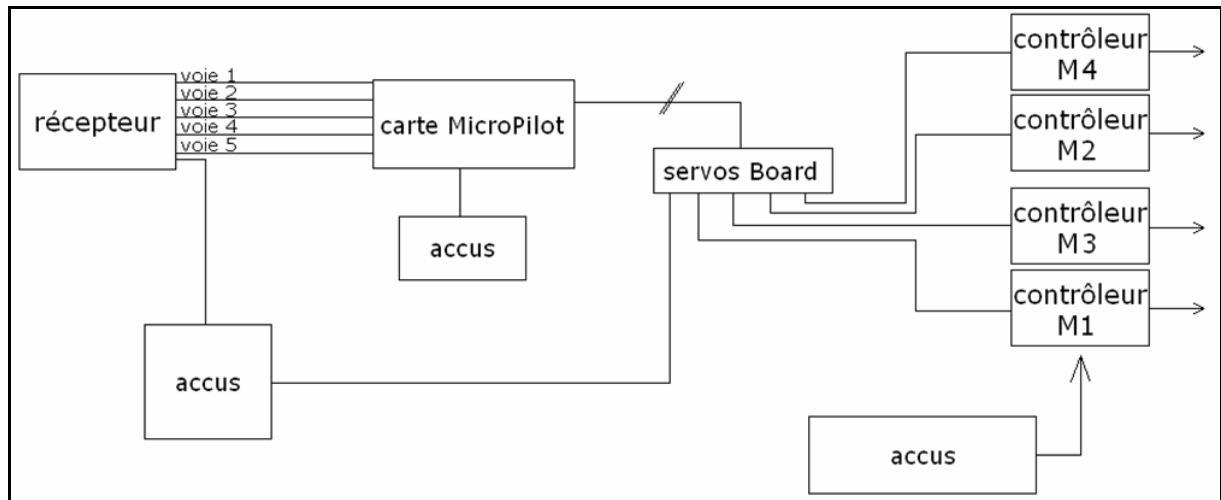


Figure 7.6 : solution carte MicroPilot

7.3.1 Branchement de la carte MicroPilot et de ses éléments additionnels

Branchement récepteur / carte :

Voie 1 : AILE (fil orange)
 Voie 2 : ELEV (fil vert)
 Voie 3 : THRO (fils gris, rouge, noir)
 Voie 4 : RUDD (fil bleu)
 Voie 5 : fil jaune

Branchement servo board / contrôleurs :

Broche 5 : Moteur 2
 Broche 6 : Moteur 3
 Broche 7 : Moteur 4
 Broche 8 : Moteur 1

Pour chaque voie, le fil signal doit être branché vers la partie avant du récepteur.

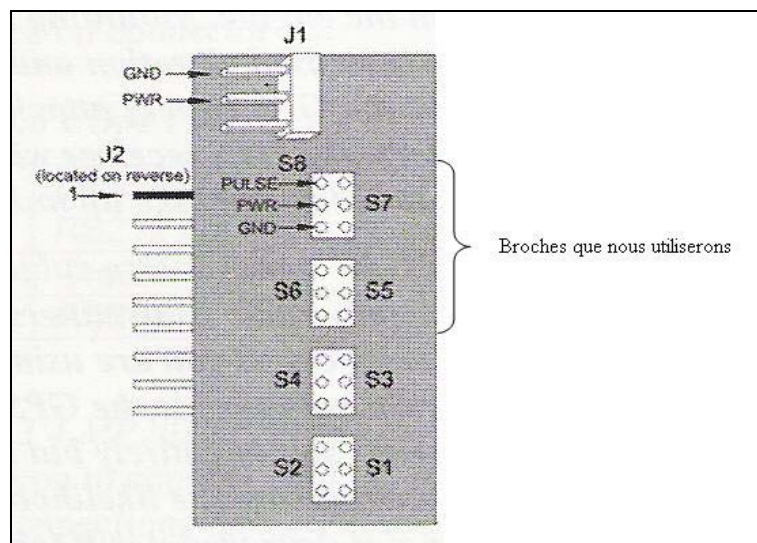


Figure 7.7 : Schéma du bloc Servo board

Les broches 1 à 4 ne sont pas utilisables dans notre cas car la carte envoie sur elles des signaux spécifiques correspondant aux réglages initiaux de la MP2028 pour le vol d'un avion. Ainsi par exemple, si l'AGL (altimètre par ultrasons) n'est pas branché, ce qui est notre cas étant donné qu'il ne nous est pas utile dans le cas d'un hélicoptère, la carte signale cette erreur en envoyant des consignes périodiques sur les broches 1 à 4 (concrètement, la carte fait battre les gouvernes de l'avion pour signaler le problème à l'utilisateur. Ceci n'étant pas acceptable dans le cas de notre étude, nous utilisons uniquement les quatre autres broches ce qui est suffisant pour un quadrirotor.

Attention : Il est très important de faire attention au sens de branchement sur le servo board car, celui-ci n'étant pas muni de détrompeurs, une erreur peut survenir ce qui entraînerait la destruction du contrôleur concerné.

Branchement carte / compas magnétique :

Le branchement doit se faire en accord avec le schéma suivant extrait de la documentation Micropilot MP2028 [8].

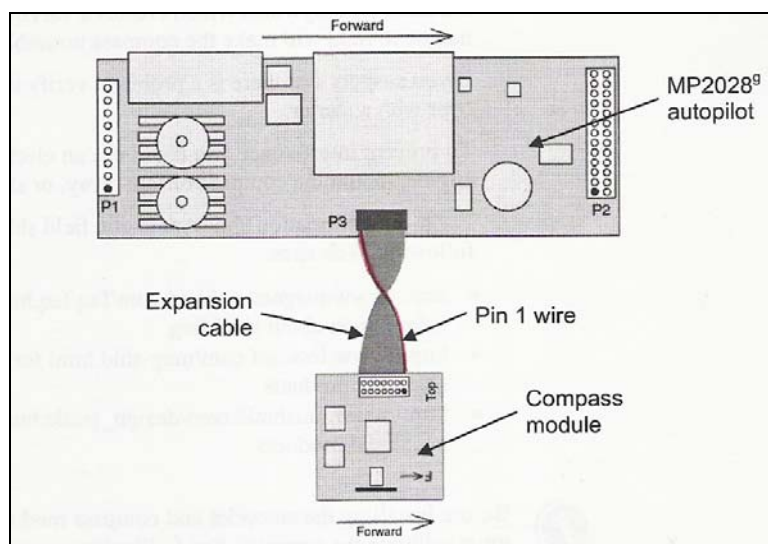


Figure 7.8 : Branchement de la carte MicroPilot au module compas magnétique

Branchement de la liaison carte / ordinateur :

Ce branchement se fait simplement par l'intermédiaire du fil COM de la carte qu'il faut relier à un câble série (avec éventuellement l'adaptation d'un câble série/USB) ou à un radio modem embarqué pour les communications en vol.

Alimentation de la carte et de ses éléments associés :

- La carte doit être alimentée idéalement en 9,6 V, ce qui est possible à l'aide d'une batterie Lithium Polymère, ce qui permet en plus de gagner en masse. L'alimentation de la carte se fait par le câble PWR en faisant une nouvelle fois très attention au sens de branchement.
- Le Servo board est alimenté en 4,8 V ce qui peut être réalisé en parallèle avec l'alimentation de récepteur lui aussi en 4,8 V. Cela permet en effet un gain de masse sans pour autant réduire l'autonomie de l'accumulateur, la puissance consommée par ces deux éléments étant relativement faible.

Cette double alimentation est réalisée à l'aide d'un câble en Y possédant une entrée (accumulateur) et deux sorties (récepteur et Servo board). Ce type de câble n'étant pas vendu dans le commerce, il suffit d'acheter la connectique et de faire soi-même les soudures.

7.3.2 Le programme de stabilisation

Nous allons décrire dans cette partie le code utilisé pour le quadrirotor, morceau par morceau afin de bien l'appréhender dans son ensemble et surtout de voir ce qui peut être modifié dans le but d'une éventuelle amélioration du programme.

Ce code est en langage C, et peut donc être lu à l'aide d'un éditeur classique du type bloc note, mais nous conseillons plutôt un éditeur capable de reconnaître le C, comme par exemple Crimson Editor (disponible gratuitement sur Internet).

Le programme s'articule autour de plusieurs parties :

- L'initialisation du code
- La déclaration des variables entrées / sorties
- Initialisation des variables + variables internes
- L'attribution des entrées/ sorties du programme
- Le code proprement dit, séparé en plusieurs USER PID [9]

- L'initialisation du code :

L'initialisation du code est propre à la carte MP2028. En effet, celle-ci dispose déjà d'un code par défaut et nous ne faisons qu'ajouter notre code à ce dernier, ce qui impose un début de code particulier.

Cette partie de code qui n'est pas modifiable, se présente de la façon suivante (dans sa version non commentée :

```
#include "usercode.h"
#include "simdll.h"
#ifdef _WIN32
#define _main
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <windows.h>
#endif /* _WIN32 */
#ifdef _WIN32
BOOL WINAPI DllMain( HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved )
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
            break;
        case DLL_THREAD_ATTACH:
            break;
        case DLL_THREAD_DETACH:
            break;
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}
#endif /* _WIN32 */
int (*saveGetVarPointer)( void **result, int id);
long getMPVarPointer( void **result, int id)
{
    return (*saveGetVarPointer)( result, id);
}
int mpUserInit( int (*function)( void **result, int id))
{
    saveGetVarPointer = function;
    return USER_RETURN_INIT_OK;    //return 1;
}
int mpUserEvent( long *result, int userEventIdentifier)
{

```

- La déclaration des variables entrées / sorties :

Voici les déclarations de variables nécessaires à notre programme, sachant qu'il est bien sûr possible d'en ajouter ou d'en enlever en cas de non utilisation. Ces déclarations ayant été faites de manière globale avant l'écriture du programme complet, il est à noter qu'elles ne sont pas toutes utilisées par la suite ; cependant, il était important de fixer les idées pour l'éventualité d'une réécriture du programme.

<code>static long *hardwareDisable;</code>		
<code>static long *ref_Roll_raw;</code>	<code>//commanded roll angle</code>	Coefficients correspondants à l'axe de roulis (<code>_p</code> pour indiqué la dérivée, ie. vitesse angulaire)
<code>static long *cRoll;</code>	<code>//current roll</code>	
<code>static long *iRoll;</code>	<code>//instantaneous roll</code>	
<code>static long *iRoll_p;</code>	<code>//instantaneous roll dot</code>	
<code>static long *ref_Pitch_raw;</code>	<code>//commanded pitch angle</code>	Coefficients correspondants à l'axe de tangage
<code>static long *cPitch;</code>	<code>//current pitch</code>	
<code>static long *cPitch_p;</code>	<code>//current pitch dot</code>	
<code>static long *iPitch;</code>	<code>//instantaneous pitch</code>	
<code>static long *iPitch_p;</code>	<code>//instantaneous pitch dot</code>	
<code>static long *iYaw_p;</code>	<code>//measured yaw rate</code>	Coefficients correspondants à l'axe de lacet
<code>static long *ref_Hdg_raw;</code>	<code>//commanded Hdg (raw value)</code>	
<code>static long *ref_ident_raw;</code>	<code>//commanded Hdg identification</code>	
<code>static long *Hdg;</code>	<code>//compass Hdg</code>	
<code>static long *Radio_raw;</code>	<code>//Radio switch (raw value)</code>	Switch de la télécommande (fil jaune) + référence de poussée
<code>static long *ref_Thrust_raw;</code>		
<code>static long *Rp0_raw;</code>	<code>//coefficient 0 of R (raw)</code>	Coefficients utilisés pour régler les PID (issus de l'interface horizon)
<code>static long *Rp1_raw;</code>	<code>//coefficient 1 of R (raw)</code>	
<code>static long *Rp2_raw;</code>	<code>//coefficient 2 of R (raw)</code>	
<code>static long *Rp3_raw;</code>	<code>//coefficient 3 of R (raw)</code>	
<code>static long *Rp4_raw;</code>	<code>//coefficient 4 of R (raw)</code>	
<code>static long *Rp5_raw;</code>	<code>//coefficient 5 of R (raw)</code>	
<code>static long *Sp1_raw;</code>	<code>//coefficient 1 of S (raw)</code>	
<code>static long *Sp2_raw;</code>	<code>//coefficient 2 of S (raw)</code>	
<code>static long *Sp3_raw;</code>	<code>//coefficient 3 of S (raw)</code>	
<code>static long *Sp4_raw;</code>	<code>//coefficient 4 of S (raw)</code>	

<code>static long Rp0;</code>	<code>//coefficient 0 of R</code>	Coefficients utilisés dans le programme pour régler le PID (après ajustement à partir des <code>_Raw</code>)
<code>static long Rp1;</code>	<code>//coefficient 1 of R</code>	
<code>static long Rp2;</code>	<code>//coefficient 2 of R</code>	
<code>static long Rp3;</code>	<code>//coefficient 3 of R</code>	
<code>static long Rp4;</code>	<code>//coefficient 4 of R</code>	
<code>static long Rp5;</code>	<code>//coefficient 5 of R</code>	
<code>static long Sp1;</code>	<code>//coefficient 1 of S</code>	
<code>static long Sp2;</code>	<code>//coefficient 2 of S</code>	
<code>static long Sp3;</code>	<code>//coefficient 3 of S</code>	Commandes pour les contrôleurs
<code>static long Sp4;</code>	<code>//coefficient 4 of S</code>	
<code>static long *fServo5;</code>		Pour faire des affichages éventuels
<code>static long *fServo6;</code>		
<code>static long *fServo7;</code>		
<code>static long *fServo8;</code>		
<code>static long *disp1;</code>		
<code>static long *disp2;</code>		
<code>static long *disp3;</code>		
<code>static long *disp4;</code>		
<code>static long *disp5;</code>		
<code>static long *disp6;</code>		
<code>static long *disp7;</code>		
<code>static long *disp8;</code>		
<code>static long *disp9;</code>		
<code>static long *disp10;</code>		

Le `*` indique qu'il s'agit d'une variable entrée / sortie.

- Initialisation des variables + variables internes :

```
//variable declaration
static long signal_min=2220;
static long signal_zero=3045;

static long Thrust_max=50000;
static float Thrust_ref;

static long Roll_err;
static long Roll_max=45*3.14/180*1024; //maximum (magnitude) commanded roll angle [rad*1024]
static float Roll_ref; //commanded roll angle (transformed value)
static long Roll_mix; //input for servo mixing (=output of roll controller)

static long Pitch_err;
static long Pitch_max=45*3.14/180*1024; //maximum (magnitude) commanded pitch angle [rad*1024]
static float Pitch_ref; //commanded pitch angle (transformed value)
static long Pitch_mix; //input for servo mixing (=output of pitch controller)

static long Yaw_mix; //input for servo mixing (=output of Yaw_p controller)
static long Hdg_err;
static long Hdg_max=30000; //maximum (magnitude) commanded Hdg [deg*1000]
static long Hdg_max_p=150; //maximum (magnitude) commanded Hdg_p [deg/(30*s)*100] or [deg/(5*s)*100]
static float Hdg_ref=0; //commanded Hdg (transformed value)
static float Hdg_ref_p; //commanded Hdg (transformed value)
```

Ces valeurs ont été obtenues par expérience ou grâce à la documentation de la carte Micropilot. On définit également ici les valeurs « ref » qui correspondront aux consignes récupérées en entrée et provenant de la télécommande après correction. Les définitions de ces termes ont été faits à cet endroit du programme plutôt que dans la partie précédente car il s'agit de variables qui ne correspondent pas à une entrée ou une sortie du programme : Il s'agit de variables internes au programme.

