

MATLAB



Institut Polytechnique
Des Sciences Avancées

TP4 – Simulation comportement drone



Modélisation du comportement d'un drone quadri-rotor

Sous la conduite de :

M. Sellami

Par :

Morin Florent – Herrewyn Anne-Laure
Promo 2013 - EP G2

INSTITUT POLYTECHNIQUE DES SCIENCES AVANCEES

7-9, rue Maurice Grandcoing - 94200 IVRY SUR SEINE

Etablissement Privé d'Enseignement Supérieur - SIRET N° 433 695 632 00011 - APE 803

Préambule

Au cours de cette étude nous allons nous intéresser au système de contrôle d'un drone quadri-rotor. Notre étude est basée sur un article de la conférence « *PID vs LQ Control Techniques Applied to an Indoor Micro Quadroter* » (Sendai, Japon ,2004) de Bouabdallah S., Noth A. et Siegwart R. (plus d'informations sur <http://infoscience.epfl.ch/record/97531>).

Au fur et à mesure de l'avancée de notre TP, nous utiliserons les équations explicitées lors de cette conférence.

I) Partie 1 - Simulation du drone

a) Schématisation :

Equations modélisées :

$$(1) \begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi} \left(\frac{I_y - I_z}{I_x} \right) - \frac{J}{I_x} \dot{\theta}\Omega + \frac{l}{I_x} U_1 \\ \ddot{\theta} = \dot{\phi}\dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) + \frac{J}{I_y} \dot{\phi}\Omega + \frac{l}{I_y} U_2 \\ \ddot{\psi} = \dot{\theta}\dot{\phi} \left(\frac{I_x - I_y}{I_z} \right) + \frac{l}{I_z} U_3 \end{cases}$$

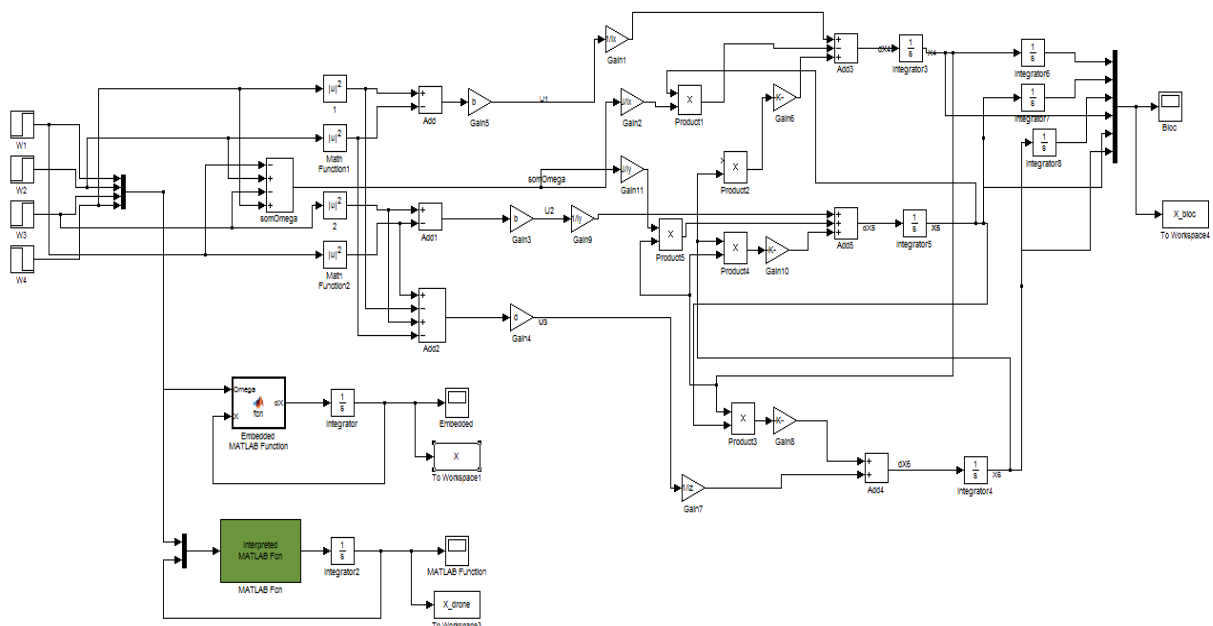
$$(2) \begin{cases} U_1 = b(\Omega_4^2 - \Omega_2^2) \\ U_2 = b(\Omega_3^2 - \Omega_1^2) \\ U_3 = d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \\ \Omega = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3 \end{cases}$$

A partir de ces équations, on peut facilement réaliser un modèle sous Simulink. Néanmoins, il existe trois procédés différents, pour obtenir le même résultat.

- Une version en utilisant uniquement des blocs Simulink tel que Gain, Sum, Product, Integrator, etc... ;
- Une version uniquement avec un bloc Embedded MATLAB function et un bloc integrator ;
- Ou encore une version avec un bloc MATLAB function.

