



RAPPORT DE PROJET

HMEE323 : ROBOTIQUE DE SERVICE

Simulation et stabilisation d'un quadrirotor par backstepping

Auteurs :
MM. ARTUSO et FAYE

Tuteur : M. JOUVENCEL
Professeur Université de Montpellier
Équipe EXPLORE (LIRMM)

Année 2015 - 2016

Table des matières

1	Introduction	2
2	Le modèle mathématique	3
3	Le schéma de contrôle sous Simulink	6
4	Mise en place du backstepping	9
5	Simulation du modèle avec backstepping	13
6	Bilan et perspectives	14

Table des figures

1	Schéma quadrirotor et ses axes.	3
2	Schéma de contrôle du quadrirotor.	6
3	Sous-système : modèle de l'effet gyroscopique.	7
4	Sous-système : modèle de l'effet de l'inertie.	7
5	Sous-système : modèle de la poussée et de la traînée.	7
6	Graphiques : altitude instable.	7
7	Structure en cascade du schéma de commande par backstepping.	9
8	Graphique : altitude désirée atteinte et stable.	13

1 Introduction

Dans le cadre d'un projet concernant la Robotique de Service, M. JOUVENCEL, Professeur à l'Université de Montpellier, nous a proposé de réaliser la simulation et la stabilisation par backstepping d'un drone de type «quadrirotor».

Dans notre cas, il est plus judicieux de parler de Robotique Aérienne. En effet, le «quadrirotor» désigne, par abus de langage, un aéronef qui dispose de quatre rotors montés d'hélices pour assurer sa sustentation. Généralement, les rotors sont positionnés en forme de croix autour du corps de l'aéronef, afin de simplifier le modèle et les calculs.

Le contexte est le suivant : dans un espace à trois dimensions, le quadrirotor est commandé grâce aux tensions envoyées sur chacun des moteurs pilotant les rotors. Celles-ci permettent au quadrirotor d'évoluer selon 6 degrés de liberté¹ : 3 pour sa position et 3 pour son inclinaison. Notons que la position est régie par le triplet de coordonnées (X,Y,Z) ² et l'inclinaison par les angles (Roulis, Tangage, Lacet).

Le cahier des charges mis en place par M. JOUVENCEL, est défini autour de deux objectifs à implémenter sous le logiciel «Matlab-Simulink». Le premier est de réaliser le schéma de câblage des différents blocs, équivalent aux équations fournies avec le modèle du quadrirotor.

Ensuite, le second objectif est de mettre en place un système permettant d'aboutir sur la stabilisation du quadrirotor. Pour ce faire, nous utilisons une commande dite «par backstepping» qui est basée sur la théorie de Lyapunov.

1. L'abréviation «DDL» est utilisée par la suite pour Degré De Liberté.

2. En aéronautique, on peut également trouver le triplet (avance, dérive, ascension)

2 Le modèle mathématique

Le modèle mathématique du quadrirotor est nécessaire afin retranscrire les équations liées à sa morphologie sous forme de différents blocs et ainsi réaliser le schéma de contrôle sous Simulink.

Avant toute chose, la notion de repère est primordiale. Pour notre simulation et nos calculs, le quadrirotor est défini dans le repère suivant (Voir Figure 1) :

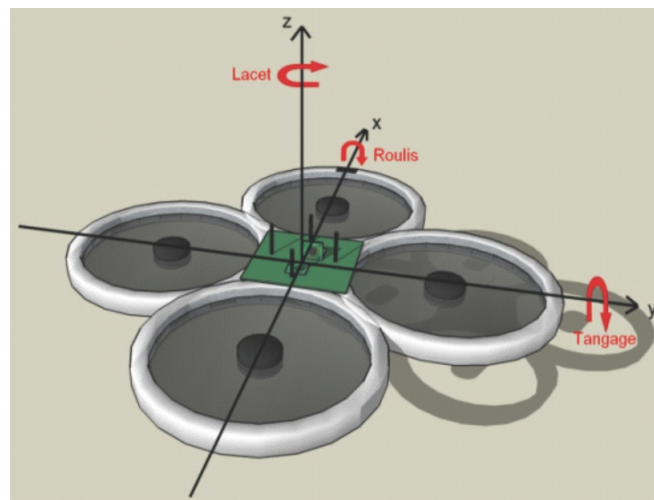


FIGURE 1 – Schéma quadrirotor et ses axes.

