

الجمهورية الجزائرية الشعبية
وزارة التعليم العالي والبحث العلمي
REPUBLIQUE ALGERIENNE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
جامعة سعد دحلب البليدة
UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

**MEMOIRE DE
PROJET DE FIN D'ETUDES**
POUR L'OBTENTION DU DIPLÔME DE
MASTER EN ELECTRONIQUE
Option : traitement de l'information et système électronique

**Réalisation et Stabilisation d'un Quadricopter
Application des Techniques Floues**

Présenté par :

Mr :Larfi Abdelmoumen.

Mr: Ferguene Oussama

Proposé par :

Mr.: Hadj Sadok M'hamed

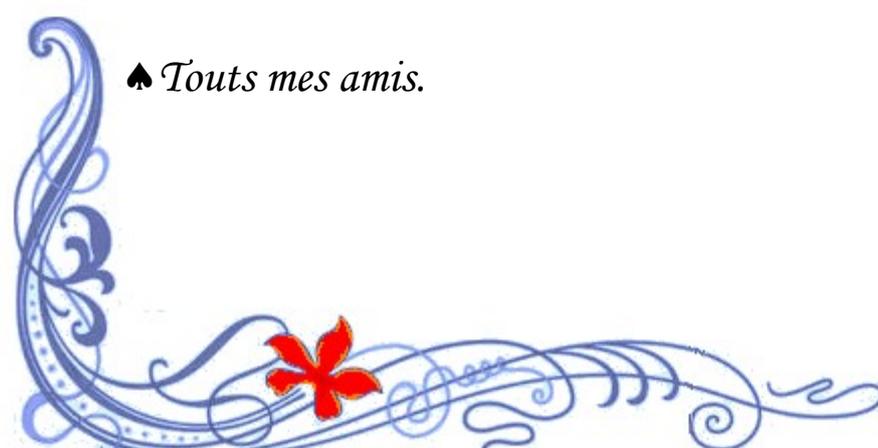
Promotion: 2011



DEDICACES

J'offre ce modeste travail avec une grande Fierté à :

- ♠ *Ma chère et aimable maman qui m'a soutenue tout le long de mes études.*
- ♠ *Mes frères et sœurs*
- ♠ *Tous les professeurs au niveau de département d'électronique*
- ♠ *Tout Les membres de la famille*
- ♠ *Mon binôme*
- ♠ *A toute la promotion TISE 2011*
- ♠ *Touts mes amis.*



Abdelmoumen



DEDICACES

J'offre ce modeste travail avec une grande Fierté à :

♠ *Mes chers et aimables parents qui m'ont soutenus tout le long de mes études.*

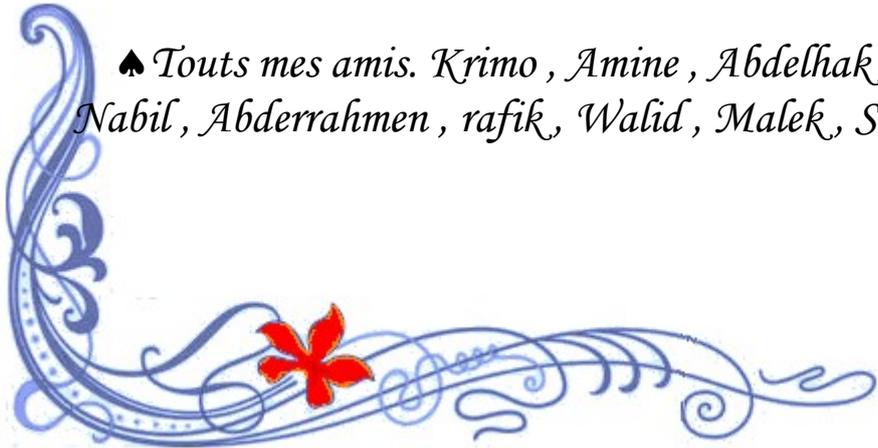
♠ *Mes frères et sœurs*

♠ *Tous les professeurs au niveau de département d'électroniques*

♠ *Tout Les membres de la famille*

♠ *Mon binôme*

♠ *A toute la promotion TISE 2011*



♠ *Tous mes amis. Krimo , Amine , Abdelhak , Toufik , Taki , Nabil , Abderrahmen , rafik , Walid , Malek , Sidali , Hamza ,*

Oussama

بسم الله والحمد لله و الصلاة و السلام على رسول الله
و بعد.

فعن أبي هريرة -رضي الله تعالى عنه- قال: قال رسول الله - صلى الله عليه وسلم-: (لا يشكر الله من لا يشكر الناس). [أخرجه أحمد وأبو داود والترمذي بإسناد صحيح]

إن كلمات قصيرة من المدح تترك أثراً عميقاً في الإنسان وقد تدفع للمزيد.

الشكر والتقدير والتحية لإخواننا من الأقارب و الجيران و الطلبة و المعلمين و كل من ساهم و أعان في هذا المشروع من قريب أو من بعيد سواء بجلب المعدات أو بالنصائح و التوجيهات أو حتى بالدعاء.

بارك الله فيكم وجزاكم الجنة بما قدمتم للجميع ونفعنا ونفعكم ونفع الجميع.

الحمد لله الذي تتم به الصالحات و صلى الله على محمد و على أهله و صحبه و سلم تسليماً كثيراً.

Ce rapport est la synthèse du travail que nous avons réalisé en quelques mois de travail, normalement le projet a une durée préfixé de six mois mais pour un projet de ce genre il faut dire que c'est insuffisant.

Ce projet est pour nous une source de grande motivation, car il s'approche le plus de notre spécialisation techniques, Traitement de l'Information et Systèmes Electroniques.

En choisissant ce projet, nous ne savions pas vraiment à quoi nous attendre, car c'était un engin et un concept que nous n'avions jamais vu auparavant. C'est lorsque nous avons récupéré les rapports des équipes précédentes que nous avons pu réellement entrer dans le vif du sujet et cerner ce que l'on attendait de nous.

Ce rapport regroupe la synthèse des travaux antérieurs, les objections faites et les objectifs résultants, ainsi que le travail que nous avons réalisé, théorique et pratique.

Avent Propos :

Une véritable mutation des systèmes aériens
Voler est et restera l'un des plus grands plaisirs de l'homme...

Laboratoire de Signal Electrique et Télécommande

Toutefois, si celui-ci était à bord de la première machine volante, pour sa plus grande fierté, ses connaissances scientifiques et technologiques lui permettent aujourd'hui de rester au sol dans certaines circonstances et ce pour son plus grand avantage.

Il étend ainsi de façon considérable le champ d'utilisation des aéronefs qui, prenant le nom de "drones", semblent être légitimement appelés à une carrière prometteuse.

Ce nouveau palier aéronautique franchi par l'homme est le fruit des plus récents progrès accomplis dans des domaines clés, tels que l'informatique, la robotique, l'optronique, l'imagerie radar, la transmission de données, etc.

Les drones occupent ainsi à juste titre une place de plus en plus importante dans les milieux aéronautiques et de la défense. Et l'on assiste à une montée en puissance des expérimentations dans le monde entier, encore plus ou moins opérationnelles.



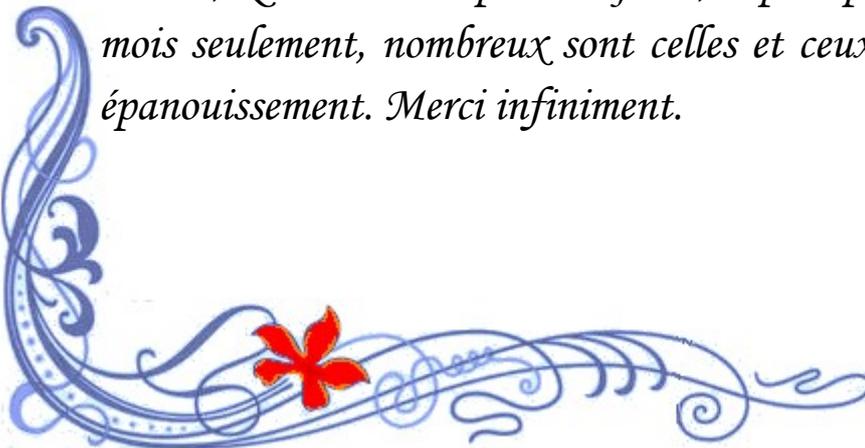
REMERCIEMENTS

Nous tenons à remercier en premier lieu le bon Dieu de nous avoir donné le Courage, la patience et la capacité de mener ce travail à terme.

Nous exprimons nos vifs remerciements à notre promoteur monsieur Hadj Sadok M'hamed de nous avoir encadré malgré la charge du travail et à exprimer notre profonde gratitude pour avoir accepté de tenir ce sujet avec nous.

Au membre de jury pour l'honneur qui nous ont accordé en acceptant de juger notre travail.

Enfin on voudrait exprimer toute notre gratitude à l'ensemble des personnes, qui ont contribué de prêt ou de loin à l'aboutissement de ce travail, Que ce soit depuis toujours, depuis plusieurs années ou quelque mois seulement, nombreux sont celles et ceux qui ont participé à notre épanouissement. Merci infiniment.



Oussama et Abdelmoumen

Résumé

Un quadrirotor est un véhicule volant constitué de quatre hélices, chacune située au bout des branches d'une croix. Grâce à ce type de structure, l'engin peut se déplacer dans toutes les directions par translation.

De nos jours, les véhicules autonomes tendent de plus en plus à remplacer l'être humain lorsque les conditions sont trop hostiles pour que celui-ci puisse y être en sécurité. C'est le cas des drones, qui sont utilisés à des fins militaires ou de reconnaissance dans des endroits sinistrés par exemple. Un drone est un véhicule volant n'embarquant aucun pilote humain à son bord. On en trouve sous plusieurs formes, allant de l'avion à l'hélicoptère, en passant par des drones en forme d'insectes battant des ailes.

Le but du projet était la réalisation d'un drone quadrirotor se stabilisant de manière autonome en vol.

Pour ce faire, le véhicule est équipé d'une centrale inertielle munie d'accéléromètres et de gyromètres. Ils sont utilisés pour déterminer les angles de tangage, roulis et lacet du véhicule ainsi que sa position dans l'espace. Ensuite, un calculateur estime les vitesses de rotation à affecter à chaque moteur pour corriger l'assiette du véhicule.

En tant que ce travail nous a permis d'étendre les principes théoriques introduits en cours de RNSF (réseaux de neurone et système flou) présenté par Mr. Hadj Sadok, afin d'adapter régulateurs flous multi variable : multi entrées multi sorties (MEMS ou MIMO) à notre système Quadri rotor.

Ce projet nous a également permis de découvrir ou d'approfondir nos connaissances concernant le monde de l'aéronautique et des drones en particulier.

Abstract

A quadrirotor is a flying vehicle made up of four propellers, each one located at the end of the branches of a cross. Thanks to this type of structure, the machine can move in all the directions by translation.

Nowadays, the autonomous vehicles tend more and more to replace the human being when the conditions are too hostile so that can be there in safety. It is the case of the drones, which are used with fine soldiers or of recognition in places disaster victims for example. A drone is a flying vehicle not taking on board any pilot human on its board. One finds some in several forms, energy of the plane to the helicopter, while passing by drones in the shape of insects beating of the wings.

The goal of the project was the realization of a being stabilized drone quadrirotor in an autonomous way in vol.

With this intention, the vehicle is equipped with an inertial power station provided with accelerometers and gyrometers. They are used to determine the angles of pitching, rolling and lace of the vehicle like its position in space. Then, one calculator estimates the number of revolutions to be assigned to each engine to correct the plate of the vehicle.

As this works allowed us to extend the theoretical principles introduced in the course of RNSF (networks of neuron and vague system) presented by Mr. Hadj Sadok, in order to adapt regulating blurs multi variable: multi entries multi left (MEMS or MIMO) to our system Quadri rotor.

This project also has allowed us to discover or look further into our knowledge concerning the world of aeronautics and the drones in particular.

ملخص

quadrotor هو عجلة السيارة التي تتكون من أربعة مراوح، كل يقع نصائح من فروع الصليب. مع هذا الهيكل، لا يمكن للآلة تحرك في كل الاتجاهات عن طريق الترجمة.

اليوم، والمركبات مستقلة تميل بشكل متزايد ليحل محل إنسان عندما تكون الظروف معادية جدا لتكون آمنة. هذا هو الحال في الطائرات بدون طيار، والتي تستخدم للاستطلاع عسكرية أو في مجالات مثل الكوارث. طائرة بدون طيار هو تشريع مركبة تحلق أي سائق الإنسان على متنها. وجدوا في أشكال كثيرة، بدءا من الطائرة المروحية، إلى طائرة بدون طيار مثل الحشرات الخفقات الأجنحة.

وكان هدف المشروع في تحقيق طائرة بدون طيار هو quadrotor استقرار مستقل في الرحلة.

للقيام بذلك، تم تجهيز الطائرة بالقصور الذاتي مع أجهزة التسارع وجيروسكوبات. يتم استخدامها لتحديد زوايا لفة، وياو الملعب من الطائرة ومكانتها في الفضاء. ثم حاسبة التقديرات إلى أن تسند إلى كل سرعات المحركات تصحيح موقف الطائرة.

لأن هذا العمل قد سمح لنا لتوسيع النظرية وعرض خلال RNSF (الشبكات العصبية ونظام غامض) قدم بواسطة السيد الصادق الحاج، للتكيف مع المتغيرات غامض تحكم متعددة : تعدد المدخلات والمخرجات المتعددة (MIMO أو ممس) لنظامنا رباعية الدوار.

هذا المشروع يسمح لنا أيضا أن تكتشف أو تعميق لدينا المعرفة حول عالم الطيران والطائرات بدون طيار في معين.

Sommaire

I. Etat de l'art:	1
1.1 Histoire :	1
1.2 Définition :	2
1.2.1 Classification :	3
❖ Par leur Autonomie au vol :	3
❖ Par leur forme :	4
❖ Par leur sustentation :	5
❖ Par leur motorisation :	5
1-3. Configuration de notre drone :	5
1-4. Prototype de Quadri rotor :	6
2. Principe de vol	7
2-1. Principe de vol d'une quadri rotor :	7
2-1.1 Sens de rotation et mouvements possibles :	7
❖ Monté – descente	8
❖ Commande de lacet	9
❖ Commande de roulis	9
❖ Commande de tangage	10
3. Modélisation.....	11
3.1 Model du Quadri rotor :	11
3.2 Identification des paramètres du système.....	12
❖ Calcul du moment d'inertie du rotor.....	13
❖ Calcul du coefficient de poussée b.....	13
❖ Calcul du coefficient de traînée d :	14
3.3 Calcul des moments d'inerties du drone :	15
3.3.1 Moment d'inertie I_z :	15
3.3.2 Moment d'inertie I_x et I_y :	16
3.4 Model Simulink :	17
4. Choix du correcteur :	19
4.1 Contrôleur Flou :	21
4.2 Structure d'un Contrôleur Flou.....	21
4.2.1 Fuzzification :	22

4.2.2 Base De Connaissances :.....	22
4.2.3 Bloc de décision :.....	23
4.2.4 Defuzzification :.....	23
4.3 La conception d'un contrôleur flou :.....	24
4.4 Application au Drone :.....	25
❖ La stratégie de contrôle :.....	27
Résultats.....	30
Discussion.....	30
5. Structure du projet :.....	31
5. 1. Choix de matériels :.....	32
5. 1.1 Structure électronique.....	32
5. 1.1.1 Calculateur :.....	32
❖ Platine Arduino "Mega 2560".....	32
5. 1.1.2 Capteur :.....	33
❖ Ultimate IMU.....	33
❖ Télémètres Ultrason.....	33
5. 1.2 Structure mécanique :.....	34
❖ Éléments utilisés :.....	34
6 . Programmation Embarquée.....	38
6 .1Calculateur Inertiel.....	38
6 .2 Calculateur Principale.....	38
6 .3 Algorithme du régulateur flou :.....	39
6 .3.1 Implémentation de l'algorithme flou sur la carte Arduino :.....	40
6 .4 Application au Drone	44
6 .5 Base de règle.....	46
Discussion.....	49
Difficultés du Projet.....	50
Conclusion.....	51

I. Etat de l'art:

1.1 Histoire :

En 1849 des bombardements sur Venise eurent lieu, à partir du *Vulcano*, au moyen de ballons sans pilote munis de bombes à retardement. 70 ans après, soit en 1916, aux États-Unis, les véhicules aériens commencés avec Kettering Bug.

Il était un bombardier biplan sans pilote en bois, il a été contrôlé par un gyroscope, son altitude mesurée par un baromètre anéroïde.



Fig1.1 Kettering Bug

Un certain nombre d'améliorations ont suivi. Pendant et après la Première Guerre mondiale, des avions sans pilote radiocommandés ont vu le jour.

Les avions sans pilote V1 et le missile V2 allemands de la seconde Guerre mondiale (1939-1945) étaient-ils des drones ? Plutôt des bombes volantes, mais le V1 à aile droite et pulsoréacteur dorsal a servi de modèle à des avions-cibles. Voir figure 1.2

Les lourdes pertes subies pendant la seconde guerre mondiale par les aviations d'observation de chacun des antagonistes ont suscité l'idée d'un engin d'observation militaire sans équipage (ni pilote, ni observateur).



Fig.1.2 Missile de croisière V1

Le drone a été développé de façon confidentielle par les États-Unis d'Amérique comme un moyen de supériorité stratégique et de rupture capacitaire devant permettre la surveillance et l'intervention militaire chez l'ennemi sans encourir les risques humains que l'opinion ne supportait pas. Cette supériorité a été acquise au travers de l'innovation technologique, surtout dans les domaines de **l'automatique** et des **transmissions**.



