



2009

# Rapport Stage

## Modélisation d'un propulseur Drone

Réalisé par :

**Natalia MENDOZA**

Tuteuré par :

**Pascal ACCO**

---

INSA Toulouse –Automatique Electronique

---

## **Résumé**

C'est document fait le Bilan de mon Projet de Stage d'étude à l'INSA de Toulouse en filière Génie Electronique Informatique option Automatique Electronique.

L'objectif principal était de construire la Modélisation d'un Propulseur Drone. Ce projet est inclus dans un projet de plus haut niveau devant également dimensionner le drone. Ce projet doit aboutir à un système autonome, léger stable et robuste.

# Sommaire

<b>1.Presentation du projet .....</b>	<b>4</b>
<b>2. Introduction .....</b>	<b>4</b>
<b>3. Les Drones .....</b>	<b>5</b>
<b>3.1Principe de vol d'un quadrirotor.....</b>	<b>6</b>
<b>4. Modélisatio .....</b>	<b>7</b>
<b>4.1 Etude statique d'un propulseur drone .....</b>	<b>7</b>
<b>4.1.1.-Calcule de K (coefficient de poussez) .....</b>	<b>7</b>
<b>4.1.2.-Calcule de Kphi (gain magnétique du moteur) .....</b>	<b>8</b>
<b>4.1.3.-Calcule d'Alpha (coefficient de traînée) .....</b>	<b>9</b>
<b>4.2 Etude dynamique du drone .....</b>	<b>10</b>
<b>4.3 Fonction de transfert statique .....</b>	<b>11</b>
<b>5. Matlab Modélisation et simulation du Drone birotor implémenté avec deux pales .....</b>	<b>13</b>
<b>5.1 Modèle Global du Drone .....</b>	<b>13</b>
<b>5.2 Modèle du moteurs dans le Drone .....</b>	<b>14</b>
<b>6. Annexe .....</b>	<b>15</b>
<b>7.Bibliographie .....</b>	<b>21</b>

# **Modélisation d'un propulseur Drone**

## **1.- Présentation du Projet**

Vous découvrirez alors dans ce rapport, un modèle dynamique caractérisé pour des conditions nécessaire par sa stabilité, d'abord, d'un Drone à quatre rotor qui a été modifié et est devenu un Drone à deux rotor opposé et deux pales opposé.

Cette modification à fait que la valeur des différents paramètres, les coefficients changent. Pour conséquence nous avons dans ce projet déterminé chaque valeur de ces coefficients et pour cela nous avons travaillé les modélisations sur un de propulseur du Drone à deux rotors et deux pales pour avoir de données.

Mais avant d'arriver, nous vous présenterons une brève explication du Drone, le principe de vol d'un quadrirotor, en suite la modélisation en statique et dynamique du système et la réponse obtenu puis de ces calculs, modélisé et simulé sur Matlab.

Vous trouverez en annexe : des plans de construction, des schémas électroniques, des programmes informatique pour les divers calculs, et des calculs théorique de linéarisations des équations.

## **2.- Introduction**

Les drones sont des aéronefs capables de voler et d'effectuer une mission sans présence humaine à bord. D'origine anglaise, le mot «drone», qui signifie «bourdon», ou «bourdonnement», est communément employé en Français en référence au bruit que font certains d'entre eux en volant.

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».

Le principe des drones peut être rapproché, toutes proportions égales par ailleurs, de celui de l'aéromodélisme, selon lequel des petites maquettes sont pilotées par télécommandes.

### 3.- Les Drones

La plupart des drones sont comparables aux avions (voir Figure (1) et (2)), sauf que leur forme n'est pas dictée par celle d'un fuselage devant abriter au moins un pilote (de façon confortable) : les combinaisons de formules aérodynamiques et de propulsion sont donc plus larges. Il existe de nombreuses configurations de drones, très différentes les uns 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.



Figure (1)



Figure (2)

Nous vous parlerons du principe de fonctionnement d'un Drone quadrirotor et sa transformation à Drone birotor implémenté avec deux pales.

#### 3.1 Principe de vol d'un quadrirotor :

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.

-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.

-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). (Voir Figure (3) et (4)).