Initialisation des autres variables internes au programme (il s'agit en majorité des variables permettant l'implémentation des PID) :

```
//variables for RST controller
static long com0=0; //heading (now)
static long hdg0=0; //heading (now)
static long hdg1=0; //previous heading
static long hdg2=0; //previous (2 steps) heading
static long hdg3=0; //previous (3 steps) heading
static long hdg0f=0; //heading (now) (filtered)
static long hdg1f=0; //previous heading (filtered)
static long hdg2f=0; //previous (2 steps) heading (filtered)
static long hdg3f=0; //previous (3 steps) heading (filtered)
static long err0=0; //heading err now
static long err1=0; //heading (1 step before)
static long err2=0; //heading (2 step before)
static long err3=0; //heading (3 step before)
static long err4=0; //heading (3 step before)
static long err5=0; //heading (3 step before)

static long yaw_pp0=0; //yaw_pp (now)
static long roll_pp0=0; //roll_pp (now)
static long pitch_pp0=0; //pitch_pp (now)

static long yaw_p0=0; //yaw_p (now)
static long yaw_p1=0; //yaw_p (1 step before)
static long yaw_p2=0; //yaw_p (2 steps before)
static long yaw_p3=0; //yaw_p (3 steps before)
static long yaw_p0f=0; //yaw_p (now) (filtered)
static long yaw_p1f=0; //yaw_p (1 step before) (filtered)
static long yaw_p2f=0; //yaw_p (2 steps before) (filtered)
static long yaw_p3f=0; //yaw_p (3 steps before) (filtered)
static long roll_p0=0; //roll_p (now)
static long roll_p1=0; //roll_p (1 step before)
static long pitch_p0=0; //pitch_p (now)
static long pitch_p1=0; //pitch_p (1 step before)
static long y0f=0; //y (now) (filtered)
static long y1f=0; //y (1 step before) (filtered)
static long y2f=0; //y (2 steps before) (filtered)
static long y3f=0; //y (3 steps before) (filtered)
```

```

static long u0=0; //commanded signal for actuator
static long u1=0; //commanded signal for actuator 1 sample before
static long u2=0; //commanded signal for actuator 2 samples before
static long u3=0; //commanded signal for actuator 3 samples before
static long u4=0; //commanded signal for actuator 4 samples before

static long thrust=0; //thrust signal (temp)
static long t=0; //correction factor (from look up table)

//variables for generation of PRBS
static long b5;
static long b9;
static long prbs_tmp;
static long sqz_count=1000;

```

- L'attribution des entrées/ sorties :

Les valeurs des attributions entrées / sorties proviennent de la documentation de la carte Micropilot.

```

switch( userEventIdentifier)
{
case USER_HARDWARE_INITIALIZED :
    getMPVarPointer( (void **) &hardwareDisable, 594);
    getMPVarPointer( (void **) &ref_Pitch_raw, 1152); //input signal com_Pitch (ELEVATOR servo chan)
    getMPVarPointer( (void **) &ref_Roll_raw, 1153); //input signal com_Roll (AILERON servo chan)
    getMPVarPointer( (void **) &ref_Hdg_raw, 1154); //input signal com_Yaw (RUDDER servo chan)
    getMPVarPointer( (void **) &ref_Thrust_raw, 1155); //input signal com_Thrust (THROTTLE servo chan)
    getMPVarPointer( (void **) &Radio_raw, 1156); //input signal Radio (yellow channel)
    getMPVarPointer( (void **) &ref_ident_raw, 6650); //input signal com_Yaw (identification)

    getMPVarPointer( (void **) &cPitch, 1057);
    getMPVarPointer( (void **) &cRoll, 1059);
    getMPVarPointer( (void **) &cPitch_p, 1058);
    getMPVarPointer( (void **) &iPitch, 1266); //[[rad*1024]
    getMPVarPointer( (void **) &iRoll, 1267); //[[rad*1024]
    getMPVarPointer( (void **) &iPitch_p, 1403); //[[rad*1024/s]
    getMPVarPointer( (void **) &iRoll_p, 1404); //[[rad*1024/s]
    getMPVarPointer( (void **) &iYaw_p, 1405); //[[rad*1024/s]
    getMPVarPointer( (void **) &Hdg, 1208); //[[deg*100]

    getMPVarPointer( (void **) &Rp0_raw, 2); //var rotation speed used
    getMPVarPointer( (void **) &Rp1_raw, 111); //var cruise speed used
    getMPVarPointer( (void **) &Rp2_raw, 5); //var climb speed used
    getMPVarPointer( (void **) &Rp3_raw, 9); //var aproach speed used
    getMPVarPointer( (void **) &Rp4_raw, 13); //var climb margin
    getMPVarPointer( (void **) &Rp5_raw, 14); //var flare altitude
    getMPVarPointer( (void **) &Sp1_raw, 122); //var waypoint diameter
    getMPVarPointer( (void **) &Sp2_raw, 150); //var waypoint reduce turn
    getMPVarPointer( (void **) &Sp3_raw, 152); //var cct altitude
    getMPVarPointer( (void **) &Sp4_raw, 528); //var descent rate used

    getMPVarPointer( (void **) &fServo3, 1229);
    getMPVarPointer( (void **) &fServo5, 1231);
    getMPVarPointer( (void **) &fServo6, 1232);
    getMPVarPointer( (void **) &fServo7, 1233);
    getMPVarPointer( (void **) &fServo8, 1234);

    getMPVarPointer( (void **) &disp1, 8004);
    getMPVarPointer( (void **) &disp2, 8005);
    getMPVarPointer( (void **) &disp3, 8006);
    getMPVarPointer( (void **) &disp4, 8007);
    getMPVarPointer( (void **) &disp5, 8008);
    getMPVarPointer( (void **) &disp6, 8009);
    getMPVarPointer( (void **) &disp7, 8010);
    getMPVarPointer( (void **) &disp8, 8011);
    getMPVarPointer( (void **) &disp9, 8012);
    getMPVarPointer( (void **) &disp10, 8013);

return USER_RETURN_IGNORE;

```

- Les USER PID :

La suite du programme est séparée en plusieurs USER PID qui font référence l'interface Horizon grâce à laquelle il est possible de choisir le USER PID et donc la partie du programme qui sera implémentée. Les différents USER PID permettent donc de tester plusieurs versions du programme sans pour autant modifier le code à chaque test et le recharger dans la carte. (Voir annexe : guide d'utilisation de la MP2028).

Le USER PID que nous allons décrire est le USER PID 1, car c'est lui qui contient le programme le plus abouti concernant la stabilisation des trois axes du quadrirotor.

La première partie du USER PID permet d'ajuster les coefficients RP. et SP. qui viennent en entrée du programme, de l'interface horizon et qui sont utilisés pour les correcteurs PID (Voir pour cela le guide d'utilisation de la MP2028).

Les corrections effectuées sur les coefficients ont été déterminées à l'aide de la notice Micropilot et permettent d'avoir des ordres de grandeur identique pour les coefficients des PID implémentés par la suite.

```

    case USER_PID1 :           //PID loop works with 5/30 Hz
//*****
//***   userPID1_1  PD Controller for Roll, Pitch and Yaw axis
//***   standard flight configuration of PID controllers

//*****
//***   userPID1_2  Adjustment of coefficients for polynomials R and S
    Rp0=*Rp0_raw*10000/9113;
    Rp1=*Rp1_raw*10000/9113;
    Rp2=*Rp2_raw*10000/9113;
    Rp3=*Rp3_raw*10000/9113;
    Rp4=*Rp4_raw*100/2624;
    Rp5=*Rp5_raw*(-100)/2624;
    Sp1=*Sp1_raw*10/264;
    Sp2=*Sp2_raw;
    Sp3=*Sp3_raw*(-10)/264;
    Sp4=*Sp4_raw;

```

La suite du USER PID consiste en l'implémentation des correcteurs PID pour les trois axes. Décrivons le correcteur de l'angle de roulis ou Roll angle.

```

//*****
//***  userPID1_3  PID Controller for Roll angle

//*****
//***  userPID1_3_1  Conversion of commanded signal (Roll axis)
Roll_ref=((float)*ref_Roll_raw-(float)signal_zero)/((float)signal_zero-(float)signal_min)
          * (float)Roll_max;

//*****
//***  userPID1_3_2  Calculation of angular acceleration (roll)

roll_p1=roll_p0;
roll_p0=*iRoll_p;
roll_pp0=(roll_p0-roll_p1)*100/24; //avec un pas de temps de 0.24

//*****
//***  userPID1_3_3  Calculation of regulation error (Roll axis)
Roll_err = Roll_ref*7 - *iRoll; //7 pour regler l'influence de la télécommande.
//*****

//*****
//***  userPID1_3_4  Calculation of signal for servo actuator (Roll axis)
Roll_mix= Roll_err* Rp3 - *iRoll_p* Rp4 - roll_pp0 * Sp2; //PID-controller
//*****

```

La première ligne de code permet de traiter le signal provenant de la télécommande pour obtenir un signal entre [- puissance Max ; + puissance Max], avec comme puissance Max la valeur souhaitée maximum en sortie du programme. Cette manipulation est effectuée car la méthode d'acquisition de la carte fausse le signal réel envoyé par la télécommande. Dans un second temps, nous calculons la donnée d'accélération angulaire en dérivant simplement la donnée de vitesse.

Ensuite, nous calculons l'erreur entre l'angle souhaité (provenant de la télécommande) et l'angle réel.

Enfin, nous implémentons le correcteur PID qui est en fait "centré" sur la vitesse car il fonde sa correction sur les données : position angulaire, vitesse angulaire et accélération angulaire.

En effet, comme nous l'expliquerons par la suite dans la partie Tests, ce type de correcteur a été retenu pour ses bons résultats théoriques.

Le correcteur pour le tangage ou Pitch angle se présente de la même façon que pour le roulis.

```

//*****
//***  userPID1_4  PID Controller for Pitch angle

//*****
//***  userPID1_4_1  Conversion of commanded signal (Pitch axis)
Pitch_ref=((float)*ref_Pitch_raw-(float)signal_zero)/((float)signal_zero-(float)signal_min)
          * (float)Pitch_max;

//*****
//***  userPID1_4_2  Calculation of angular acceleration (pitch)

pitch_p1=pitch_p0;
pitch_p0=*iPitch_p;
pitch_pp0=(pitch_p0-pitch_p1)*100/24; //avec un pas de temps de 0.24

//*****
//***  userPID1_4_3  Calculation of regulation error (Pitch axis)
Pitch_err = Pitch_ref*7 - *iPitch; //Calculation of error (input for PID loop)
//*****

//*****
//***  userPID1_4_4  Calculation of signal for servo actuator (Pitch axis)
Pitch_mix= Pitch_err* Rp5 - *iPitch_p* Sp1 - pitch_pp0 * Sp3; //PID-controller
//*****

```


Le correcteur pour l'axe de lacet ou Yaw angle est très proche des deux autres à quelques détails près. En effet, pour plus de facilités au pilotage, la télécommande agit sur la vitesse de rotation de référence. Ainsi, une action sur la télécommande fait tourner le drone en lacet plus ou moins vite, et si l'action n'est pas maintenue, le drone s'arrête dans la position ou nous l'avons amené.

De plus nous avons mis en place un système de protection du programme au cas où l'angle de lacet dépasserait les 360°, le correcteur sera donc toujours valable dans ce cas de figure.

```

//*****
//**** userPID1_5
//*****
//**** userPID1_5_1 PD Controller for Yaw axis
//*****
//**** userPID1_5_1_1 Conversion of commanded signal (Yaw axis)
// (on commande la dérivée du cap pour plus de facilité de pilotage)
Hdg_ref_p=((float)*ref_Hdg_raw-(float)signal_zero)/((float)signal_zero-(float)signal_min)
* (float)Hdg_max_p;

//*****
//**** userPID1_5_1_2 Integration of commanded signal in order to command an angle (yaw axis)
Hdg_ref=Hdg_ref+Hdg_ref_p;
//*****

//*****
//**** userPID1_5_3 Reparation of overflow problem of commanded signal
if(Hdg_ref < 0)
{
    Hdg_ref=Hdg_ref+36000;
}
else
{
    if(Hdg_ref > 36000)
    {
        Hdg_ref=Hdg_ref-36000;
    }
}

```

Un même système de protection a été mis en place pour l'erreur angulaire.

```

//*****
//**** userPID1_5_4 Calculation of angular acceleration (yaw)

yaw_p1=yaw_p0;
yaw_p0=*iYaw_p;
yaw_pp0=(yaw_p0-yaw_p1)/3;

//*****
//**** userPID1_5_5 Calculation of regulation error (Yaw axis)
Hdg_err=Hdg_ref - *Hdg;
//*****

//*****
//**** userPID1_5_6 Adjustment of regulation error (Yaw axis)
if(Hdg_err > 18000)
{
    Hdg_err=Hdg_err-36000;
}
else
{
    if(Hdg_err < -18000)
    {
        Hdg_err=Hdg_err+36000;
    }
    else
    {
        //nothing to do
    }
}

//*****
//**** userPID1_5_7 Calculation of signal for servo actuator (Yaw axis)
Yaw_mix = Hdg_err * Rp0 - *iYaw_p *Rp1 - Rp2 * yaw_pp0;
Yaw_mix = Yaw_mix/100;

```


Enfin, nous effectuons une correction d'ordre de grandeur sur les coefficients et spécifions les sorties du programme qui seront envoyées vers les variateurs.

```

//*****
//***  userPID1_6  Correction of decimals servo command
    Roll_mix=Roll_mix/1000;
    Pitch_mix=Pitch_mix/1000;
    Yaw_mix=Yaw_mix/1000;
//*****
//***  End of userPID1_6
//*****

//*****
//***  userPID1_7 Contrôle de puissance

    Thrust_ref=(float)*ref_Thrust_raw;  //donnée de puissance

//***  end of userPID1_7
//*****

//*****
//***  userPID1_8 Servo mixing
//*****
//***  userPID1_8_1  Servo mixing (Roll, Pitch, Yaw axis)

    *fServo5 = Thrust_ref - Roll_mix + Yaw_mix;
    *fServo6 = Thrust_ref - Pitch_mix - Yaw_mix;
    *fServo7 = Thrust_ref + Roll_mix + Yaw_mix;
    *fServo8 = Thrust_ref + Pitch_mix - Yaw_mix;

```

Le USER PID est ainsi implémenté et il reste à conclure cette partie du programme par :

```
return USER_RETURN_REPLACE;
```

Remarque :

Les PID sur les trois axes se présentent de la façon suivante en fonction des coefficients RP et SP :

Correction Lacet (yaw) = $RP0 \cdot \text{erreur} - RP1 \cdot \text{vitesse} - RP2 \cdot \text{accélération}$

Correction Roulis (roll) = $RP3 \cdot \text{erreur} - RP4 \cdot \text{vitesse} - SP2 \cdot \text{accélération}$

Correction Tangage (pitch) = $RP5 \cdot \text{erreur} - SP1 \cdot \text{vitesse} - SP3 \cdot \text{accélération}$

8. Tests

8.1 Test de la solution mixeurs – gyroscopes

Dans un premier temps, nous avons testé le schéma de commande avec les mixeurs uniquement, sans les gyroscopes, pour vérifier le bon fonctionnement et la réalisation effective des quatre équations présentées au paragraphe 7.2.3. Nous avons pu constater que les différentes commandes (puissance, lacet, tangage et roulis) avaient une influence correcte sur la rotation des quatre hélices.

Mais, lors des tests, nous avons constaté qu'un moteur pouvait s'arrêter lors d'une commande de roulis ou de tangage trop importante. De plus, une faible action sur le manche de la radiocommande modifiait beaucoup trop les vitesses des rotors. Ainsi, pour améliorer le pilotage, nous avons décidé de limiter les effets sur les axes de roulis et de tangage à la plage [25% , 75%] au lieu de [0% , 100%], ceci grâce à la radiocommande programmable. Cette modification a ainsi permis d'augmenter la résolution autour de la position d'équilibre qui est théoriquement à 50%, puis d'empêcher l'arrêt des moteurs lors d'une consigne de roulis ou de tangage maximum.

Nous avons aussi la possibilité d'utiliser une commande non-linéaire en fonction de la position du manche, ceci encore à l'aide de la radiocommande programmable. Ceci permettrait par exemple une plus forte variation de la vitesse de rotation des moteurs autour du neutre 50%.

Ensuite, nous avons connecté un gyroscope sur l'axe de roulis. En ayant attaché le drone par les moteurs 1 et 3 et fait tourner uniquement les moteurs 2 et 4, nous avons testé la stabilité du quadrirotor autour de l'axe de roulis. Après introduction d'une perturbation, nous avons constaté que les oscillations s'amplifiaient, ceci quel que soit le réglage du gain du gyroscope. En fait, la correction s'effectuait dans le bon sens, mais bien trop tard et de manière trop forte.

Nous avons déduit des tests réalisés que les contrôleurs introduisaient des retards, qui ne pouvaient être gérés correctement par les gyroscopes.

8.2 Test de la solution carte MicroPilot

Compte tenu du temps qui nous était imparti et la volonté d'effectuer des essais en toute sécurité, nous nous sommes limités dans notre étude aux tests des correcteurs sur un axe (axe de roulis ou de tangage).

Il n'est pas illusoire de penser que cette stabilisation sur un seul axe sera transposable pour la stabilisation sur les trois axes relativement facilement, et ceci de part la symétrie du drone et grâce au fait que l'axe de lacet soit le plus simple à stabiliser.

8.2.1 Dispositif de test

Le dispositif de test mis en place est relativement simple ; en effet nous avons attaché le drone par les tiges des rotors de deux moteurs opposés. Ce mode d'attache a pour but de se rapprocher le plus possible des conditions de vol réel du drone.

Cependant nous avons remarqué expérimentalement que le drone avait tendance à effectuer un mouvement de balancier avec les câbles l'attachant. Ce mouvement étant dû exclusivement à notre dispositif de test et ne pouvant avoir lieu lors d'un vol normal futur du drone, nous avons décidé d'annuler cet effet en bloquant latéralement le quadrirotor au niveau de ses deux

attaches. Ces dispositifs de blocage latéraux ne gênent bien sûr en rien la rotation du drone autour de l'axe de test.

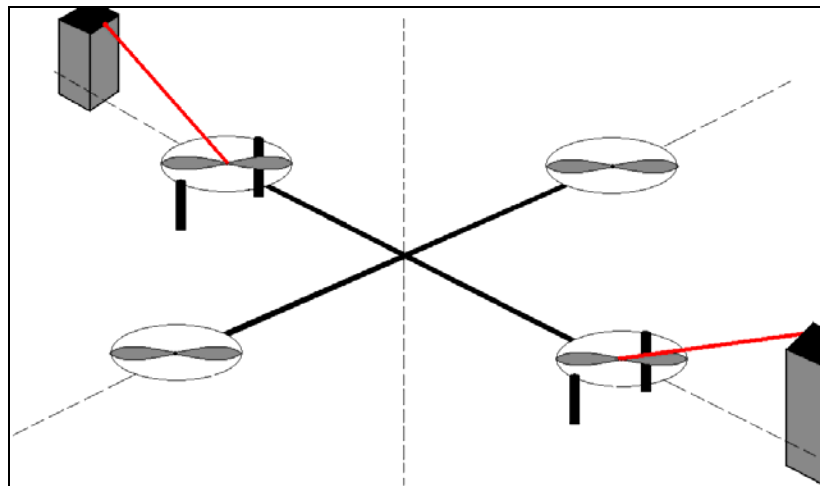


Figure 8.1 : Schéma du dispositif de test

8.2.2 Tests réalisés

A l'aide de ce dispositif et de la carte Micropilot par l'intermédiaire du programme décrit dans la partie 7.3.2, nous avons pu tester différents types de correcteur en partant d'un simple proportionnel jusqu'à un correcteur PID sur la vitesse angulaire. Le but de ces tests était de nous rendre compte à quel point notre modèle théorique établi dans les parties 5 et 6 est en accord avec l'expérience et peut s'appliquer à la réalité.