Fig.1.3 Predator tirant un missile hellfire

Les drones ont été ensuite dans tous les **conflits** et **opérations de maintien de la paix**. Il a notamment été utilisé au **Kosovo** ou au **Tchad**, lors des **attaques aériennes américaines au Pakistan** ou contre la **piraterie** maritime, par les Américains qui l'ont introduit en 2009.

1.2 Définition :

Un **drone** (« faux bourdon » en anglais), ou UAV (Unmanned Aerial Vehicle) est un **aéronef** sans pilote humain à bord,

La désignation de drone est très limitative puisqu'elle ne recouvre qu'un véhicule aérien. Le drone n'est en fait qu'un des éléments d'un système, conçu et déployé pour assurer

une ou plusieurs missions. C'est la raison pour laquelle les spécialistes parlent de «*systèmes de drones*».

1.2.1 Classification :

On distingue toutefois plusieurs manières pour classer les drones:

❖ Par leur Autonomie au vol :

- Ceux qui requièrent effectivement l'assistance d'un pilote au sol,



Fig1.4 Pilotage d'un Predator depuis une station au sol

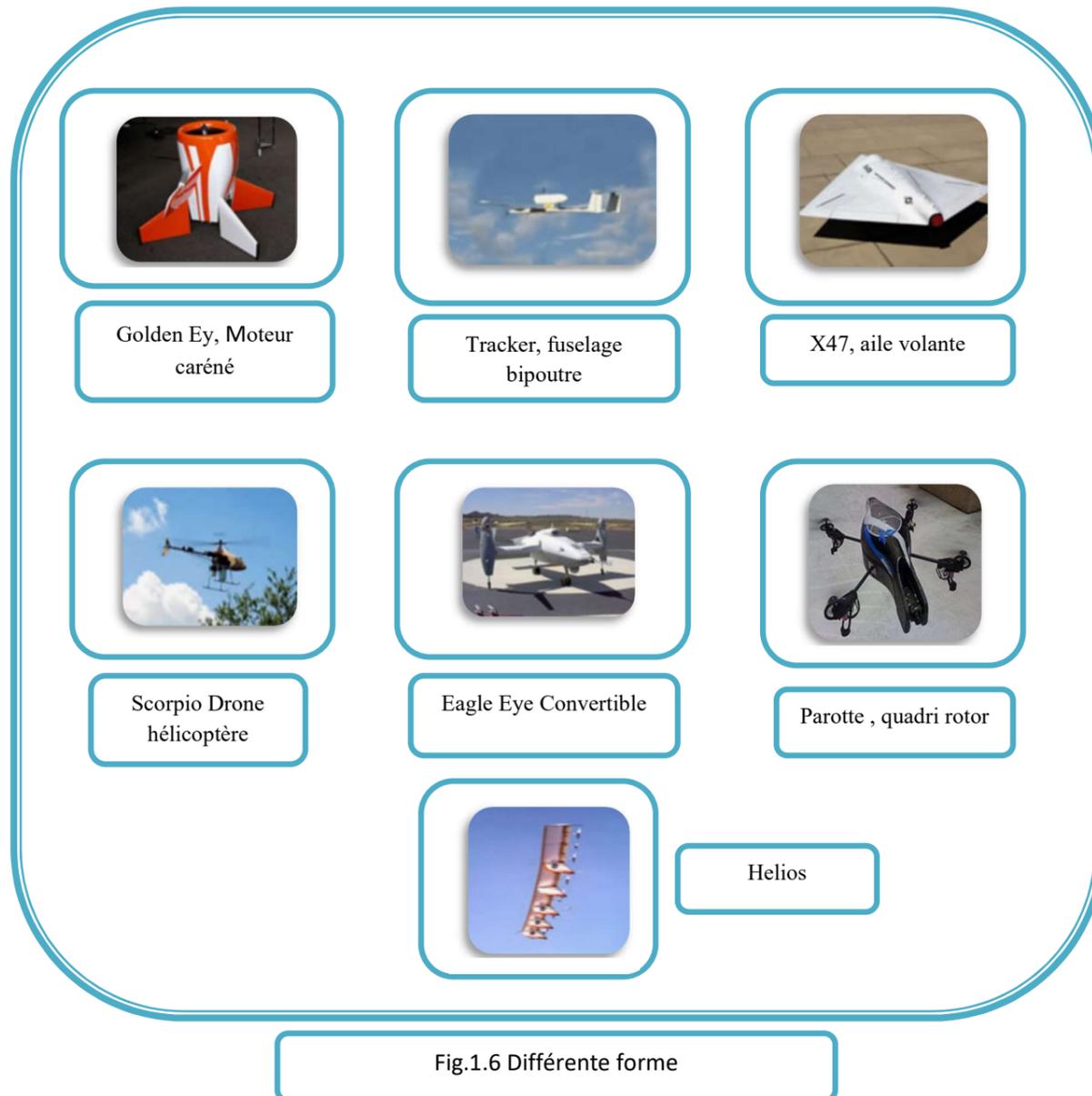
- Ceux qui sont entièrement **autonomes**. Cette autonomie de pilotage peut s'étendre à la prise de décision opérationnelle pour réagir face à tout événement aléatoire en cours de mission



Fig1.5 Prowler à l'atterrissage

❖ **Par leur forme :**

Il existe de nombreuses configurations de drones, très différentes les unes des autres – pratiquement une pour chaque machine – et dont certaines sont très novatrices. En fait, la forme d'un drone sera déterminée par la nature et le profil de sa mission, ainsi que par sa charge utile : à chaque demande correspond pratiquement une solution spécifique.



❖ **Par leur sustentation :**

- Sustentation assurée par une voilure fixe(ou simplement **l'aile fixe**) : comme sur les avions
- Sustentation assurée par une voilure tournante : comme sur l'hélicoptère
- Sustentation assurée par des ailes battantes : comme celles des oiseaux ou de certains insectes
- Il existe également des drones convertibles : qui associent, grâce à leurs rotors basculants, les capacités de vol vertical de l'hélicoptère à celles des grandes vitesses de l'avion.

❖ **Par leur motorisation :**

On retrouve sur les drones toute la palette des motorisations possibles pour les avions

- moteurs à pistons, avec ou sans turbocompresseur, turbines à hélices, turboréacteurs
- moteurs électriques

Pour conclure C'est la mission du drone et sa charge utile qui vont déterminer sa configuration en matière d'autonomie, forme, sustentation et de motorisation.

1 - 3. Configuration de notre drone :

Pour des missions à caractère civile on peut citer :

- l'exploration de tombaux et des sites en ruine (Archéologie)
- l'exploration de station nucléaire, tunnel contaminé ou immeuble (Pompier)
- Première observation avant toute intervention humaine (Police, Gendarmerie).

Autonomie au vol	Forme	Sustentation	Motorisation
Autonome	Quadri rotor	Voilure tournante	Electrique

1- 4. Prototype de Quadri rotor :

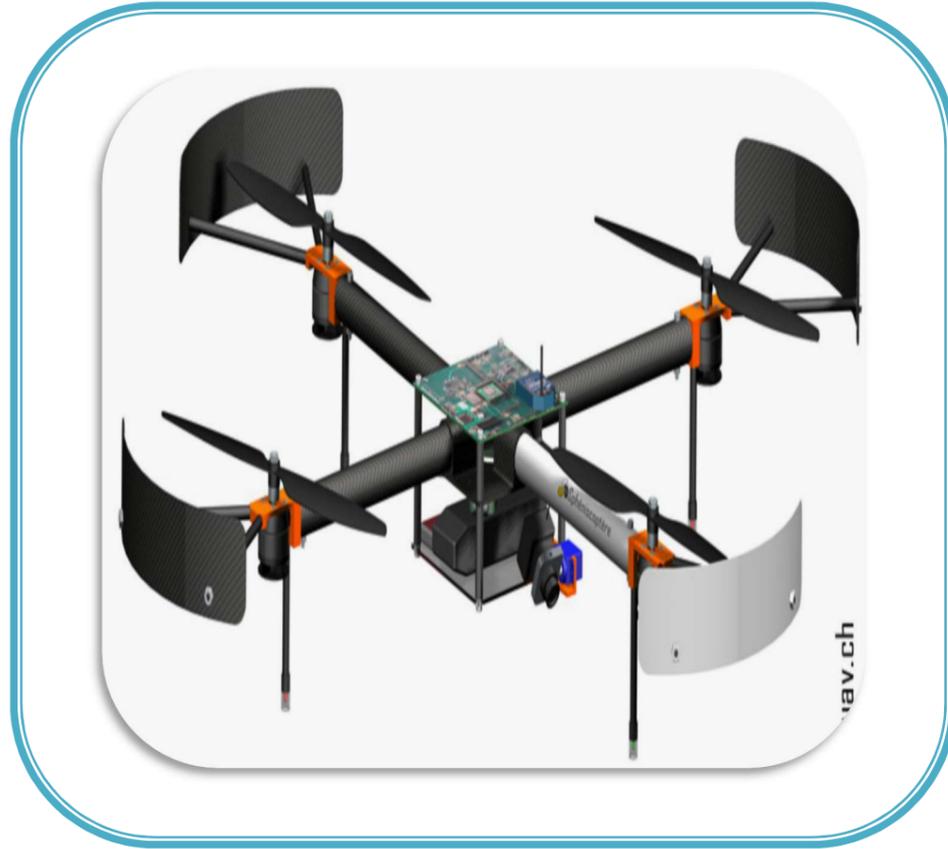


Fig.1.7 Prototype de Quadri rotor

2. Principe de vol

2- 1. Principe de vol d'une quadri rotor :

2- 1. 1Sens de rotation et mouvements possibles :

Un quadri rotor 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 la quadri rotor dans le sens inverse de sa rotation.

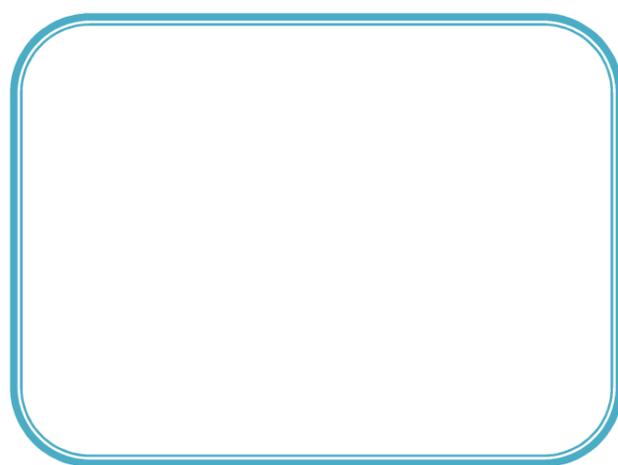


Fig.2.1 Moment de lact

De plus, pour faciliter la gestion des commandes de tangage et de roulis, on choisit l'avant de la quadri rotor 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.

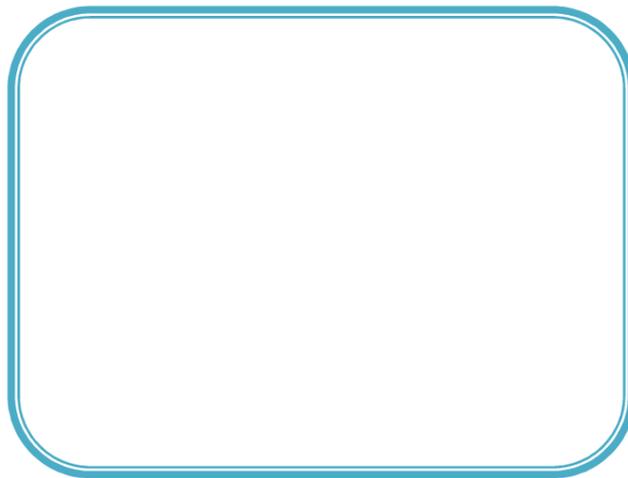


Fig.2.2 Sens de rotation des moteurs

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

❖ Monté - descente

L'altitude doit être stable et cela est facilement réalisé en jouant sur la vitesse de rotation de l'ensemble des moteurs. On diminue la vitesse des quatre moteurs pour faire descendre le quadri-rotor tandis qu'on l'augmente pour le faire monter.

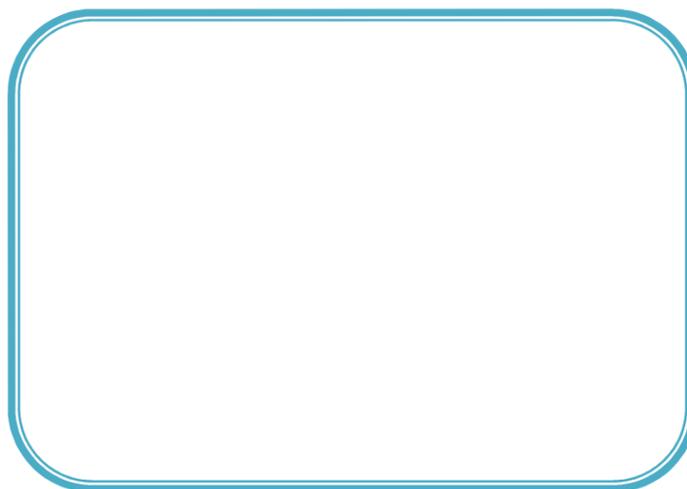


Fig.2.3 ascension et descente

❖ Commande de lacet

Axe de lacet. On réalise cela en agissant sur les moteurs par paires : on augmente la vitesse de rotation de deux moteurs face a face (avant et arrière) et on diminue la vitesse des deux autres (gauche et droite).



Fig.2.4 Commande de lacet

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

❖ Commande de roulis

Axe de roulis : en agissant sur les moteurs gauche et droit, on peut maintenir l'horizontalité. Par exemple, si le quadri-rotor penche a gauche, on augmente la vitesse du moteur gauche tout en diminuant le droit.

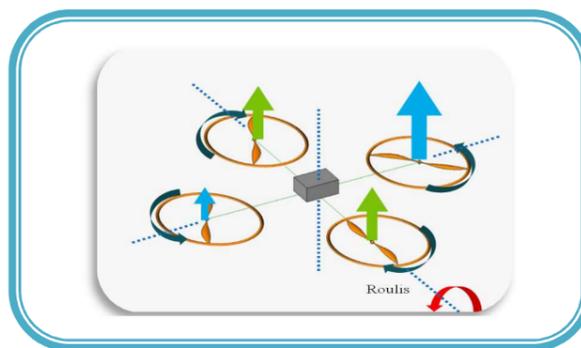


Fig.2.5 Commande de roulis

❖ **Commande de tangage**

Axe de tangage : les moteurs avant et arrière permettent de garder une assiette stable sur cet axe. Si le quadri-rotor est orienté vers l'arrière, il faut donner plus de vitesse au moteur arrière et baisser celle du moteur avant.

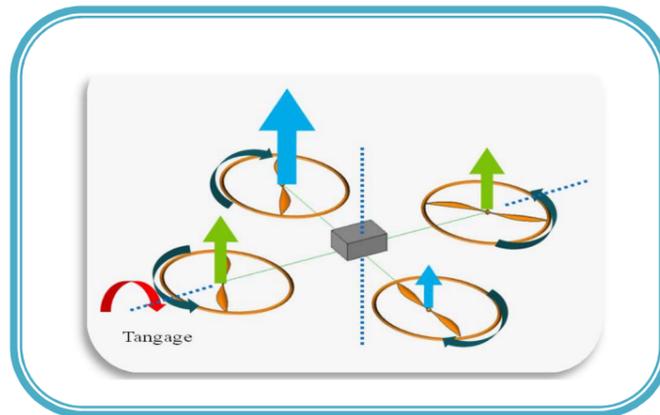


Fig.2.6 Commande de tangage

3. Modélisation

Un modèle mathématique est une approximation simple du monde réel, décrivant l'influence d'une entrée donnée sur ce système.

3.1 Model du Quadri rotor :

Ce modèle mathématique dynamique est basé sur les équations de la mécanique représentant les forces appliquées au drone.

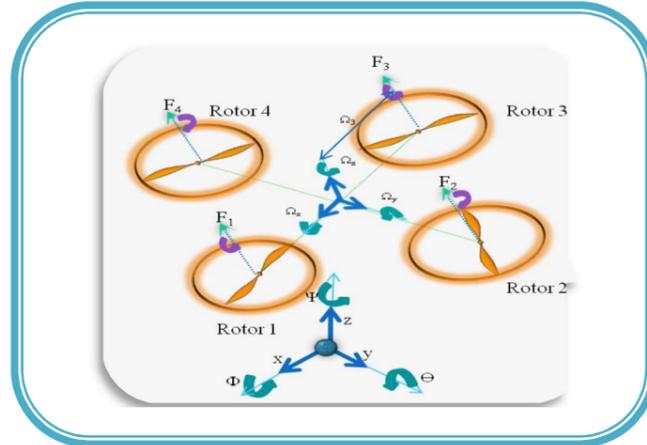


fig. 3 .1 les forces appliquées au drone.