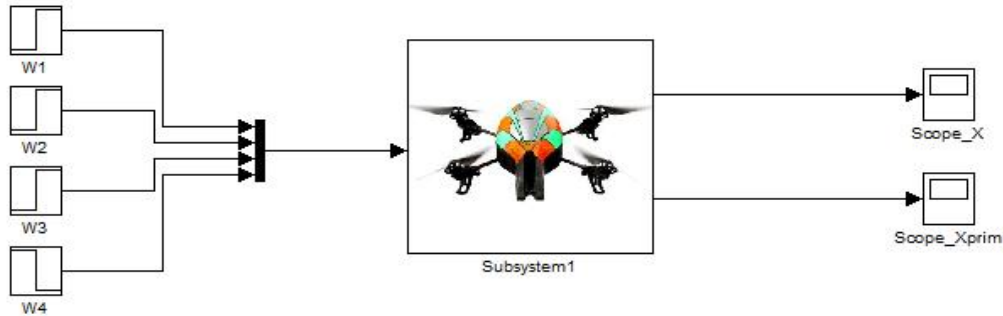
Les entrées des différentes versions sont les mêmes. La sortie de chaque version doit être théoriquement la même, les courbes de sortie doivent être les mêmes.

Après schématisation, on obtient donc le câblage suivant :



Remarque : Le fichier .m du bloc MATLAB Function est « drone.m » fournit avec ce compte-rendu.

On peut mettre la version réalisé uniquement qu'avec des blocs dans un unique subsystem.



Désormais, il est facile de modéliser le comportement de notre quadri-rotor, en modifiant les vitesses de rotations de chacun de ses moteurs, pour lui faire réaliser un mouvement souhaité. Les mouvements possibles sont les suivants :

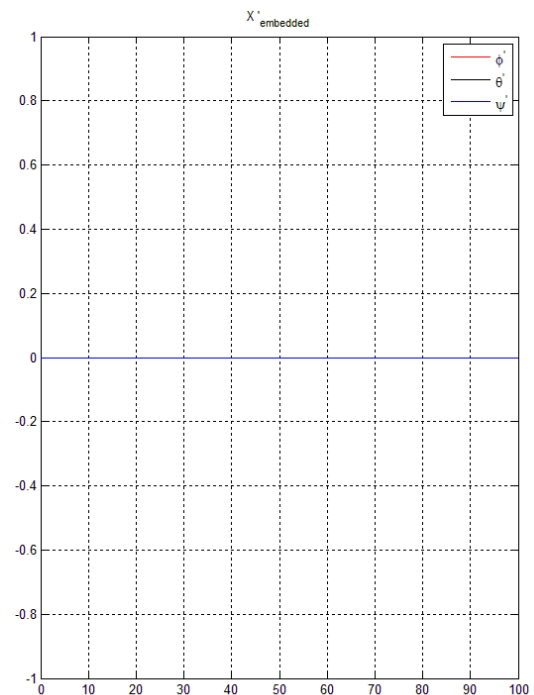
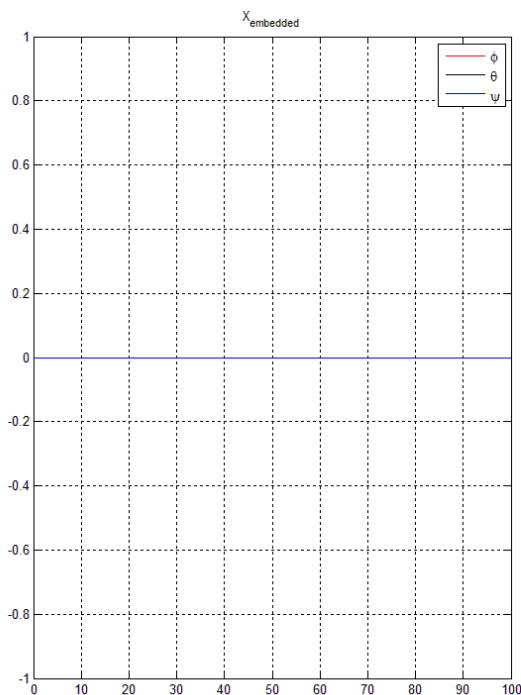
- Aucun mouvement : On ne touche à aucun paramètre ;
- Mouvement de lacet : Augmentation de W1/W3 et diminution de W2/W4 ;
- Mouvement de roulis : Augmentation de W4 et diminution de W2 ou inversement ;
- Mouvement de tangage : Augmentation de W1 et diminution de W3 ou inversement

b) Première simulation :

Nous allons donc débuter par la simulation où les vitesses de rotations sont constantes.