Ces tests ont été réalisés sans les batteries moteurs en utilisant une alimentation annexe. Un simple contrôle du programme informatique et des réglages de base était effectué à l'aide d'un ordinateur directement lié à la carte Micropilot.

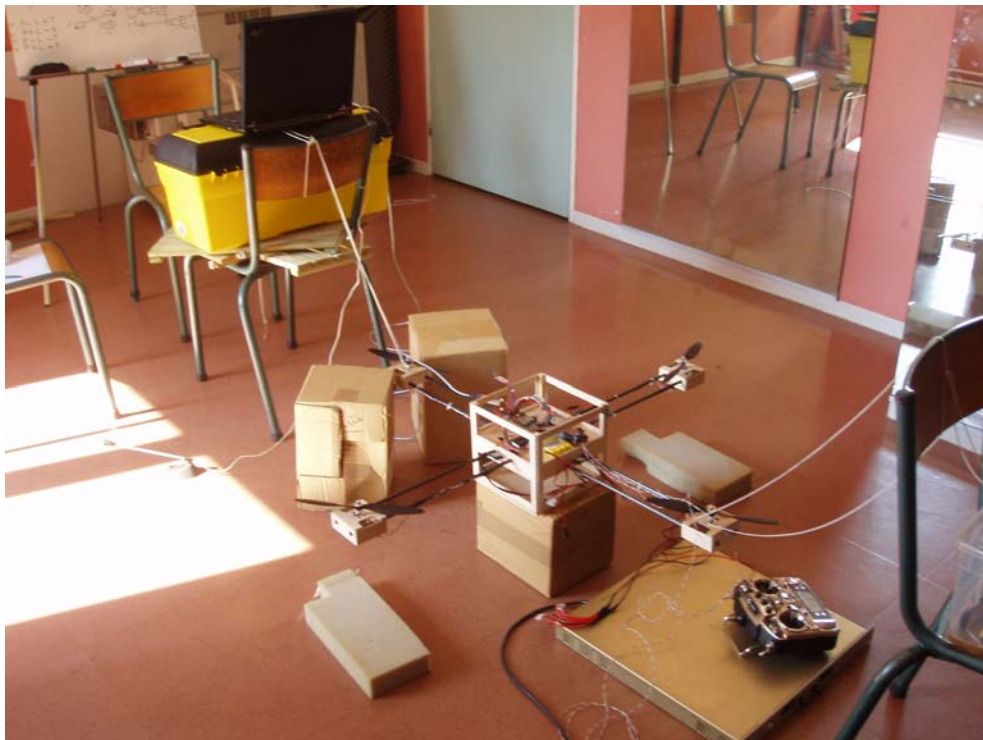


Figure 8.2 : Photographie du dispositif de test

8.2.2 Résultats des tests

Les coefficients proportionnel, vitesse et accélération utilisés par la suite sont ceux que l'on peut régler dans l'interface Horizon (voir guide d'utilisation de la MP2028 et la partie 7.3.2).

La solution du correcteur PID déterminé théoriquement a été mise en place, mais entraîne des oscillations du drone, plus ou moins amplifiées en fonction de la modification apportée aux coefficients. Cette différence entre pratique et théorie est sûrement due aux approximations du modèle ou plus précisément à une sous-estimation du retard dans la boucle de commande engendrée par l'ensemble moteur/variateur et pas les organes de mesure.

La donnée d'accélération a donc été utilisée afin d'obtenir une information plus rapide sur la position du drone [10]. Cette technique nous permet ainsi d'anticiper sur la modification de vitesse angulaire en observant et en corrigeant directement en se référant à l'accélération angulaire.

Nous allons donc ici effectuer un raisonnement par expérience afin de déterminer les meilleurs coefficients pour le correcteur. 5000 est une valeur moyenne pour les coefficients. Les conclusions des tests sont tirées du retour ou non du drone à sa position d'équilibre à la suite d'une perturbation de commande de type dirac et échelon.

Prop.	Vitesse	Acc.	Résultats du test
5000	0	0	Oscillations car retard à la correction
0	5000	0	Oscillations identiques aux problèmes connus pour la solution avec gyroscopes (même donnée).
0	0	5000	Semble plus stabilisable, avec des réactions rapides à des échelons de commande.
5000	0	1000	Bonne stabilisation
5000	0	5000	Très bonne stabilisation sauf à très forte inclinaison → Coefficient retenu pour l'accélération = 5000
5000	0	7000	Oscille à haut régime
5000	0	10000	Oscille
1000	0	5000	Ne stabilise pas autour de la position demandée
3000	0	5000	Bonne stabilisation mais moins bien que 5000
6000	0	5000	Oscille faiblement
7000	0	5000	Oscille faiblement → Coefficient retenu pour le proportionnel = 5000 (nous n'avons pas trouvé de résultats clairement meilleurs avec d'autre coefficients).
5000	5000	5000	Oscille
5000	2500	5000	Oscille après un certain nombre de petites perturbations répétitives.
5000	1000	5000	Oscillations faibles
5000	500	5000	Très bonne stabilisation → Coefficient retenu pour la vitesse = 500

A la suite des tests, le correcteur proportionnel/vitesse/accélération optimal a pour coefficients : 5000/500/5000.

La suite logique de ce travail expérimental pourra consister à trouver un dispositif de test permettant de libérer le drone sur ses trois axes et de voir quelles sont les modifications à apporter aux coefficients précédents pour par exemple compenser des problèmes de couplage entre correcteurs.

8.3 Courbe Masse-Puissance

A l'aide des mixeurs uniquement (sans les gyroscopes connectés), nous avons pu tracer une courbe de puissance en fonction du poids. Le point de mesure correspond à la limite de sustentation du quadrirotor. Les résultats sont donnés dans le tableau 8.1 et la courbe est présentée figure 8.1.

Masse (g)	Tension (V)	Intensité (A)	Puissance totale (W)	Puissance par moteur (W)
75	10	16	160	40
100	10	17	170	42,5
125	10	17,8	178	44,5
135	10	18	180	45
200	10	20	200	50
270	10	22	220	55
350	10	23	230	57,5
405	10	26	260	65
540	10	30	300	75
700	12	27	324	81

Tableau 8.1 : Données expérimentales

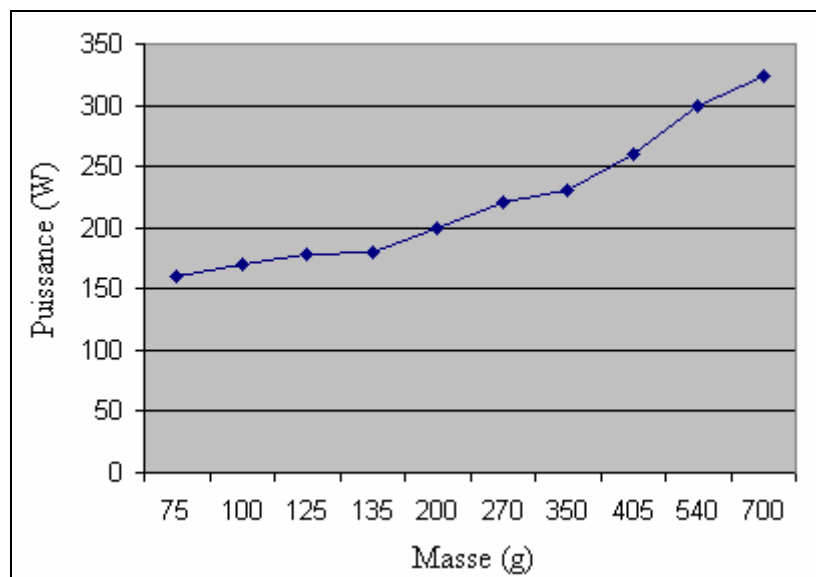


Figure 8.3 : Courbe puissance = f (masse)

9. Conclusion

Ainsi nous avons lors de ce PIP construit deux structures de quadrirotor dont une est opérationnelle. Cette dernière pouvant être équipée de deux plateformes différentes : l'une contenant la solution mixeurs-gyroscopes qui nous a pas apporté les satisfactions espérée et l'autre constituée de la carte Micro-Pilot qui a permis de stabiliser (grâce à un correcteur PID) le drone de roulis et donc par symétrie le tangage est aussi stabilisé. Nous n'avons pas effectué de test pour l'axe de lacet, on peut penser que cette stabilisation sera relativement aisée.

Pour la suite du projet, il est nécessaire de se demander quelles performances doit réaliser le drone. En effet, le drone se doit de posséder une bonne stabilité mais aussi une bonne manœuvrabilité. Dans notre projet, nous nous sommes seulement intéressés à la stabilité. Cependant, la stabilité dégrade la manœuvrabilité. Pour déplacer le drone, il faut en effet s'écarter de la position d'équilibre afin de déclencher le mouvement. Il serait intéressant de trouver un compromis entre stabilité et manœuvrabilité. L'automatisation complète du drone serait alors possible en implémentant la carte MicroPilot.

10. Annexes

10.1 Annexe : Schémas électroniques de la solution à base d'AOP [11]

Cette solution non-réalisée est donnée en annexe à titre indicatif. Elle présente un schéma qui fonctionne en théorie, mais qui n'est pas réalisable en pratique, notamment à cause du filtrage initial pour récupérer la valeur moyenne du signal émis par le récepteur, mais aussi à cause des ordres de grandeurs des tensions.

La figure 10.1 présente le schéma général du système de commande. Un filtre passe-bas qui peut être utilisé est donné figure 10.2.

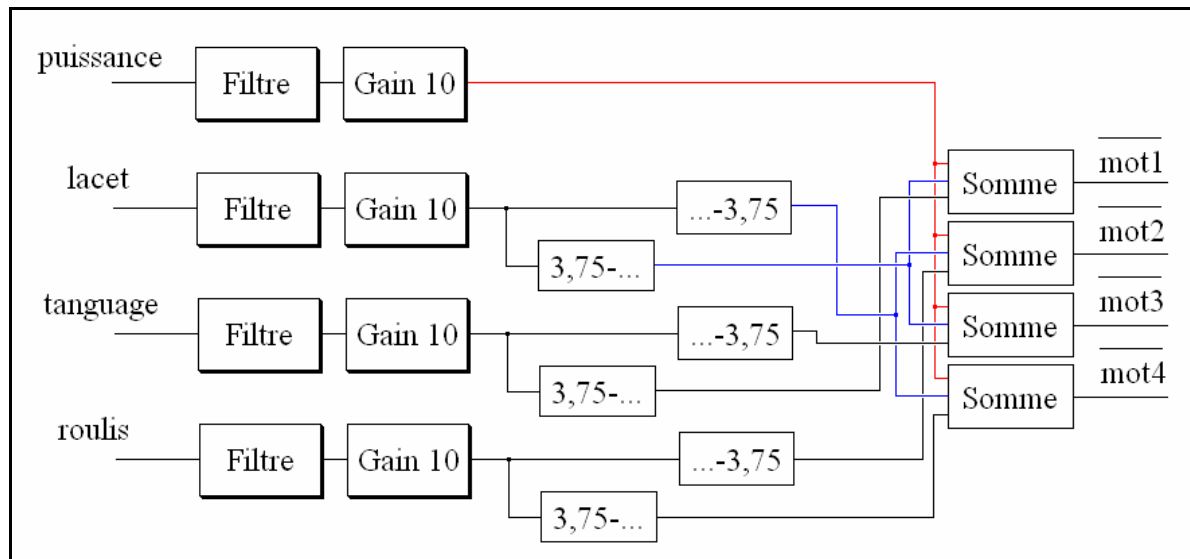


Figure 10.1 : Schéma général

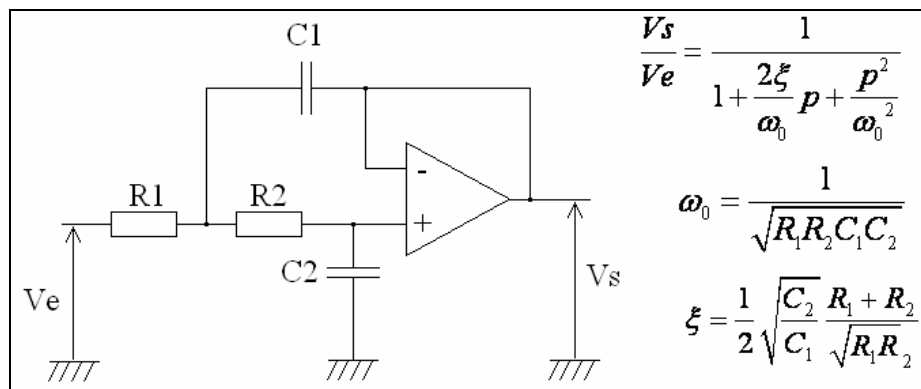


Figure 10.2 : Filtre passe-bas

Un gain de 10, figure 10.3, permet d'obtenir une tension entre 2,5V (0%) et 5V (100%). Ensuite, des soustractions sont effectuées avec une tension de 3,75V (qui peut être obtenue précisément à l'aide du régulateur et d'un pont diviseur de tension) pour obtenir des tensions variant de -1,25V à 1,25V ou de 1,25V à -1,25V lorsque la commande varie de 0% à 100% (voir figure 10.4). Un gain plus élevé pourrait fournir des tensions plus fortes, en utilisant une tension de référence plus grande que 3,75V.

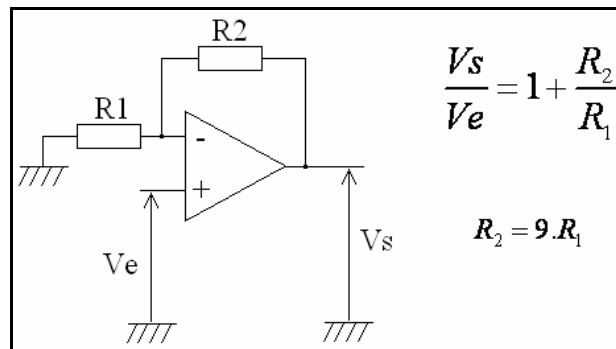


Figure 10.3 : Gain de 10

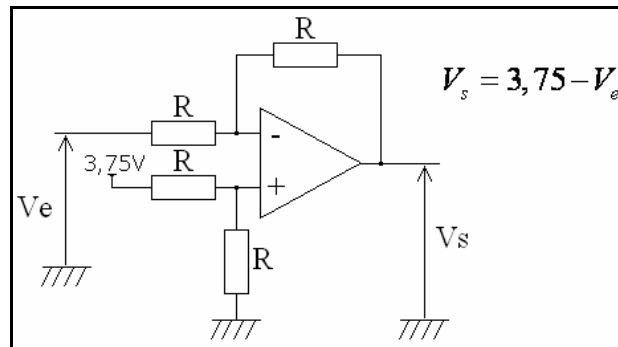


Figure 10.4 : Soustracteurs

Des sommes sont en suite effectuées pour réaliser les mêmes équations qu'au paragraphe 7.2.3. Le schéma figure 10.5 présente une manière d'effectuer la somme de trois tensions. Les tensions en sortie des sommateurs varient alors de 0V à 7,5V, sachant que l'on souhaite qu'elles restent comprises entre 2,5V et 5V (pour satisfaire ce critère, on peut utiliser un AOP qui sature à 5V puis un comparateur avec un transistor pour la limite basse de 2,5V).

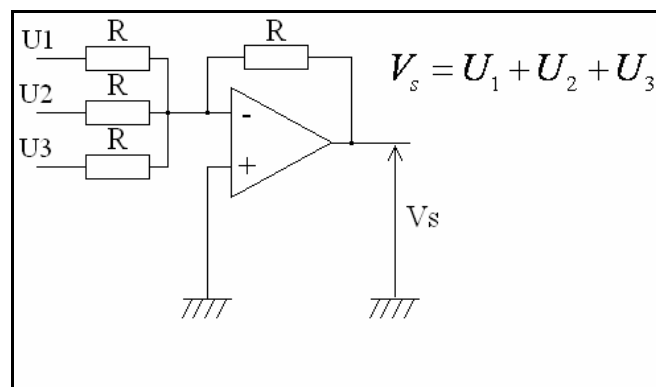


Figure 10.5 : Sommateur

Il est alors possible de concevoir des PID (à l'aide d'AOP), placés juste après les gains de 10.

Mais il reste le signal créneau à générer en sortie du circuit. Le plus simple est d'utiliser un oscillateur (figure 10.6) et quatre comparateurs dont les amplificateurs opérationnels saturent à $+V_{sat} = 5V$ et à $-V_{sat} = 0V$ (alimentation dissymétrique).

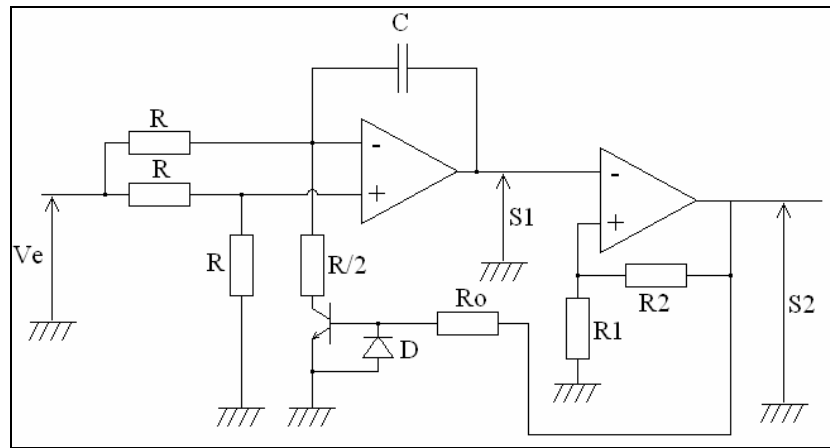


Figure 10.6 : oscillateur commandé en tension (VCO)

La figure 10.7 présente les courbes de S1 et S2. Il reste alors à régler les composants pour obtenir la période de 10 ms souhaitée ainsi qu'un rapport cyclique de 0,05 à 0,1 lorsque la tension de commande varie de 5V à 2,5V.