Qui peuvent être décomposées en différents types d'accélération :

$$\ddot{\phi} = \dot{\theta} \psi \frac{(Iy - Iz)}{Ix} + bl \frac{(\Omega_4^2 - \Omega_2^2)}{Ix} + \dot{\theta} I_{rotor} \frac{(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)}{Ix}$$

3.1

Equation de l'accélération angulaire suivant l'axe de roulis

$$\ddot{\theta} = \dot{\psi} \dot{\phi} \frac{(Iz - Ix)}{Iy} + bl \frac{(\Omega_3^2 - \Omega_1^2)}{Iy} + \dot{\phi} I_{rotor} \frac{(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)}{Iy}$$

3.2

Equation de l'accélération angulaire suivant l'axe de tangage

$$\ddot{\psi} = \dot{\phi} \dot{\theta} \frac{(Ix - Iy)}{Iz} + d \frac{(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)}{Iz} \quad 3.3$$

Equation de l'accélération angulaire suivant l'axe de lacet

$$\ddot{x} = \frac{(\cos\Psi \sin\theta \cos\Phi + \sin\Psi \sin\Phi)}{m} \sum_{i=1}^4 T_i \quad 3.4$$

Equation de l'accélération suivant l'axe x

$$\ddot{y} = \frac{(\sin\Psi \sin\theta \cos\Phi + \sin\Psi \sin\Phi)}{m} \sum_{i=1}^4 T_i \quad 3.5$$

Equation de l'accélération suivant l'axe y

$$\ddot{z} = \frac{(\cos\theta + \cos\Phi)}{m} \sum_{i=1}^4 T_i \quad 3.6$$

Equation de l'accélération suivant l'axe z

Avec $T_i = b\Omega^2$; $i \in \{1, 2, 3, 4\}$

3.2 Identification des paramètres du système

A fin de calculer les paramètres, il faut faire une série d'expériences physiques. On place le rotor (moteur et hélice) relié à un tachymètre sur un axe et le tout sur un capteur de force. Et on relève les mesures de façon à avoir une courbe $F = f(\Omega)$ voire la figure III.2.

Pour cela on a pris les résultats du site où on a acheté le système de propulsion

Vitesse (tr/min)	Poussée (gram)
3150	145
4000	290
4950	430
5850	615

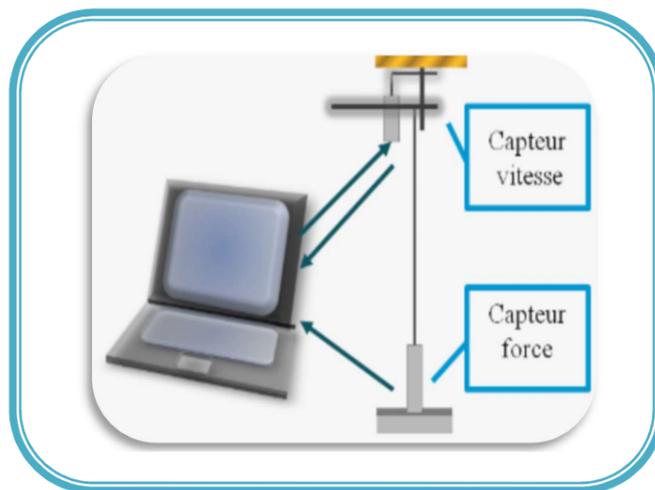


Fig.3 .2 mesure vitesse et force résultante

❖ Calcul du moment d'inertie du rotor

Pour I rotor, le moment d'inertie de chaque moteur, il faut trouver les caractéristiques de couple et de résistance d'un moteur et d'une pale afin de retrouver le coefficient. Au final, pour chaque moteur.

$$I_r = 2.85704073752 \cdot 10^{-5} \text{kg.m}^2$$

❖ 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 5850 tr/min et peuvent soulever 615 grammes chacun.

Donc

$$F_i = 6,37 \text{ N} ; \Omega_i = 613 \text{ rad/s}$$

On obtient alors

$$b = 1,69 \cdot 10^{-5} \text{kg.m.rad}^{-2}$$

❖ 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 :

$$\ddot{\psi} = \dot{\phi} \dot{\theta} \frac{(I_x - I_y)}{I_z} + d \frac{(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)}{I_z} \quad 3.8$$

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 obtient alors :

$$\ddot{\psi} = \frac{d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)}{I_z} \quad 3.9$$

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).

$$\psi = \frac{d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) t^2}{2 I_z} \quad 3.10$$

Lors de l'expérience, on met les moteurs 1 et 3 à mi-puissance, c'est-à-dire 2925 tr/min et les moteurs 2 et 4 à l'arrêt. On effectue un quart de tour.

$$\frac{\pi}{2} = \frac{d \Omega^2 T^2}{I_z} \quad 3.11$$

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

On obtient alors

$$d = 8.92.10^{-5} kg.m^2.rad^{-2}$$

3.3 Calcul des moments d'inerties du drone :

3.3.1 Moment d'inertie Iz :

Pour mesurer l'inertie z quatre fils sont utilisés pour créer une rotation au tour de l'axe Z Fig.III.3 (expérience du pendule). Cette méthode nécessite la connaissance des paramètres définis dans la table

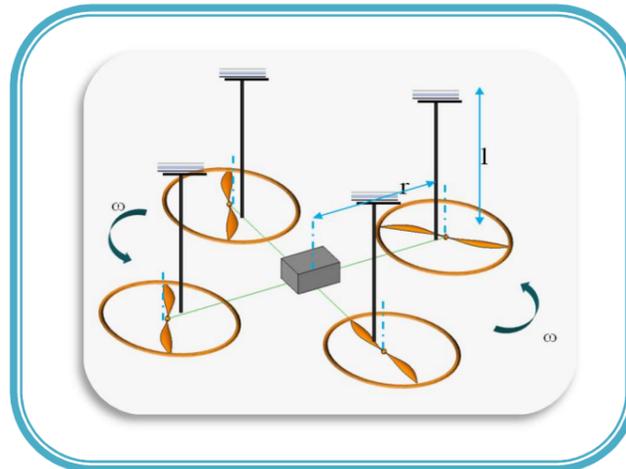


Fig.3.3 suspension du drone par quatre fils pour la mesure du moment d'inertie Iz

Symbole	Définition
I	l'inertie de l'axe de rotation
m	la masse de la plateforme
r	la distance entre le fil de suspension et le C.G.
ω	la fréquence d'oscillation (rad/s)
l	la longueur du fil de suspension

Tab3.1 Table Définition des Paramètres d'Inertie

L'inertie de rotation autour de l'axe Z est donné par :

$$I_z = \frac{mgr^2}{\omega^2 l}$$

3.3.2 Moment d'inertie I_x et I_y :

Les moments d'inertie autour des axes x et y (I_x , I_y) sont donnés de la même manière que pour I_z . Toutefois, dans cette expérience nous n'avons utilisé que deux fils et la plate-forme est suspendue verticalement.

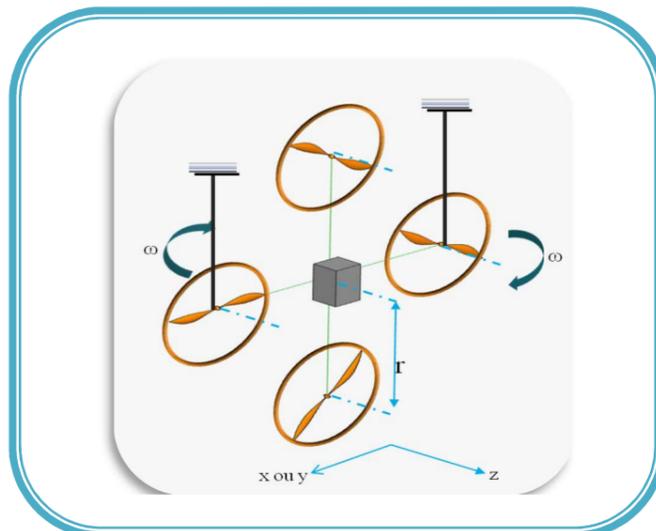


Fig.3 .4 suspension du drone par deux fils pour la mesure des moments d'inertie I_x et I_y

En effectuant les expériences on trouve :

$$I_x = 0.0622 \text{kg} \cdot \text{m}^2$$

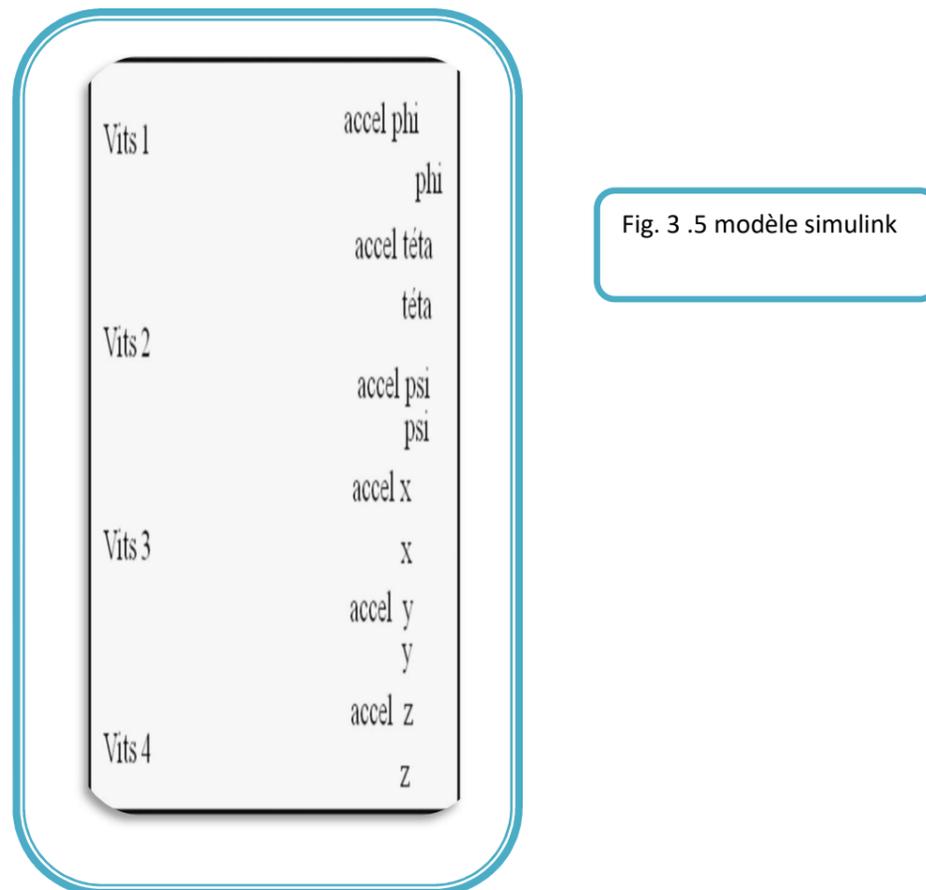
$$I_y = 0.0733 \text{kg} \cdot \text{m}^2$$

$$I_z = 0.0964 \text{kg} \cdot \text{m}^2$$

3.4 Model Simulink :

Dans cette section, nous nous intéresserons à l'introduction du modèle dans l'environnement Matlab »Simulink, Ce modèle servira à l'analyse et à la simulation du comportement du drone modélisé ainsi qu'à la synthèse du correcteur.

On définit notre modèle avec en sorties les trois angles et accélérations angulaires (ϕ , θ , ψ) ainsi que les trois positions et accélérations en x, y et z, et en entrées les quatre vitesses moteurs



L'intérieur de cette boîte de sous-système sous Matlab se présente comme suit :

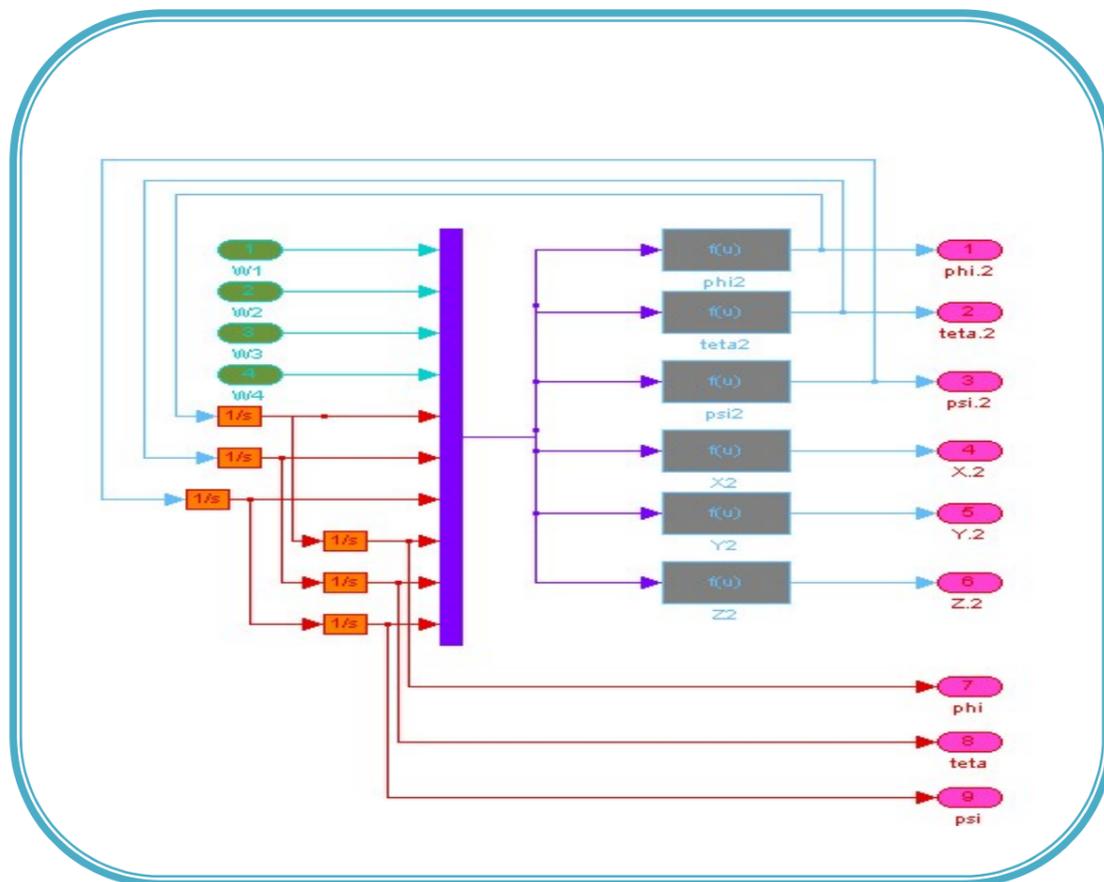


Fig. 3 .6 représentation sous simulink des équations du modèle

Cette représentation semble compliquée mais est en fait assez simple. Chaque partie du schéma constitue une des équations vues plus haut. Ces six équations sont liées entre elles par les variables φ , θ , et ψ . Elles possèdent toutes au moins un terme variant en fonction de vitesses motrices. De cette manière, si on injecte les quatre vitesses en entrée du système, les équations déduiront automatiquement les angles, accélérations angulaires, positions et accélérations qui en résultent.

4. Choix du correcteur :

La position dans l'espace d'une quadri rotor est généralement contrôlé par un opérateur par le biais d'un système radiocommande, tandis que son attitude peut être automatiquement stabilisées par l'intermédiaire d'un pilote automatique qui permet au véhicule de conserver une orientation donnée et, par conséquent, empêche le véhicule de s'écraser lorsque le opérateur effectue les manœuvres désirés.

La quadri rotor et l'attitude hautement instable que représente ce corps rigide a attiré l'attention de nombreux chercheurs au cours des dernières années. Une large classe de contrôleurs a été proposée.

La commande de tel procédés complexes se heurtes a plusieurs difficultés :

- ❖ Présence de non linéarités.
- ❖ Les incertitudes.
- ❖ La complexité au niveau de la modélisation.
- ❖ L'élaboration des lois de commande convenables.

Ce qui exige de nouvelles méthodes de contrôle plus sophistiquée ayant de grandes capacités d'apprentissage et de prise de décision dans des milieux incertains et perturbés.

L'homme a beaucoup appris à partir d'études des systèmes naturels, en utilisant ce qu'on a appris pour développer de nouveaux modèles algorithmiques pour résoudre des problèmes complexes. Ces algorithmes intelligents comprennent les réseaux neuronaux artificiels (NN), calcul évolutive (EC), l'intelligence en essaim (SI), les systèmes immunitaires artificiels (AIS), et les systèmes flous (FS).

Artificial neural networks (NN),
Evolutionary computation (EC),
swarm intelligence(SI),
artificial immune systems (AIS), and
fuzzy systems (FS)

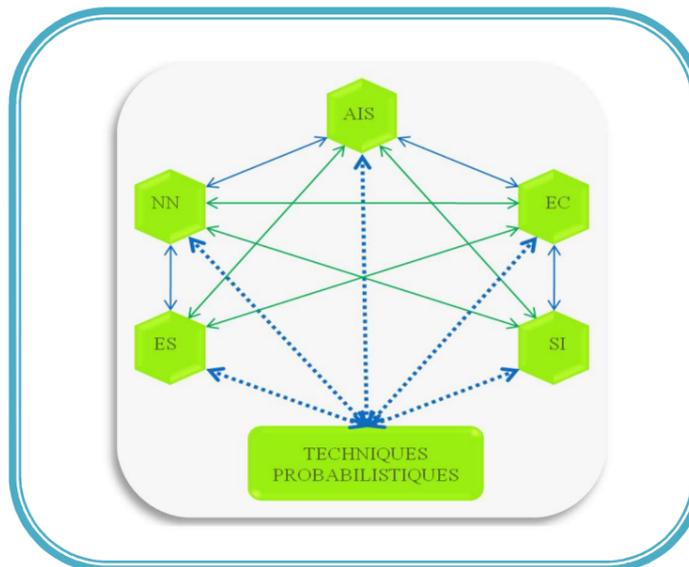


Fig.4.1 Algorithmes intelligents et interaction possibles

Les flèches indiquent que les techniques de paradigmes différents peuvent être combinées pour former systèmes hybrides.

“La raison pour laquelle le langage naturel s’exprime en termes flous n’est pas que la pensée humaine est floue, mais que le monde est flou ” John F.SOWA

La résolution des problèmes passe par la recherche d’un modèle le plus certain que possible. Vu que notre environnement est incertain il y a des cas où on n’arrive pas à définir un modèle représentant la réalité avec certitude. Les modèles dans notre cerveau peuvent être assez complexes et également vagues ou flous.

Pourquoi le flou ?