On pose :

$$\Omega_2 = \Omega_4 = \Omega_1 = \Omega_3 = 150 \text{ rad/s}$$



On remarque donc que, si on impose quatre vitesses de rotations de la même valeur, le quadri-rotor reste stable, et aucune perturbation n'apparaît. Néanmoins, on peut affirmer que le drone ne pourra pas voler dans ces conditions. En effet, si on effectue le calcul de la portance fournie par les quatre rotors, et qu'on le compare au poids total du système, on remarque qu'il ne peut pas décoller.

Vérification : Force de Portance (F_t) = $\sum_1^4 F_i$ avec $F_i = b \cdot \Omega_i$

On a donc dans ce cas-là :

$$F_i = 150 \cdot b = 150 \times 3,13 \times 10^{-5}$$
$$F_i = 4,695 \times 10^{-3} N$$

D'où :

$$F_t = \sum_1^4 F_i = 18,78 \times 10^{-3} N$$

Or la masse totale du quadri-rotor, moteur compris est de :

$$M_t = 4m_{motor} + m_{gearbox} + m_{propeller} + m_{OS4}$$
$$M_t = 0,7102 \text{ kg}$$

D'où le poids total :

$$P = 0,7102 \times 9,81 = 6.967062 \text{ N}$$

On peut donc affirmer que le drone ne pourra s'élever :

$$P > F_t$$

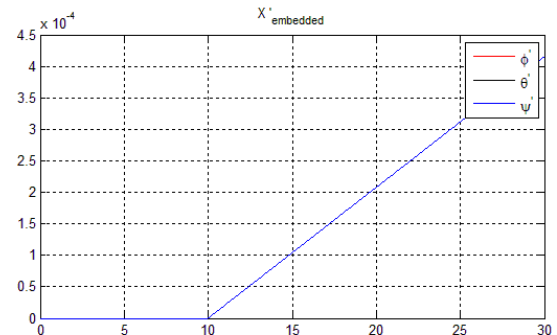
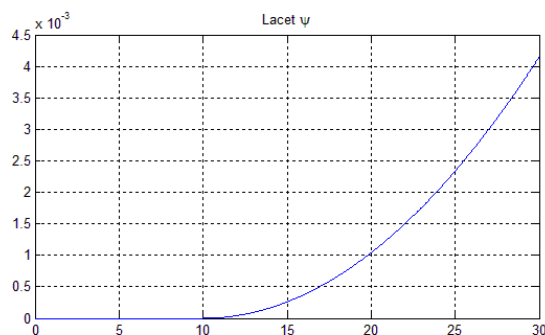
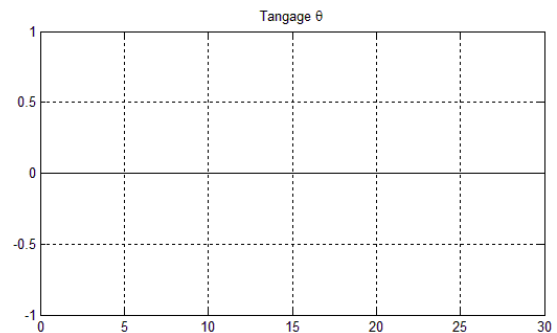
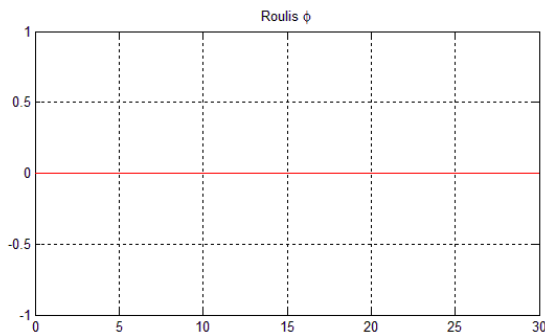
c) Deuxième simulation

Mouvement de lacet :

On simule le mouvement de lacet ainsi :

```
om=(150/500)^2;  
w1=sqrt((150^2)+om);  
w2=sqrt((150^2)-om);  
w3=sqrt((150^2)+om);  
w4=sqrt((150^2)-om);
```

On obtient les courbes suivantes :