Nous avons choisi de positionner l'axe Z vertical dirigé vers le haut puisque nous considérons que le quadrirotor est initialement à l'altitude zéro sur le sol et qu'il doit «s'élever» pour atteindre une altitude positive désirée.

Nous pouvons ainsi agir sur les quatre moteurs pour stabiliser le quadrirotor sur ses différents axes :

- L'axe de roulis X : Pour maintenir l'horizontalité ou «assiette» en terme d'aviation, nous pouvons solliciter les moteurs gauche et droit. Par exemple, si le quadrirotor penche d'un côté, on peut augmenter la vitesse du moteur côté penché et baisser celle du moteur du côté surélevé.
- L'axe de tangage Y : Pour maintenir son inclinaison sur cet axe, les moteurs avant et arrière sont stimulés.
- L'axe de lacet Z : Pour ne pas que le quadrirotor tourne constamment sur lui-même autour de son axe Z, on agit par paire de moteurs : on peut augmenter la vitesse de rotation des deux moteurs face à face (avant/arrière ou droit/gauche) et diminuer celle de l'autre paire.

Le tout premier but est d'atteindre l'altitude désirée et y rester, c'est-à-dire avoir un critère de stabilité sur l'altitude du quadrirotor. Pour ce faire, il faut solliciter la vitesse de rotation de l'ensemble des moteurs.

Le modèle mathématique est régi par les équations de la dynamique permettant de faire évoluer le quadrirotor selon 6 DDL :

- La portance : est générée par les quatre moteurs, elle permet au quadrirotor de s'élever si elle est supérieure à la traînée. Elle est exprimée par les relations :

$$T_x = b.l.(\Omega_4^2 - \Omega_2^2) \quad \text{sur l'axe du roulis X}$$

$$T_y = b.l.(\Omega_3^2 - \Omega_1^2) \quad \text{sur l'axe du tangage Y}$$

avec :

- Ω_i^2 : la vitesse de rotation de chaque moteur au carré $[(rad/s)^2]$
- b : le coefficient de portée $[kg.m/rad^2]$
- l : la demi-envergure du quadrirotor $[m]$

- La traînée : est la résultante des forces de frottement de l'air sur le quadrirotor. Elle est parallèle à la trajectoire mais de sens opposé. Elle est exprimée par la relation :

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

avec :

- d : le coefficient de traînée $[kg.m^2/rad^2]$

- L'effet gyroscopique : est une force qui apparaît lorsque le quadrirotor est en rotation simultanée sur deux axes. On dit généralement avec abus que "deux rotations simultanées = trois rotations", la troisième rotation est en fait induite par les deux autres et s'applique sur le troisième axe. L'effet gyroscopique tend à résister aux mouvements du quadrirotor. son expression s'écrit :

$$T_x = I_{Rotor}.\omega_y.(\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4)$$

$$T_y = I_{Rotor}.\omega_x.(-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4)$$

avec :

- I_{Rotor} : le moment d'inertie du moteur $[kg.m^2]$
- ω_x : la vitesse angulaire selon l'axe X $[rad/s]$
- ω_y : la vitesse angulaire selon l'axe Y $[rad/s]$

Si l'on projette les forces ainsi exprimées et que l'on ajoute l'effet des moments d'inertie sur l'accélération, nous pouvons exprimer les relations comportementales en roulis, tangage et lacet du quadrirotor de la façon suivante :

L'équation de l'accélération angulaire suivant l'axe de roulis X :

$$\ddot{\Phi} = \dot{\theta} \cdot \dot{\Psi} \cdot \frac{I_y - I_z}{I_x} + b.l. \cdot \frac{\Omega_4^2 - \Omega_2^2}{I_x} + \dot{\theta} \cdot I_{Rotor} \cdot \frac{\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4}{I_x}$$

L'équation de l'accélération angulaire suivant l'axe de tangage Y :

$$\ddot{\theta} = \dot{\Psi} \cdot \dot{\Phi} \cdot \frac{I_z - I_x}{I_y} + b.l. \cdot \frac{\Omega_3^2 - \Omega_1^2}{I_y} + \dot{\Phi} \cdot I_{Rotor} \cdot \frac{\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4}{I_y}$$

