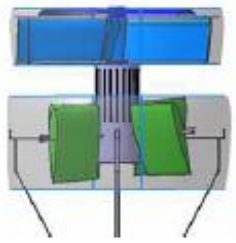
Signaux et Systèmes Linéaires Espace d'état
TP N° 1

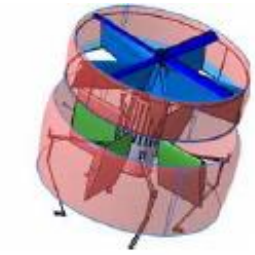
Dans le cadre du projet mini-drones lancé par la DGA et l'ONERA, l'ENSAM (Ecole Nationale Supérieure des Arts et Métiers) a conçu le drone représenté ci-dessous.

Ce drone peut présenter 3 modes de fonctionnement différents :

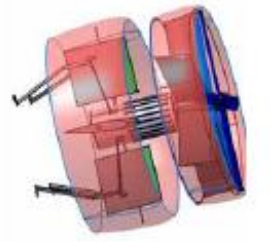
- Mode hélicoptère : orientation verticale de l'axe longitudinal de l'appareil et déplacements en translation selon l'axe vertical.
- Mode avion
- Phases de basculement entre le mode avion et le mode hélicoptère



Mode hélicoptère



Phase de basculement



Mode avion

Dans ce TP, on ne s'intéressera qu'au mode hélicoptère

L'objectif de ce TP est de synthétiser une commande par retour d'état afin d'asservir l'altitude du drone.

Ce drone est équipé d'un moteur brushless actionnant une hélice. Le signal de commande est le courant d'entrée du moteur, $I(t)$. Le drone est muni d'un télémètre é ultrasons permettant de mesurer son altitude par rapport au sol, $Z(t)$. Lorsque le drone est posé sur le sol, on a $Z = 0$. On note $V_z(t) = \dot{Z}(t)$ la vitesse du drone selon l'axe vertical. La vitesse de rotation de l'hélice est notée $R(t)$. Le vecteur d'état considéré est :

$$X(t) = \begin{bmatrix} Z(t) \\ V_z(t) \\ R(t) \end{bmatrix}$$

Le comportement du drone en mode hélicoptère peut être modélisé par l'équation d'état non linéaire suivante :

$$\begin{cases} \dot{Z}(t) = V_z(t) \\ \dot{V}_z(t) = \dfrac{k_h\, I(t)\, (R(t))^2}{M\, K_c} - g \\ \dot{R}(t) = \dfrac{I(t)}{I_h\, K_c} - \dfrac{k_1\, (R(t))^2\, (1 + k_2\, (I(t))^2)}{I_h} \end{cases} \tag{1}$$

oé

- La masse du drone est $M = 1\,\text{kg}$
- $g = 9.81\,\text{m.s}^{-2}$
- La constante de couple du moteur est $K_c = 185\,\text{A.N}^{-1}\text{.m}^{-1}$
- Le moment d'inertie de l'hélice est $I_h = 0.00025\,\text{kg.m}^2$
- k_h, k_1 et k_2 sont des coefficients aérodynamiques que l'on supposera constants. On prendra : $k_h = 0.0026$, $k_1 = 3.3568\,10^{-7}$ et $k_2 = 4.95\,10^{-5}$

On cherche à asservir l'altitude du drone, donc l'équation de sortie est :

$$y(t) = C\,X(t) \qquad \text{avec} \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

1 Etude en boucle ouverte

1. Montrer que les points d'équilibre de l'équation d'état (1) sont de la forme $(I_e; X_e)$ avec :

$$I_e = \sqrt{\frac{k_1\, M\, g\, K_c^2}{k_h - k_1\, k_2\, M\, g\, K_c^2}} \qquad X_e = \begin{bmatrix} Z_e \\ V_{ze} \\ R_e \end{bmatrix}$$

$$Z_e \text{ est quelconque}, \quad V_{ze} = 0 \quad \text{et} \quad R_e = \sqrt{\frac{M\, g\, K_c}{k_h\, I_e}}$$

Dans la suite du TP, on prendra $Z_e = 0$
On considère qu'é l'instant initial le système est é l'équilibre i.e. $I(0) = I_e$ et $X(0) = X_e$