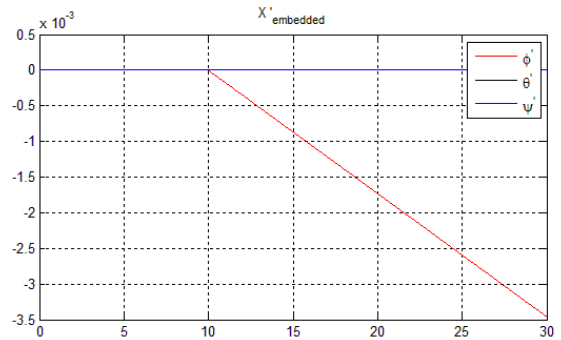
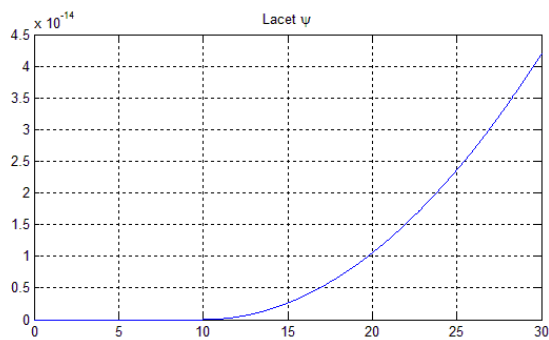
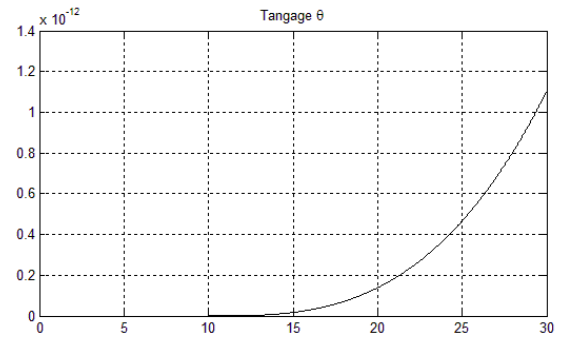
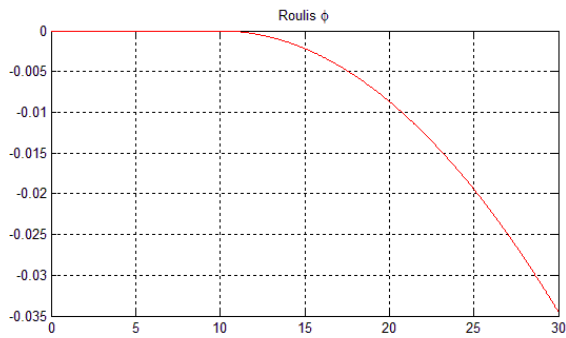
On remarque que le mouvement en lacet est stable. Pour un tel mouvement, seul l'angle de lacet est modifié. Le tangage ainsi que le roulis reste à 0.

Mouvement de roulis :

On simule le mouvement de roulis ainsi :

```
om=(150/500)^2;  
w1=sqrt((150^2));  
w2=sqrt((150^2)-om);  
w3=sqrt((150^2));  
w4=sqrt((150^2)+om);
```

On obtient les courbes suivantes :



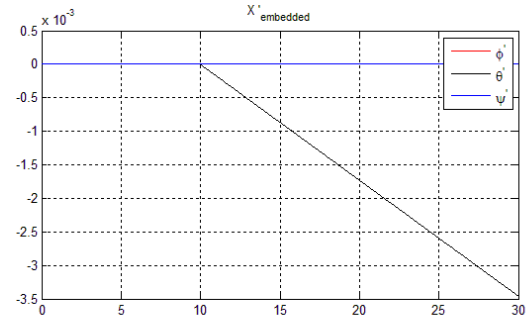
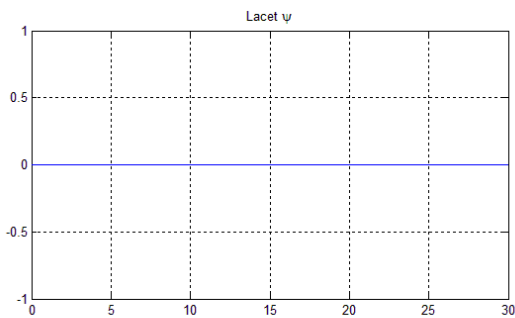
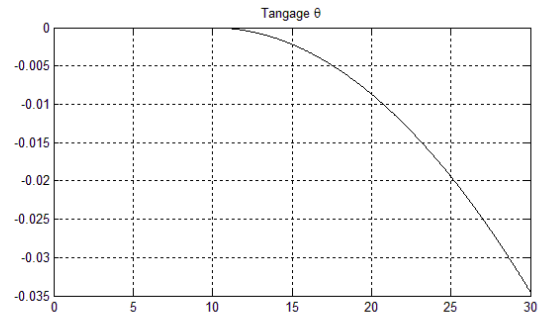
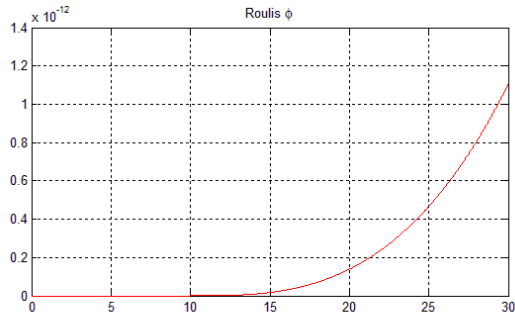
Contrairement au cas précédent, on remarque que les trois angles varient au lieu d'un. Néanmoins, les valeurs des deux angles (tangage et lacet) sont faibles, en 10^{-12} mais croient à l'infini.