L'équation de l'accélération angulaire suivant l'axe de lacet Z :

$$\ddot{\Psi} = \dot{\Phi} \cdot \dot{\theta} \cdot \frac{I_x - I_y}{I_z} + d. \cdot \frac{-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2}{I_z}$$

Et les relations comportementales en X, Y et Z telles que :

L'équation de l'accélération suivant l'axe X :

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

L'équation de l'accélération suivant l'axe Y :

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

L'équation de l'accélération suivant l'axe Z :

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

avec :

- T_i : La portance générée par les moteurs $T_i = b \cdot \Omega_i^2$ avec $i \in [1;2;3;4]$ qui représente les moteurs

3 Le schéma de contrôle sous Simulink

Le logiciel Simulink, extension à Matlab, nous permet ici de simuler le modèle du quadrirotor vu dans la section précédente. Il faut définir la position désirée ainsi que son orientation en fin de simulation. Il s'agit ici de nos commandes. Une fois toutes les équations transcrites sous forme de schéma bloc sous Simulink, les commandes imposées et les variables initialisées, la simulation peut alors commencer.

Le schéma de contrôle réalisé est illustré ci-dessous (Voir Figure 2) :

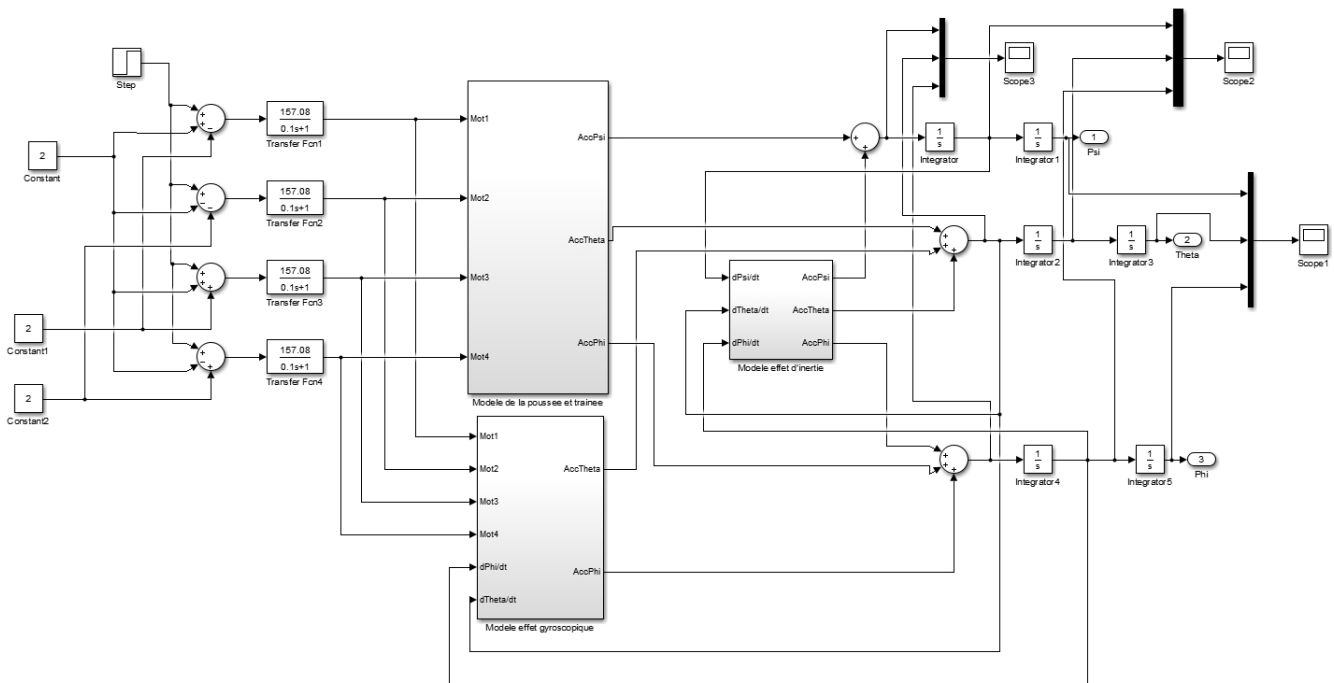


FIGURE 2 – Schéma de contrôle du quadrirotor.

Pour ne pas alourdir le schéma et de ce fait le rendre illisible, les 3 blocs principaux «Modèle de la poussée et de la traînée», «Modèle effet de l'inertie» et «modèle de l'effet gyroscopique» sont composés respectivement de sous-systèmes (Voir Figures 3, 4 et 5).