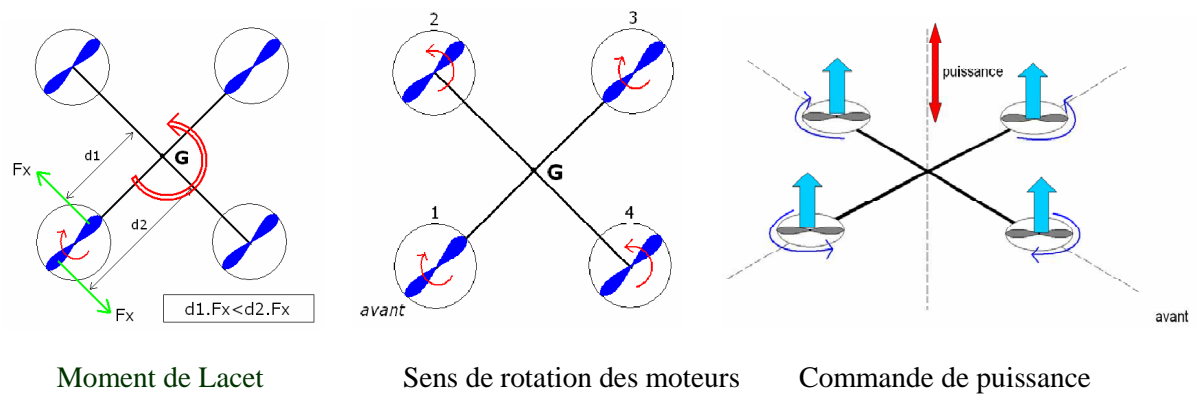


Figure (3)

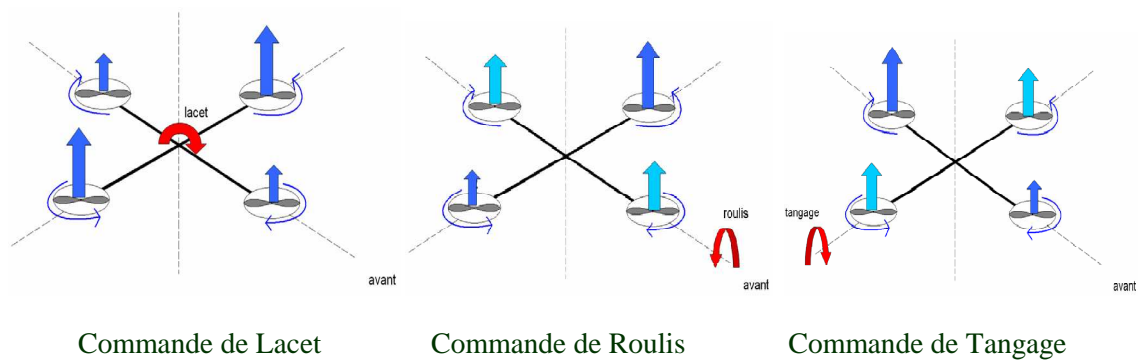


Figure (4)

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).

Nous venons de voir le principe de vol d'un drone à 4 rotors, et maintenant nous comprenons son fonctionnement.

En utilisant cette idée nous avons changé la structure de ce drone pour un autre de seulement deux rotors et à la place des autres deux nous avons mis deux pales. (Voir Figure (5))



Figure (5)

Cette nouvelle structure change les valeurs des coefficients statique et dynamique qui a en principe le drone à 4 rotor, pour cela dans ce projet nous avons pris un propulseur de cette structure pour pouvoir

ainsi modéliser et recalculer les valeurs des coefficients de poussé, traînée, gain magnétique du moteur, l'inertie du rotor, et la valeur de la bobine du moteur.

Nous avons fait des essayes pour pouvoir obtenir de façon expérimental, données, qu'après nous avons utilisé en la modélisation.

#### 4.- Modélisation

Pour pouvoir obtenir ces nouvelles valeurs des coefficients nous avons divisé l'étude en deux parties, statique et dynamique.

##### 4.1 Etude statique d'un propulseur drone :

Pour cette modélisation nous avons pris un propulseur drone qui constat d'un moteur DC et une pale, et nous avons fait les suivants expériences :

Nous avons donnée un courant au propulseur qui a été posé sur une balance qui nous a indiqué la force de poussez, aussi, nous avons pu mesurer la vitesse de rotation de la pale.

- Dans cette partie nous allons calculer les suivants coefficients : K, Kphi, Alpha.

Pour cela, de façon expérimentale nous avons obtenue les données suivantes :

Voltage (V)	I (A)	Fréquence (Hz)	Pois (gr)	W0 (rad/seg)
1	0.56	13.02	4	40.9
1.5	0.72	22.93	10	72.03
2	0.92	33.33	12	104.7
2.5	1.12	42.37	22	133.10
3	1.37	56.81	36	178.47
3.5	1.58	64.1	50	201.37
4	1.92	76.92	72	241.65
4.5	2.33	87.72	96	275.58
5	2.7	94.34	118	296.37
5.5	3.15	106.38	148	334.20
6	3.57	114.94	170	361.09
7	4.53	129.87	222	407.99
8	5.94	147.06	284	462.00
9	6.9	156.25	312	490.87
Pois Moteur + hélice = 110gr				
R = 0.4 $\Omega$				
L = 0.05mH				

#### 4.1.1.-Calcul de K (coefficient de poussez) :

La force générée par l'ensemble moteur et hélice est de la forme : (voir Figure (6)).