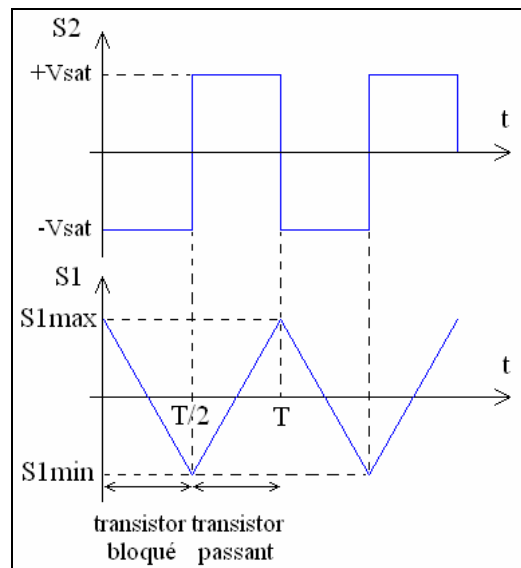


Figure 10.7 : Allure des tensions de l'oscillateur

Nous avons les relations :

$$S1_{\min} = \frac{R_1}{R_1 + R_2} \cdot (-V_{\text{sat}}) ; S1_{\max} = \frac{R_1}{R_1 + R_2} V_{\text{sat}} \text{ et } T = 8 \frac{V_{\text{sat}}}{E} RC \frac{R_1}{R_1 + R_2}$$

Il faut : $\frac{5 - S1_{\min}}{S1_{\max} - S1_{\min}} = \frac{T/2 - 0,5}{T/2} = 0,95$ et $\frac{2,5 - S1_{\min}}{S1_{\max} - S1_{\min}} = \frac{T/2 - 1}{T/2} = 0,9$

On obtient avec ces deux équations à deux inconnues : $S1_{\max} = 7,5V$ et $S1_{\min} = -42,5V$, d'où la difficulté de réaliser un tel schéma. Pour rendre la valeur de $S1_{\min}$ acceptable, il faudrait en fait avoir un écart faible entre les tensions qui correspondent à 0% et à 100% de la commande. Ce qui est difficilement acceptable en pratique, à cause des tensions parasites qui sont inévitables.

10.2 Annexe : Guide d'utilisations de la carte MP2028

Ce guide reprend celui proposé par Andréas Goërmer [12] en ce qui concerne l'utilisation de la carte MP2028 pour le drone Casper ; nous y avons ajouté les manipulations à effectuer en se fondant sur notre expérience et de la documentation de la carte MP2028g fournie par Micropilot.

Nous supposons que les logiciels sont installés dans le répertoire : C:\Program Files\MicroPilot (que nous nommerons rep \ par la suite). Il sera également nécessaire d'ouvrir des fichiers codés en texte, pour cela nous conseillons d'utiliser Crimson editor téléchargeable gratuitement sur Internet ce qui permettra en plus de lire les fichiers en C.

1^{ère} étape : Branchement de la carte Micropilot

Pour charger le programme, il suffit de brancher les câbles power et COM. Le premier doit être branché sur un générateur ou une batterie délivrant 9.6V (tension optimale pour la carte), et le second doit être relié à l'ordinateur. Pour ce faire il faut disposer d'un câble série afin de relier la carte au port série du PC (ce port est en général le port N°1 de l'ordinateur, ce qui ne posera pas de problème). Si l'ordinateur ne dispose pas de port série (ordinateur portable), il faut utiliser comme intermédiaire un câble série USB. Dans ce dernier cas, le chargement du programme sera impossible par défaut, car initialement le port USB de l'ordinateur n'est pas le port N°1 et l'exécutable permettant de charger le programme dans la carte comme nous le verrons plus en détail par la suite, ne travaille que sur le COM 1. La solution pour s'en sortir et d'aller dans le gestionnaire de périphériques (clic droit sur poste de travail, Gérer), de désactiver l'utilisateur du port N°1 après avoir vérifié qu'il ne sert à rien d'important, et régler le port USB sur COM 1.

2^{ème} étape : Compilation du code

Pour la 1^{ère} utilisation :

- Copier le contenu du dossier « rep \XTENDER3\examples\userCode » vers le nouveau dossier « rep \XTENDER3\examples\Quadrirotor »
- Ouvrir « rep\XTENDER3\examples\userCode\userCode.c » et le sauvegarder sous le nom qui sera utilisé plus tard. Nous utiliserons « Quadrirotor_userCode.c ».

Modifier le fichier de la manière suivante:

ligne 72, remplacer « int » par « long »

ligne 34, insérer: « #include "usercode.h »

sauver sous « Quadrirotor_userCode.c »

Implémenter votre propre code, le sauver (toujours dans Quadrirotor_userCode.c).

- A l'aide d'un éditeur de texte modifier le fichier (clic droit, modifier) :
« rep\XTENDER3\examples\ Quadrirotor \gdb.bat » Changer le répertoire
« %MicroPilotSDK3%/examples/userCode/gnuc/ »
par: « %MicroPilotSDK3%/examples/ Quadrirotor /gnuc/ »

Sauver «gdb.bat»

- Ouvrir le fichier « rep\XTENDER3\examples\ Quadrirotor \ makefile » à l'aide d'un éditeur de texte. A la ligne 42: remplacer « exampleUserCode.c » par « Quadrirotor_userCode.c ».
- Modifier le fichier « appendUser.bat » en lui ajoutant juste avant la ligne “pause”, la ligne : copy gnu\mflyanduser.bin “%MicroPilotSDK3%bin\”
- Dans le répertoire “rep\XTENDER3\bin” créer le fichier “flashld-mflyanduser.bat” en copiant le fichier “flashld-mfly.bat” en changeant son nom et en l’ouvrant avec l’éditeur texte : changer à la première ligne “mfly.bin ” en “ mflyanduser.bin ” et enlever le pause (2^{ème} ligne).

A effectuer à chaque nouvelle modification du code :

Il est possible pour compiler le code de double cliquer sur le fichier make-usercode.bat, mais la console ne restera pas assez longtemps sur l'écran pour se rendre compte d'une erreur dans le programme. Pour remédier à cet inconvénient, nous conseillons d'ouvrir une console (invite) et d'utiliser la commande : “cd rep\MicroPilot\XTENDER3\examples\ Quadrirotor ” afin de se placer dans le bon dossier, puis exécuter “make-usercode.bat” pour compiler le code. Vérifier qu'il n'y ait pas d'erreur : si il y a simplement signalement de variables non utilisées, cela ne posera pas de problème, en revanche le signalement d'une variable non déclarée est plus grave et dans ce cas il faudra revoir le code.

Exécuter ensuite “appendUser.bat” (double cliquer sur le fichier ne pose ici pas de problème mais puisque la console est ouverte...) Cette commande permet de créer un fichier “mflyanduser.bin” dans le fichier “rep\MicroPilot\XTENDER3\examples\ Quadrirotor\gnu” et de le copier dans le dossier “rep\MicroPilot\XTENDER3\bin\”.

Aller dans ce dossier et vérifier la création de ce fichier (regarder la date) afin d'être sûr que cela soit la dernière version du fichier.

Vérifier le branchement de la MP2028g à l'aide du câble au port série de l'ordinateur et couper l'alimentation de la carte.

Dans le répertoire “rep\XTENDER3\bin” et exécuter le batch “flashld-mflyanduser.bat” (en double cliquant dessus par exemple).

Mettre la carte sous tension quand cela est demandé par l'invite de commande : une tension de 9V est recommandée au minimum. Si cela marche, l'invite de commande indique “loading.....” Il faut patienter environ 3 minutes afin que le programme soit complètement chargé dans la carte (la fin du chargement est indiquée par un bip).

Lorsque le code est actualisé dans la carte, mettre la carte hors tension et fermer l'invite de commande.

3ème étape : Session Hyperterminal

Ouvrir une session Hyperterminal, patienter quelques secondes puis mettre la carte MP2028 sous tension, et patienter le temps de l'initialisation (test des gyros,...).

Si rien ne se passe lorsque vous mettez la carte sous tension c'est que la liaison a été rompue, ce qui arrive souvent en cas d'utilisation du port USB. La solution est de débrancher et rebrancher la prise USB.

Lors de l'initialisation, il est possible de voir si le GPS est activé ou non. Lors d'essais en intérieur, il est recommandé de le désactiver. En effet, la détection risque de ne pas avoir lieu et donc l'initialisation peut échouer. Comme la procédure d'initialisation du GPS est bloquante, la carte restera en l'état et il ne sera pas possible de poursuivre. Pour désactiver l'étalonnage du GPS presser "ffff" et attendre que le GPS se désactive.

Une fois le GPS désactivé, accéder au menu en pressant "qqqq".

Changer la variable n° 588 en tapant "p", entrer le numéro de variable : "588", et changer sa valeur de 0 à 2. (Enable user defined Code) mettre à jour le code en appuyant sur "w".

Lorsque le code est actualisé, mettre la carte hors tension pendant quelques secondes, puis rallumer la carte et laisser l'initialisation se poursuivre (cette dernière étape est facultative mais permet tout de même de voir si le 0 a bien été transformé en 2).

Une fois ceci terminé, éteindre l'Hyperterminal.

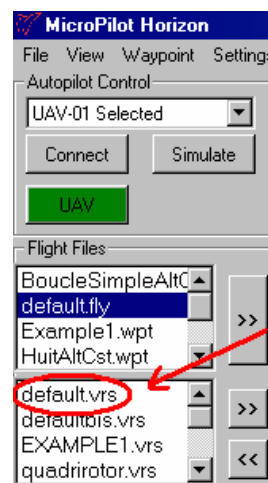
4ème étape : Utilisation de Horizon

Il est possible que lors de votre utilisation de Horizon il y ait des problèmes de communication qui sont dus encore une fois à la liaison USB. La technique pour y remédier est de sortir de Horizon, débrancher la prise USB, couper l'alimentation de la carte, rebrancher la prise USB, remettre la carte sous tension, et enfin revenir dans Horizon (cette technique devrait marcher mais si tel n'est pas encore le cas, juste débrancher et rebrancher la prise USB). En revanche il ne devrait pas y avoir de problème en utilisant la liaison série directement sur le port série d'un ordinateur fixe.

Avant de se connecter la carte (à l'aide de l'icône « Connect »), il faut effectuer les réglages suivants :

- A la première utilisation :

Double clique sur default.vrs

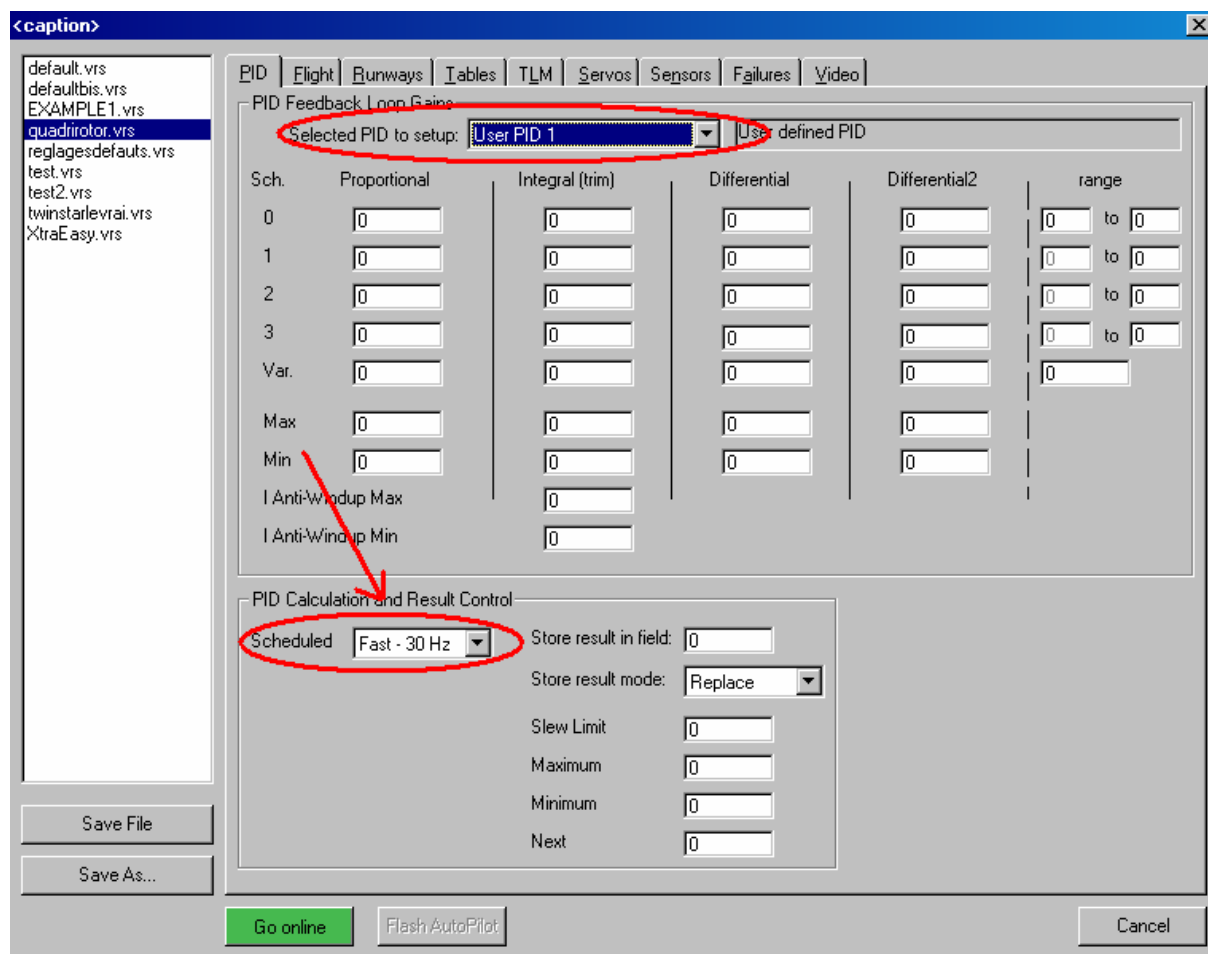


Ce fichier n'est pas modifiable et il faut donc créer un fichier quadrirotor.vrs en cliquant sur Save As. Une fois ce fichier crée, quitter le fichier default.vrs.

- A chaque utilisation :
 - Ouvrir « quadrirotor.vrs » en double cliquant dessus, puis cliquer sur **Go online**. S'il n'y a pas de problème de liaison, le bouton doit devenir vert.

- Dans l'onglet PID, faire défiler Selected PID to setup et mettre à 30Hz le User PID qui contient la version du programme à tester (le programme étant en effet découpé en User PID).

Par exemple ici avec le User PID 1 :



Vérifier que les autres User PID sont bien sur *Scheduled : Never*, afin de ne pas créer d'interactions dans le programme.

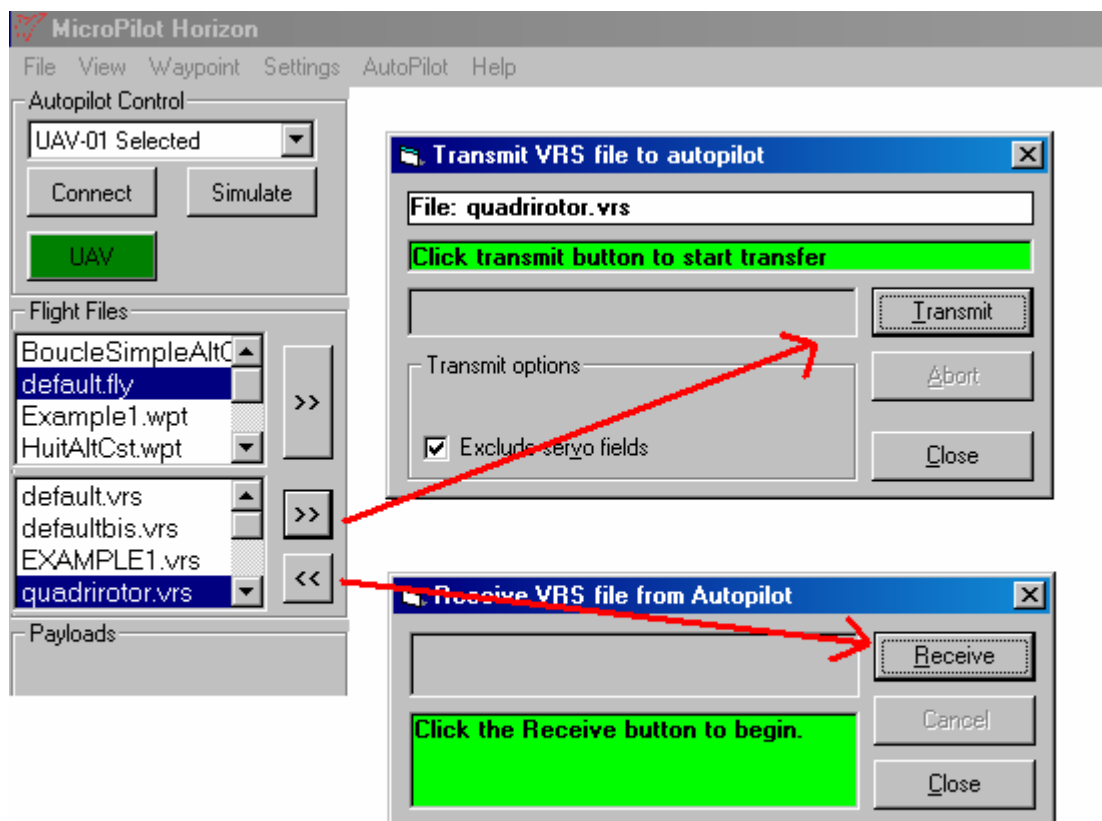
- Dans l'onglet Servos, il est possible de régler la course des servos.
- Dans l'onglet Sensors, décocher Enable AGL Sensor (on n'utilise pas l'AGL pour le quadrirotor) et cocher Enable Compass.
- L'onglet Flight permet de régler les coefficients des différents correcteurs PID implémentés dans le programme.