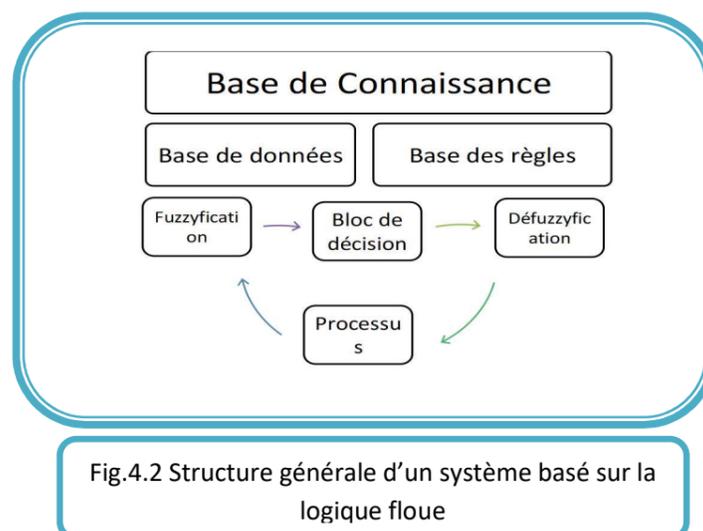
Il s’agit plutôt d’une branche mathématique rigoureuse qui offre des solutions très intéressantes pour le réglage. Ces solutions sont particulièrement intéressantes quand il s’agit des SYSTÈMES NON LINÉAIRES, DIFFICILES À MODÉLISER et le Quadrator est un !

4.1 Contrôleur Flou :

Un système flou est un système basé sur les concepts de raisonnement approximatif : variable linguistique, les propositions floues, les règles linguistiques si-alors, le modus ponens généralisé. Un contrôleur flou est un cas particulier du système flou consacré à des tâches de contrôle.

Nous allons présenter un aperçu des contrôleurs flous. Il commence par une description de la structure du contrôleur flou. Choix des variables linguistiques, les ensembles de termes, les méthodes de défuzzification, et quel type de raisonnement flou considéré.

4.2 Structure d’un Contrôleur Flou



Chaque système basé sur la logique floue est composé de 4 Blocs principaux :

- ❖ Base de Connaissance [règle et paramètre des fonctions d'appartenances]
- ❖ Bloc de décision ou le Moteur d'inférence [opération sur les règles]
- ❖ Fuzzification [transformation des entrées précises en degrés d'appartenance]
- ❖ Défuzzification [transformation les résultats flous en sorties précises]

4.2.1 Fuzzification :

Le module fuzzification effectue deux tâches.

- Normalisation de l'entrée, c'est à dire cartographie à savoir des valeurs d'entrée dans des univers de discours normalisé
- Transformation des valeurs état nettes du processus dans des ensembles flous, afin de les rendre compatibles avec les parties antérieures des règles linguistiques qui seront appliquées dans le moteur d'inférence floue

4.2.2 Base De Connaissances :

La base de connaissances consiste en une base de données et une base de règles.

- la base de données contient les facteurs de normalisation et les paramètres des ensembles flous qui déterminent le sens des valeurs linguistiques de la variable linguistique.

La conception de la base de donnée se fait par :

- la sélection du facteur d'échelle. il joue un rôle similaire à celui du gain dans un contrôleur conventionnel.
- le choix des variables linguistiques, Essentiel pour la réussite de la commande floue. habituellement, les variables linguistiques sont l'Etat, d'une erreur de l'Etat, et dérivée d'erreur de l'Etat
- la définition des ensembles des termes linguistiques. Soit le nombre de valeurs que peut prendre une variable linguistique. la taille de cet ensemble détermine la granularité de l'action de contrôle et le nombre de règles
- le choix des fonctions d'appartenance. il comprend le choix des formes, Distribution, et les paramètres fonctions d'appartenance pour chaque variable linguistique

- la base de règles représente la stratégie de contrôle sous la forme d'linguistiques si-alors. Chaque règle a deux parties, Partie antécédente (prémisse ou condition) exprimé par *Si...* description de l'état du système, *Entrée*, Partie conséquente (conclusion) exprimé par *... Alors...* l'action a exécuté par l'opérateur, *Sortie*
La conception de la base de règles se compose du choix de la source et le contenu de l'ensemble des règles. Il existe quatre méthodes, rapportées dans la littérature, pour dériver des règles linguistiques:
 - Modélisation de la connaissance de contrôle de l'ingénieur. l'approche la plus courante consiste à une verbalisation introspective de l'expertise humaine.
 - Modélisation des actions de l'opérateur humain et de son expérience. Cette approche comprend un interrogatoire d'opérateurs expérimentés en utilisant un questionnaire soigneusement organisé Modélisation floue du système Contrôlé. Le modèle flou du processus est la description linguistique de ses caractéristiques dynamique. Basé sur ce modèle, les règles contrôleur flou sont générés
 - Approche d'apprentissage. le régulateur flou est constitué de deux bases de règles:
l'une est générale et l'autre a un rôle de superviseur pour créer et de modifier la base de règles générale dans la fonction de la performance souhaitée du système
 -

4.2.3 Bloc de décision :

Ce module est de déterminer le sens de l'ensemble des règles linguistiques, c'est à dire la valeur globale de la variable de sortie de contrôle basé sur les contributions individuelles de chaque réglage les paramètres de conception du moteur d'inférence sont les suivants :

- Choix de l'opérateur d'implication qui détermine le sens d'une règle linguistique. Les plus utilisés sont l'implication de Mamdani et de Larsen
- Choix de la méthode de raisonnement flou pour fusionner plusieurs règles. Les plus utilisés sont Min-Max et la méthode de Takagi Sugeno

4.2.4 Defuzzification :

La transformation d'une information floue en une information déterminée soit la concrétisation de l'inférence floue. Il y a plusieurs méthode de défuzzyfication on citera Centre de gravité, Moyenne des maximum et Moyenne pondérée.

Voire la figure 4.3 représente les méthodes de résonnement des systèmes flous.

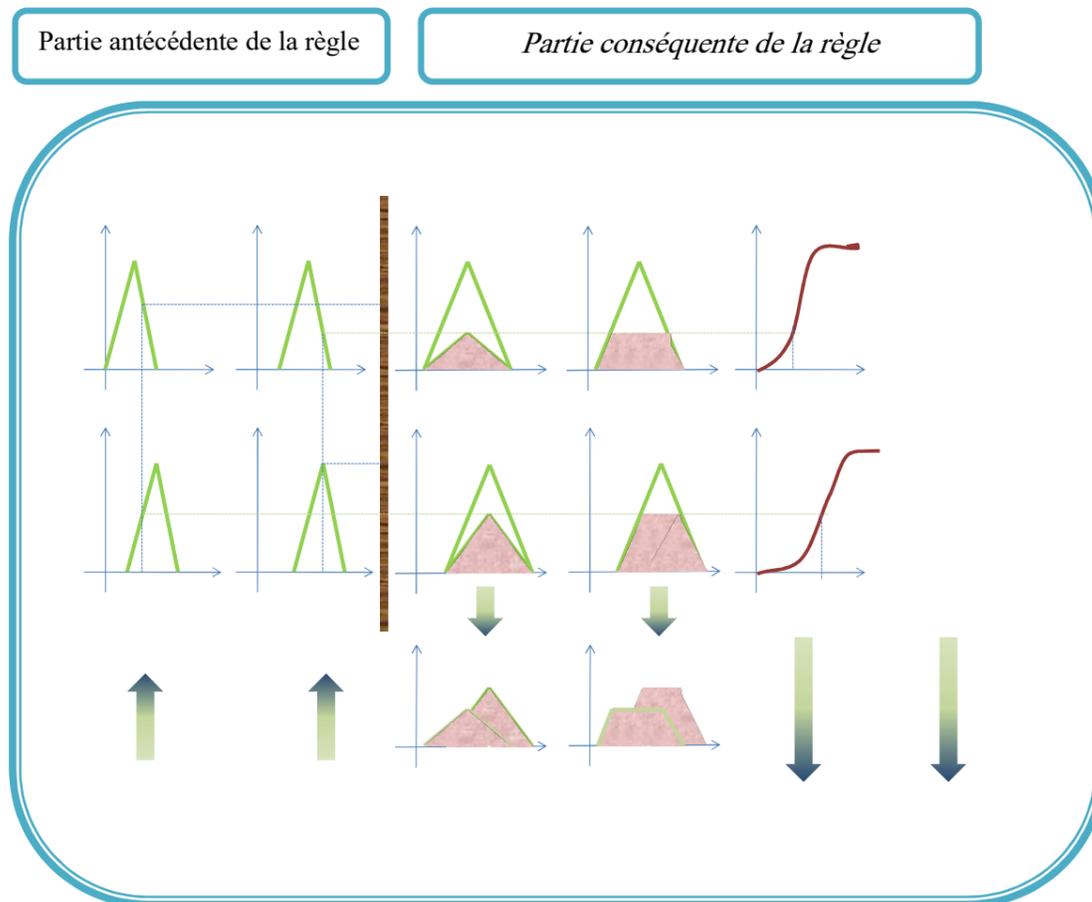


Fig.4.3 Les méthodes de raisonnement des systèmes flous

4.3 La conception d'un contrôleur flou :

La conception d'un contrôleur flou comporte les aspects suivants : un univers de discours doit être défini, ainsi que les ensembles flous des fonctions d'appartenance pour les entrées et des espaces de sortie doivent être conçus. Avec l'aide d'un expert humain, les règles linguistiques qui décrivent le comportement dynamique doivent être définies. Le moteur d'inférence, fuzzification et defuzzification doivent être mises en œuvre par le concepteur.

❖ **Contrôleur flou Mamdani** : Mamdani et Assilian ont produit le premier contrôleur flou. Il suit les étapes suivantes

1. Identifier et nommer les variables linguistiques d'entrées et définir leurs intervalles numériques.
2. Identifier et nommer les variables linguistiques de sortie et définir leurs intervalles numériques.

3. Définir un ensemble de fonctions d'appartenance floue pour chacune des variables d'entrée et les variables de sortie.
4. Construire la base de règles qui représente la stratégie de contrôle.
5. Effectuer la fuzzification des valeurs d'entrée.
6. Effectuer l'inférence pour déterminer le poids d'activation des règles activées.
7. Defuzzifier, en utilisant la méthode du centre de gravité, afin de déterminer l'action correspondante à exécuté.

4.4 Application au Drone :

Les variables linguistiques d'entrées et leurs intervalles numériques :

- ❖ Angle Tangage (θ) $(-\frac{\pi}{2} \leq \theta < \frac{\pi}{2})$ rad.
- ❖ Angle Roulis (ϕ) $(-\frac{\pi}{2} \leq \phi < \frac{\pi}{2})$ rad.
- ❖ Angle Lacet (ψ) $(-\pi \leq \psi < \pi)$ rad.
- ❖ L'altitude (Z) $(0 \leq Z < 600)$ cm

Les variables linguistiques de sorties et leurs intervalles numériques :

- ❖ Commande moteur 1 ($m1$) $(0 \leq m1 < 100)$ cm.
- ❖ Commande moteur 2 ($m2$) $(0 \leq m2 < 100)$ cm.
- ❖ Commande moteur 3 ($m3$) $(0 \leq m3 < 100)$ cm.
- ❖ Commande moteur 4 ($m4$) $(0 \leq m4 < 100)$ cm

Fonctions d'appartenance floue sur **Simulink** pour chacune des variables d'entrée et de sortie.

Fonction d'appartenance des variables d'entres : angle Tangage, angle Roulis et angle Lacet
Variable linguistique (NG), (NP), (EZ), (PP), (PG)

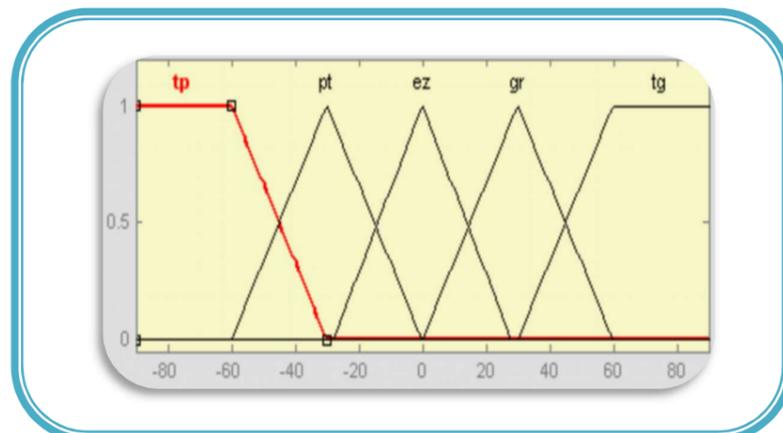


Fig.4.4 Fonction d'appartenance des variables d'entres : angle Tangage, angle Roulis et angle Lacet

Fonction d'appartenance des variables d'entre altitude Z

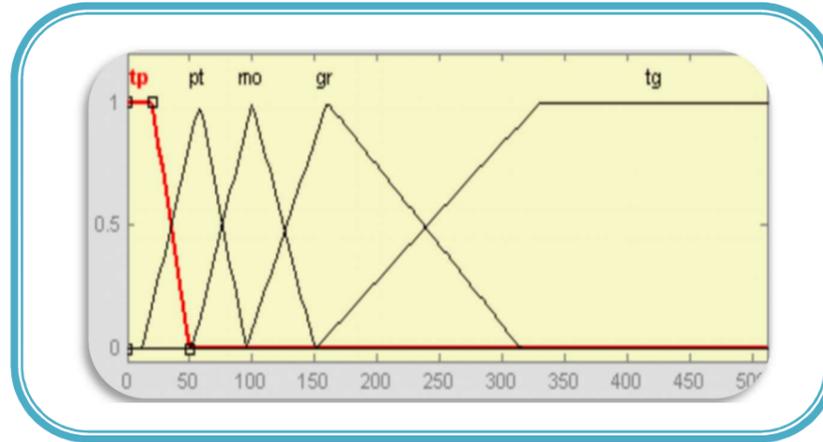


Fig4.5 Fonction d'appartenance des variables d'entre altitude Z

Fonction d'appartenance des variables de sorties m1, m2, m3, m4

Variable linguistique (TP), (P), (M), (G), (TG).

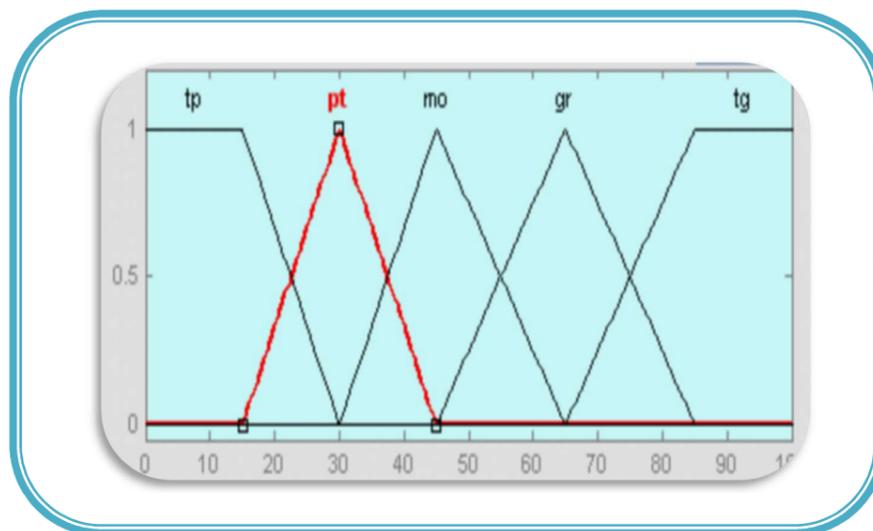


Fig.4.6 Fonction d'appartenance des variables de sorties m1, m2, m3, m4

❖ La stratégie de contrôle :

- Si Tang est EZ et Rouli est EZ et Rouli est EZ et Z est P, alors M1 égal TG et M2 égal TG et M3 égal TG et M4 égal TG ou
- Si Tang est EZ et Rouli est EZ et Rouli est EZ et Z est M, alors M1 égal G et M2 égal G et M3 égal G et M4 égal G ou
- Si Tang est EZ et Rouli est EZ et Rouli est EZ et Z est G, alors M1 égal M et M2 égal M et M3 égal M et M4 égal M ou
- Si Tang est PP et Rouli est EZ et Rouli est EZ et Z est M, alors M1 égal TG et M2 égal G et M3 égal G et M4 égal G ou
- Si Tang est NP et Rouli est EZ et Rouli est EZ et Z est M, alors M1 égal G et M2 égal G et M3 égal TG et M4 égal G ou
- Si Tang est EZ et Rouli est PP et Rouli est EZ et Z est M, alors M1 égal G et M2 égal G et M3 égal G et M4 égal TG ou
- Si Tang est EZ et Rouli est NP et Rouli est EZ et Z est M, alors M1 égal G et M2 égal TG et M3 égal G et M4 égal G ou
- Si Tang est EZ et Rouli est EZ et Lacet est P et Z est M, alors M1 égal G et M2 égal TG et M3 égal G et M4 égal TG ou
- Si Tang est EZ et Rouli est EZ et Lacet est N et Z est M, alors M1 égal TG et M2 égal G et M3 égal TG et M4 égal G

On peut alors établir l'inférence suivante en forme de tableau

Tang	Rouli	Lacet	Z	M1	M2	M3	M4
EZ	EZ	EZ	P	TG	TG	TG	TG
EZ	EZ	EZ	M	G	G	G	G
EZ	EZ	EZ	G	M	M	M	M
PP	EZ	EZ	M	TG	G	G	G
NP	EZ	EZ	M	G	G	TG	G
EZ	PP	EZ	M	G	TG	G	G
EZ	NP	EZ	M	G	TG	G	G
EZ	EZ	P	M	G	TG	G	TG
EZ	EZ	N	M	TG	G	TG	G

Tab.4.1 Table d'inférence

Type de moteur d'inférence : Type Mamdani "MAX-MIN"

Le poids d'activation est attribué par l'opérateur "et" soit le MIN des prémisses

Agrégation ou fusion, le résultat flou est la combinaison de tous les résultats en utilisant un opérateur (max).