$w_0$  = vitesse de rotation du moteur

$F_p$  = Force du poussez

$$F_p = K(w_0.^2)$$

A pleine puissance, le moteur a une vitesse de rotation de rad/seg et peuvent soulever une quantité de masse en grammes.

$$K = F_p / (w_0.^2) \text{ gr.s}^2/\text{rad}$$

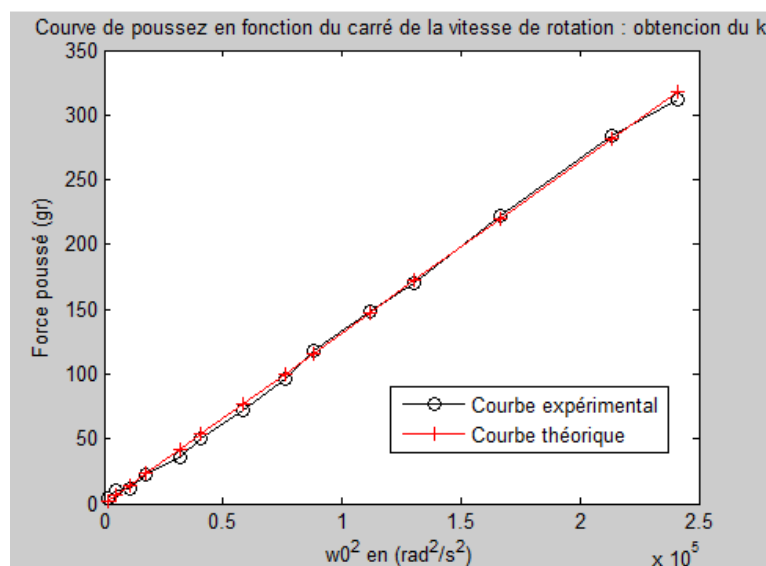


Figure (6)

La valeur de K obtenue est :  $K = 0.0013$

#### 4.1.2.-Calcul de Kphi (gain magnétique du moteur) : (voir Figure (7)).

$$V = Res \times I + Kphi \times w_0.$$

$$Kphi = \frac{(V - Res \times I)}{w_0.}$$



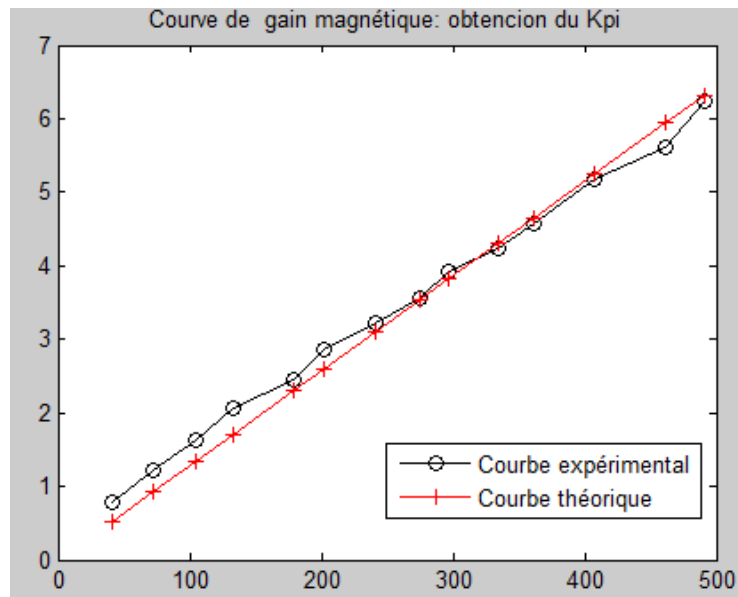


Figure (7)

$$K\phi = 0.0129$$

4.1.3.-Calcul de Alpha (coefficient de traînée) : (voir Figure (8)).