Mouvement de tangage :

On simule le mouvement de tangage ainsi :

```
om=(150/500)^2;  
w1=sqrt((150^2)+om);  
w2=sqrt((150^2));  
w3=sqrt((150^2)-om);  
w4=sqrt((150^2));
```

On obtient les courbes suivantes :



L'angle de lacet n'est pas modifié. Cependant, le roulis varie (mais valeur en 10^{-12}). L'angle tangage, comme prévu, varie lui aussi et part à l'infini.

II) Partie 2 - Ajout d'un modèle des moteurs à courant continu :

Dans cette partie, on cherche à modéliser les moteurs du drone. Les entrées de notre schéma ne seront donc plus les vitesses de rotation de chaque hélice, mais les tensions d'entrées de chaque moteur. On relie la vitesse de rotation du moteur à la vitesse de rotation de l'hélice par :

$$\omega_{m,i} = r\Omega_i$$

Avec r : facteur de réduction des moteurs, $\omega_{m,i}$: vitesse de rotation du $i^{\text{ème}}$ moteur et Ω_i : vitesse de rotation de l'hélice associée.

Nos moteurs sont régis par le système d'équation suivant :

$$(3) \quad \begin{cases} L \frac{di}{dt} = u - Ri - k_e \omega_m \\ J \frac{d\omega_m}{dt} = \tau_m - \tau_d \end{cases}$$

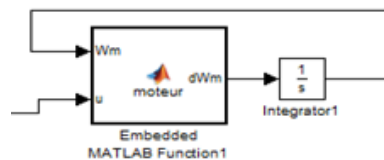
L'inductance de nos moteurs étant très faible, on peut en déduire l'équation suivante :

$$J \frac{d\omega_m}{dt} = -\frac{k_m^2}{R} \omega_m - \tau_d + \frac{k_m}{R} u$$

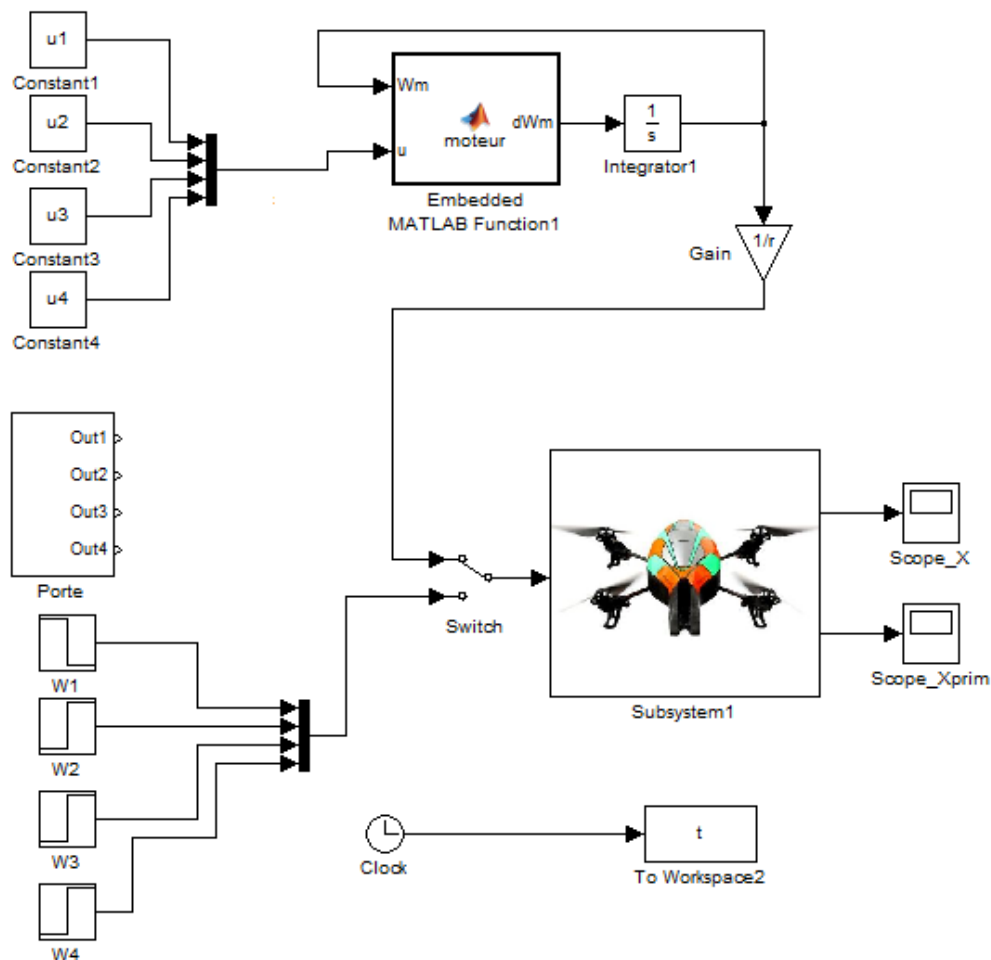
En ajoutant l'équation de l'hélice et de la boîte de vitesse, le système d'équation (3) devient donc :

$$(4) \quad \begin{cases} \dot{\omega}_m = -\frac{1}{\tau} \omega_m - \frac{d}{\eta r^3 J_t} \omega_m^2 + \frac{1}{k_m \tau} u \\ \frac{1}{\tau} = \frac{k_m^2}{R J_t} \end{cases}$$

Ces équations sont facilement modélisables grâce à un bloc Embedded MATLAB Function.