Défuzzification : Moyenne pondérée



Et voici la figure qui représente la Simulation de la vole stationnaire :

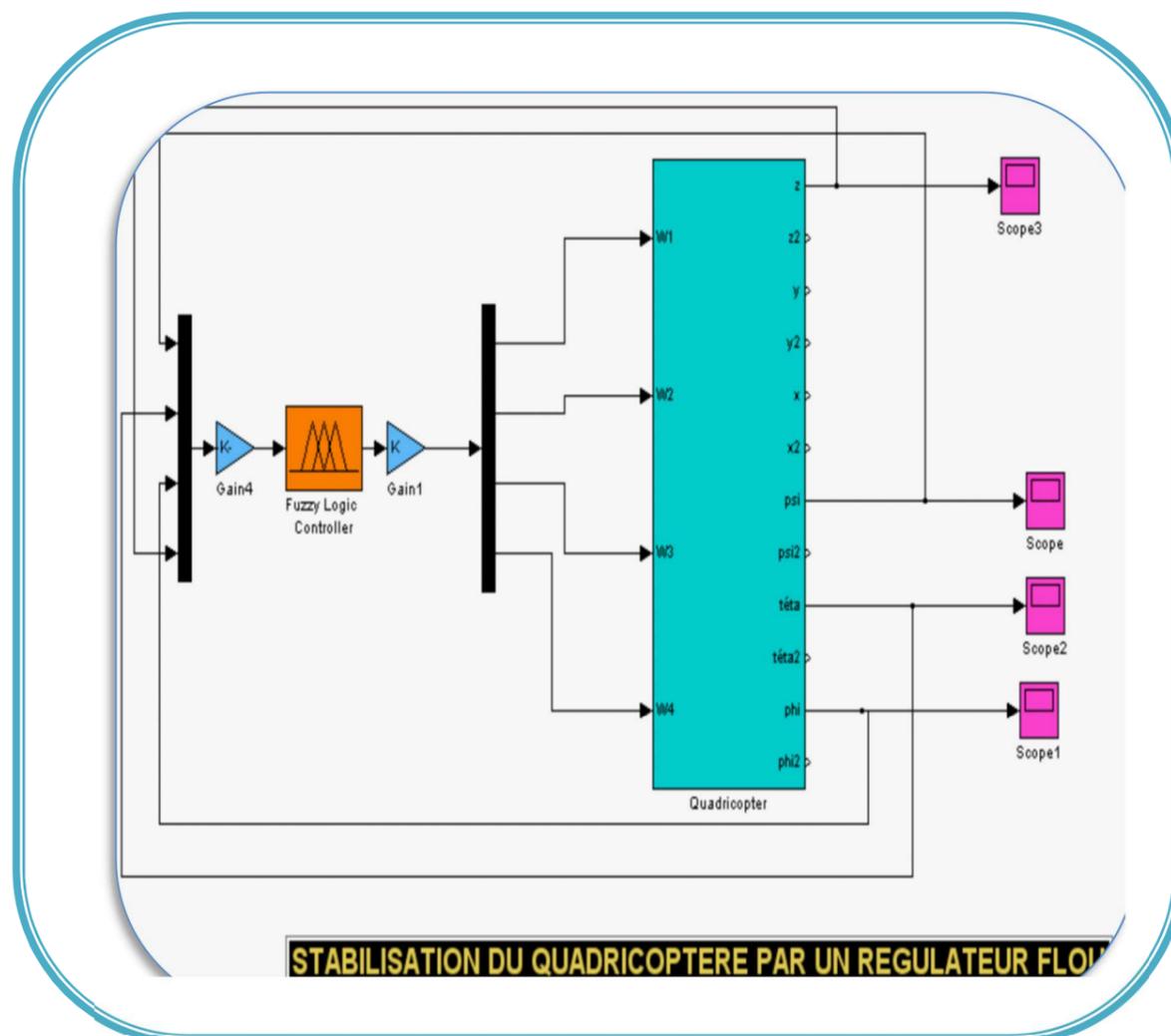


Fig.4.7 Boucle de régulation sur Simulink

Base de règle avec comme entrées Angle Roulis, Angle Tangage et l'Altitude Z

N°	anglroul	angltang	altitude z	m 1	m2	m3	m4
1	NP	NP	MO	TG	TG	TG	TG
2	NP	NP	MO	GR	GR	GR	MG
3	NP	PP	MO	TG	GR	GR	MG
4	NP	NP	MO	GR	GR	TG	MG
5	PP	NP	MO	GR	TG	GR	MG
6	NP	PP	P	GR	MO	MO	MO
7	NP	NM	P	MO	MO	GR	MO
8	NP	NM	P	MO	MO	GR	MO
9	NP	PP	P	MO	MO	MO	MG
10	NM	NP	P	MO	GR	MO	MO
11	NP	PP	MO	TG	GR	TG	MG
12	PP	PP	MO	TG	GR	GR	TG
13	NM	NM	MO	GR	TG	TG	MG
14	NM	PP	MO	TG	TG	GR	MG
15	PP	NM	MO	GR	GR	TG	TG
16	EZ	EZ	TP	MG	MG	MG	MG
17	NG	NG	P	GR	GR	GR	GR
18	EZ	EZ	MO	MO	MO	MO	MO
19	EZ	EZ	G	PT	PT	PT	PT
20	EZ	EZ	TG	TP	TP	TP	TP
21	EZ	NG	MO	TP	MO	TG	MO
22	EZ	PM	MO	PT	MO	GR	MO
23	EZ	PM	MO	MG	MO	MP	MO
24	PP	EZ	MO	MO	MP	MO	MG
25	PP	EZ	MO	MO	GR	MO	PT
26	MP	EZ	MO	MO	MG	MO	MP
27	NG	EZ	TP	MG	GR	MG	TG
28	EZ	PM	P	MO	GR	MG	GR
29	NG	EZ	TG	TP	MO	TP	PT
30	PP	EZ	G	MP	MO	PT	MP
31	EZ	NG	TG	MP	TP	PT	TP
32	EZ	NP	G	PT	TP	MP	TP
33	PG	PG	P	MO	MO	PT	PT
34	PP	PP	MO	MP	MP	TP	TP
35	NP	NP	TG	TP	TP	MP	MP
36	PM	PM	MO	GR	GR	MO	MO
37	PP	PP	P	MG	MG	GR	GR
38	PM	PM	GR	PT	PT	MP	MP
39	NM	NM	TG	MP	MP	PT	PT
40	EZ	PP	MO	GR	MO	PT	MO

Tab.4.2 Table d'inférence

Résultats

Les meilleurs résultats obtenus :

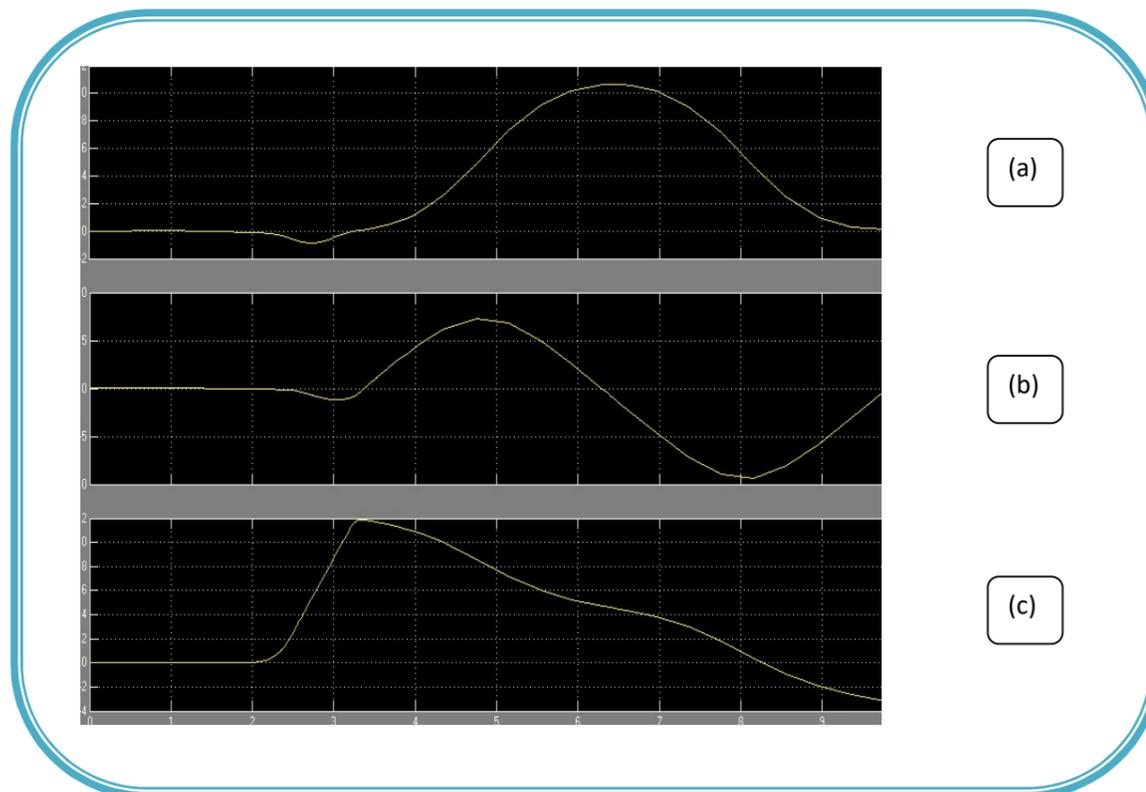


Fig.4.8 résultats de la simulation

Avec (a) axe de Roulis, (b) axe de Tangage et (c) axe de Lacet.

Discussion

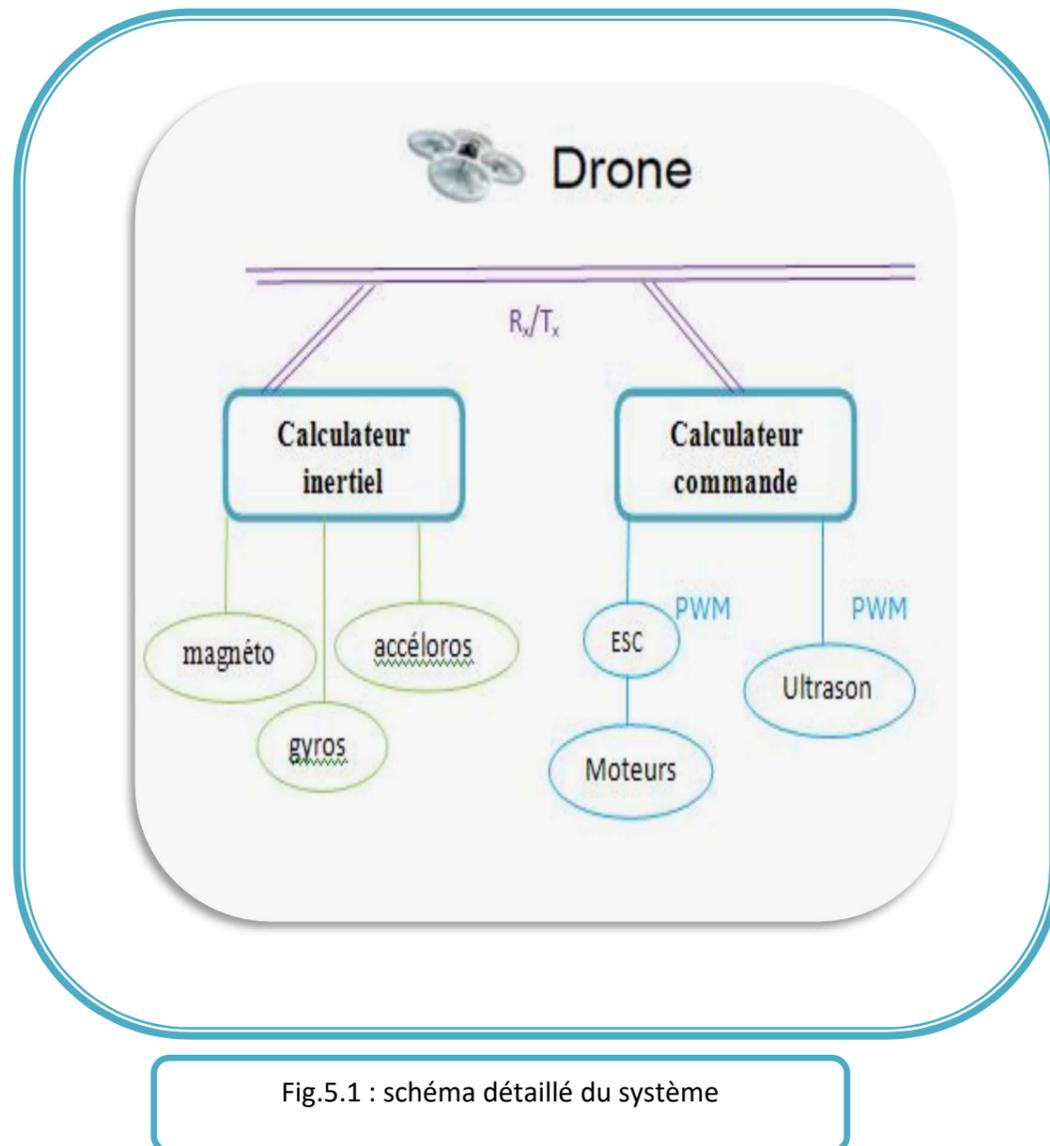
Les résultats de ce premier régulateur furent incohérents, on remarque que l'appareille a tendance à se corriger, mais à la 3^{em} seconde on remarque des oscillations qui signifient probablement que l'appareille fait des tonneaux.

L'appareille est fortement instable cas où il n'y a aucune règle active.

Cela est dû à un mauvais calibrage de la base de règle et base de connaissances.

5. Structure du projet :

Voici un schéma détaillé de l'architecture matérielle retenue pour le projet



5. 1. Choix de matériels :

5. 1.1 Structure électronique

5. 1.1.1 Calculateur :

❖ Platine Arduino "Mega 2560"

Afin de disposer d'une puissance de calcul suffisante pour réaliser l'ensemble des Tâches incombant au calculateur, notre choix s'est porté vers la **Platine Arduino "Mega 2560"**

Les modules arduino™ sont des plateformes de prototypage micro contrôlées "open-source" spécialement conçues pour les artistes, les concepteurs ou les hobistes. Cette version est architecturée autour d'un ATmega2560 associé à une interface USB.

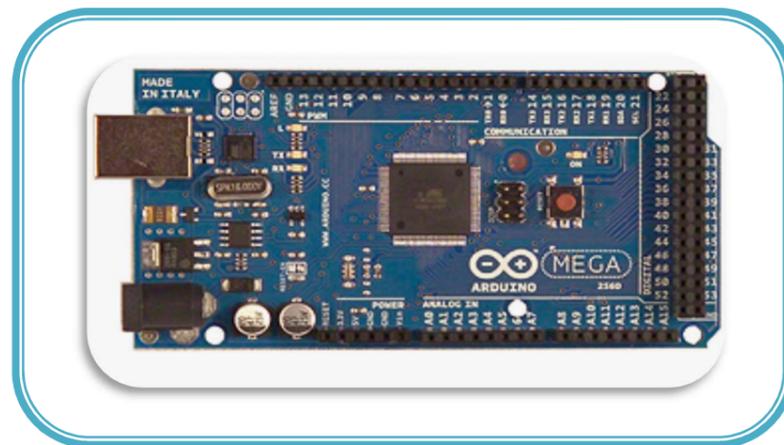


Fig.5.2 Platine Arduino "Mega 2560"

5.1.1.2 Capteur :

La qualité et la précision des capteurs sont très importantes pour assurer une bonne connaissance de l'attitude du drone dans l'espace. La centrale inertielle est bien sûr l'élément le plus important et le plus complexe, et donc le plus cher. Ainsi le choix des capteurs résulte d'un compromis entre le prix et la qualité des mesures.

❖ Ultimate IMU

Cette petite platine combine un gyroscope "ITG-3200", un accéléromètre "ADXL345" ainsi qu'une boussole électronique "HMC-5843". L'ensemble est contrôlé par un puissant microcontrôleur "LPC2148"

La platine dispose également d'un support pour carte MicroSD™, d'un support pour module de communication radio XBee™ optionnel ainsi qu'un connecteur permettant de recevoir un module GPS "EM-408" optionnel

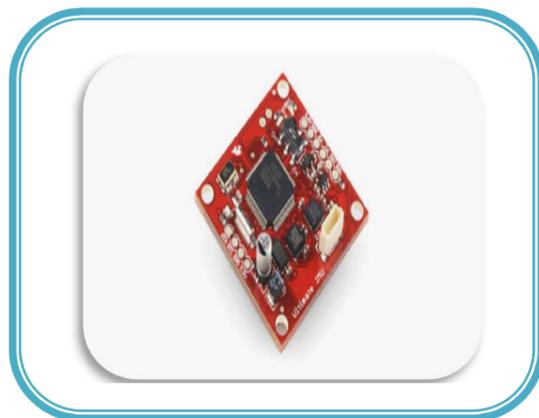


Fig.5.3 Centrale inertielle Ultimate IMU

❖ Télémètres Ultrason

Télémètres ultrason LVEZ4 de la marque Maxbotix, détectant les obstacles de 0 à 6,45m, et ce avec une résolution de 512 valeurs. Cette valeur peut-être récupérée via une liaison série RS232, une ligne analogique, ou un signal PWM. L'intérêt d'avoir choisi ces télémètres est qu'ils possèdent des cônes de détection : un cône large pour l'évaluation de la distance au sol et au plafond



Fig.5.4 Télémètres ultrason LVEZ4

5.1.2 Structure mécanique :

❖ Éléments utilisés :

- 4 moteurs TowerPro Brushless Outrunner 2410-09 13A / 104W donnés pour 840kv (tours par minute par volt). Ces moteurs peuvent soulever chacun une charge d'environ 600g, soit 2,4kg pour l'ensemble du drone. Le fonctionnement en 10V des moteurs nécessite l'utilisation de batteries 3S (3 cellules de 3,7V en série) fournissant du 11,1V.