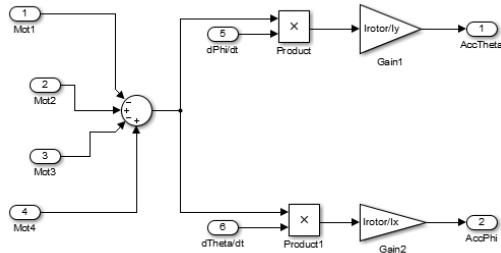


FIGURE 3 – Sous-système : modèle de l'effet gyroscopique.

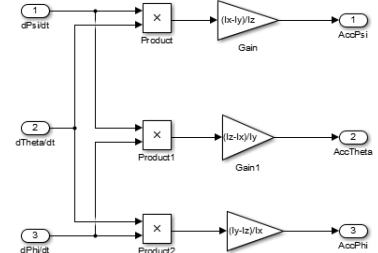


FIGURE 4 – Sous-système : modèle de l'effet de l'inertie.

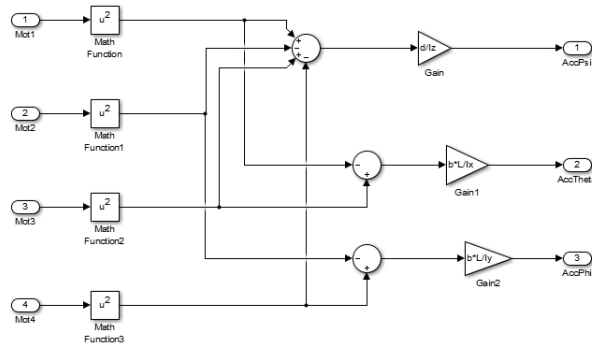


FIGURE 5 – Sous-système : modèle de la poussée et de la traînée.

Mais les résultats en fin de simulation sont sans appel (Voir Figure 6).

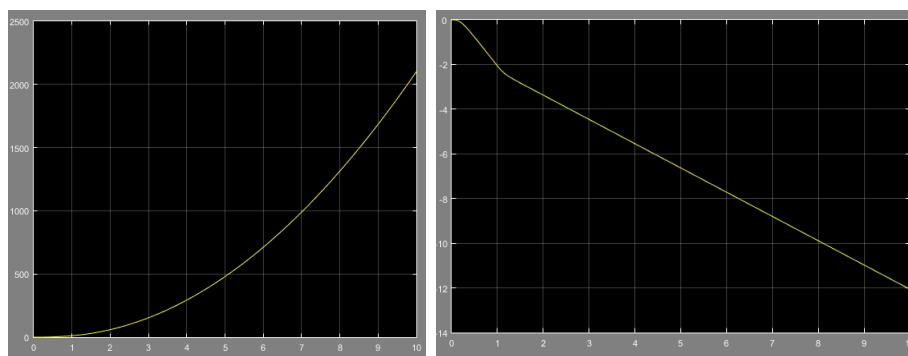


FIGURE 6 – Graphiques : altitude instable.

Le schéma de contrôle, disposant seulement d'une boucle de contre-réaction pour réinjecter les valeurs en sortie sur un comparateur, ne permet pas d'assurer la stabilité du système. Dans le meilleur des cas, le quadrirotor prend de

l'altitude de manière exponentielle, dépassant ainsi l'altitude désirée. Dans un cas plus déplorable, n'ayant pas modélisé le sol, le quadrirotor évolue en altitude négative.

L'instabilité du système peut être expliquée par le fait que nous ne réinjectons seulement les vitesses de déplacement et les vitesses angulaires dans le système donc nous n'utilisons en réalité que la cinématique du modèle. Pour une meilleure stabilité, il faut descendre au niveau inférieur, c'est-à-dire de réinjecter les accélérations calculées et ainsi utiliser la dynamique du modèle.

Au vu de l'instabilité du système et des résultats décevants de la première approche, nous avons procédé à la stabilisation du quadrirotor en imposant au système une commande par backstepping.