Tableau des correspondances Horizon / coefficients du programme :

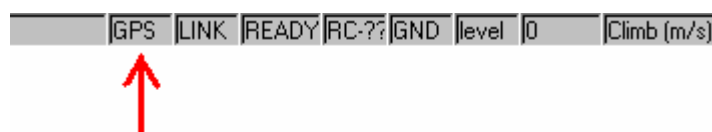
Flight dans Horizon	Coefficients dans le programme
Rotation Speed	RP0
Cruise Speed	RP1
Climb Speed	RP2
Approach Speed	RP3
Climb Margin	RP4
Flare Altitude	RP5

Waypoint Dia.	SP1
Wpt. Reduce Turn	SP2
Cct Alt	SP3

- Une fois tout ceci réglé, cliquer sur **Flash AutoPilot**, puis **Go Offline** pour enfin quitter la fenêtre de réglage.
- Comme cela sera indiqué, afin de sauvegarder les réglages, qui viennent d'être effectué sur la carte, dans le fichier quadrirotor.vrs, il faut cliquer sur << puis **Receive**. (Pour réutiliser la carte sans changer les réglages, il suffira alors de cliquer sur >> : **Transmit**).



- Cliquer sur l'icône « Connect ». Horizon doit alors communiquer avec la carte, l'icône de statut « ready » apparaît rouge. L'initialisation du GPS peut être longue, afin de passer outre il faut double cliquer sur l'icône de statut « GPS » (pouvant apparaître « OSAT »), elle doit passer au jaune. Lorsque « READY » est vert alors tout est prêt pour utiliser le code utilisateur.



10.3 Annexe : Programme informatique

```

/*****

```

```

Filename:  quadrirotor_userCode.c

```

```

Project:  1) Simulator
          2) AutoPilot firmware codebase
          3) MicroPilot SDK

```

```

Copyright: (c) 1998-2003 MicroPilot

```

```

Author:   Howard Loewen

```

```

Synopsis: <hh-todo>

```

```

CVS-TAG : $Id: userCode.c,v 1.1 2003/09/17 14:52:13 hloewen Exp $

```

```

Comments: This header file is part of an SDK that gets shipped to
          end users, it should ONLY contain code that is deemed
          necessary for SDK based applications to compile.
          All code and data clasified as private to MicroPilot should
          reside in source files that are NOT included with the SDK.

```

```

          Note that this is a stub function and NOT an example so it should
          contain null versions of all functions.

```

```

History:  HH:2003-Jan-08:refactored code cleanup

```

```

*****/

```

```

#include "usercode.h"
#include "simdll.h"

```

```

#ifdef _WIN32
#define _main
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <windows.h>
#endif /* _WIN32 */

```

```

/* the code in this file is compiled in two different fashions: to run with the MicroPilot simulator and
to run on the actual MP2028 */

```

```

#ifdef _WIN32
BOOL APIENTRY DllMain( HANDLE hModule, DWORD  ul_reason_for_call, LPVOID
lpReserved )
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
            break;

        case DLL_THREAD_ATTACH:
            break;

        case DLL_THREAD_DETACH:
            break;
    }
}

```

```

    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}

#endif /* _WIN32*/

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

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

int mpUserInit( int (*function)( void **result, int id))
{
    saveGetVarPointer = function;
    return USER_RETURN_INIT_OK;           //return 1;
}

/* mpUserEvent - This is the function that is called when the user code is implimented */
int mpUserEvent( long *result, int userEventIdentifier)
{
    //variable declaration (pointers)
    static long *hardwareDisable;
    static long *ref_Roll_raw;           //commanded roll angle (raw value)
    static long *cRoll;                 //current roll
    static long *iRoll;                 //instantaneous roll
    static long *iRoll_p;               //instantaneous roll dot

    static long *ref_Pitch_raw;         //commanded pitch angle (raw value)
    static long *cPitch;                //current pitch
    static long *cPitch_p;              //current pitch dot
    static long *iPitch;                //instantaneous pitch
    static long *iPitch_p;              //instantaneous pitch dot

    static long *iYaw_p;                //measured yaw rate
    static long *ref_Hdg_raw;           //commanded Hdg (raw value)
    static long *ref_ident_raw;         //commanded Hdg identification (raw value)
    static long *Hdg;                   //compass Hdg

    static long *Radio_raw;             //Radio switch (raw value)
    static long *ref_Thrust_raw;        //Radio switch (raw value)

    static long *Rp0_raw;               //coefficient 0 of R (raw)
    static long *Rp1_raw;               //coefficient 1 of R (raw)
    static long *Rp2_raw;               //coefficient 2 of R (raw)
    static long *Rp3_raw;               //coefficient 3 of R (raw)
    static long *Rp4_raw;               //coefficient 4 of R (raw)
    static long *Rp5_raw;               //coefficient 5 of R (raw)
    static long *Sp1_raw;               //coefficient 1 of S (raw)
    static long *Sp2_raw;               //coefficient 2 of S (raw)
    static long *Sp3_raw;               //coefficient 3 of S (raw)

```



```

static long *Sp4_raw;           //coefficient 4 of S (raw)

static long Rp0;                //coefficient 0 of R
static long Rp1;                //coefficient 1 of R
static long Rp2;                //coefficient 2 of R
static long Rp3;                //coefficient 3 of R
static long Rp4;                //coefficient 4 of R
static long Rp5;                //coefficient 5 of R
static long Sp1;                //coefficient 1 of S
static long Sp2;                //coefficient 2 of S
static long Sp3;                //coefficient 3 of S
static long Sp4;                //coefficient 4 of S

static long *fServo3;
static long *fServo5;
static long *fServo6;
static long *fServo7;
static long *fServo8;

static long *disp1;
static long *disp2;
static long *disp3;
static long *disp4;
static long *disp5;
static long *disp6;
static long *disp7;
static long *disp8;
static long *disp9;
static long *disp10;

//variable declaration
static long signal_min=2220;    //2222 or
static long signal_zero=3045;

static long Thrust_max=50000;   //par tests
static float Thrust_ref;

static long Roll_err;
static long Roll_max=45*3.14/180*1024; //maximum (magnitude) commanded roll angle
[rad*1024]
static float Roll_ref;          //commanded roll angle (transformed value)
static long Roll_mix;          //input for servo mixing (=output of roll controller)
static long Pitch_err;
static long Pitch_max=45*3.14/180*1024; //maximum (magnitude) commanded pitch angle
[rad*1024]
static float Pitch_ref;        //commanded pitch angle (transformed value)
static long Pitch_mix;         //input for servo mixing (=output of pitch controller)

static long Yaw_mix;           //input for servo mixing (=output of Yaw_p
controller)
static long Hdg_err;
static long Hdg_max=30000;      //maximum (magnitude) commanded Hdg [deg*1000]
static long Hdg_max_p=150;     //maximum (magnitude) commanded Hdg_p
[deg/(30*s)*100] or [deg/(5*s)*100]
static float Hdg_ref=0;        //commanded Hdg (transformed value)
static float Hdg_ref_p;       //commanded Hdg (transformed value)

```

```
static long hdgsimu=0;
static long pitchsimu=0;
static long rollsimu=0;
```

//variables for RST controller

```
static long com0=0;
static long hdg0=0;
static long hdg1=0;
static long hdg2=0;
static long hdg3=0;
static long hdg0f=0;
static long hdg1f=0;
static long hdg2f=0;
static long hdg3f=0;
static long err0=0;
static long err1=0;
static long err2=0;
static long err3=0;
static long err4=0;
static long err5=0;
```

```
static long yaw_pp0=0;
static long roll_pp0=0;
static long pitch_pp0=0;
```

```
static long yaw_p0=0;
static long yaw_p1=0;
static long yaw_p2=0;
static long yaw_p3=0;
static long yaw_p0f=0;
static long yaw_p1f=0;
static long yaw_p2f=0;
static long yaw_p3f=0;
static long roll_p0=0;
static long roll_p1=0;
static long pitch_p0=0;
static long pitch_p1=0;
```

```
static long y0f=0;
static long y1f=0;
static long y2f=0;
static long y3f=0;
static long u0=0;
static long u1=0;
static long u2=0;
static long u3=0;
static long u4=0;
```

```
static long thrust=0;
static long t=0;
```

//variables for generation of PRBS

```
static long b5;
static long b9;
```

```
//heading (now)
//heading (now)
//previous heading
//previous (2 steps) heading
//previous (3 steps) heading
//heading (now)(filtered)
//previous heading (filtered)
//previous (2 steps) heading (filtered)
//previous (3 steps) heading (filtered)
//heading err now
//heading (1 step before)
//heading (2 step before)
//heading (3 step before)
//heading (3 step before)
//heading (3 step before)
```

```
//yaw_pp (now)
//roll_pp (now)
//pitch_pp (now)
```

```
//yaw_p (now)
//yaw_p (1 step before)
//yaw_p (2 steps before)
//yaw_p (3 steps before)
//yaw_p (now) (filtered)
//yaw_p (1 step before) (filtered)
//yaw_p (2 steps before) (filtered)
//yaw_p (3 steps before) (filtered)
//roll_p (now)
//roll_p (1 step before)
//pitch_p (now)
//pitch_p (1 step before)
```

```
//y (now) (filtered)
//y (1 step before) (filtered)
//y (2 steps before) (filtered)
//y (3 steps before) (filtered)
//commanded signal for actuator
//commanded signal for actuator 1 sample before
//commanded signal for actuator 2 samples before
//commanded signal for actuator 3 samples before
//commanded signal for actuator 4 samples before
```

```
//thrust signal (temp)
//correction factor (from look up table)
```

```

static long prbs_tmp;
static long sqz_count=1000;

switch( userEventIdentifier)
{
case USER_HARDWARE_INITIALIZED :
    getMPVarPointer( (void **)&hardwareDisable, 594);
    getMPVarPointer( (void **)&ref_Pitch_raw, 1152);    //input signal com_Pitch
(ELEVATOR servo chan)
    getMPVarPointer( (void **)&ref_Roll_raw, 1153);    //input signal
com_Roll (AILERON servo chan)
    getMPVarPointer( (void **)&ref_Hdg_raw, 1154);    //input signal
com_Yaw (RUDDER servo chan)
    getMPVarPointer( (void **)&ref_Thrust_raw, 1155); //input signal com_Thrust
(THROTTLE servo chan; power supply RC unit)
    getMPVarPointer( (void **)&Radio_raw, 1156);    //input signal Radio
(yellow channel)
    getMPVarPointer( (void **)&ref_ident_raw, 6650); //input signal com_Yaw
(identification)

    getMPVarPointer( (void **)&cPitch, 1057);
    getMPVarPointer( (void **)&cRoll, 1059);
    getMPVarPointer( (void **)&cPitch_p, 1058);
    getMPVarPointer( (void **)&iPitch, 1266);    //rad*1024
    getMPVarPointer( (void **)&iRoll, 1267);    //rad*1024
    getMPVarPointer( (void **)&iPitch_p, 1403); //rad*1024/s
    getMPVarPointer( (void **)&iRoll_p, 1404); //rad*1024/s
    getMPVarPointer( (void **)&iYaw_p, 1405);  //rad*1024/s
    getMPVarPointer( (void **)&Hdg, 1208);    //deg*100

    getMPVarPointer( (void **)&Rp0_raw, 2);    //var rotation speed used
    getMPVarPointer( (void **)&Rp1_raw, 111); //var cruise speed used
    getMPVarPointer( (void **)&Rp2_raw, 5);    //var climb speed used
    getMPVarPointer( (void **)&Rp3_raw, 9);    //var approach speed used
    getMPVarPointer( (void **)&Rp4_raw, 13);   //var climb margin
    getMPVarPointer( (void **)&Rp5_raw, 14);   //var flare altitude
    getMPVarPointer( (void **)&Sp1_raw, 122);  //var waypoint diameter
    getMPVarPointer( (void **)&Sp2_raw, 150);  //var waypoint reduce turn
    getMPVarPointer( (void **)&Sp3_raw, 152);  //var cct altitude
    getMPVarPointer( (void **)&Sp4_raw, 528);  //var descent rate used

    getMPVarPointer( (void **)&fServo3, 1229);
    getMPVarPointer( (void **)&fServo5, 1231);
    getMPVarPointer( (void **)&fServo6, 1232);
    getMPVarPointer( (void **)&fServo7, 1233);
    getMPVarPointer( (void **)&fServo8, 1234);

    getMPVarPointer( (void **)&disp1, 8004);
    getMPVarPointer( (void **)&disp2, 8005);
    getMPVarPointer( (void **)&disp3, 8006);
    getMPVarPointer( (void **)&disp4, 8007);
    getMPVarPointer( (void **)&disp5, 8008);
    getMPVarPointer( (void **)&disp6, 8009);
    getMPVarPointer( (void **)&disp7, 8010);
    getMPVarPointer( (void **)&disp8, 8011);
    getMPVarPointer( (void **)&disp9, 8012);

```

[illegible]

```

//*****

//*****
//***  userPID1_4    PID Controller for Pitch angle

//*****
//***  userPID1_4_1  Conversion of commanded signal (Pitch axis)
Pitch_ref=((float)*ref_Pitch_raw-(float)signal_zero)/((float)signal_zero-(float)signal_min) *
(float)Pitch_max;    //transformation of input signal in commanded Pitch angel

//*****
//***  userPID1_4_2  Calculation of angular acceleration (pitch)

pitch_p1=pitch_p0;
pitch_p0=*iPitch_p;
pitch_pp0=(pitch_p0-pitch_p1)*100/24; //avec un pas de temps de 0.24 indentique à
celui de l'integrateur

//*****
//***  userPID1_4_3  Calculation of regulation error (Pitch axis)
Pitch_err = Pitch_ref*7 - *iPitch;    //Calculation of error    (input for PID loop)

//*****
//***  userPID1_4_4  Calculation of signal for servo actuator (Pitch axis)
Pitch_mix= Pitch_err* Rp5 - *iPitch_p* Sp1 - pitch_pp0 * Sp3; //PID-controller

//***  End of userPID1_4
//*****

//*****
//**** userPID1_5

//*****
//***  userPID1_5_1  PD Controller for Yaw axis

//*****
//***  userPID1_5_1_1  Conversion of commanded signal (Yaw axis) (on commande
la dévivée du cap pour plus de facilité de pilotage)
Hdg_ref_p=((float)*ref_Hdg_raw-(float)signal_zero)/((float)signal_zero-
(float)signal_min) * (float)Hdg_max_p; //transformation of input signal in commanded Hdg

//*****
//***  userPID1_5_1_2  Integration of commanded signal in order to command an
angle (yaw axis)
Hdg_ref=Hdg_ref+Hdg_ref_p;

//*****
//***  userPID1_5_3  Reparation of owerflow problem of commanded signal
if(Hdg_ref < 0)
{
    Hdg_ref=Hdg_ref+36000;
}

```

```

else
{
    if(Hdg_ref > 36000)
    {
        Hdg_ref=Hdg_ref-36000;
    }
}

//*****
//***  userPID1_5_4  Calculation of angular acceleration (yaw)

yaw_p1=yaw_p0;
yaw_p0=*iYaw_p;
yaw_pp0=(yaw_p0-yaw_p1)/3;

//*****
//***  userPID1_5_5  Calculation of regulation error (Yaw axis)
Hdg_err=Hdg_ref - *Hdg;

//*****
//***  userPID1_5_6  Adjustment of regulation error (Yaw axis)
if(Hdg_err > 18000)
{
    Hdg_err=Hdg_err-36000;
}
else
{
    if(Hdg_err < -18000)
    {
        Hdg_err=Hdg_err+36000;
    }
    else
    {
        //nothing to do
    }
}

//*****
*disp4=Hdg_err;          //monitoring Hdg_ref_m (variable 8004 is used)

//*****
//***  userPID1_5_7  Calculation of signal for servo actuator (Yaw axis)
Yaw_mix = Hdg_err * Rp0 - *iYaw_p * Rp1 - Rp2 * yaw_pp0;

//*****
//***  userPID1_5_8  Adjustment of decimals of Yaw_mix
Yaw_mix = Yaw_mix/100; //à voir...
//*****

//*** end of userPID1_5
//*****

//*****
//***  userPID1_6    Correction of decimals servo command
Roll_mix=Roll_mix/1000;
Pitch_mix=Pitch_mix/1000;
Yaw_mix=Yaw_mix/1000;
//*****