2. On fournit les fichiers *DefParam.m* et *SimNLbo.mdl*. Le fichier *DefParam.m* contient les déclarations des paramètres ainsi que le calcul de I_e et de X_e. Le fichier *SimNLbo.mdl* est une simulation du comportement du drone sous simulink. Il faut exécuter *DefParam.m* au moins une fois avant de pouvoir lancer la simulation *SimNLbo.mdl*. L'état initial utilisé dans *SimNLbo.mdl* est défini dans *DefParam.m* : les variables *Z0*, *Vz0* et *R0* représentent respectivement $Z(0)$, $V_z(0)$ et $R(0)$. Après la simulation, on dispose du vecteurs *tempsNL* qui contient tous les temps de calcul et de la matrice *Xnl* qui contient les valeurs du vecteur d'état pour chaque temps de calcul. Les valeurs du vecteur d'état sont stockées dans *Xnl* de la façon suivante :

$$\begin{bmatrix} Z(0) & V_z(0) & R(0) \\ Z(t_1) & V_z(t_1) & R(t_1) \\ Z(t_2) & V_z(t_2) & R(t_2) \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Ainsi, la colonne numéro j de la matrice *Xnl* contient les valeurs de la $j^{\text{éme}}$ composante du vecteur d'état au cours du temps. Par conséquent, afin de tracer, la $j^{\text{éme}}$ composante du vecteur d'état en fonction du temps, on tapera, sous Matlab, la commande suivante : *plot(tempsNL,Xnl(:,j))*.

(a) Effectuer une simulation en prenant une intensité d'entrée constante : $I(t) = I_e$. La valeur du point d'équilibre $(I_e; X_e)$ calculée é la question précédente est-elle correcte?

(b) Prendre encore $I(t) = I_e$ mais choisir un état initial légèrement différent de l'état d'équilibre X_e (prendre, par exemple *Z0=0*, *Vz0=0* et *R0 = Re + 1e-5*). L'état d'équilibre X_e vous semble-t-il stable?

(c) Prendre $X(0) = X_e$ et $I(t) = I_e + 0.1\,\mathcal{H}(t)$ oé $\mathcal{H}(t)$ est l'échelon de Heavyside. Le point d'équilibre $(I_e; X_e)$ vous semble-t-il stable?

3. Linéariser l'équation (1) autour du point d'équilibre $(I_e; X_e)$. Montrer que l'on obtient la représentation d'état linéarisée :

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,i(t) \\ y(t) = C\,x(t) \end{cases}$$

oé

$$x(t) = X(t) - X_e \qquad i(t) = I(t) - I_e$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \dfrac{2k_h I_e R_e}{MK_c} \\ 0 & 0 & \dfrac{-2R_e k_1\,(1 + k_2 I_e^2)}{I_h} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \dfrac{k_h R_e^2}{MK_c} \\ \dfrac{1}{I_h K_c} - 2\dfrac{k_1 k_2 R_e^2 I_e}{I_h} \end{bmatrix}$$

4. Créer un fichier *Tp1.m*. En début de ce fichier, écrire la commande *eval('DefParam');* afin de lancer l'exécution du fichier *DefParam.m*. Puis, définir les matrices A et B.

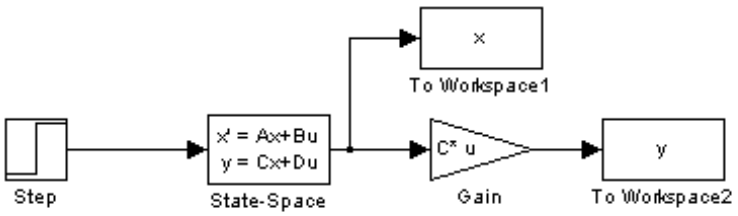
5. A l'aide de Matlab, étudier la stabilité, la communicabilité et l'observabilité du système linéarisé. Interpréter ces résultats.
On pourra utiliser les fonctions Matlab :

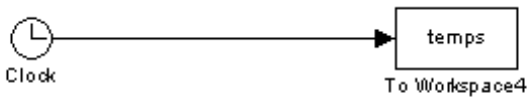
   - *eig* : cette fonction permet de calculer les valeurs propres d'une matrice
   - *ctrb* : permet de calculer la matrice de communicabilité, Q_S, associée é la paire (A, B). Cette fonction s'utilise de la façon suivante : *Qs = ctrb(A,B)*
   - *obsv* : permet de calculer la matrice d'observabilité, Q_B, associée é la paire (A, C). Cette fonction s'utilise de la façon suivante : *Qb = obsv(A,C)*
   - *rank* : permet de calculer le rang d'une matrice