Fig5.5 Moteur brushless TowerPro

- 2 couples d'hélices rotatives et contrarotatives aux dimensions 10"x4, 5". L'utilisation de ces deux types d'hélices satisfaites les caractéristiques propres aux quadri rotors dont les 2 moteurs d'un axe tournent dans le sens opposé à celui des 2 moteurs de l'autre axe.



Fig.5.6 Hélice

- 2 batteries Rhino 4900mAh 3S1P 11.1v 20C Lipoly Pack assurant les besoins énergétiques du drone. Une batterie 20C de 4900mAh offre un ampérage maximum de $4,9 \times 20$ soit 98A. Les 4 moteurs demandent au maximum $4 \times 13,5$ A soit 54A. La batterie est volontairement surdimensionnée car les habitués de batteries Lipo conseillent une utilisation à 2/3 des capacités de la batterie pour ne pas l'endommager. D'après les tests réalisés, on voit que le drone décolle à mi-throttle, donc à la moitié des possibilités des moteurs. On considère donc un ampérage nécessaire de $4 \times 13,5 / 2$ soit 27A, la batterie pouvant fournir 4,7A par heure, on obtient donc une autonomie approximative de 10 minutes en vol stationnaire.



Fig.5.7 batteries Rhino

- 4 contrôleurs TowerPro *SS Series 18A ESC* assurant la gestion des besoins électriques des moteurs vis-à-vis de la batterie, besoins définis par les commandes moteurs en sortie de calculateur. Ces contrôleurs supportent un ampérage constant de 18A. Les moteurs ne consomment chacun que 13,5A maximum ce qui offre une marge de sécurité satisfaisante par rapport aux caractéristiques

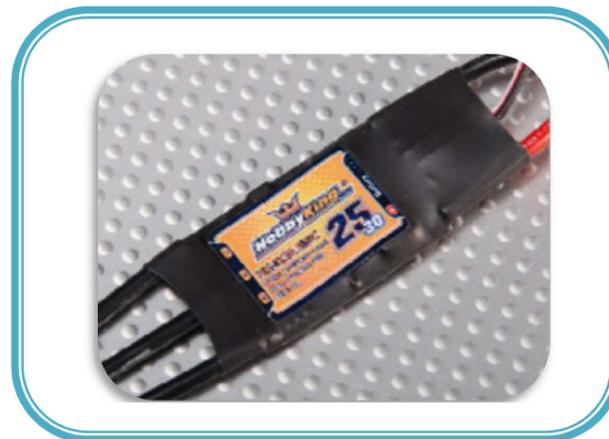


Fig.5.8 contrôleurs TowerPro

- Des tubes 12 mm et des feuilles en aluminium pour l'ossature du drone. L'aluminium, de part sa structure particulière, offre une grande résistance pour un poids réduit.

Fig.5.9 Tubes en Aluminium

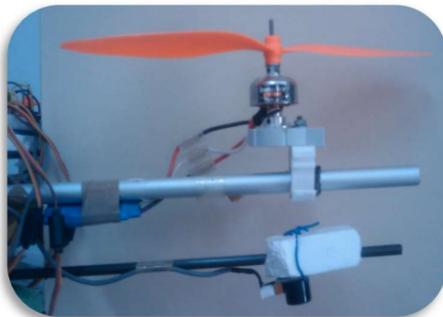


Fig.5.10 Fixation du moteur

Fig.5.11 Quadricopter soutenu par des files



Fig.5.12 Dispositif électronique

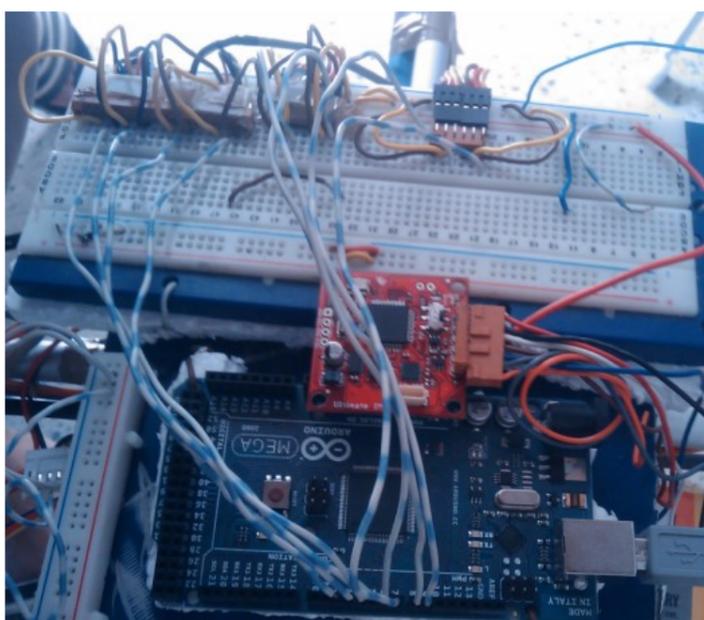


Fig.5.13 Vue d'ensemble

6.1 Calculateur Inertiel

La centrale inertielle présente un bon rapport qualité prix et est donc abordable pour les particuliers. C'est sûrement ce qui explique qu'une communauté de développeurs et de passionnés de modélisme ait vu le jour sur Internet. Le constructeur de l'IMU met à disposition des firmwares complets à programmer sur l'IMU, des fonctionnalités élémentaires sont déjà implémentées. De cette manière, en étudiant le code fourni nous avons pu utiliser les fonctions suivantes :

- ❖ acquisition des valeurs capteurs gyroscope "ITG-3200", accéléromètre "ADXL345" magnétomètre "HMC-5843".
- ❖ Calcule des angles de Roulis, Tangage et Lacet, cette fonction est implémentée dans le firmware et permet d'interpréter la valeur de chaque angle et la vitesse angulaire à partir des capteurs afin des les exploiter facilement.
- ❖ Transmettre les données au calculateur Principale, par liaison série (Tx/Rx).

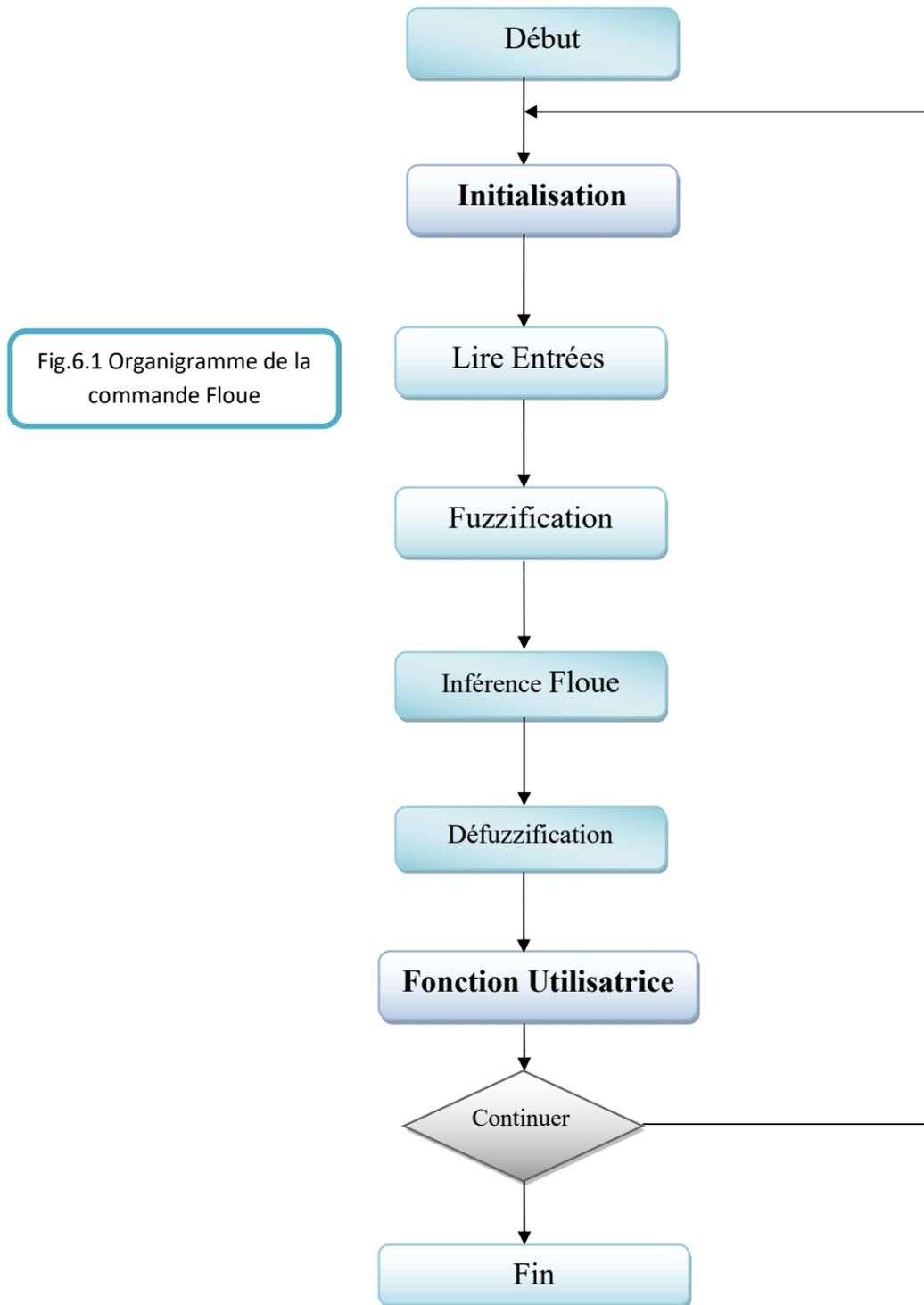
6.2 Calculateur Principale

Comme déjà présenté nous avons utilisé une carte Arduino Mega, afin de réaliser les taches suivantes :

- ❖ Réception des angles calculer par la centrale inertielle (IMU), par liaison série (Tx/Rx).
- ❖ Faire l'acquisition et le traitement (calcul de l'altitude) des signaux PWM des télémètres ultrasons.
- ❖ Dérouler l'algorithme de commande (régulateur flou).
- ❖ générer les 4 signaux PWM à destination du contrôleur de chaque moteur. Ces signaux PWM représentent les commandes envoyées aux moteurs.

6.3 Algorithme du régulateur flou :

Les définitions données précédemment permettent de présenter l'algorithme général.



➤ **Initialisation** : dresser la base de règle et la base de données

6 .3.1 Implémentation de l'algorithme flou sur la carte Arduino :

Implémenter un algorithme flou avec un langage C n'est pas une tâche facile et nécessite des heures et des heures de programmation à jouer avec des pointeurs et des structures, heureusement pour des novices comme nous en matière de programmation il existe des générateurs de code (FuzzyTHECH) (FuzzC), mais ils sont que sous licence, avec quelques journées de recherche sur internet on a trouvé un téléchargeable sans licence (Fudge) de la compagnie Motorola

Fudge :

Un logiciel doté d'une interface graphique proche de celui proposé par Fuzzy Logic ToolBox qui permet de définir complètement le système flou et d'évaluer l'inférence mais le plus important c'est qu'il en génère un code C

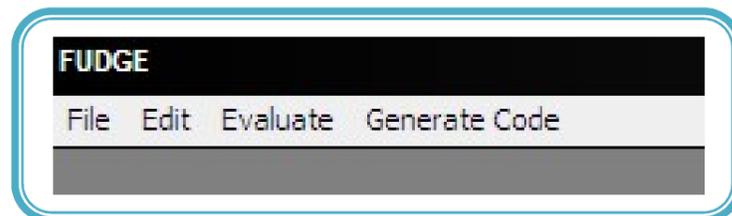


Fig.6.2 Barre d'outils de fudge

On décrit ci-après les différents menus et leurs options

Menu File :

New : nouveau système flou

Open : lecture d'un système flou sauvegardé

Save as : sauvegarde sur disque du système en cours

Menu Edit :

Crisp Inputs : ajout des variables d'entrées.

Inputs Membership Function : éditeur de fonction d'appartenance pour les variables d'entrées.

Crisp Outputs : ajout des variables de sorties.

Outputs Membership Function : éditeur de fonction d'appartenance pour les variables d'entrées.

Rules : éditeur de règles floues.

Menu Evaluate:

Fuzzy Logic Evaluator : permet d'évaluer le degré d'activation de la règle et la valeur que va la sortie en fonction des entrées

Control Surface : affiche la surface représentant la variable de sortie en fonction de la variable de sortie

Menu Generate Code:

Comme c'est un produit Motorola il faut s'attendre à ce que les codes générés sont destinés aux microcontrôleurs de la firme mais ils ont pensé à générer un code pur C.

(MC68HC05), (MC68HC11), (MC68HC16) et (MC68000) : génération de code Basic 1

Ansi C : génération de code C

Avec quelques modifications il peut être compatible avec notre microcontrôleur Atmega

En cliquant sur le **Menu Edit** après sur *Crisp Inputs*, elle permet de créer la variable, l'unité de mesure et le nombre de fonctions d'appartenance. Voir la figure 6.3

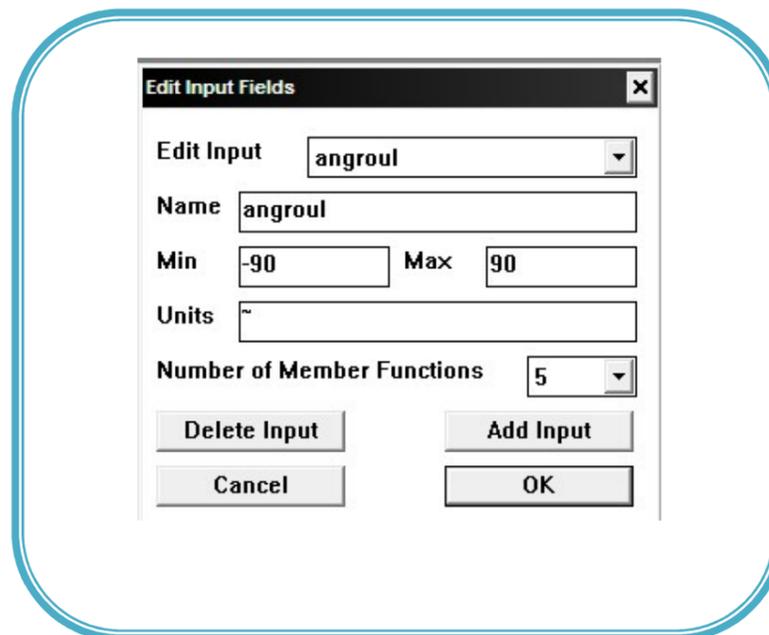


Fig.6.3. fenêtre Edit Crisp

On fait de même pour *Crisp Inputs*.