$$\alpha = \frac{(K\phi \times I)}{(w_0^2)} \text{ gr.cm.s}^2 / \text{rad}^2$$

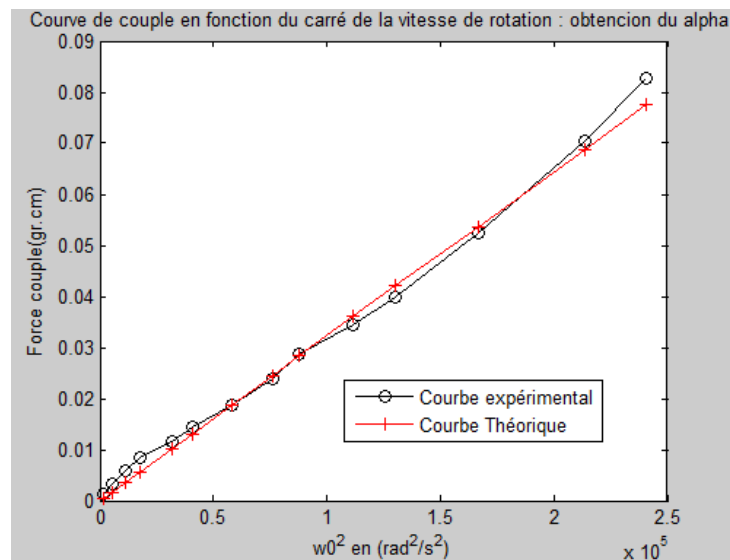


Figure (8)

$$\alpha = 3.2249e-007$$

Nous avons mis les données et la formules sur Matlab et nous avons observé que les courbes théoriques et expérimentales nous donnent des bons résultats assez proche l'un de l'autre.

## 4.2 Etude dynamique du drone

Dans cette partie nous allons calculer les coefficients J et L des équations qui régulent le moteur d'un propulseur drone. (Voir les équations associées à ces deux coefficients et sa résolution dans l'annexe (A)).

Pour cette étude dynamique nous avons mis un moteur du propulseur drone en série avec une résistance de valeur égal à  $0.576\Omega$ . Comme signal d'entrée nous avons donnée un échelon de 1V. Puis nous avons mesuré et enregistré le voltage sur la résistance par différentes valeurs de signal d'entrée pour ainsi avoir une donnée dans le temps de la courant.

Avec l'aide de Matlab nous avons pu faire de calculs et avoir ainsi la courant qui passé pour le circuit. (Voir Figure (9)).

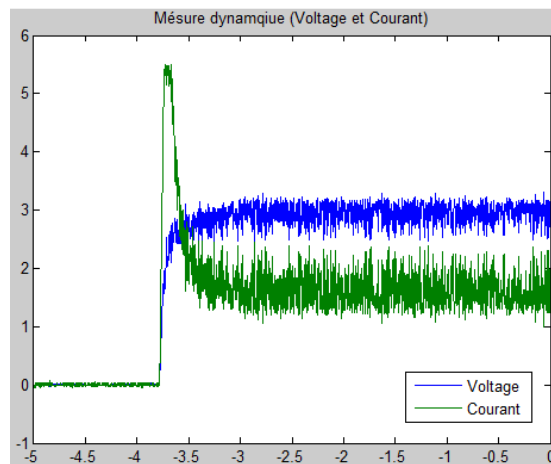


Figure (9)

Moyen statique entre le courant et le voltage : (voir Figure (10)).

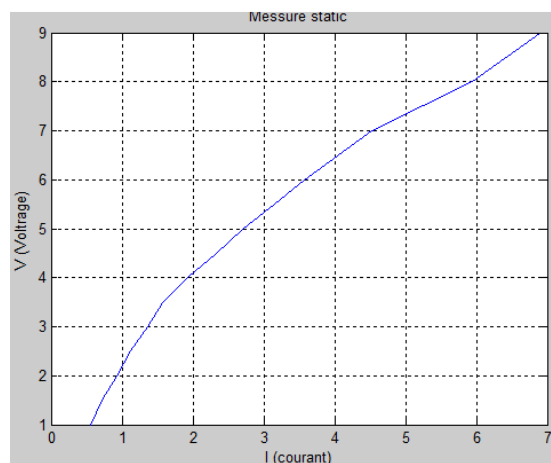


Figure (10)

### 4.3 Fonction de transfert statique :

Si :  $S \rightarrow 0$  Dans l'équation (0.5) de l'annexe (A) nous auront le résultat suivant : (voir Figure (11)).

$$FT_{statique} = \frac{w0.(2\alpha)}{w0.(0.8\alpha) + K\phi i^2}$$

$$I_{statique} = V \times FT_{statique}$$

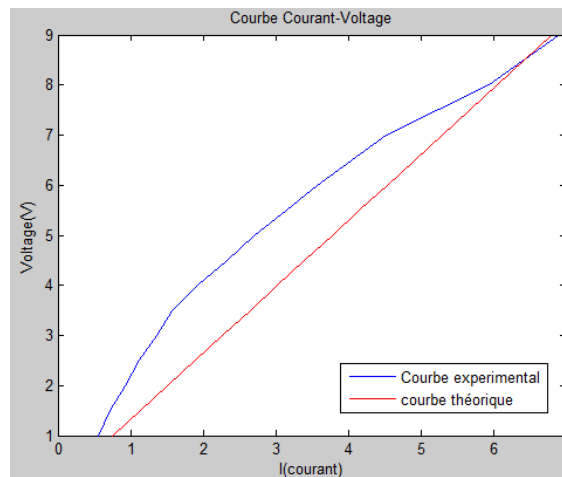


Figure (11)