6. Tracer la réponse indicielle du système linéarisé à un échelon de courant, $i(t)$, d'amplitude 0.3 Ampère. Pour cela, on définira, tout d'abord, la représentation d'état à l'aide de la commande *ss* de la façon suivante : *Sys=ss(A,B,C,0)*. Puis on tracera la réponse indicielle é l'aide de la commande : *step(0.3*Sys)*.
D'après le tracé obtenu, le système linéarisé est-il stable?

7. Calculer la fonction de transfert du système linéarisé. On pourra, pour cela, utiliser la commande *tf*.

8. Simuler é l'aide de Simulink, le comportement du système linéarisé de la façon représentée ci-dessous.





- Le bloc *State-Space* permet de simuler le comportement d'un système décrit par une représentation d'état de la forme :

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t) \\ y(t) = C\,x(t) + D\,u(t) \end{cases}$$

  afin de définir les paramètres du bloc *State-Space*, cliquer deux fois sur ce bloc, puis dans le champ A, écrire A, dans le champ B, écrire B. Afin de pouvoir accéder aux valeurs de chaque composante du vecteur d'état, on mettra dans le champ C, la matrice identité de dimension 3. On mettra dans le champ D une matrice nulle de dimensions adéquates. Ainsi le signal de sortie du bloc *State-Space* est $x(t)$. Le champ *Initial conditions* du bloc *State-Space* permet de définir la valeur initiale du vecteur d'état.
- Dans le champ *Multiplication* du bloc *gain*, choisir le mode de multiplication : *Matrix(K*u)*.
- Dans le champ *Save format* des blocs *To Workspace*, choisir *Array*.
- On veillera é définir l'état initial $x(0)$ et l'entrée $i(t)$ de façon cohérente avec l'état initial $X(0)$ du système non linéaire et l'entrée $I(t)$ du système non linéaire.

9. Afin d'évaluer la validité de la linéarisation effectuée, on souhaite comparer la réponse temporelle du système non linéaire et celle du système linéaire. On prendra $I(t) = I_e + 0.3\,\mathcal{H}(t)$ et $X(0) = X_e$. Tracer :
   - sur une même figure $Z(t)$ obtenu par simulation du système non linéaire et $Z(t)$ obtenu par simulation du système linéaire.
   - sur une même figure $V_z(t)$ obtenu par simulation du système non linéaire et $V_z(t)$ obtenu par simulation du système linéaire.
   - sur une même figure $R(t)$ obtenu par simulation du système non linéaire et $R(t) = r(t) + R_e$ obtenu par simulation du système linéaire.

   Evaluer la validité de la linéarisation.

2 Commande par retour d'état

On considère, dans cette partie, que l'on a accès à toutes les composantes du vecteur d'état.
On considère, tout d'abord, le système linéarisé et on souhaite asservir l'altitude $Z(t)$ é l'altitude désirée Z_d. Pour cela, on mettra en œuvre une commande par retour d'état de la forme :

$$i(t) = M_1\,Z_d - K\,x(t)$$

On souhaite que l'erreur en régime permanent soit nulle, que le temps de réponse soit d'environ 6 secondes et que la réponse temporelle ne présente pas d'oscillations.

1. On souhaite que les pôles de la boucle fermée soient -0.5, -5 et -5. Ce choix vous semble-t-il approprié?
2. Calculer K et M_1 à l'aide de Matlab. On pourra, afin de calculer K, utiliser la commande *acker*.
3. Tracer, à l'aide de Matlab, la réponse indicielle du système asservi à un échelon de consigne d'amplitude 1 m. Le résultat obtenu est-il satisfaisant?
4. Simuler le comportement du système linéaire asservi à l'aide de simulink
5. Remarquons que d'après la loi de commande considérée : $I(t) = i(t) + I_e = M_1\,Z_d - K\,x(t) + I_e$

$$I(t) = M_1\,Z_d - K\,(X(t) - X_e) + I_e$$

   Simuler le comportement du système non linéaire asservi au moyen de cette loi de commande.
6. Comparer la réponse du système non linéaire et celle du système linéaire pour un échelon de consigne d'amplitude 1 mètre.