4 Mise en place du backstepping

Le «backstepping» est un algorithme de contrôle récursif qui fonctionne en concevant des lois de contrôle intermédiaires pour certaines des variables d'état. Ces variables d'état sont appelées «variables de commande virtuelles» pour le système. Contrairement à d'autres algorithmes de commande qui ont tendance à linéariser les systèmes non-linéaires, le backstepping n'annule pas les non-linéarités du système puisque certains des termes non-linéaires peuvent contribuer à assurer la stabilité du système.

Le principe est le suivant :

- Définir une variable d'erreur.
- Définir une fonction dite de Lyapunov à partir de l'erreur.
- Dériver la fonction de Lyapunov.
- Déterminer la variable de commande virtuelle.
- Poursuivre jusqu'à faire apparaître des variables de commande réelles.

Le backstepping a aussi pour avantage d'une part de posséder une structure en cascade ce qui permet de bien séparer l'action de chaque bloc sur le schéma de commande (Voir Figure 7) et d'autre part de simplifier l'expression de la commande.

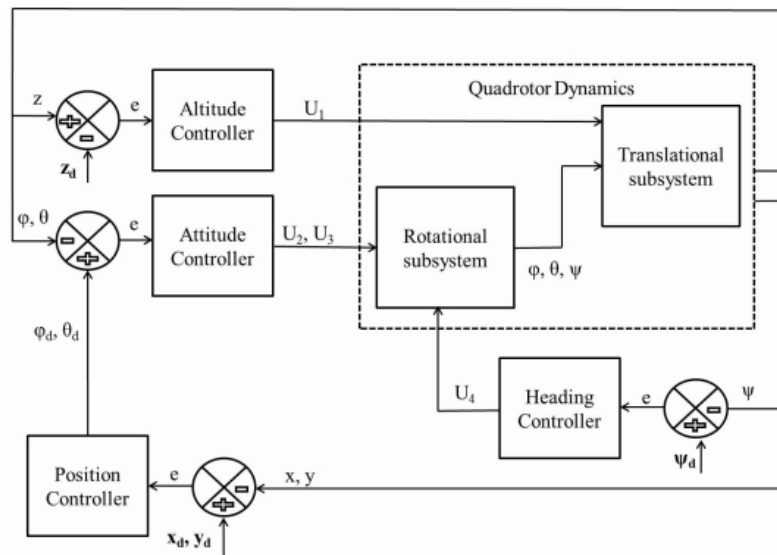


FIGURE 7 – Structure en cascade du schéma de commande par backstepping.

En vue de la mise en place du backstepping, nous avons repris le modèle des accélérations du quadrirotor exprimé dans la littérature.

Pour simplifier les équations à venir, considérons le vecteur d'état X tel que :
 $X = [\Phi \ \dot{\Phi} \ \theta \ \dot{\theta} \ \Psi \ \dot{\Psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}]^T$

Ce qui peut être exprimé par :

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T$$

Notons les variables intermédiaires suivantes :

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}} \quad a_2 = \frac{I_{Rotor}}{I_{xx}} \quad a_3 = \frac{I_{zz} - I_{xx}}{I_{yy}} \quad a_4 = \frac{I_{Rotor}}{I_{yy}}$$

$$a_5 = \frac{I_{xx} - I_{yy}}{I_{zz}} \quad b_1 = \frac{l}{I_{xx}} \quad b_2 = \frac{l}{I_{yy}} \quad b_3 = \frac{l}{I_{zz}}$$

Nous avons ainsi les accélérations angulaires du roulis, tangage et lacet :

$$\ddot{\Phi} = b_1.U_2 - a_2.x_4.\Omega_{Rotor} + a_1.x_4.x_6$$

$$\ddot{\theta} = b_2.U_3 + a_4.x_2.\Omega_{Rotor} + a_3.x_2.x_6$$

$$\ddot{\Psi} = b_3.U_4 + a_5.x_2.x_4$$

Et les accélérations du mouvement selon X, Y et Z :

$$\ddot{x} = \frac{-U_1}{m} . (\sin(x_1) . \sin(x_5) + \cos(x_1) . \cos(x_5) . \sin(x_3))$$

$$\ddot{y} = \frac{-U_1}{m} . (\cos(x_1) . \sin(x_5) . \sin(x_3) - \cos(x_5) . \sin(x_1))$$

$$\ddot{z} = g - \frac{U_1}{m} . (\cos(x_1) . \cos(x_3))$$

L'objectif étant maintenant de calculer la valeur des contrôleurs U_1 (contrôleur de l'altitude), U_2 (contrôleur du roulis), U_3 (contrôleur du tangage) et U_4 (contrôleur du lacet).