En cliquant sur *Inputs Membership Function*. Cette fenêtre permet de configurer les valeurs linguistiques et leur sous ensembles respectifs voir la figure 6.4. le seul inconvénient est qu'il est limitée a des fonction d'appartenance rectangle (en donne les valeurs des 4 points qui caractérise le rectangle, pour avoir une fonction triangulaire les deux point centraux doivent avoir la même valeur)

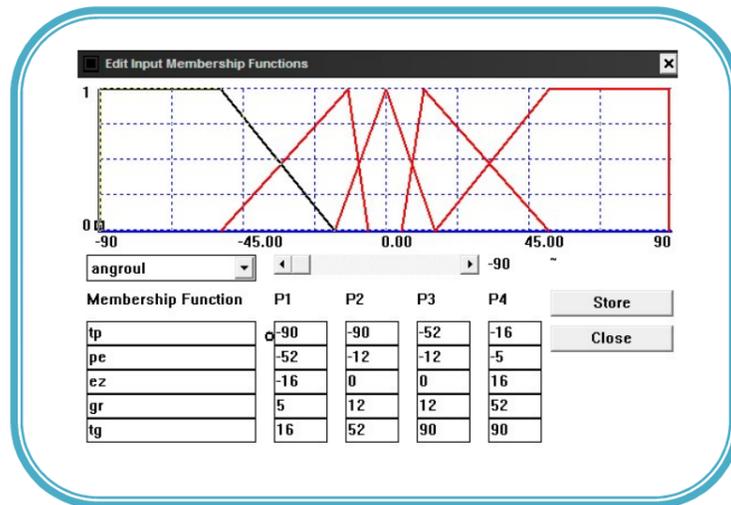


Fig .6.4.fenêtre Edit Crisp

En cliquant sur *Outputs Membership Function*. Cette fenêtre permet de configurer les valeurs linguistiques et leur sous ensembles respectifs voir la figure 6.5, les sous ensembles des variables de sorties sont des fonctions singletons (en pratique on préfère les fonctions de type singleton pour alléger les calculs)

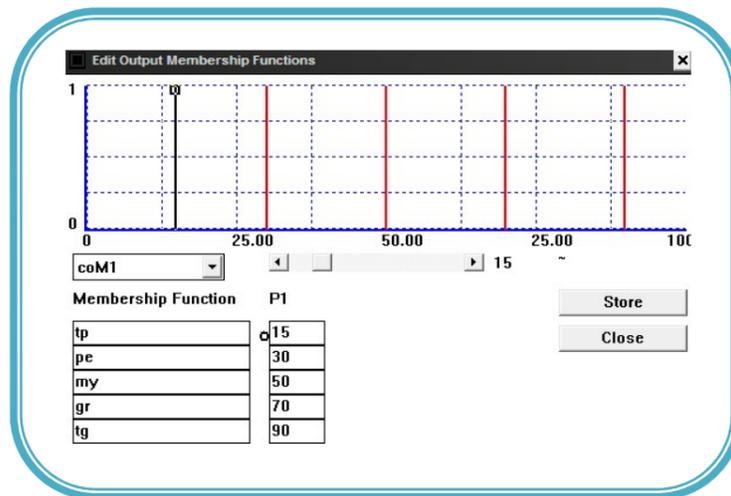


Fig .6.5.fenêtre Edit Crisp

En cliquant sur *Rules* la fenêtre permet l'ajout et la suppression des règles, Voir figure 6.6

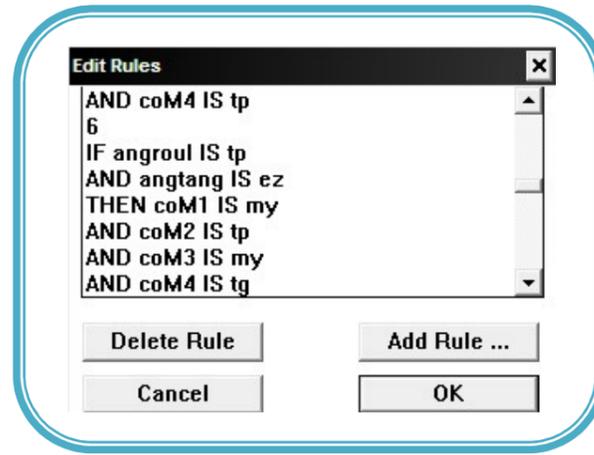


Fig .6.6.fenêtre Edit Rules

Menu Evaluate Fuzzy Logic Evaluator cette fenêtre permet d'évaluer l'inférence flou, l'inférence est base sur le contrôleur de Mamdani (Min/Max), Voir figure 6.7

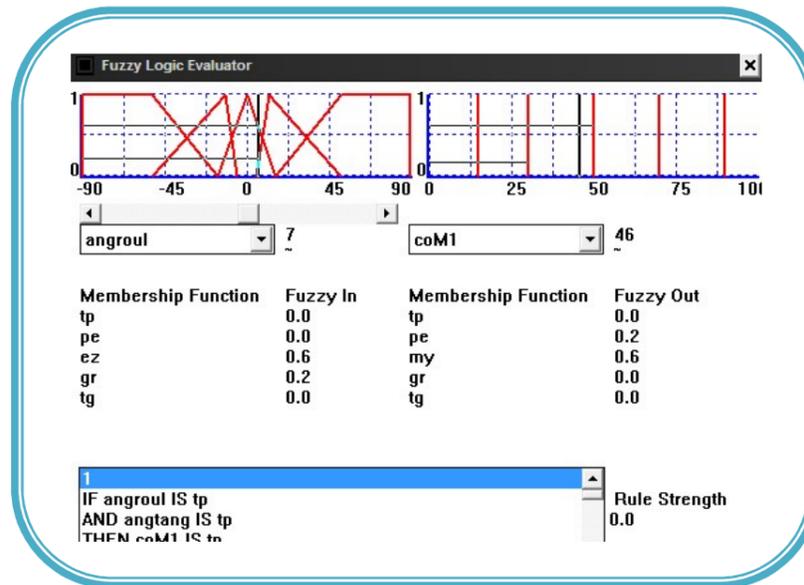


Fig .6.7.fenêtre Fuzzy logic

Après avoir défini les variables d'entrées et variables de sorties, leur univers de discours, leur ensemble de sous-ensembles flous (valeur linguistique), définit la base de règle et évalué le système flou, On clique sur **Menu Generate Code** pour avoir le fichier code source C, un fichier '.h' (header)

6.4 Application au Drone :

Les variables linguistiques d'entrées et leurs intervalles numériques : les angles calculés par la centrale inertielle sont en degrés

- ❖ Angle Tangage (θ) $(-90 \leq \theta < 90)$ deg.
- ❖ Angle Roulis (ϕ) $(-90 \leq \phi < 90)$ deg.
- ❖ Angle Lacet (ψ) $(-180 \leq \psi < 180)$ deg.
- ❖ L'altitude (Z) $(0 \leq Z < 600)$ cm

Les variables linguistiques de sorties et leurs intervalles numériques : les commande moteur sont 0 à 100, ils sont générés par une fonction programmer sur l'environnement de programmation de l'Arduino

- ❖ Commande moteur 1 (m1) $(0 \leq m1 < 100)$.
- ❖ Commande moteur 2 (m2) $(0 \leq m2 < 100)$.
- ❖ Commande moteur 3 (m3) $(0 \leq m3 < 100)$.
- ❖ Commande moteur 4 (m4) $(0 \leq m4 < 100)$

Fonctions d'appartenance floue sur **Fudge** pour chacune des variables d'entrée et de sortie.

- Fonction d'appartenance des variables angle Tangage et angle Roulis
Variable linguistique (NG), (NP), (EZ), (PP), (PG)

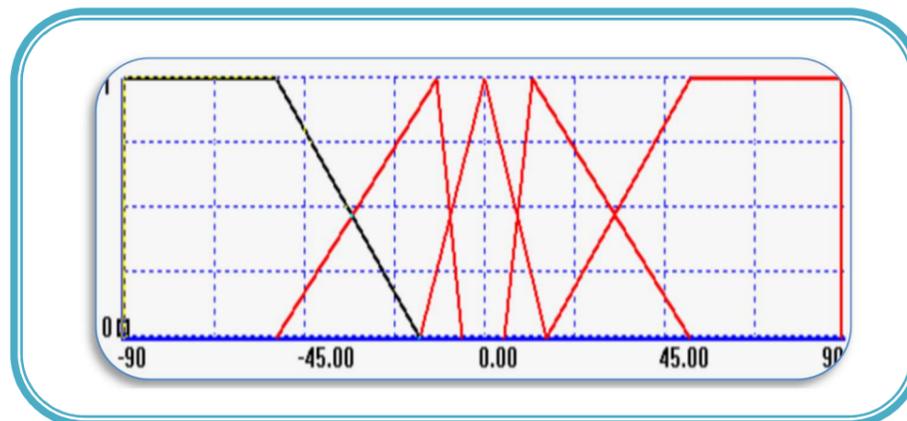


Fig.6.8 fonctions d'appartenances des angles Roulis et Tangage

- Fonction d'appartenance de la variable angle Lacet
Variable linguistique (N), (EZ), (P)

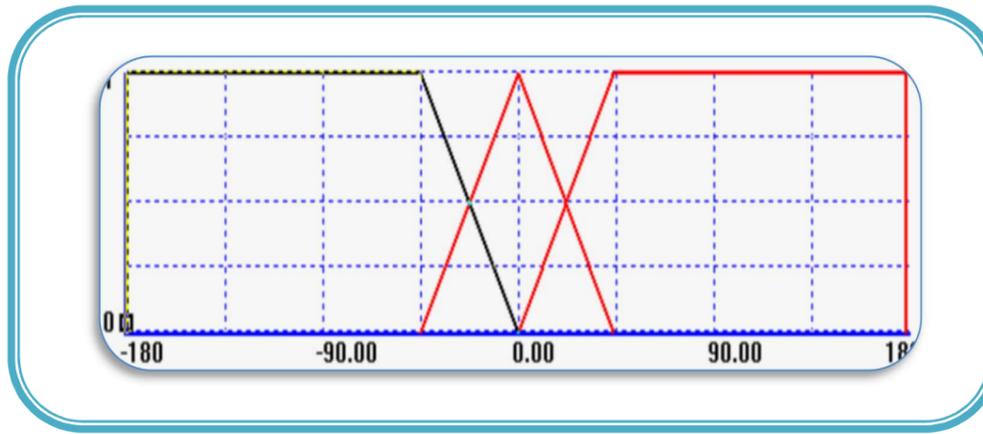


Fig.6.9.fonction d'appartenance de l'angle Lacet

- Fonction d'appartenance des variables d'entre altitude Z
Variable linguistique (TP), (P), (M), (G), (TG)

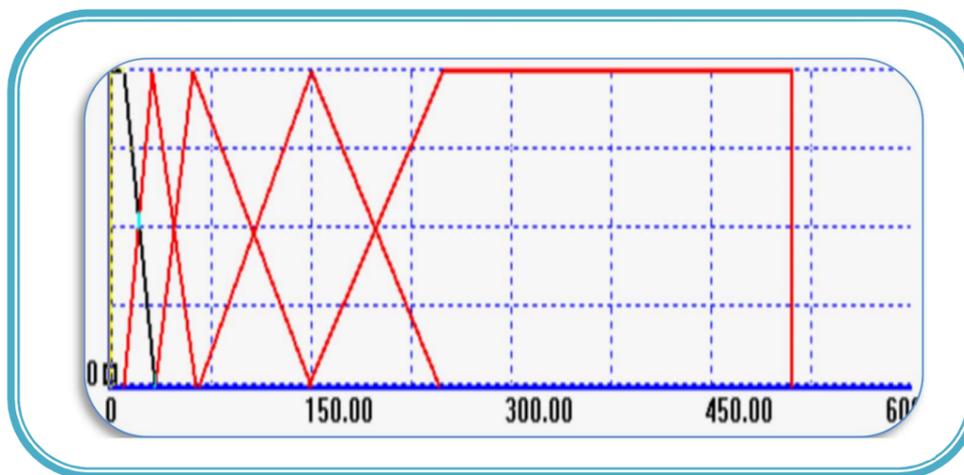


Fig.6.10.Fonction d'appartenance de

- Fonction d'appartenance des variables de sorties m1, m2, m3, m4
Variable linguistique (TTP), (TP), (P), (M), (G), (TG), (TTG).

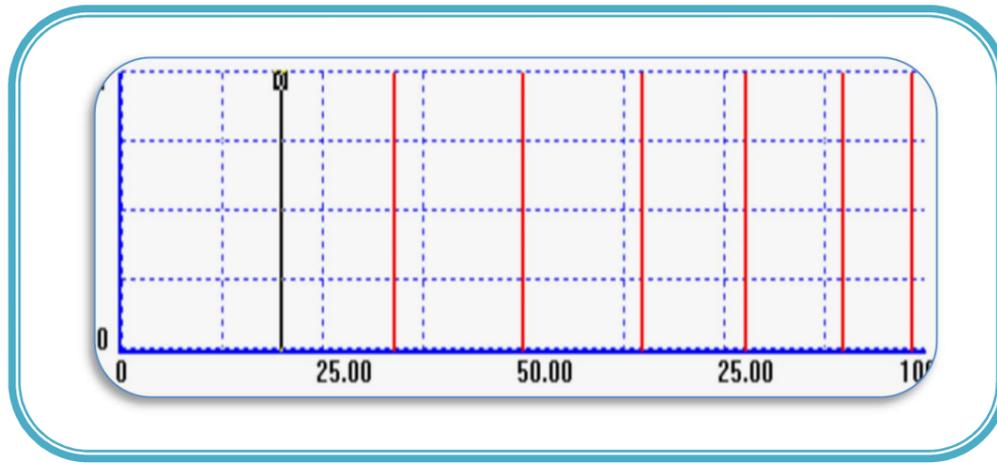


Fig.6.11.Fonction d'appartenance des sorties Commande

✚ Avant la réalisation de la structure on a essayé un petits régulateur flou pour un seul moteur avec comme capteur un potentiomètre, le régulateur fonctionne très bien en a vus la variation de la commande suivant les règles établis, Lorsque la structure a était faite on ai tombé sur un problème de taille, tout marché merveilleusement bien seulement une chose la plus importante les données envoyées par la centrale inertielle arrivées codés ou un truque dans le genre, on a passé beaucoup de temps a cherche une solution pour réceptionner la bonne information mais rien de bon.

Que faire dans ce cas ? On a un quadri copter, un calculateur, une centrale inertielle et des capteurs ultrasoniques on veut stabiliser le système mais la pièce maitresse (information capteur) ne marche pas ?

Solution : On a placé un ultrason en dessous de chaque moteur qui donne une information sur l'altitude de ce dernier, et on a définit une commande pour garder les 4 moteurs à la même altitude.

Soit le système flou suivant :

Les variables linguistiques d'entrées et définir leurs intervalles numériques :
les distance par apport au sol traité par les ultrasons

- ❖ La valeur capteur1 (sen1) $(0 \leq sen1 < 600)$ cm.
- ❖ La valeur capteur1 (sen2) $(0 \leq sen2 < 600)$ cm.
- ❖ La valeur capteur1 (sen3) $(0 \leq sen3 < 600)$ cm.
- ❖ La valeur capteur1 (sen4) $(0 \leq sen4 < 600)$ cm.

Les variables linguistiques de sorties et définir leurs intervalles numériques:

Les commandes motrices sont 0 à 100, ils sont générés par une fonction programmée sur l'environnement de programmation de l'Arduino.

- ❖ Commande moteur 1 (m1) $(0 \leq m1 < 100)$.
- ❖ Commande moteur 2 (m2) $(0 \leq m2 < 100)$.
- ❖ Commande moteur 3 (m3) $(0 \leq m3 < 100)$.
- ❖ Commande moteur 4 (m4) $(0 \leq m4 < 100)$

Fonctions d'appartenance floue sur **Fudge** pour chacune des variables d'entrée et de sortie.

- Fonction d'appartenance des variables d'entrées
Variable linguistique (NG), (NP), (EZ), (PP), (PG)

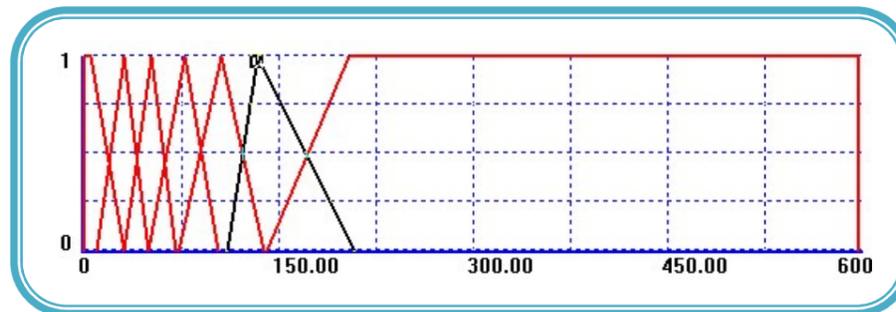


Fig.6.12.fonctions d'appartenances définissant l'altitude de chaque moteur

- Fonction d'appartenance des variables de sorties m1, m2, m3, m4
Variable linguistique (TTP), (TP), (P), (M), (G), (TG), (TTG).

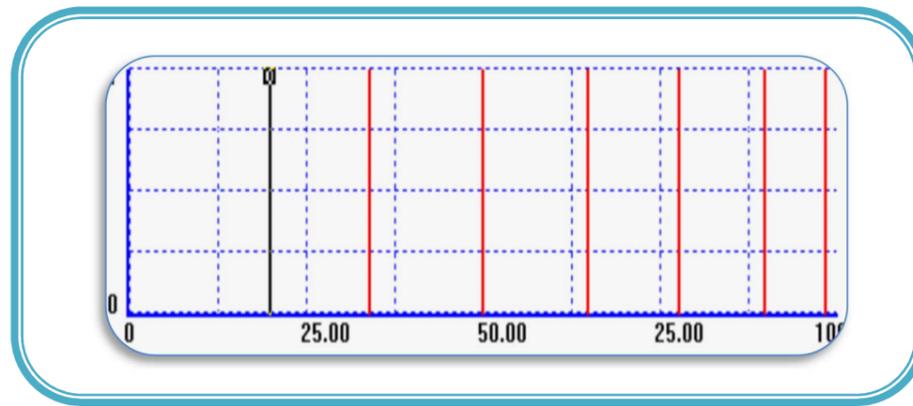


Fig.6.13.Fonction d'appartenance des sorties Commandes

6.5 Base de règle

Les choix des règles semble facile et naturel, parce que l'on peut les définir d'une manière presque intuitive, et vu le nombre d'entrées et le fait que chaque règle en plus diminue la vitesse du réglage et complique l'algorithme, nous allons choisir le moins de règles possible

SEN1	SEN2	SEN3	SEN4	M1	M2	M3	M4
TTP	TTP	TTP	TTP	TTG	TTG	TTG	TTG
TP	TP	TP	TP	TTG	TTG	TTG	TTG
P	P	P	P	TTG	TTG	TTG	TTG
M	M	M	M	TG	TG	TG	TG
G	G	G	G	M	M	M	M
TG	TG	TG	TG	P	P	P	P
TTG	TTG	TTG	TTG	TP	TP	TP	TP
TP	P	M	P	TTG	TG	G	TG
M	P	TP	P	TG	TTG	TG	G
P	TP	P	M	TG	TTG	TG	G
P	M	P	TP	TG	G	TG	TTG
M	P	P	TP	G	TG	TG	TTG
P	TP	M	P	TG	TTG	G	TG
P	M	TP	P	TG	G	TTG	TG
TG	P	P	M	TTG	TG	TG	G