Pour pouvoir obtenir un meilleur résultat, avec l'aide de ident (boite à outils qui propose de nombreuses fonctions qui permettent d'extraire un ou des modèles à partir de différents campagnes de mesures. (voir Figure (12)).) de Matlab nous allons obtenir une expression mathématique que nous utiliserons pour comparer ce résultat avec l'équation (0.5) de l'annexe (A) et pouvoir ainsi déterminer le valeur de J et L.

Mise au format "ident" des vecteurs d'entrée input (Voltage(V)) et output (Idynamique).

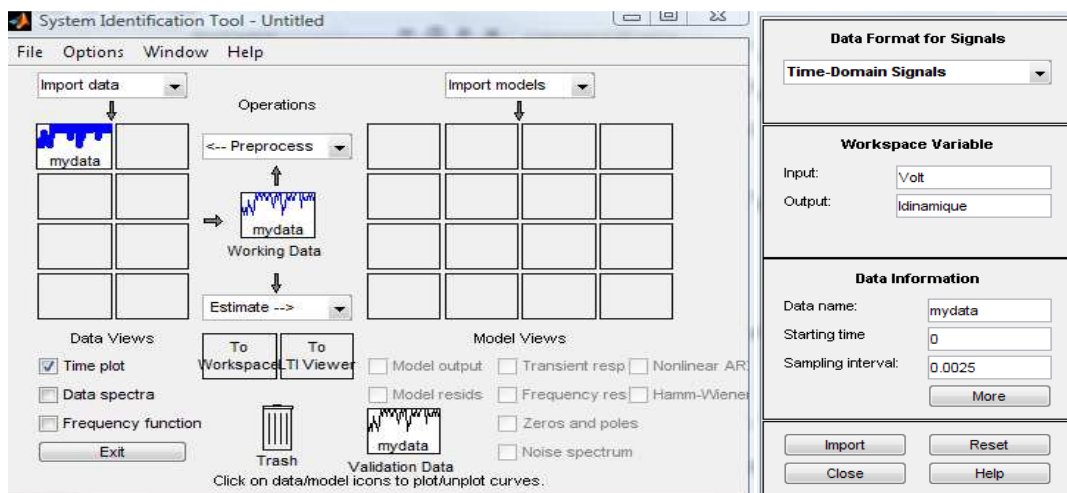


Figure (12)

Signal d'entre (Voltage) et sortie (Courant). (Voir Figure (13)).

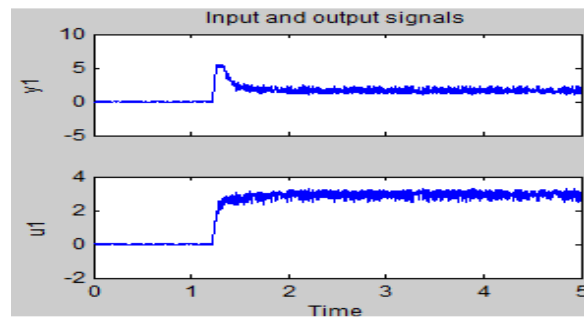


Figure (13)

Nous avons appliqué le model au processus : (Voir Figure (14)).

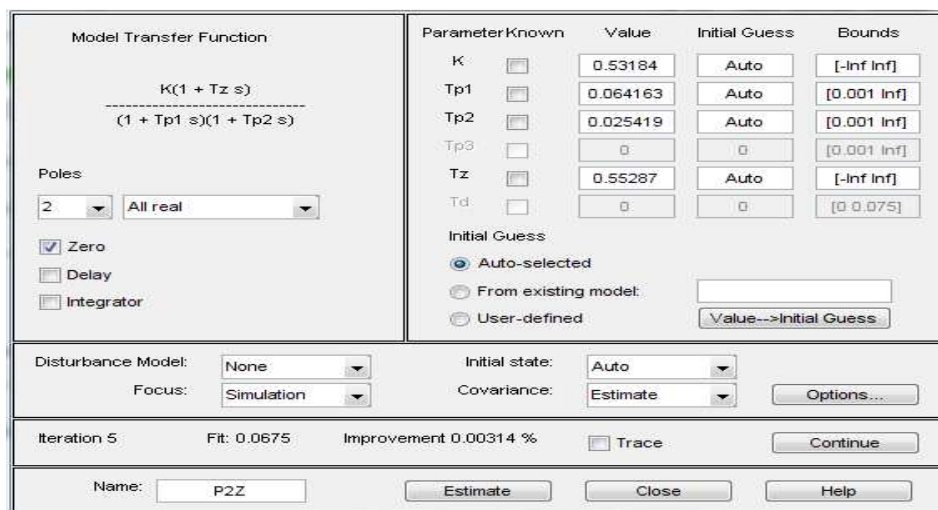


Figure (14)