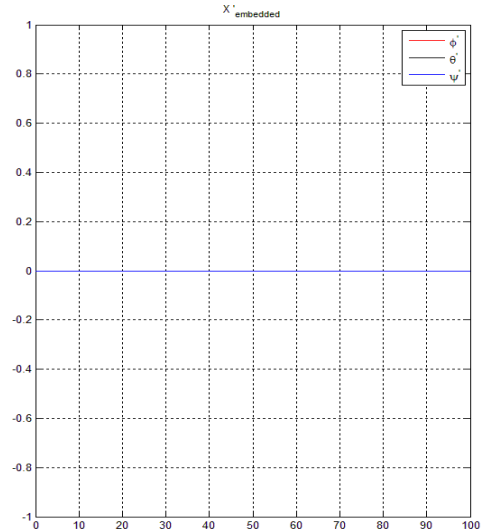
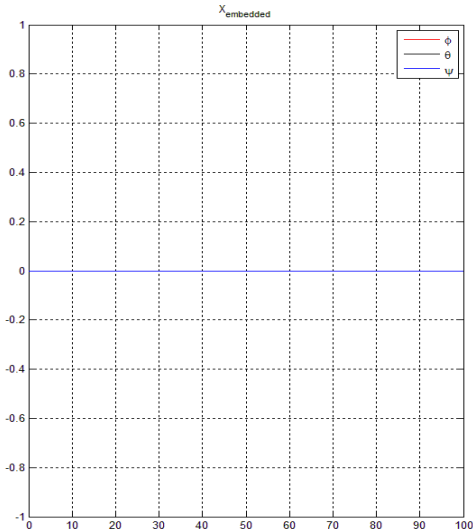


Après intégration de cette partie de la modélisation, on obtient la schématisation finale suivante :



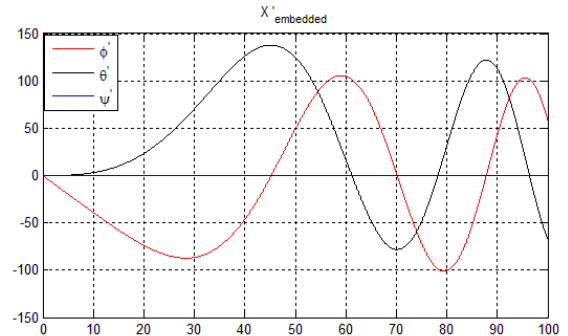
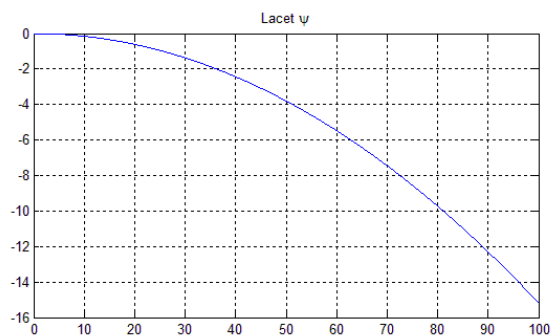
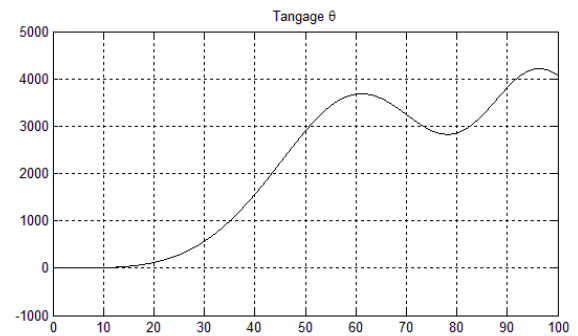
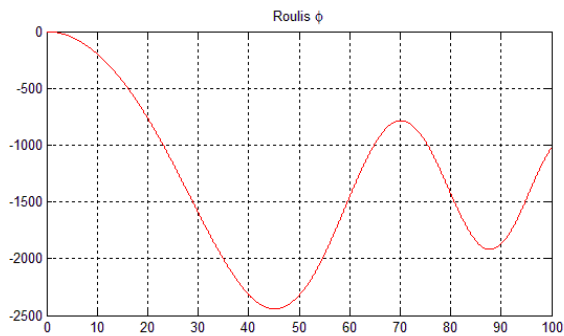
a) Première simulation

On génère une première simulation avec une tension de 4,2 volts pour chacun des moteurs. Pour cela, on utilise un switch pour choisir comme entrée la sortie des moteurs. Le résultat de la simulation est le même que dans la première partie de notre étude. En effet, les tensions étant constantes et toutes égales, les vitesses de rotations des hélices le sont aussi. C'est pourquoi il n'y a aucune variation des différents angles.