Tab.6.1 Tableau d'inférence

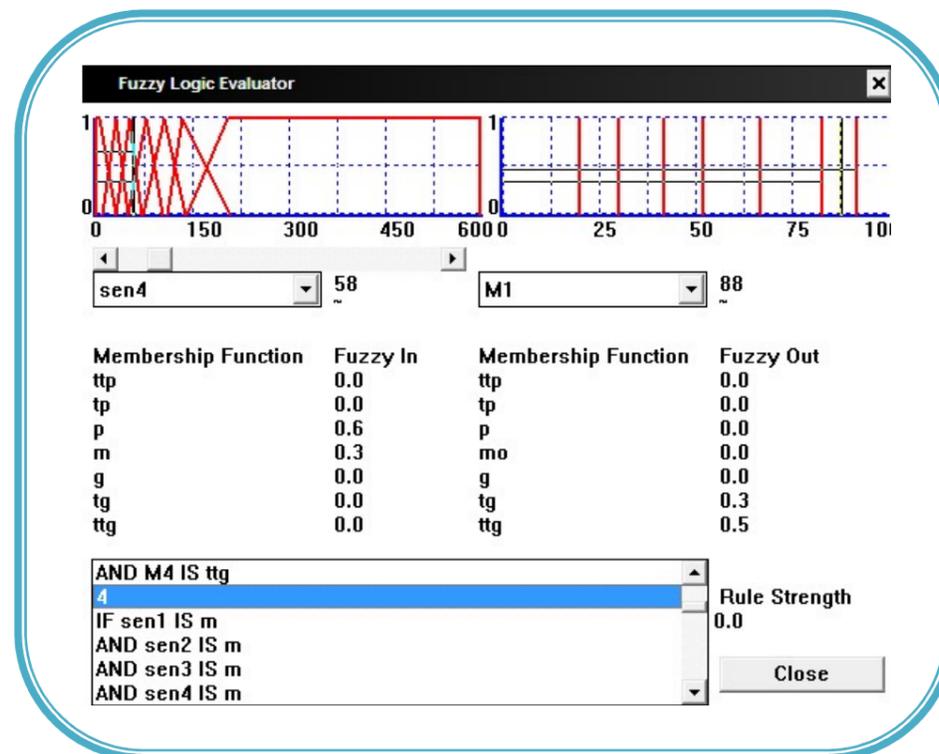


Fig.6.14.Fenêtre d'évaluation

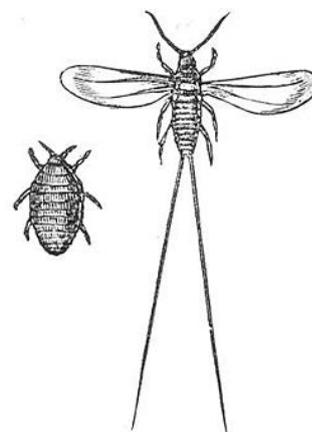
Discussion

Lors de l'implémentation du code généré par Fudge l'appareille décolle ce qui montre que le calculateur principale a fait le traitement des données acquits par les ultrasons et il a déroulé l'algorithme flou mais en raison de capteurs bruités le drone a un comportement instable et il peut être du a un mauvais calibrage des paramètres du régulateur, ce qui explique pourquoi la conception d'un régulateur flou se termine habituellement par l'ajustement des paramètres à la main.

E120 Une teinture d'un **E471** Le E471 provient en général de graisse végétale, mais il n'est pas exclu d'en trouver provenant de graisse animale. La composition chimique du composé étant identique, l'indication de provenance (végétale ou animale) n'est pas obligatoire.

Il est possible que le [porc](#) entre dans sa composition, aussi, les [Juifs](#) et les [Musulmans](#) éviteront les produits contenant cet additif sauf s'il est certifié d'origine végétale ou possédant le label Casherout ou Hallal.

Le pourpre profond est extraite des femelles cochenille. La femelle *Dactylopius coccus* est utilisée pour produire différentes teintes de rouges, d'orange et d'écarlate. Cette coloration provient de l'acide carminique. Les cochenilles ont une teneur en acide carminique naturel extrait de 19 à 22%. Les insectes sont tués par immersion dans l'eau chaude (après quoi elles sont séchées) ou par l'exposition aux rayons du soleil, de vapeur ou la chaleur d'un four. Chaque méthode produit une couleur différente. Les insectes doivent être séchés à environ 30 pour cent de leur poids initial avant de pouvoir être stockés sans se désintégrer. Il faut environ 70.000 insectes pour produire une livre de teinture de cochenille.



Il existe deux formes principales de teinture de cochenille: l'extrait de cochenille est un colorant dont la base est l'état brut séché et pulvérisé du corps des insectes, alors que le carmin est une coloration plus épurée. Pour préparer le carmin, les corps d'insectes en poudre sont bouillis dans de l'ammoniaque ou une solution de carbonate de sodium, l'insoluble est éliminée par filtrage, et de l'alun est ajouté à la solution de sel de l'acide carminique clair pour précipiter le sel d'aluminium rouge. La pureté des couleurs est assurée par l'absence de fer. Chlorure d'étain, de l'acide citrique, du borax, ou de la **gélatine** peut-être ajoutée afin de réguler la formation du précipité. Pour les tons de pourpre, de la chaux est ajoutée à l'alun.

En 2005, le Pérou a produit 200 tonnes de teinture de cochenille et les îles Canaries en ont produit 20 tonnes. Le Chili et le Mexique ont aussi récemment commencé à exporter cette teinture. La France est considérée comme le plus gros importateur mondial, le Japon et l'Italie importent aussi les insectes. Une grande partie de ces importations sont transformées et réexportées vers les autres pays développés. En 2005, le prix du marché de la cochenille était compris entre 50 et 80 USD par kilogramme, tandis que les colorants alimentaires synthétiques bruts sont disponibles pour 10 à 20 USD par kilogramme.

Origine du E120 :



Issu d'insectes

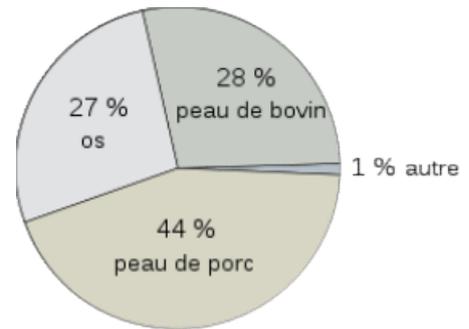
Régimes spéciaux :

- [Musulman](#) Cet additif est douteux (parfois halal, parfois pas)
- Substance obtenue à partir d'insectes. Théoriquement c'est autorisé.
- [Végétarien](#) Cet additif n'est pas végétarien
- Substance obtenue à partir de cochenilles (un insecte)
- [Juif](#) Cet additif n'est pas casher

E 441

À l'échelle industrielle, la gélatine est fabriquée à partir de sous-produits de l'industrie du **cuir** et de la **viande**, principalement les os et **dépouilles** des **bovins**.

Les **phoques** et les **requins** sont aussi d'excellentes sources de collagène. Les peaux de **porcs** sont également utilisées ce qui peut poser problème pour les juifs⁴ et les musulmans respectueux des règles imposées par leur religion car l'origine de la gélatine ne figure généralement pas dans l'étiquetage des produits alimentaires. Les os de porcs et autres os de seconde qualité, dits « os verts » ne sont pas utilisés⁵. Contrairement à l'idée reçue, la corne n'est pas utilisée. Les matériaux bruts sont préparés par des procédés de séchage, des procédés **acides** et des procédés **basiques**, qui sont employés pour extraire le **collagène**, ce qui peut prendre plusieurs semaines. La production mondiale est de l'ordre de 250 000 tonnes par an.



Origine de la gélatine

E440

Les **pectines** (du **grec ancien** πηκτός / *pêktós*, « épaissi, caillé »), ou plus largement les substances pectiques, sont des **polyosides**, rattachées aux **glucides**. Ce sont des substances **exclusivement d'origine végétale**. La pectine est présente en grande quantité dans les pépins et les zestes de **groseilles**, **pommes**, **coings** et **agrumes**.

E452

Origine du E452 :

Origine inconnue

Régimes spéciaux :

- **Musulman**Cet additif est halal
- **Végétarien**Cet additif n'est pas fait à partir d'animaux (ni de leur lait ou oeufs)
- **Juif**Cet additif est casher

E330

Origine du E330 :



Issu de végétaux

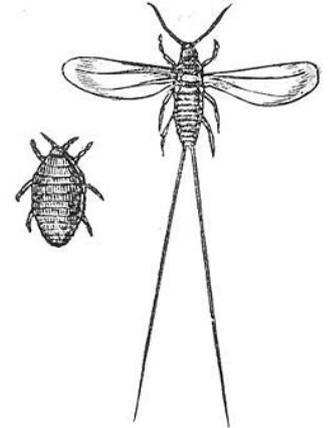
Régimes spéciaux :

- **Musulman**Cet additif est halal
- **Végétarien**Cet additif n'est pas fait à partir d'animaux (ni de leur lait ou oeufs)
- **Juif**Cet additif est casher

E471 Le E471 provient en général de graisse végétale, mais il n'est pas exclu d'en trouver provenant de graisse animale. La composition chimique du composé étant identique, l'indication de provenance (végétale ou animale) n'est pas obligatoire.

Il est possible que le [porc](#) entre dans sa composition, aussi, les [Juifs](#) et les [Musulmans](#) éviteront les produits contenant cet additif sauf s'il est certifié d'origine végétale ou possédant le label Casherout ou Hallal.

E120 Une teinture d'un pourpre profond est extraite des femelles cochenille. La femelle *Dactylopius coccus* est utilisée pour produire différentes teintes de rouges, d'orange et d'écarlate. Cette coloration provient de l'acide carminique. Les cochenilles ont un teneur en acides carminique naturel extrait de 19 à 22%. Les insectes sont tués par immersion dans l'eau chaude (après quoi elles sont séchées) ou par l'exposition aux rayons du soleil, de vapeur ou la chaleur d'un four. Chaque méthode produit une couleur différente. Les insectes doivent être séchés à environ 30 pour cent de leur poids initial avant de pouvoir être stocké sans se désintégrer. Il faut environ 70.000 insectes pour produire une livre de teinture de cochenille.



Il existe deux formes principales de teinture de cochenille: l'extrait de cochenille est un colorant dont la base est l'état brut séché et pulvérisé du corps des insectes, alors que le carmin est une coloration plus épurée. Pour préparer le carmin, le corps d'insectes en poudre sont bouillies dans de l'ammoniaque ou une solution de carbonate de sodium, l'insoluble est éliminée par filtrage, et de l'alun est ajouté à la solution de sel de l'acide carminique clair pour précipiter le sel d'aluminium rouge. La pureté des couleurs est assurée par l'absence de fer. Chlorure d'étain, de l'acide citrique, du borax, ou de **la gélatine** peut-être ajoutée afin de réguler la formation du précipité. Pour les tons de pourpre, de la chaux est ajoutée à l'alun.

En 2005, le Pérou a produit 200 tonnes de teinture de cochenille et les îles Canaries en ont produit 20 tonnes. Le Chili et le Mexique ont aussi récemment commencé à exporter cette teinture. La France est considéré comme le plus gros importateur mondial, le Japon et l'Italie importent aussi les insectes. Une grande partie de ces importations sont transformées et réexportés vers les autres pays développés. En 2005, le prix du marché de la cochenille était compris entre 50 et 80 USD par kilogramme, tandis que les colorants alimentaires synthétiques bruts sont disponibles pour 10 à 20 USD par kilogramme.

Origine du E120 :



Issu d'insectes

Régimes spéciaux :

- [Musulman](#) Cet additif est douteux (parfois halal, parfois pas)
- Substance obtenue à partir d'insectes. Théoriquement c'est autorisé.
- [Végétarien](#) Cet additif n'est pas végétarien
- Substance obtenue à partir de cochenilles (un insecte)
- [Juif](#) Cet additif n'est pas casher

Conclusion

Dans ce travail on a prité par la réalisation d'un système drone quadri copter et pour faire sa stabilisation par un contrôleur flou.

La mise en œuvre de régulateur Flou souffre de problèmes d'expertise (dressage de la base de données et base de règles).

L'utilisation de Matlab Fuzzy Logic Toolbox peut éviter beaucoup de crache, on s'approche du contexte voulu par la régulation après il reste quelque ajustement à la main pour l'analyse du système réel.

L'implémentation de contrôleur flou peut devenir très facile en utilisant des générateurs de code tel que Fudge ou autre.

Le bon choix du matériel utilisé a un effet considérable dans la réussite du projet.

L'application de filtre pour les données capteurs tel que le filtre de Kalman est indispensable.

L'exploitation d'une supervision basée sur un système hybride Neuro-Flou à optimisation génétique en vue de l'adaptation des paramètres de synthèse du régulateur, permet d'offrir une alternative forte intéressante dont l'objectif est de satisfaire les critères de performances désirées.

Même si on n'a pas pu stabiliser le quadri copter une suite pourra être donnée à ce projet, en intégrant un algorithme d'optimisation pour le régulateur de stabilisation, un algorithme de guidage et de prise de décision.

Difficultés du Projet

Dans un premier temps, compte-tenu de la nature du projet, il nous a été difficile de déterminer l'ampleur de la tâche. La réalisation d'un drone quadri rotor offre de nombreuses possibilités en matière d'applications.

Nous avons rencontré plusieurs difficultés dans différents points du projet :

- L'ossature et matérielles utiliser pour la réalisation du quadricopter, nous étions obligés d'attendre jusqu'à que le matérielle soit en main.
- Familiarisation avec certain composants notamment avec la centrale inertielle où nous nous sommes longtemps penchés sur le problème de communication avec le calculateur principale.
- Essai libre sans banc de teste ce qui a causé beaucoup de craches.
- Absence de filtrage des données capteurs ultrasonique.
- L'implémentation de l'algorithme flou avant de trouver l'option générateur de code

D'une manière générale nous avons mal estimé la durée à attribuer à certaines tâches, ce qui au final ne nous a pas permis de finir tout ce que souhaitions entreprendre.

Bibliographie:

- L. Derafa, L. Fridman, A. Benallegue and A. Ouldali . **Super Twisting Control Algorithm for the Four Rotors Helicopter Attitude Tracking Problem.** Mexico City, Mexico, June 26 - 28, 2010.
- L. Derafa, T. Madani, A. Benallegue . **Dynamic Modelling and Experimental Identification of Four Rotors Helicopter Parameters**
- M. Mokhtari, M. Marie. **Applications de MATLAB 5 et SIMULINK 2. Logique floue. Réseaux de neurones .traitement de signal.**
- Jean-Camille DE BARROS, Daniel HISSEL, Pascal MAUSSION, Michel THOLOMIER, André TITLI. **Systèmes et régulateurs flous.** Techniques de l'ingénieur.
- Dmien Poinsot. **Modélisation et commande d'un VTOL UAV par déflexion de flux.**
- Yassine HADDAB, Bernard LANG, **Guillaume LAURENT. Matlab / simulink pour l'analyse et la commande de systèmes.**
- Wei-Yen Wang , Yi-Hsing Chien , Yih-Guang Leu , Tsu-Tian Lee .**Automatica . Adaptive T-S fuzzy-neural modeling and control for general MIMO unknown nonaffine nonlinear systems using projection update laws.**
- Guy STUTZER, Eddie SMIGIEL. **Modélisation et commande du moteur triphasé. Commande vectorielle des moteur synchrones commande numérique par contrôleur DSP.**
- Jean-louis AMAT, Gérard YAHIAOUI. **Techniques avancées pour le traitement de l'information. Réseaux de neurones, logique floue, algorithmes génétiques.**
- A. SOUKKOU, A. KHELAF, S. LEULMI. **Supervision neuro-floue a apprentissage génétique d'un PID robuste.**
- Camille PARRA, Bingwei SU, Yves BRIERE, Doel BORDENEUVE GUIBE .**modélisation and instrumentation of a mini UAV : PEGASE. Journées doctorales d'Automatiques, Valenciennes 25, 26,27 juin 2003**
- Norberto Eiji Nawa , Tomonori Hashiyama , Takeshi Fruhashi , Yoshiki Uchikawa . **Fuzzy Logic Controllers Generated by Pseudo-Bacterial Genetic Algorithm with Adaptive Operator.**
- Jon C. Ervin, Sema E. Alptekin. **Fuzzy Logic Control of a Model b Airplane**
- Jelena GODJEVAC. **Idées nettes sur la logique floue. Presses polytechniques et universitaires romandes**
- Jelena GODJEVAC. **Neuro-Fuzzy Controllers, Design and Application . presses polytechniques et universitaires romandes.**
- Jacob Oursland . **The Design and Implementation of a Quadrotor Flight Controller Using the QUEST Algorithm.**
- Philippe Martin Erwan Salaün. **Estimation d'état pour les drones aériens.**

- Manel Moussaoui, Mohammed Ben mohammed, Djokhrab Ala **Eddine système flou pour la navigation autonome avec évitement d'obstacles d'un robot mobile de type tricycle.**
- Alexis FRENOT, Anthony GOSSMANN, Romaric GUILLERM. **Stabilization d'un quadric rotor**
- Thèse de Doctorat : Alexis MOUHINGOU. **Modélisation Dynamique d'un Engin Volant Tenant Compte des Flexibilités Structurelles.** L'Université d'Evry Val d'Essonne Décembre 2007.
- Thèse de Doctorat : Arben Cela, Abdellatif Reama. **Modélisation et interface de contrôle d'une quadri rotor.** École de la chambre de commerce et d'industrie de paris
- Projet de fin d'études CORIOLIS .école centrale d'électronique PARIS.
- Liens internet utiles : vous trouverez toutes les informations sur le matérielles utilisés
<http://www.sparkfun.com/>
<http://www.sparkfun.com/tutorials/94>
<http://www.hobbycity.com/>
<http://www.lextronic.fr/>
<http://www.arduino.cc/>
- Vous pouvez télécharger le logiciel Fudge, le fichier .h et un code C comme exemple depuis ce lien :
<http://pobot.org/Robot-Flou.html>