Il apparaît un "taux de recouvrement" exprimé en 67.94% qui indique la bonne adéquation des courbes. (Voir Figure (15)).

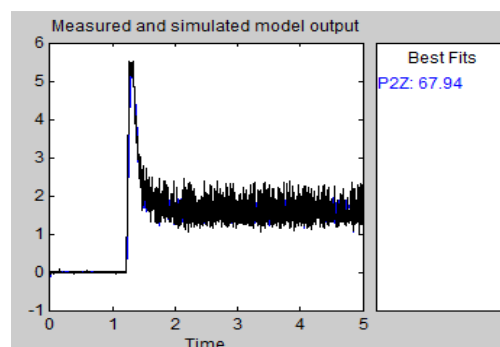


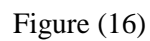
Figure (15)

Nous avons pris le moyen de toutes les valeurs obtenues et les coefficients J et L ont le résultat suivant :

$$J = 5.39645e-005$$

$$L = 0.0075 H$$

### 5.1 Modèle Global du Drone : (Voir Figure (16)).



Des évolutions dans le temps de la vitesse, l'accélération, et la position. (Voir Figure (17)).



Figure (17)

## 5.2 Modèle du moteurs dans le Drone : (Voir Figure (18)).

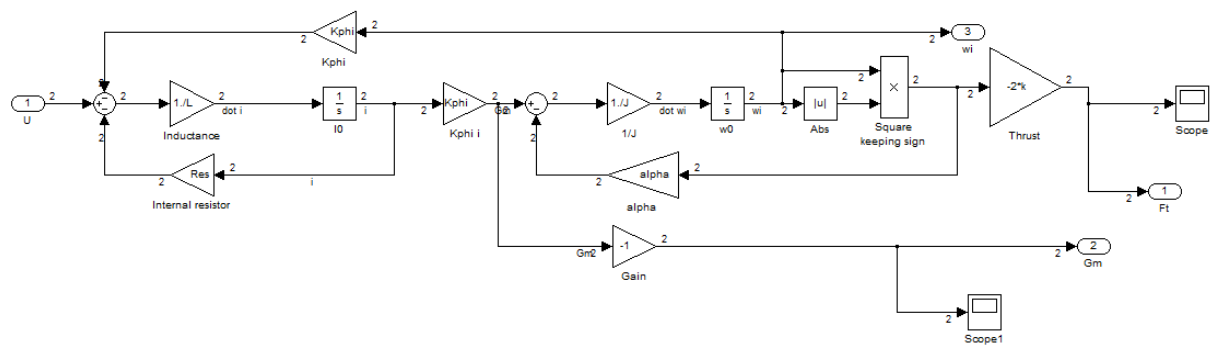


Figure (18)

La réponse des forces de pousser dans les circuits de moteurs : (Voir Figure (19)).

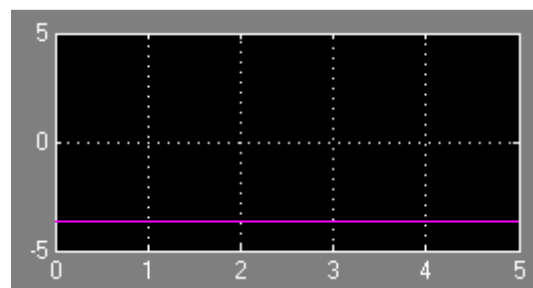


Figure (19)