A titre d'exemple, nous allons détailler la méthode de calculs pour le contrôleur du tangage U_3 .

Le système est le suivant :

$$\begin{aligned}\dot{x}_3 &= x_4 \\ \dot{x}_4 &= b_2.U_3 + a_4.x_2.\Omega_{Rotor} + a_3.x_2.x_6\end{aligned}$$

Nous définissons l'erreur \mathcal{E}_3 sur x_3 telle que

$$\mathcal{E}_3 = x_{3d} - x_3$$

A partir de laquelle nous exprimons une fonction de Lyapunov candidate :

$$V_3 = \frac{1}{2}.\mathcal{E}_3^2$$

Puis nous la dérivons :

$$\dot{V}_3 = \dot{\mathcal{E}}_3.\mathcal{E}_3 = \mathcal{E}_3.(\dot{x}_{3d} - x_4)$$

Ensuite, il faut créer la fonction de délimitation (ou dite de bornage) :

$$W_3(\mathcal{E}) = -K_3.\mathcal{E}_3^2 \quad \text{avec } K_3 > 0$$

Nous nous intéressons désormais à l'erreur \mathcal{E}_4 sur x_4 telle que

$$\mathcal{E}_4 = x_{4d} - x_4 \quad \text{or} \quad \dot{x}_3 = x_4 \quad \text{d'où : } x_{4d} = \dot{x}_{3d} + K_3.\mathcal{E}_3$$

$$\text{Donc : } \mathcal{E}_4 = x_4 - \dot{x}_{3d} - K_3.\mathcal{E}_3$$

Nous reprenons donc l'expression de la dérivée de la fonction de Lyapunov candidate :

$$\dot{V}_3 = \dot{\mathcal{E}}_3.\mathcal{E}_3 = \mathcal{E}_3.(\dot{x}_{3d} - (\mathcal{E}_4 + \dot{x}_{3d} + K_3.\mathcal{E}_3))$$

$$\text{D'où : } \dot{V}_3 = -\mathcal{E}_3.\mathcal{E}_4 - K_3.\mathcal{E}_3^2$$

Nous créons ainsi une fonction de Lyapunov candidate :

$$V_4 = V_3 + \frac{1}{2}.\mathcal{E}_4^2$$

Il faut également définir une nouvelle fonction de délimitation :

$$W_4(\mathcal{E}) = -K_3.\mathcal{E}_3^2 - K_4.\mathcal{E}_4^2 \quad \text{avec } K_4 > 0$$

Nous dérivons la nouvelle fonction de Lyapunov :

$$\dot{V}_4 = -\mathcal{E}_3.\mathcal{E}_4 - K_3.\mathcal{E}_3^2 + \mathcal{E}_4.(x_4 - \ddot{x}_{3d} - K_3.\mathcal{E}_3)$$

Celle-ci doit respecter l'inégalité suivante :

$$\dot{V}_4 = -\mathcal{E}_3.\mathcal{E}_4 - K_3.\mathcal{E}_3^2 + \mathcal{E}_4.(x_4 - \ddot{x}_{3d} - K_3.\mathcal{E}_3) \leq -K_3.\mathcal{E}_3^2 - K_4.\mathcal{E}_4^2$$

En remplaçant la valeur de x_4 par :

$$\dot{x}_4 = b_2.U_3 + a_4.x_2.\Omega_{Rotor} + a_3.x_2.x_6$$

Nous aboutissons enfin sur l'expression du contrôleur du tangage U_3 :

$$U_3 = \frac{1}{b_2}.(-K_4.\mathcal{E}_4 + \mathcal{E}_3 - x_2.x_6.a_3 - x_2.\Omega_{Rotor}.a_4 + \ddot{x}_{3d} + K_3.\dot{x}_{3d} - K_3.x_4)$$

De la même manière, nous avons obtenu :

Pour le contrôleur de l'altitude :

$$U_1 = \frac{m}{\cos(x_1).\cos(x_3)}.(-\mathcal{E}_7 + g - \ddot{x}_{7d} - K_7.\dot{x}_{7d} + K_7.x_8 + K_8.\mathcal{E}_8)$$