```

```

        *disp1=Roll_mix;
        *disp2=Pitch_mix;
        *disp3=Yaw_mix;

    /***   End of userPID1_6
    /*******

    /*******
    /***   userPID1_7 Contrôle de puissance

        Thrust_ref=((float)*ref_Thrust_raw-(float)signal_zero)/((float)signal_zero-
(float)signal_min) * (float)Thrust_max;

    /***   end of userPID1_7
    /*******

    /*******
    /***   userPID1_8 Servo mixing

        /*******
        /***   userPID1_8_1 Servo mixing (Roll, Pitch, Yaw axis)

            if(*Radio_raw==1){

                *fServo5 = Thrust_ref - Roll_mix + Yaw_mix; //à revoir surtout les coeffs
                *fServo6 = Thrust_ref - Pitch_mix - Yaw_mix;
                *fServo7 = Thrust_ref + Roll_mix + Yaw_mix;
                *fServo8 = Thrust_ref + Pitch_mix - Yaw_mix;

            }
            else{
                *fServo5 = Thrust_ref - Roll_mix;
                *fServo6 = Thrust_ref - Pitch_mix;
                *fServo7 = Thrust_ref + Roll_mix;
                *fServo8 = Thrust_ref + Pitch_mix;

            }

        /*******

    /***   end of userPID1_8
    /*******

    /*******
    /***   userPID1_9 End of userPID1
        return USER_RETURN_REPLACE;

```

10.4 Annexe : Schéma SimuLink

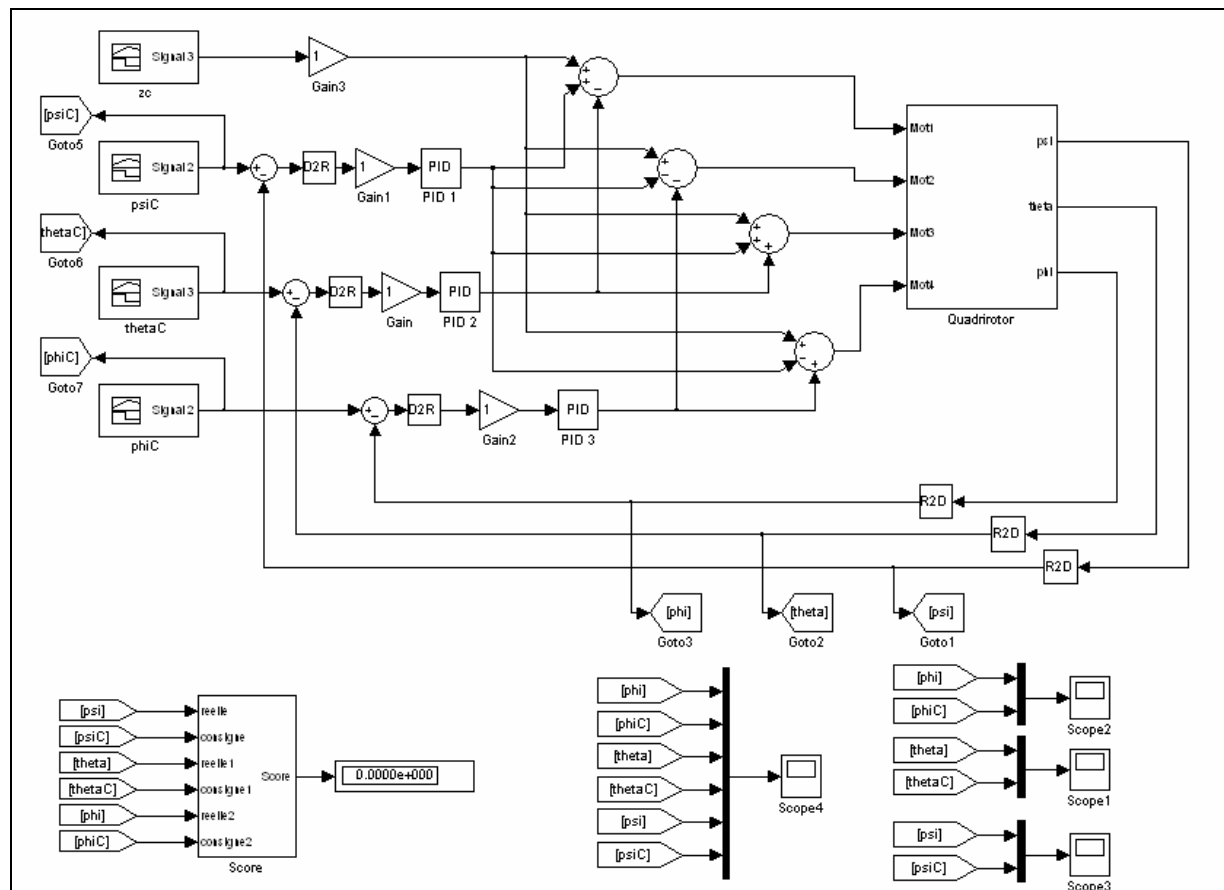


Schéma globale de commande du quadrirotor

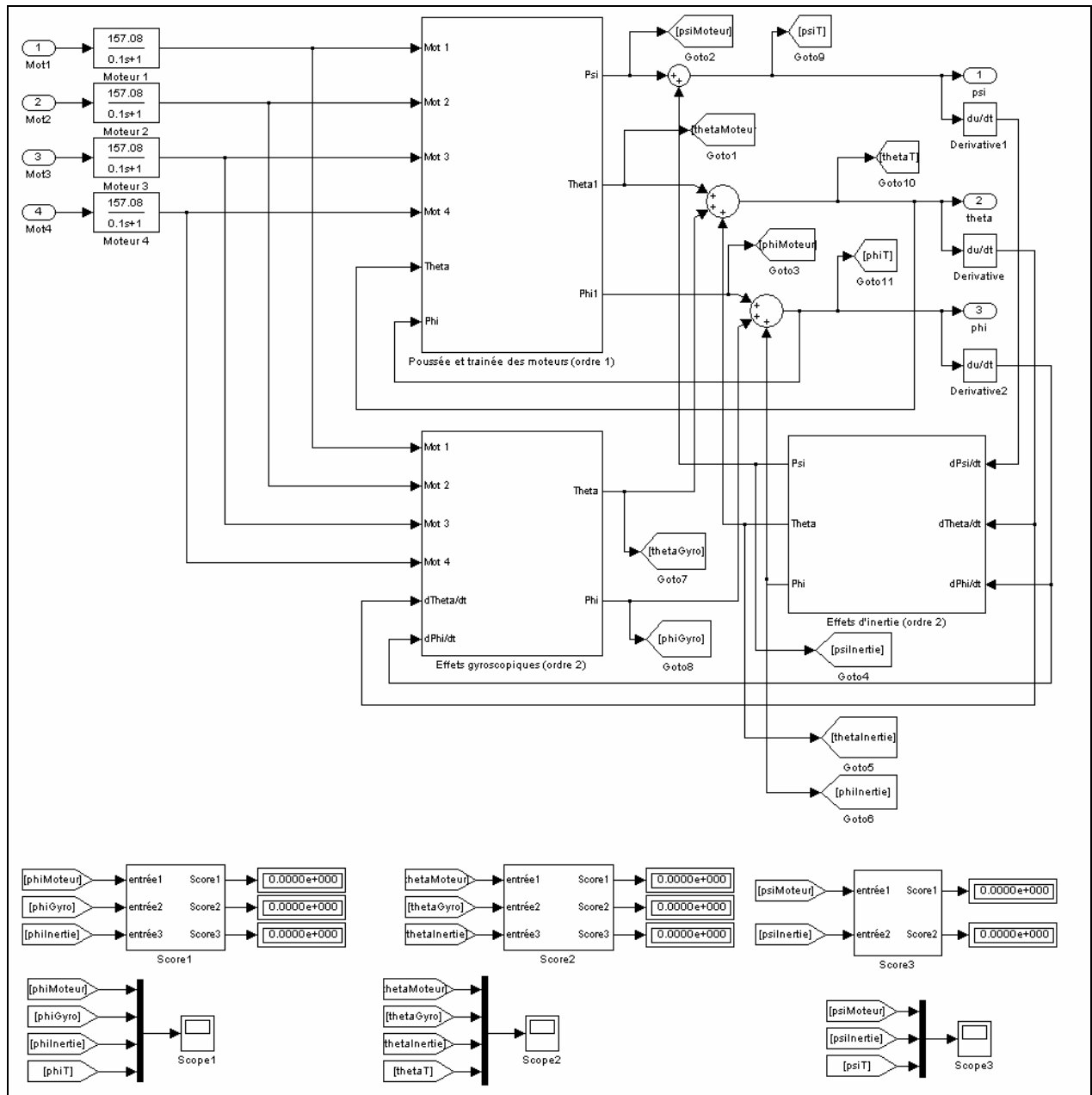


Schéma principale de la dynamique du quadrirotor

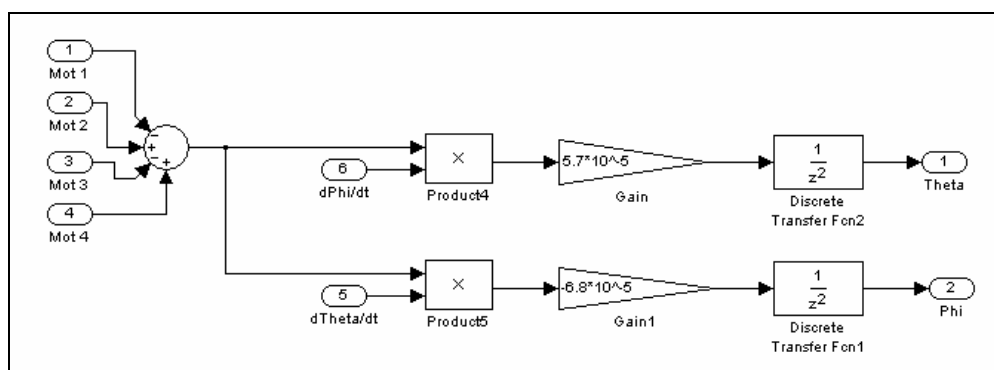


Schéma des effets gyroscopiques

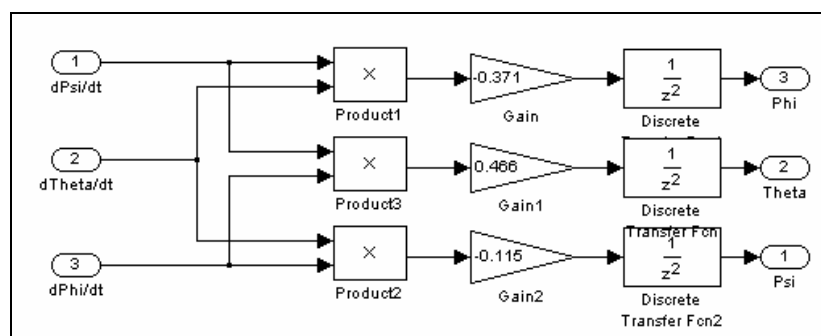


Schéma des effets d'inerties

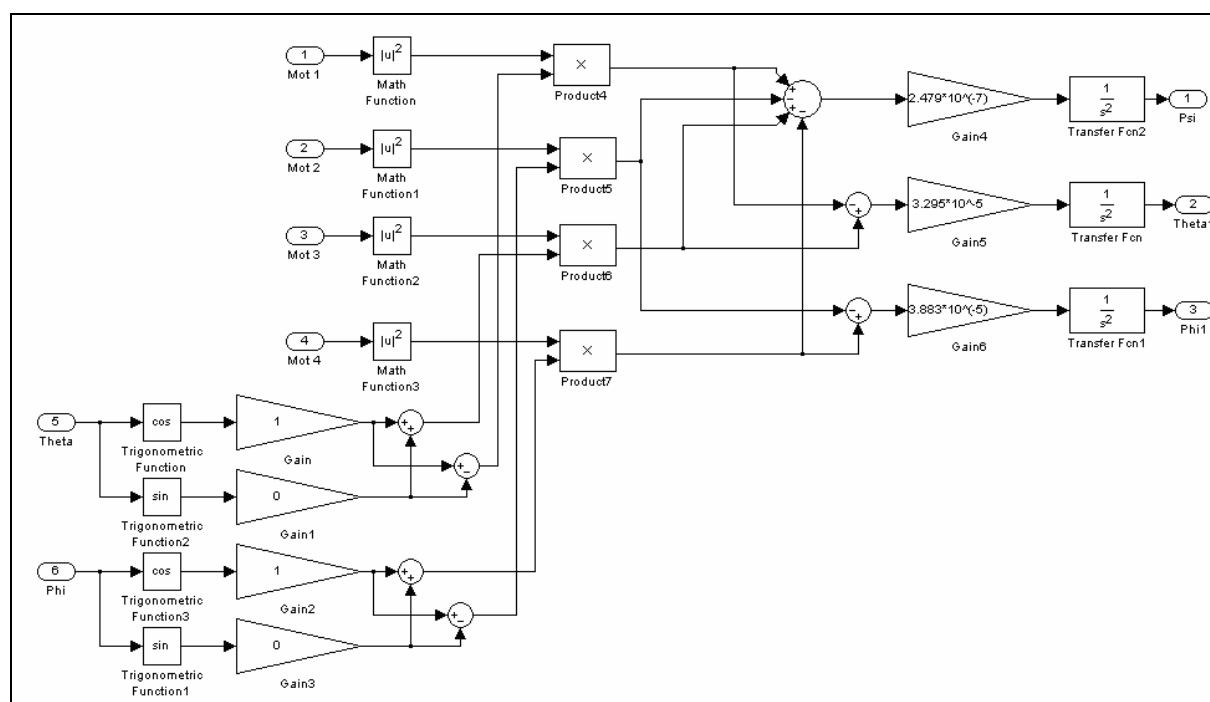


Schéma de la poussée moteur et de la trainée, incluant les effets d'angles des moteurs

10.5 Annexe : Budget

Le tableau ci-dessous présente les prix des différents composants utilisés, certains ayant été achetés cette année, d'autres étant hérités des années précédentes (comme la radiocommande ou le récepteur par exemple).

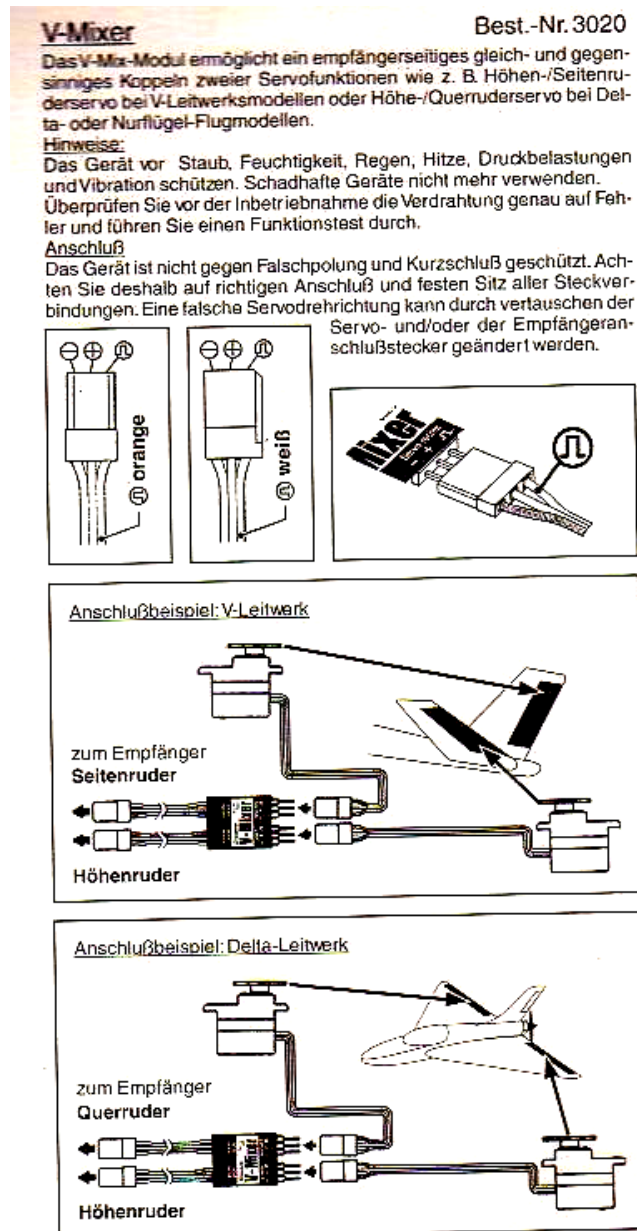
<i>Composant</i>	<i>prix</i>	<i>quantité</i>	<i>Solution mixeur/Gyro</i>	<i>Solution carte MP</i>
Radio Futaba 9CAP/9CHP	640 €	1	X	X
Récepteur Graupner R700	60 €	1	X	X
Pale 10x4,5	4\$ (3,23€)	4	X	X
Brushless typhon-micro 6-23	53 €	4	X	X
Contrôleur Wena – sinus-12-bec	62 €	4	X	X
Accu récepteur de 4,8 V et 800mAh	14 €	1	X	X
Li-Po 2000	50 €	4	X	X
Mixeur Graupner - Best.-Nr.3020	29 €	3	X	
Gyroscope Piezo Gyro G400	89 €	1	X	
Gyroscope Heading-lock MS-04	69,50 €	2	X	
Carte MicroPilot	1500 €	1		X
Total	3202 €		1702 €	2887 €

Nous avons également eu besoin de barres de carbones et de contre-plaqué pour construire la structure du quadrirotor.

En fait, nous avons dépensé cette année environ 870 € de pour notre PIP, ce qui inclus les composants sur fond jaune clair dans le tableau ci-dessus.

10.6 Annexe : documents associés aux composants achetés

Mixeur :



Contrôleur :

Félicitations, pour l'achat du nouveau contrôleur pour moteur Brushless "Flyware". Vous disposez d'un produit à la pointe de la technologie et profitant d'excellentes qualités pour valoriser vos modèles. Une attention particulière a été adoptée sur la robustesse et la facilité de programmation par l'émetteur.