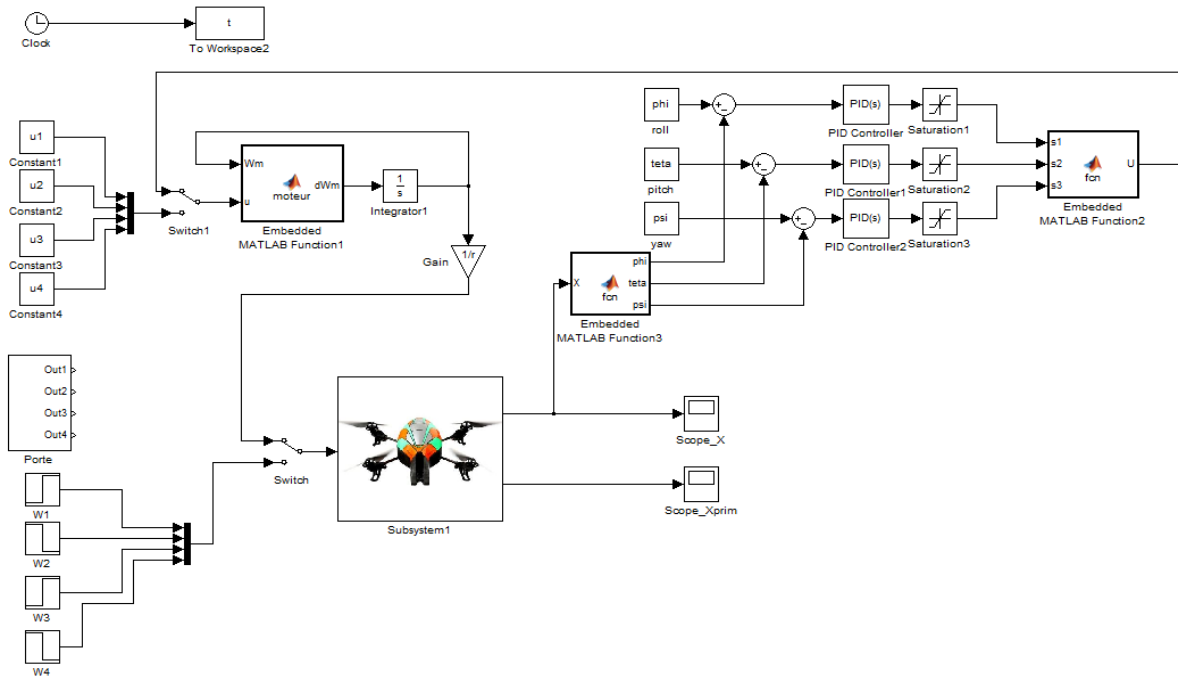
b) Deuxième simulation

On choisit, arbitrairement, de simuler le mouvement de roulis. On note l'instabilité grâce aux oscillations qui perdurent avec le temps.



On remarque qu'il existe un amortissement mais que la valeur de l'angle est stable pour $t \gg 100s$, ce qui n'est pas acceptable pour ce type de système. Un système de régulation est donc nécessaire pour assurer une bonne tenue de vol de ce système.

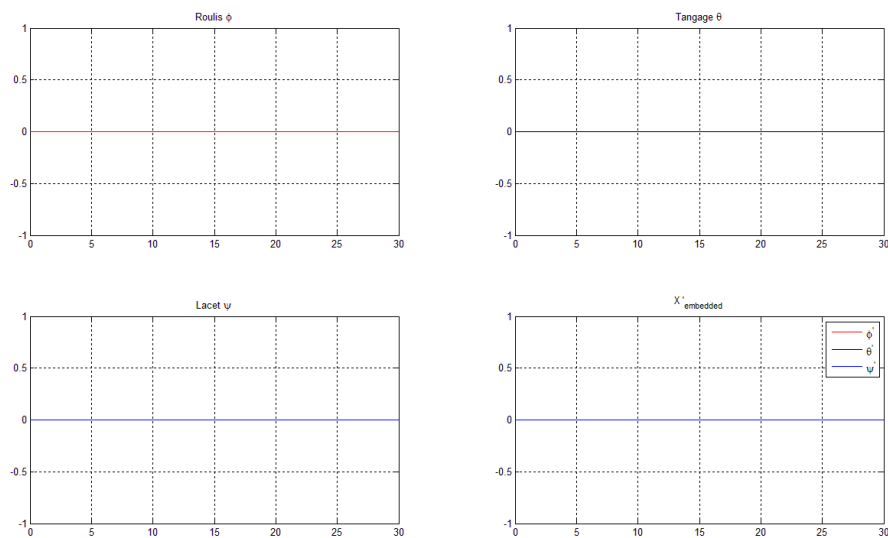
III) Partie 3 – Ajout d'un régulateur PID :