Pour le contrôleur du roulis :

$$U_2 = \frac{1}{b_1}.(-K_2.\mathcal{E}_2 + \mathcal{E}_1 - x_4.x_6.a_1 + x_4.\Omega_{Rotor}.a_2 + \ddot{x}_{1d} + K_1.\dot{x}_{1d} - K_1.x_2)$$

Pour le contrôleur du lacet :

$$U_4 = \frac{1}{b_3}.(-K_6.\mathcal{E}_6 + \mathcal{E}_5 - x_2.x_4.a_5 + \ddot{x}_{5d} + K_5.\dot{x}_{5d} - K_5.x_6)$$

5 Simulation du modèle avec backstepping

Nous utilisons encore le logiciel Simulink pour la simulation de notre système. Dans cette partie en revanche, il n'est envisageable de mettre une illustration du schéma de contrôle réalisé. Le schéma complet étant trop conséquent, nous avons fourni le fichier Simulink en pièce attachée à ce présent rapport.

Concernant les résultats, ceux-ci sont tout à fait acceptables puisque la stabilisation du quadrirotor à l'altitude désirée est assurée (Voir Figure 8).

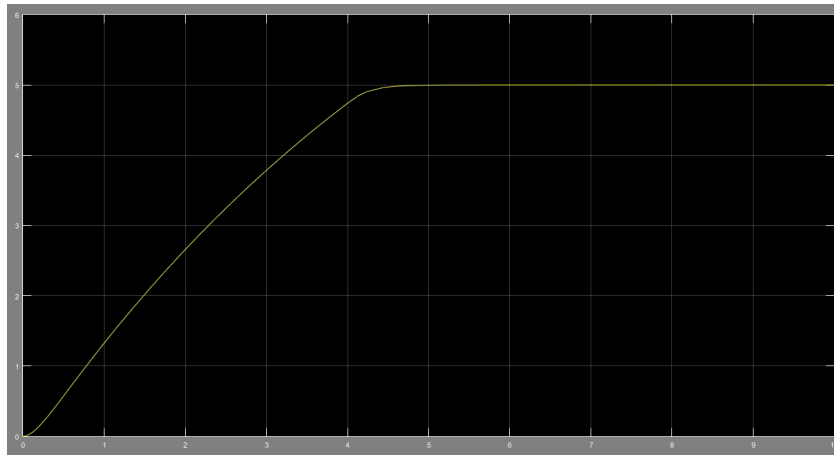


FIGURE 8 – Graphique : altitude désirée atteinte et stable.

Ici l'altitude désirée est de 5m, elle est atteinte en 4,6 secondes. Bien entendu, cela n'est pas très parlant puisque le système ne prend pas en compte les limites physiques concernant les moteurs : par exemple, la vitesse de rotation maximale des rotors. Mais en jouant les paramètres de gain dans le paramétrage du backstepping (les différents K) nous pouvons tout de même modifier le temps de réponse du système en faisant attention en revanche de respecter les conditions sur les fonctions de Lyapunov.

6 Bilan et perspectives

Bien que le système réel du quadrirotor soit bien plus complexe, les travaux réalisés lors de ce projet nous ont permis de faire nos premiers pas en matière de commande de robot aérien. Et d'éprouver une réelle satisfaction lorsque le drone suit nos instructions, prouvant ainsi que le système de contrôle mis en place est fonctionnel.

Néanmoins, nous ne contrôlons pas tout : en pilotant uniquement l'élévation en Z du quadrirotor, nous ne sommes pas en mesure de le déplacer librement selon ses 6 degrés de liberté. De plus, la stabilité est assurée fortement par la commande par backstepping ce qui rend le quadrirotor moins manœuvrable.

Nous pouvons émettre quelques pistes quant aux améliorations qui pourraient être mises en place sur notre projet. Notamment pour contrôler le quadrirotor selon tous ses degrés de liberté :

- Réinjecter en entrée les valeurs en X et Y , devrait aboutir au contrôler total du quadrirotor. Il sera sûrement nécessaire de retravailler les équations générales sur la commande du backstepping.
- Spécifier les valeurs maximales de rotation des rotors permettrait de se rapprocher du système réel .

Plus humainement, ce projet a également été l'occasion de réaliser un travail collaboratif ce qui permet de nous détacher un peu de notre statut d'étudiant et de mieux nous préparer à la vie active.

Le présent rapport et les fichiers Matlab-Simulink sont disponibles sur :

<https://github.com/ArtusoAlexis/RobotiquedeService>