## 6.- Annexe

### Annexe B

Programme général en Matlab de modèle d'un Drone deux moteur et deux pales :

```
clear all
close all
% This script initialises the drone.mdl Simulink model to zero
% (ground, no angle, etc.)

% Number of engines
nbe = 2;

disp(['Initialising Drone at ground, all zero, for ', num2str(nbe), ...
    ' engines...']);

% General parameters

Mb = 0.72; % Body mass
Mp = 0.01 * [1,1]; % Propeller mass
Mm = 0.04 * [1,1]; % Motor mass
l = 0.10 * [1,1]; % Propeller radius
lm = 0.05 * [1,1]; % Motor radius
hm = 0.05 * [1,1]; % Motor heights
direction = [1,1]; % Engine rotation directions

% Weight
M = Mb + sum(Mp) + sum(Mm); % Total mass
g = 9.81; % Gravity constant

%Wz velocidad angular initial

%Wz = w0 * (sqrt(alpha/aplphapa));
%Om0 = [0,0,Wz]; % Drone's angular rotation speed

% Initial conditions

v0 = [0,0,0]; % Drone velocity
x0 = [0,0,0]; % Drone position
E0 = [0,0,0]; % Drone angle
%Om0 = [0,0,Wz]; % Drone's angular rotation speed

% Engine parameters
% Other calculated parameters
J = (Mp .* l.^2) ./ 2; % Propeller inertia
Kphi = 0.24; % Magnetic gain
L = 100e-3 * [1,1]; % Inductance
Res = 0.1 * [1,1]; % Internal resistance

%wnom = 200*2*pi/60 * [1,1]; % Nominal rotation
f = 10 * [1,1]; % Lift-to-drag coefficient
k = 0.0039 .* direction; % coef. de elevacion fuerza
```

```

alpha = 1./f.*abs(k); % Drag coefficient

% Pales parameters
fpa = 12;
Kpa = 3*k(1);
alphapa = 1./fpa.*abs(Kpa); % el wnom para la fuerza inicial de 2 motores

%wnom = sqrt((M*g) ./ (2.*k));

% Position of engines from center of gravity of the model (in the mobile
frame)
r = 0.20; % Distance between an engine and the
center % of gravity

%
%      | GP1 | GP3 |
GP = [ [ r , -r ] ; ...
      [ 0 , 0 ] ; ...
      [ 0 , 0 ] ];
%      | GP1 | GP3 |

% Body inertia matrix
Ib = [ [ Mb*r^2/6 , 0 , 0 ] ; ...
      [ 0 , Mb*r^2/6 , 0 ] ; ...
      [ 0 , 0 , Mb*r^2/3 ] ];

% Engines inertia matrices
Iei = {};
for i=1:nbe
    % Propeller inertia at Pi
    Ip = [ [ Mp(i)*l(i)^2/4 , 0 , 0 ] ; ...
          [ 0 , Mp(i)*l(i)^2/4 , 0 ] ; ...
          [ 0 , 0 , Mp(i)*l(i)^2/2 ] ];

    % Motor inertia at Pi
    lm2 = lm(i)^2; hm2 = hm(i)^2;
    Im = [ [ Mm(i)*(lm2/4+hm2/12) , 0 , 0 ] ;
    ...
          [ 0 , Mm(i)*(lm2/4+hm2/12) , 0 ] ;
    ...
          [ 0 , 0 , Mm(i)*(lm2/2) ] ];

    % Transport to G
    a = GP(1,i); b = GP(2,i); c = GP(3,i);
    T = [ [ (b^2+c^2) , -a*b , -a*c ] ; ...
          [ -a*b , (c^2+a^2) , -b*c ] ; ...
          [ -a*c , -b*c , (a^2+b^2) ] ];

    % Total inertia at G
    Ie = (Ip+Mp(i)*T) + (Im+Mm(i)*T);
    % Add to cell
    Iei = [Iei, { Ie }];
end

% Total inertia
I = Ib; % Body
for i=1:nbe; I=I+Iei{i}; end; % Engines

% Inertia inverse
Iminv = inv(I);

```



```

%% equilibrium values : steady state
% Input for equilibrium (To counter gravity)

%el w0 necesario para aplicar a 2 motores c/u
%w0 = wnom;

w0 = sqrt( (M*g)/ (2*(k(1) + ((Kpa(1)/alphapa(1))* alpha(1)))));
I0 = w0.^2.*abs(alpha)./Kphi;
u0 = direction .* (Res.*I0+Kphi.*abs(w0));

%Wz velocidad angular initial

Wz0 = w0 * (sqrt(alpha(1)/alphapa(1)));
Om0   = [0,0,Wz0];           % Drone's angular rotation speed
%Om0   = [0,0,0];

% initial value for simulation
I_init = I0.*direction;      % Current in motors
w_init = w0.*direction;      % Rotation speed of motors

%% Load simulation
disp(['Preloading simulation...']);
load_system('drone');
disp(['You can now type ''sim drone''.']);
disp(['Initialisation done.']);