Il est cependant nécessaire de disposer de quelques connaissances de base pour aboutir à une bonne utilisation du contrôleur. Grâce à cette notice, vous comprendrez assez rapidement le fonctionnement du contrôleur. Pour avancer le plus rapidement possible, nous vous conseillons de lire attentivement cette notice et surtout les conseils de sécurité, avant de mettre le contrôleur en route. Nous vous souhaitons avec ce contrôleur de la série "Sinus", beaucoup de succès et de plaisirs.

Glossaire:

	Page
1. Sécurité / Exclusion de responsabilité	2
2. Caractéristiques techniques	3
3. Divers indications	3
4. Mise en route	4
5. Programmation	4/5
6. Indication d'erreurs	7
7. Clause de garantie	7
1. Sécurité / Exclusion de responsabilité	

La mise en route et le pilotage des modèles RC, nécessite une multitude de connaissances techniques, une prudence élevée et une attitude responsable. Des négligences lors de la conception ou le pilotage du modèle, pourront avoir des répercussions graves sur le matériel ou les personnes. Travaillez toujours de façon pointue et utilisez votre modèle avec précautions.

Un moteur électrique avec son accu branché, pourra suite à un problème technique ou mécanique se mettre en route accidentellement. Cela pourra aussi se produire si la réception est active et l'émetteur est éteint. Dans ce cas il y a risque de blessure grave, par une hélice ou une pale qui s'est mise à tourner accidentellement. Toute pièce susceptible de tourner par propulsion électrique représente un danger. Evitez de vous placer à un endroit dangereux, près d'une hélice. Veillez aussi, à ce qu'aucun objet ne puisse être en contact avec l'hélice ou les pales en rotation !

Protégez le contrôleur contre la poussière, la saleté, l'humidité et les pressions mécaniques. Evitez de le placer dans la chaleur excessive ou le froid (respectez les indications techniques) ou les vibrations anormales. Pensez à contrôler de temps en temps le bon fonctionnement du contrôleur. Lors d'utilisation d'accus et de chargeur d'autres fabricants, respectez les indications d'utilisation de ces derniers.

Les contrôleurs ne sont pas protégés contre les inversions de pôles, un branchement inversé des câbles au niveau de l'accu entraînera la destruction du contrôleur. Si le moteur tourne à l'envers, vous pourrez inverser le sens de rotation en inversant le branchement de deux des trois fils de phases du moteur (C'est égal les quels). N'inversez jamais les fils d'alimentation de l'accu. Nos produits sont conçus pour une utilisation dans les modèles réduits, leur utilisation dans des modèles de grandeur réel est interdite.

Ne détachez jamais l'accu du contrôleur quand le moteur est en route, dans ce cas il y a risque de surtension (Effet générateur), ce qui pourrait détruire le contrôleur. Si vous n'utilisez pas le modèle, détachez l'accu du contrôleur. Sur les contrôleurs BEC (Sans coupleur Opto invariable), le risque d'interférence est plus élevé.

Effectuez dans tous les cas avant tout vol, des tests de portée de votre système de réception. Validez à utiliser des câbles le plus court possible entre le contrôleur et l'accu. Des câbles trop long pourraient entraîner la destruction du contrôleur. Vérifiez quotidiennement les bornes liaisons du contrôleur vers l'accu, du moteur et du récepteur.

En tant que fabricant, nous ne disposons d'aucun contrôle sur la bonne utilisation du modèle par le client final. Nous ne pouvons que prévenir des risques éventuels encourus lors d'une mauvaise utilisation du matériel. De ce fait nous déclinons toutes responsabilités. Nous n'assurons ni les coûts des dégâts matériels, ni les risques de blessure causés par une mauvaise utilisation de notre matériel.

FlyWARE



Notice d'utilisation

Contrôleur Brushless

Série - Sinus NC/Lipo

WEIMA Elektronik – Wendelhofstraße 6 – 78120 Furtwangen
www.flyware.de

WEIMA

<p>Mode Accu 4 - 4 éléments Lipo (uniquement sur Sinus 18/25A)</p> <ul style="list-style-type: none">Fonction de contrôle spécifique pour mode LipoIndication de défaut lors de sur/sous tensionArrêt progressif Automatique si la tension est inférieure à 3V/élémentSurveillance du chargement des accus, pendant le branchement de la batterie. (Indications d'erreur sur accu déchargé) < 10%			
<p>4. <u>Mise en route</u></p> <p>Lorsque vous branchez l'accu au contrôleur, vous allez entendre un premier son court vous indiquant la reconnaissance de la tension (Accu). Deux secondes après, un deuxième signal sonore vous indiquera le mode de fonctionnement et après le mode d'accu choisit. Vous entendrez un ou deux sons se suivant rapidement pour le mode de fonctionnement et après deux secondes; un, deux, trois ou quatre sons se suivant rapidement vous indiquant que vous êtes dans le mode d'accu. Le nombre des sons, dépendra du mode programmé. Les sons seront émis par le moteur !</p> <p>Veillez noter, que les émissions sonores se feront uniquement après la reconnaissance d'un signal d'émission correct, avec la manette des gaz sur position arrêt "0". Si le contrôleur ne reçoit pas de signal d'émission, 10 secondes après avoir branché la tension électrique (Accu), une indication de défaut sera émise.</p> <p>Lors de l'installation dans le modèle, veillez que les câbles et branchements ne soient pas tendus mécaniquement. Veillez qu'une bonne circulation d'air soit établi autour du contrôleur, cela évitera que la protection thermique ne se mette en route. Eloignez le contrôleur le plus loin possible par rapport au récepteur.</p>			
<p>5. <u>Programmation</u></p> <p>Les contrôleurs sortent d'usine, préprogrammé en mode de fonctionnement 1 et en mode d'accu 1, fonctionnant ainsi directement avec la plus part des émetteurs du marché où les courses des servos sont réglées à 100%. Si vous désirez modifier un mode ou reconnaître une nouvelle configuration des "Gaz", il faudra lancer une nouvelle programmation. Définissez auparavant les modes souhaités, puis suivez les indications suivantes.</p> <p>Allumez l'émetteur Montez la manette des gaz à 100% Branchez l'accu de propulsion</p> <p>Patientez jusqu'à ce qu'une suite de 3 sons soient audibles. Dans un délai de 3 secondes: Placez la manette des gaz à l'arrêt "0" (Stop)</p>			
<p>Programmation du mode de fonctionnement</p>			
<p>Mode de fonctionnement 1</p> <p>Manette sur mi-Gaz Attendez jusqu'à ce qu'une suite de deux sons suite de deux sons soient audibles Baissez la manette des gaz à "0" Manette sur plein Gaz Attendez qu'une série de 3 sons soient audibles Placez la manette sur "0" gaz</p>		<p>Mode de fonctionnement 2</p> <p>Manette sur mi-Gaz Attendez jusqu'à ce qu'une suite de deux sons soient audibles Baissez la manette des gaz à "0" Manette sur mi-Gaz Attendez qu'une série de 2 sons soient audibles Placez la manette sur "0" - arrêt Mettez la manette sur plein Gaz Attendez qu'une série de 3 sons soient audibles Placez la manette sur "0" - arrêt</p>	

7. Garantie

Les contrôleurs Flyware sont garanties pour une durée de 24 Mois. La facture d'achat faisant office de bon de garantie. Une éventuelle réparation ne prorogera pas la durée de garantie.

Si pendant la période de garantie, une panne ou un dysfonctionnement dû à la fabrication du produit est constaté, tous les frais de réparation seront à notre charge. Les pannes ou destruction dû à un mauvais emploi du produit, sont exclues de la garantie.

Expédiez le produit à l'adresse indiquée à la fin de cette notice. Le produit sera expédié à vos frais et les frais de port pour le renvoi après la réparation sous garantie seront à notre charge.

Des envois en port dû, seront refusés. L'assurance pour le produit renvoyé en SAV sera à la charge du client, nous déclinons toute responsabilité, même en cas de perte du colis.

Pour que la réparation sous garantie puisse être acceptée, veuillez respecter ces quelques points :

- Rajoutez dans votre colis la facture d'achat.
- Le produit aura été utilisé conforme à la notice d'utilisation.
- Le contrôleur n'aura pas été ouvert ou réparé par un autre service que le nôtre.
- Les dommages dû à l'humidité, à un objet étranger, à une inversion de pôle, une surtension ou une pression mécanique, seront exclus de la garantie.
- Joignez un papier, en y indiquant le problème ou le dysfonctionnement constaté.

Attestation de conformité

La société WEMA Elektronik - Wendelhostr. 6 - D-78120 FURTWANGEN, déclare que tous les contrôleurs Brushless de la gamme Sinus, Easy wave, Wave et Power wave, respectent les normes EN55014 de 1993 et EN55104 de 1995 sur la compatibilité électro-magnétique, selon la directive européenne 89/336 EWG.

Distributeur exclusif pour la France et la Belgique :

ASE Distribution
ZA - 7, rue des Paliens
F-67720 HOERDT
Email: ASE@tele2.fr

Programmation du mode d'accu

Mode 1	Mode 2	Mode 3	Mode 4
Manette demi-gaz Attente suite 2 sons Manette sur "0" gaz	Manette demi-gaz Attente suite 2 sons Manette sur "0" gaz	Manette demi-gaz Attente suite 2 sons Manette sur "0" gaz	Manette demi-gaz Attente suite 2 sons Manette sur "0" gaz
Manette sur plein gaz Attente suite 3 sons Manette sur "0" gaz	Manette demi-gaz Attente suite 2 sons Manette sur "0" gaz	Manette demi-gaz Attente suite 2 sons Manette sur "0" gaz	Manette demi-gaz Attente suite 2 sons Manette sur "0" gaz
	Manette sur plein gaz Attente suite 3 sons Manette sur "0" gaz	Manette sur demi-gaz Attente suite 2 sons Manette sur "0" gaz	Manette sur demi-gaz Attente suite 2 sons Manette sur "0" gaz
		Manette sur plein gaz Attente suite 3 sons Manette sur "0" gaz	Manette sur demi-gaz Attente suite 2 sons Manette sur "0" gaz
			Manette plein gaz Attente suite 3 sons Manette sur "0" gaz

Fin de programmation

Après la fin de la programmation, un rappel des signaux va suivre vous indiquant le programme mémorisé. Vérifiez si c'est le bon.

Attention :

Après le rappel de programmation, les modes sont mémorisés et le moteur sera prêt à démarrer. Respecter les indications de sécurité !

6. Indications d'erreurs

Il est impossible de faire démarrer le moteur, après la programmation du contrôleur. Le moteur émet un signal d'erreur. Cela peut avoir plusieurs causes ;

- Contrôlez le signal d'émission
 - Contrôlez la tension électrique
 - Programmez le bon mode d'accu.
- Sur les éléments Lipo, une sécurité extrême est nécessaire, c'est pour cela que le contrôleur effectuera un auto contrôle de la tension, dès le branchement de l'accu. Si le programme du mode d'accu n'est pas le bon, il n'y aura pas de démarrage et le contrôleur signalera un message d'erreur.

Moteur Brushless :

High-end Technology RC[®]

Brushless E-motors and ESC's ARF Models DEM Components

Typ	Typhoon micro 6/13	Typhoon micro 6/23	Typhoon micro 6/24	Typhoon micro 6/2014	Typhoon micro 15/13	Typhoon EDF 2W	Typhoon EDF 3W
Best.-Nr.							
Voltage	7,2-14,4V	7,2-14,4V	7,2 14,4V	7,2-14,4V	7,2...12V	7,2 -11,1V	8,4-15V
KV	2450	1500	1450	1500	1200	4580	2980
No of winds	13	23	24	20	10	2	3
Curr draw max efficiency	5-8	6-10	6-12	6-12	10-15	38	30
Max power ***	140 Watt	90 Watt	85 Watt	100 Watt	200 Watt	350 Watt	350Watt
Max efficiency	80 %	80%	80%	80%	82 %	78 %	78%
Directon	R and L	R and L	R and L	R and L	R and L	R and L	R and L
No poles	10	10	10	14	10	6	6
Length	40	40	40	40	50,5	49	49
Length (without axle)	25mm	25mm	25mm	25mm	34mm	37mm	37mm
Diameter in mm	29	29	29,2	29,2	29,2	28	28
Front axle length	14	13	12	12	13	12	12
Shaft diameter	3,17	3,17	3,17	3,17	3,17	3,17	3,17
Weight aprox	43g	43g	43g	43g	73g	78g	78g
Recommended props	4.7x4.7 3S1P 5x5 3S1P 5.2x5.2 3S1P	8x3.8 3S1P 9x3.8 3S1P	8x3.8 3S1P 9x3.8 3S1P	8x3.8 3S1P 9x3.8 3S1P	11 x8 8C	4.7 x 4.7 on 7-8 cells 480 size fan on 3S Lipo	4.7 x 4.7 on 7-8 cells 480 size fan on 3S Lipo
A/C type	Pylon	3D	3D	3D	Glider	Pylon racer EDF JET	Pylon racer EDF JET
Modelweight	300-500 gram	250- 700 gram	250-700 gram	250-700 gram	700-2000 gram	500 -1200 gram	500-1200 gram
Recomended speed controller	Tsunami 10 Tsunami-18	Tsunami-10	Tsunami-10 Tsunami -18	Tsunami-10 Tsunami -18	Tsunami-18 Tsunami -30	Tsunami 35 Tsunami 45	Tsunami-30

**** If you go over these Wattage you risk overheating and damage the motor use sufficient cooling

Gyroscope Piezo Gyro G400 :


Piezo Gyro G 400

No.
8067

Attention: Nouveau principe de fonctionnement, lire impérativement le mode d'emploi avant de mettre l'appareil en service.

1. Préface

Le gyroscope piézo G 400 est un système gyroscopique révolutionnaire de précision pour le pilotage de modèles réduits d'avions, de planeurs et d'hélicoptères avec capteur piézo. Ce gyroscope piézo moderne dispose de deux modes de service différents:

Mode normal

Avec ce mode, la stabilisation intervient sur un axe (par exemple le rotor arrière), comme pour les gyroscopes communs. Lorsque le modèle est tiré de son assiette de vol par des incidences externes telles que le vent, une modification rapide du régime ou des changements de pas, le capteur piézo mesure cette valeur, la transforme en signal et la communique au servo du rotor arrière comme valeur correctrice, afin de compenser la différence apparue;

Mode tenue de cap

Tenue de cap signifie tenue de cap. Dans ce mode, le gyroscope piézo G 400 travaille initialement comme un gyroscope normal, en plus toutefois, il „repère“ son assiette actuelle. Lorsque le modèle est tiré de son assiette de vol au niveau de l'axe dont se charge le gyroscope, par des incidences externes telles que le vent, etc., l'électronique totalise ces écarts et ramène le modèle à son assiette initiale. Cet instruction de retour à l'assiette initiale est exécutée si rapidement qu'elle n'est pratiquement pas assimilée par le pilote. Le modèle semble „cloué“ à sa position, qu'il se trouve de travers par rapport au vent, face à lui ou avec vent arrière, le gyroscope piézo G 400 s'efforce, quelles que soient les circonstances de conserver la position.

À noter

Pour atteindre sa puissance maximale, il faut que le gyroscope piézo G 400 soit associé à un servo extrêmement rapide pour le rotor arrière. Il est naturellement possible de l'exploiter également avec des servos à vitesse de positionnement normale. Toutefois les effets de stabilisation ne seront alors pas aussi rapides qu'avec un servo super rapide.

Servos recommandés

Alimenté par 4,8 volts (4 éléments Cd-Ni, Ni-MH):
servo S 9253 réf. F 1278

Alimenté par 6 volts (5 éléments Cd-Ni, Ni-MH):
servo S 9254 réf. F 1266

Servos pour petits hélicoptères électriques:

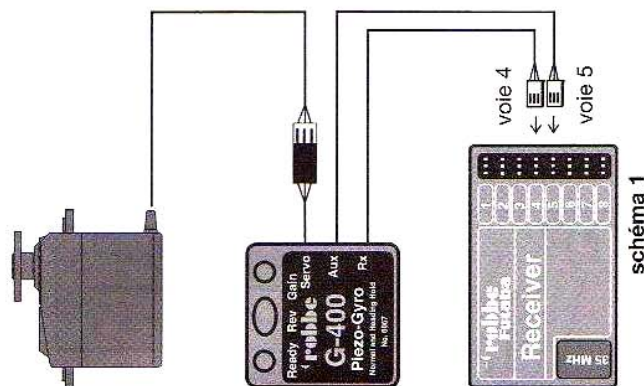
S 3101, réf. F 1258 ou FS 500 Micro BB, réf. 8460.