On pose les paramètres K_p , K_i , K_d , avec les valeurs suivantes:

Pour le roulis: $K_{pid1} = [0.9 \ 0.3 \ 2]$
 Pour le tangage: $K_{pid2} = [0.9 \ 0.3 \ 2]$
 Pour le lacet: $K_{pid3} = [10 \ 0.03 \ 20]$

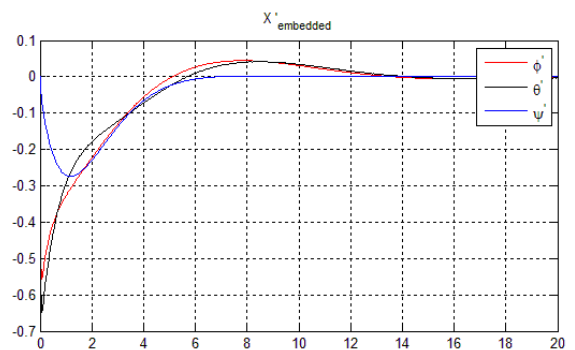
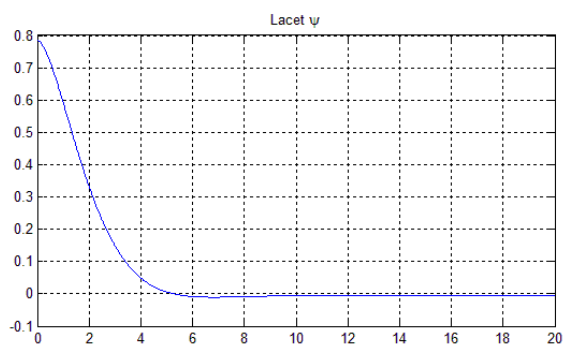
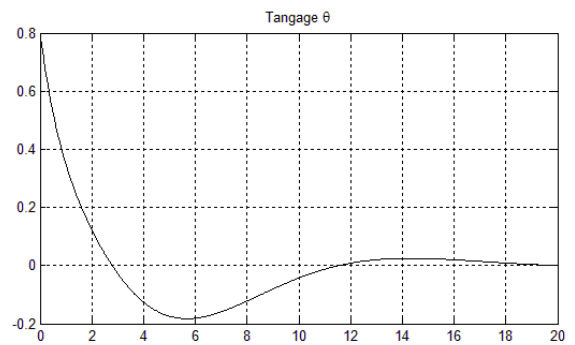
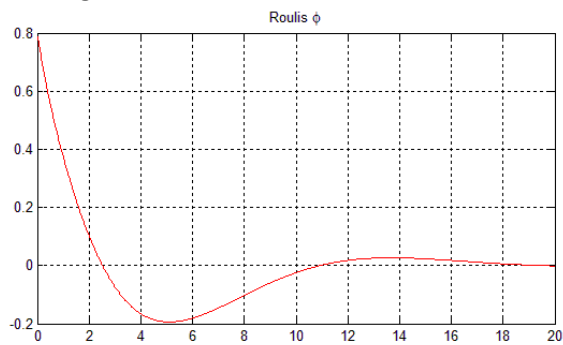
a) Première simulation



Aucune variation, ce qui est encore une fois logique, vu qu'on impose des conditions initiales nulles.

b) Deuxième simulation

On choisit de simuler la variation des angles, en posant $\frac{\pi}{4}$ comme condition initial pour chacun de ces angles. On obtient donc les courbes suivantes :



On remarque donc que les courbes se stabilisent après une dizaine de secondes. Cependant, le tangage met beaucoup plus de temps que les autres à se stabiliser (environ 15 secondes). Ces valeurs restent relativement importantes, pour des angles relativement faibles.

Conclusion :

Notre quadri-rotor est un système régi par plusieurs équations dynamiques. Nous avons réussi à modéliser ces équations de plusieurs manières différentes lors de notre première partie. Néanmoins, nous ne pilotions que les vitesses de rotations de nos hélices. En deuxième partie, nous avons donc pu piloter ces dernières en jouant directement sur les tensions d'entrées des moteurs à courant continu de chaque hélice. Cependant, à chaque essai, pour une variation d'angle donnée, notre système volant était instable. Nous avons donc pu implémenter des PID's ainsi que des blocs de saturations pour stabiliser, dans la mesure du possible, le quadri rotor pour une variation d'angle donnée.