```

## Annexe C

Programme général en Matlab pour l'obtention de coefficients K, Kphi, Alpha.

```
clear all;
close all;

% Calcule de coefficient K, Kphi, Alpha, d'un propulseur Drone et sa
fonction de transfert en statique :

%Données obtenues de façon experimental :

%Voltage (V)
V = [1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 7 8 9];

%Courrent I(A)
I=[0.56 0.72 0.92 1.12 1.37 1.58 1.92 2.33 2.7 3.15 3.57 4.53 5.94 6.9];

%Frequence F(Hz)
F= [13.02 22.93 33.33 42.37 56.81 64.1 76.92 87.72 94.34 106.38 114.94
129.87 147.06 156.25];

%Pousse (gr)
Poussez=[4 10 12 22 36 50 72 96 118 148 170 222 284 312];

%Resistance (ohmios)
Res = 0.4 ;

%% identification de K

%Frequence w0(rad/seg)
w0 = pi*F;

Vk = Poussez ./ w0.^2

Polynome=polyfit(w0.^2,Poussez,1);
K=Polynome(1);

figure(1)
plot(w0.^2,Poussez,'k-o'); hold on;
plot(w0.^2, K*w0.^2,'r-+');
title('Courbe de poussez en fonction du carré de la vitesse de rotation :
obtention du k ');
xlabel('w0^2 en (rad^2/s^2)')
ylabel('Force poussez (gr)')

%% identification de Kphi

% V = Res*I + Kpi*w0 ;

Kphicalcule = (V - Res*I)./w0

Pe = fminsearch( @(P) sum(((V - Res*I)-(w0*P(1))).^2) ,
[mean(Kphicalcule)])
```

```

Kphi = Pe(1)

figure(2)

plot(w0,(V - Res*I),'k-o'); hold on;
plot(w0, Kphi*w0,'r-+');

title('Courbe de gain magnetique du moteur: obtencion du Kphi ');

%% identification de Alpha

Alphacalcul = ((Kphi*I)-(0.006*(ones(1,14))))./w0.^2

Ae = fminsearch( @(P) sum((((Kphi*I)-(0.006*(ones(1,14))))-
(w0.^2*P(1))).^2) , [mean(Alphacalcul)])

Alpha = Ae(1)

figure(3)

plot(w0.^2,((Kphi*I)-(0.006*(ones(1,14))))),'k-o'); hold on;
plot(w0.^2, Alpha*w0.^2,'r-+');
title('Courbe de couple en fonction du carré de la vitesse de rotation :
obtencion du Alpha ');
xlabel('w0^2 en (rad^2/s^2)')
ylabel('Force couple(gr.cm)')

%% Fonction de Transfer

FTstatic= ( (w0.*(2*Alpha)) / ( (w0.*(2*Alpha*(0.4))) + Kphi^2 ) )

Ift=V*FTstatic

figure(4)
plot(I,V)
hold on
plot(Ift,V, 'r')

```

## Annexe D

Programme général en Matlab pour l'obtention de coefficients J et L.

```
clear all;
close all;

%Calcul des coefficients J et L:

%coefficients obtenues par Ident:

%Fréquence F(Hz)

F= [13.02 22.93 33.33 42.37 56.81 64.1 76.92 87.72 94.34 106.38 114.94
129.87 147.06 156.25];

%Fréquence w0(rad/seg)

w0 = pi*F;

Kphi= 0.0129;
Alpha = 3.2249e-007

A= ((w0.*(2*Alpha*(0.4))) + Kphi^2 );

Aprom = mean(A);

Tp1= 0.053184;
Tp2= 0.025419;
K= 0.53184;
Tz= 0.55287;

J= Aprom*K*Tz

L= (Tp1*Tp2*Aprom)/J
```

## Bibliographie

- [1] Richard Feynman. Rogers Commission Report on the Space Shuttle Challenger Accident, appendix Personal observations on the reliability of the Shuttle. Rogers Commission, June 1986. Available from: <http://www.ralenz.com/old/space/feynman-report.html>.
- [2] The Object Management Group. Omg sysml [online]. Available from: <http://www.omgsysml.org/>.
- [3] The Object Management Group. Omg unified modelling language [online]. Available from: <http://www.uml.org/>.
- [4] The Object Management Group. Sysml and modelica integration [online]. Available from: <http://www.omgwiki.org/OMGSysML/doku.php>
- [5] Randy Miller. Practical uml: A hands-on introduction for developers [online]. Available from: <http://edn.embarcadero.com/article/31863>.
- [6] Aurélie Souletis. Mini Unmanned Aerial Vehicles Dynamic Model. Technical report, Institut National des Sciences Appliquées de Toulouse, January 2008.
- [7] Tim Weilkiens. System Engineering with SysML/UML. The Morgan Kaufmann /Object Management Group Press, 2007.
- [8] Wikipédia (en) [online]. Available from: [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page).
- [9] Wikipédia (fr) [online]. Available from: <http://fr.wikipedia.org/wiki/Accueil>.