Piezo Gyro G 400

No. 8067

Einbau- und Bedienungsanleitung
Installation and operating instructions
Notice de montage et d'utilisation



2. Notice abrégée pour les pilotes très pressés avant la première mise en service

- réaliser le raccordement selon les indications du schéma 1
- mettre l'émetteur en marche
- mettre le récepteur en marche
- ne pas déplacer** le modèle ni le gyroscope piézo G 400 pendant approx. **10 secondes**, le gyroscope piézo G 400 réalise un calibrage à zéro
- calibrage achevé** -> le servo du rotor arrière se déplace en position initiale et la LED rouge "Ready" du gyroscope piézo G 400 s'allume
- amener le curseur de la voie 5 en fin de course
- amener le manche du rotor arrière dans une de ses fins de course et le relâcher
- lorsque le servo revient à sa position initiale -> **mode normal**
- lorsque le servo demeure dans sa position -> **mode tenue de cap**
- le curseur 5 fait office de „commutateur“ entre les deux modes et simultanément de dispositif de réglage de la sensibilité.

en position médiane:
en fin de course I. direction:
en fin de course II. direction:

0%

0% - 100% mode „normal“

0% - 100% mode „tenue de cap“

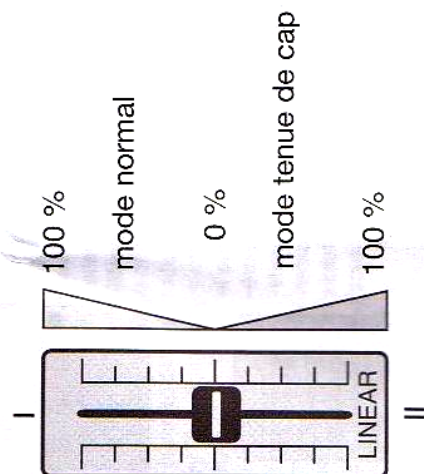
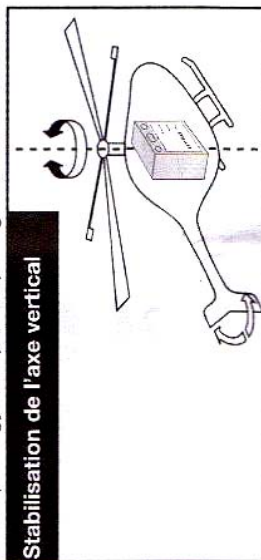


schéma 2

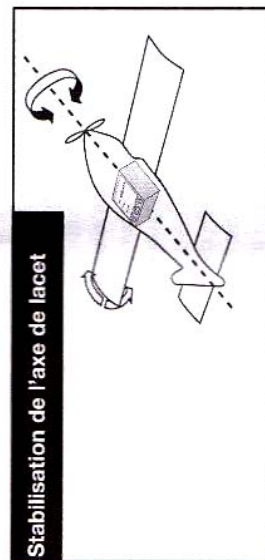
3. Mise en place

Important: tous les systèmes gyroscopiques à capteur piézo sont très sensibles aux vibrations. Aucun gyroscope piézo ne donnera de bons résultats dans un modèle dont les charges vibratoires sont élevées. Si nécessaire, tester également son implantation avec d'autres types de mousse plastique. De bons résultats ont également été obtenus avec le double face de 3,2 mm d'épaisseur de la Sté 3M (robbe réf. 5014). Nous vous prions donc instamment d'accorder une attention très particulière à ce point.

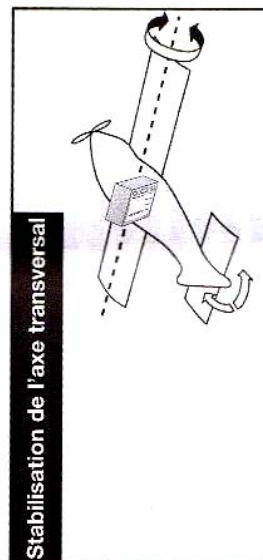
Coller un morceau de mousse plastique autocollante fournie avec l'appareil sur le fond du carter du gyroscope. Disposer ensuite le carter à un endroit à **faibles vibrations** dans le modèle (dans le modèle par ex. dans la coupole sous le servo de pas). Installer le gyroscope de sorte que l'axe qu'il est susceptible de stabiliser le traverse verticalement (cf. schéma 3). La position d'implantation n'est pas critique, qu'elle soit proche de l'arbre du rotor ou de la superstructure avant qui porte les servos, cela n'a guère d'importance, par contre il est essentiel que le gyroscope soit protégé des vibrations.



Stabilisation de l'axe vertical



Stabilisation de l'axe de lacet



Stabilisation de l'axe transversal

schéma 3

4. Branchement et mise en service

A noter: dès que la voie „Aux” est raccordée, le variateur „Gain” est hors fonction.

4.1. Branchement du gyroscope piézo G 400 en „ mode normal” avec réglage de la sensibilité à partir de l'émetteur

Stabilisation du rotor arrière comme pour les autres systèmes gyroscopiques piézo actuels. Ce mode est adapté au vol normal et pour l'entraînement des figures de voltige. Contrairement au mode tenue de cap décrit ultérieurement, la flèche du modèle se tournera systématiquement dans le vent sous l'effet de girouette.

Important

- le gyroscope piézo G 400 détecte et corrige les moindres écarts de telle sorte qu'il n'exige pas de mixage de soutien pas -> rotor arrière (Revo-Mix), nous recommandons de couper ce mixage.
- pour piloter des pirouettes rapides, avec les systèmes gyroscopiques communs leur sensibilité était discriminée (mixage Gyro-Sens). Ceci n'est plus nécessaire avec le gyroscope piézo G 400, car avec lui, le signal du capteur est simplement surmodulé par le manche. L'importance de cette surmodulation est déterminée par le réglage de la réduction de la course du servo (ATV). Voilà pourquoi il faut absolument couper le mixage de discrimination (Gyro-Sens).
- il faut également couper toutes les autres fonctions de mixage qui présentent un effet sur le rotor arrière (par ex. mixage plateau cyclique -> rotor arrière)

Comme indiqué sur le schéma 1, installer le gyroscope piézo G 400 entre le servo de rotor arrière et le récepteur et raccorder le connecteur „AUX” à une voie à curseur libre (voie 5).

- mettre l'émetteur en marche
- mettre le récepteur en marche
- ne pas bouger** le modèle ni le gyroscope piézo G 400 pendant approx. **10 secondes**, le gyroscope piézo G 400 effectue son étalonnage à zéro
- calibrage achevé** -> le servo de rotor arrière se déplace en position initiale et la LED rouge du gyroscope piézo G 400 s'allume
- si le modèle a malgré tout bougé pour une raison ou pour une autre ou lorsque la LED clignote, il faut refaire une initialisation en coupant l'alimentation du récepteur puis en la rétablissant
- amener le curseur de la voie 5 en fin de course
- amener le manche du rotor arrière dans une de ses fins de course et le relâcher
- lorsque le servo revient à sa position initiale -> **mode normal**
- si le servo reste en place, cela signifie que le gyroscope se trouve en mode tenue de cap, amener le curseur 5 en fin de course dans la position opposée
- disposer le trim du rotor arrière en position médiane
- disposer le manche du servo mécaniquement de manière à ce qu'il présente un angle de 90° pratiquement par rapport à la tringle
- accrocher la tringle au mécanisme du rotor arrière dans le trou le plus à l'extérieur
- disposer le manche du rotor arrière en position de débattement maximum
- accrocher la tringle au guignol du servo de manière à obtenir la course la plus longue sans butée mécanique contre le rotor arrière
- contrôle du sens de rotation des servos: déplacer le manche de commande du rotor arrière vers la **droite** l'arête avant des pales du rotor arrière va vers la **gauche**? Si ce

n'est pas le cas, inverser la course du servo au niveau de l'émetteur

- contrôle de la direction de l'effet du gyroscope (commutateur REV) : déplacer la flèche vers la **droite**, l'arête avant des pales du rotor arrière va vers la **gauche**? Si ce n'est pas le cas, commuter le commutateur „REV” sur le gyroscope piézo G 400. **À noter** : la commutation ne devient efficace qu'après avoir coupé puis rétabli le gyroscope
- Il est dès lors possible avec le curseur 5, à partir de l'émetteur, de régler la sensibilité progressivement de 0 - 100 %.

À noter: contrairement à ce qui se passait pour les systèmes gyroscopiques communs jusqu'à présent, avec le gyroscope piézo G 400 la course du servo est aussi réduite avec la sensibilité du gyroscope, car, lorsque la sensibilité du gyroscope est réduite il suffit d'une plus petite course du rotor arrière bien que la capacité d'exécution des instructions reste la même. Par ailleurs, le gyroscope dispose d'une caractéristique de pilotage exponentielle. Un réglage inférieur à 30 % n'est pas approprié à la pratique et ne doit pas être programmé car la course de pilotage du rotor arrière est considérablement réduite.

Pour se familiariser avec le gyroscope lors du premier vol, nous recommandons les réglages initiaux suivants:

- réglage de la course du servo du rotor arrière (ATV) : approx. 70 %
- sensibilité du gyroscope (ATV): approx. 70 %

4.2. Branchement du gyroscope piézo G 400 en „mode tenue de cap” avec réglage de la sensibilité au niveau de l'émetteur

Comme décrit précédemment la position de la flèche est parfaitement maintenue. Ce nouveau mode offre les avantages suivants:

- le débutant a dès lors la possibilité de se concentrer exclusivement sur les autres fonctions de base de l'hélicoptère, étant donné que l'arrière conserve systématiquement la position dans laquelle il a été programmé.
- pour les exercices pratiqués avec vent de travers, il n'est plus nécessaire de rectifier systématiquement au niveau du rotor arrière. Le modèle vole sans effet girouette. C'est-à-dire que la flèche ne se tourne pas dans le vent. Pour „voler sur une trajectoire rectiligne” il faut toutefois maintenant disposer l'hélicoptère dans la position souhaitée avec de légères corrections du manche de pilotage.
- idéal pour le vol 3D et le vol dos qui prennent de plus en plus d'importance, il n'est pas nécessaire de rectifier au niveau du rotor arrière. La direction sélectionnée est préservée.

Important:

- le gyroscope piézo G 400 détecte et corrige les moindres écarts de telle sorte qu'il n'exige pas de mixage de soutien pas -> rotor arrière (**Revo-Mix**), nous recommandons de couper ce mixage.
- pour piloter des pirouettes rapides, avec les systèmes gyroscopiques communs leur sensibilité était discriminée (mixage Gyro-Sens). Ceci n'est plus nécessaire avec le gyroscope piézo G 400, car avec lui, le signal du capteur est simplement surmodulé par le manche. L'importance de cette surmodulation est déterminée par le réglage de la réduction de la course du servo (ATV). Voilà pourquoi il faut absolument couper le mixage de discrimination (Gyro-Sens).
- il faut également couper toutes les autres fonctions de mixage qui présentent un effet sur le rotor arrière (par ex. mixage plateau cyclique -> rotor arrière).

5. Généralités

5.1 Condition préalables imposées au modèles

Étant donné la haute sensibilité du gyroscope et la force et la rapidité de réaction du servo de rotor arrière et donc la réaction proportionnelle du rotor arrière, les efforts imposés à la flèche sont nettement supérieurs aux charges habituelles. Ceci exige un entraînement du rotor arrière insensible aux efforts de torsion et un mécanisme de rotor arrière particulièrement solide. Une transmission rigide est de ce fait recommandée pour absorber les réactions de transfert de charge très importantes. Par ailleurs, la tringle de liaison doit être absolument absente de jeu et particulièrement souple.

5.2 Adaptation thermique

Bien que le système gyroscopique ait été pourvu d'un dispositif de compensation de la température bénéficiant des recherches techniques les plus récentes, une adaptation s'impose en présence de nuances de température entre la température du gyroscope et la température ambiante. Voilà pourquoi il est recommandé de laisser le gyroscope sous tension 5 minutes environ afin qu'il s'adapte à la température ambiante. Le couper ensuite puis le remettre sous tension.

5.3 Autres remarques:

- sur les ensembles de radiocommande dont le réglage ATV est fonction de la phase de vol, par exemple l'ensemble FC 28, il est possible, pour chaque phase de vol, de définir une course autonome du servo du rotor arrière (ATV) et une sensibilité particulière également qu'il est possible de requérir automatiquement et simultanément à l'aide des commutateurs de phase de vol „Idle-Up 1“, „Idle-Up 2“.
- en fonction de la combinaison rotor principal/rotor arrière, il peut se produire qu'en mode normal la flèche oscille légèrement. Cette réaction n'est pas obligatoirement causée par une sensibilité trop élevée, mais peut également être provoquée par des pales de rotor arrière trop grandes ou dont la corde est trop importante. Dans ce cas, raccourcir les pales du rotor arrière de 3 à 5 mm.
- la configuration des voies rotor arrière et „Aux“, de même que la position du commutateur Rev (inversion de la course de servo) sont lues lorsque le gyroscope est mis en marche puis sauvegardées. Si l'ordre d'affectation des voies „Aux“ ou „Rx“ est modifié sur le récepteur ou si la fonction d'inversion est actionnée, il faut couper puis remettre le gyroscope sous tension afin que la configuration soit saisie puis mise en mémoire.

6. Caractéristiques techniques:

tension de service: 3 à 8 volts fournis par l'accu du récepteur
consommation: approx. 25 mA (à 4,8V)
encombrement: approx. 41 x 41 x 20 mm
poids: approx. 30 g

7. Contenu du kit

1 x gyroscope piézo G 400
2 x 2 morceaux de mousse plastique autocollants d'amortissement
1 x mode d'emploi

Sous réserve de modification technique

Pour la mise en service procéder comme indiqué sous 4.1.

- activer le mode „tenue de cap“ (voie à curseur 5 en butée)
 - attention au rotor arrière, il se déplace dans un sens ou dans l'autre en fonction de l'impulsion neutre de l'ensemble de radiocommande
 - compenser au niveau du trim jusqu'à ce que le rotor arrière ou le servo s'immobilise (Un conseil : si nécessaire, réduire électroniquement le trim (par ex. TRIM RATE 30 - 40 %) pour permettre un ajustement plus précis
 - activer le mode „normal“
 - amener le modèle en vol stationnaire, le nez dans le vent
 - lorsque le modèle se déplace autour de son axe vertical, en vol stationnaire, atterrir et corriger au niveau de la tringle du rotor arrière - **pas au niveau du trim**
 - si nécessaire, reprendre plusieurs fois cette procédure
 - lorsque le modèle cesse de se dégager de son axe vertical, l'ajustement électronique et l'ajustement mécanique du modèle sont parfaitement adaptés.
- Maintenant, il est possible d'effectuer un vol circulaire en mode „normal“ ou en mode „tenue de cap“. Tous les autres réglages comme la sensibilité maximale du gyroscope et le réglage de la course du servo du rotor arrière (ATV) pour la vitesse de rotation doivent être déterminés par l'expérience accumulée en volant !

- Il est dès lors possible avec le curseur 5, à partir de l'émetteur, de régler la sensibilité progressivement de 0 - 100 %.

À noter: Contrairement à ce qui se passait pour les systèmes gyroscopiques communs jusqu'à présent, avec le gyroscope piézo G 400 la course du servo est aussi réduite avec la sensibilité du gyroscope. Étant donné que le gyroscope dispose d'une caractéristique de pilotage exponentielle, la réduction de la course est à peine repérable entre approx. 30 à 100 %. Un réglage inférieur à 30 % n'est pas approprié à la pratique et ne doit pas être programmé car la course de pilotage du rotor arrière est considérablement réduite.

4.3. Raccordement du gyroscope piézo G 400 en „mode normal“ avec réglage manuel de la sensibilité sur le gyroscope.

Sur les ensembles de radiocommande disposant de peu de voies, il ne reste souvent pas de voie disponible pour le réglage de la sensibilité du gyroscope. Dans ce cas, ne pas raccorder le connecteur „Aux“ du gyroscope. Il est alors possible de régler sur le gyroscope la sensibilité progressivement de 0 - 100 %, la sensibilité augmentant si on tourne dans le sens des aiguilles d'une montre et diminuant dans le sens opposé.

À noter: Contrairement à ce qui se passait pour les systèmes gyroscopiques communs jusqu'à présent, avec le gyroscope piézo G 400 la course du servo est aussi réduite avec la sensibilité du gyroscope, car, lorsque la sensibilité du gyroscope est réduite il suffit d'une plus petite course du rotor arrière bien que la capacité d'exécution des instructions reste la même. Par ailleurs, le gyroscope dispose d'une caractéristique de pilotage exponentielle. Un réglage inférieur à 30 % n'est pas approprié à la pratique et ne doit pas être programmé car la course de pilotage du rotor arrière est considérablement réduite.

Comme indiqué sur le schéma 1, installer le gyroscope piézo G 400 entre le servo du rotor arrière et le récepteur. Dans cette variante, le connecteur de raccordement „Aux“ reste non affecté. Lorsque la voie complémentaire ne peut être raccordée par l'ensemble de radiocommande, le gyroscope travaille alors automatiquement en mode standard. Dans ce cas, il **n'est pas** possible d'activer le mode tenue de cap.

11. Bibliographie

- [1] **Tarek Hamel - Philippe Souères**
Modélisation, estimation et contrôle des drones à voilures tournantes : Un aperçu des projets de recherche français
UNSA-CNRS, LAAS-CNRS, Toulouse, France
- [2] Site contenant de nombreuses informations de fabrication concernant le drone quadrirotor
<http://perso.wanadoo.fr/grzflyer/>
- [3] **Nicolas Albert**
Certification du code embarqué d'un micro-drone
Laboratoire d'Optimisation Globale SDER/ENAC
Février à Juillet 2005
- [4] **André Noth**
Synthèse et Implémentation d'un contrôleur pour micro hélicoptère 4 rotors
Ecole Polytechnique Fédérale de Lausanne, Février 2004.
- [5] **X. Dufresne**
Support de cours de Mécanique Générale Version 3.1
ENSICA, Toulouse, 2003
- [6] Forum de discussions consacré en grande partie aux drones quadrirotor
<http://forum.xbird.org/>
- [7] **R. Lozan - P. Castillo - S. Salazar - D.Lara**
Stabilisation de véhicules aériens à décollage vertical : Théorie et application
Heudiasyc, UMR CNRS
- [8] MicroPilot MP2028g Installation and Operation
MicroPilot MP2028g Installing and Using Analog to Digital Converters, Working with Radio Modems, Electronic Compass User Guide.
- [9] *Régulateur PID numérique v1.2*
Laboratoire de régulation numérique – 28 mars 2005
- [10] **Pedro Castillo Garcia**
Rapport d'activité de recherche : Commande non linéaire d'objets volants miniatures
(page 12)
Laboratoire HeuDiaSyc UMR CNRS - 2005
- [11] **V. Pommier-Budinger**
Support de cours d'électronique analogique
ENSICA, Toulouse, 2004 / 2005
- [12] **Andréas Goërmer**
Flight dynamics of a mini UAV,
Rapport de stage ENSICA