

# Commande linéaire des systèmes dynamiques

## Projet de synthèse

Thierry Miquel, Florian Sansou, Amélie Carrierou,  
Jim Sharples, Jean-Philippe Condomines, Antoine Drouin  
ENAC

2 décembre 2025



# Table des matières

<b>1 Objectif</b>	<b>5</b>
<b>2 Liste - non exhaustive - de commandes Matlab® pouvant être utiles au projet</b>	<b>7</b>
2.1 Organisation du dossier de travail . . . . .	7
2.2 Commandes Matlab® . . . . .	8
2.2.1 Organisation des fichiers . . . . .	8
2.2.2 Commandes pour l'affichage de données . . . . .	8
2.2.3 Commandes utiles pour l'automatique . . . . .	8
2.3 Blocs Simulink® . . . . .	9
2.3.1 Debug et organisation de fichiers Simulink® . . . . .	9
2.3.2 Blocs Simulink® pour l'automatique . . . . .	9
<b>3 Projet Tutoré : commande du micro-drone DonuT</b>	<b>11</b>
3.1 Introduction . . . . .	11
3.2 Modélisation . . . . .	11
3.2.1 Relations cinématiques . . . . .	12
3.2.2 Expression des forces et des moments . . . . .	12
3.2.3 Relation entre les commandes du drone et les forces et moments appliqués	13
3.2.4 Équations de Newton-Euler . . . . .	14
3.2.5 Représentation d'état . . . . .	15
3.3 Point d'équilibre . . . . .	16
3.3.1 Indications . . . . .	16
3.4 Linéarisation . . . . .	16
3.4.1 Indications . . . . .	16
3.5 Commande classique . . . . .	17
3.5.1 Méthode de Wang & all . . . . .	17
3.6 Commande modale . . . . .	18
3.6.1 Calcul de la pré-commande . . . . .	19
3.6.2 Ajout d'un intégrateur . . . . .	19
3.6.3 Observateur d'état . . . . .	20
3.7 Commande optimale (cas MIMO) . . . . .	21
3.7.1 Méthode <i>Linear Quadratic Tracker</i> (LQT) . . . . .	22
<b>4 Projet de Synthèse</b>	<b>27</b>
4.1 Scénario . . . . .	27
4.2 Organisation . . . . .	28
4.3 Travail à effectuer . . . . .	28
4.4 Prise en main du modèle Matlab® / Simulink® . . . . .	29

4.4.1	Point d'équilibre . . . . .	29
4.4.2	Linéarisation . . . . .	29
4.4.3	Modes . . . . .	29
4.5	Commande classique : boucle de tenue d'assiette . . . . .	30
4.6	Commande modale : boucle de tenue de la vitesse équivalente . . . . .	30
4.7	Commande optimale (cas MIMO) . . . . .	32
4.8	Et après... . . . . .	33
4.9	Évaluation . . . . .	33
<b>5</b>	<b>Annexe : Modélisation de la dynamique longitudinale d'un avion</b>	<b>35</b>
5.1	Hypothèses et notations . . . . .	35
5.2	Modèle atmosphérique . . . . .	36
5.3	Modèle d'état . . . . .	37
5.4	Rappel sur la vitesse équivalente . . . . .	38
5.5	Poussée . . . . .	38
5.6	Rappel sur le PHR . . . . .	38
5.7	Paramètres . . . . .	40

# Chapitre 1

## Objectif

L'objectif de ce projet de synthèse est de concevoir les différentes lois du pilote automatique permettant de simuler ce scénario en utilisant le logiciel de simulation numérique Matlab® / Simulink®.

Selon Wikipédia<sup>1</sup> *Un pilote automatique est un dispositif de guidage automatique d'un véhicule sans intervention humaine. La conduite d'un véhicule exige l'attention continue d'un pilote afin de permettre un déplacement en toute sécurité. L'amélioration des techniques de construction aéronautique et navale a conduit à une inévitable augmentation de la durée des trajets. L'attention constante induisait ainsi une fatigue grave chez les pilotes. Au-delà de certaines durées de trajets, les limites humaines ne permettent pas de maintenir cette attention constante. Un pilote automatique est ainsi conçu pour effectuer certaines tâches du pilote, et le soulager. Le premier pilote automatique pour avion a été développé par Lawrence Sperry en 1912. Le pilote automatique permettait à l'avion de voler à cap constant et altitude constante sans l'attention du pilote, ce qui réduisait considérablement la charge de travail du pilote.*

Un pilote automatique d'avion est organisé autour des modes de base et des modes supérieurs. Les modes de base gèrent la position de l'avion autour de son centre de gravité et assurent le maintien de l'attitude de l'avion. Ce sont des modes de stabilisation, appelés aussi modes de pilotage. Les modes supérieurs gèrent la position du centre de gravité de l'avion dans l'espace et permettent le guidage de l'avion sur une trajectoire.

Dans un premier temps, vous serez accompagnés dans la synthèse de lois de contrôle pour le micro-drone DonuT. Puis vous travaillerez en équipe et en autonomie pour concevoir et implémenter différentes lois de commande relatives à la dynamique longitudinale d'un avion.

---

1. [https://fr.wikipedia.org/wiki/Pilote\\_automatique](https://fr.wikipedia.org/wiki/Pilote_automatique)



FIGURE 1.1 – FCU et PFD

## Chapitre 2

# Liste - non exhaustive - de commandes Matlab® pouvant être utiles au projet

Ce chapitre a pour but de présenter une liste non exhaustive de commandes Matlab®, de blocs Simulink® ainsi que divers conseils pouvant être utiles dans le cadre de ce projet ou de façon plus générale en automatique.

La première commande à connaître est *help* qui peut être tapée seule dans la fenêtre de commande Matlab® ou suivie d'une fonction (*help <commande>*) pour avoir sa documentation.

Voici quelques liens utiles :

- Site de Mathworks avec une documentation très détaillée pour Matlab® / Simulink® : <https://fr.mathworks.com/help/matlab/index.html>
- Formation Matlab® Onramp pour débuter : <https://matlabacademy.mathworks.com/fr/details/matlab-onramp/gettingstarted>
- Formation Simulink® Onramp pour débuter avec Simulink® : <https://matlabacademy.mathworks.com/fr/details/simulink-onramp/simulink>

### 2.1 Organisation du dossier de travail

Quelques conseils de base pour l'organisation de vos projets Matlab® / Simulink® (ou autre de façon plus générale) :

- Avoir des noms clairs et explicatifs pour vos fichiers
- Avoir des noms compréhensibles et si possible concis pour vos variables
- Avoir des sous dossiers pour différentes catégories (ex : données issues de simulations, figures)
- Avoir des fichiers Simulink® séparés pour chaque cas de figures (ex : pour chaque type de correcteurs et chaque type de modèles linéaire et non linéaire)

Pour aller plus loin dans l'organisation des dossiers de travail et la collaboration entre plusieurs personnes il est possible d'utiliser des *projets matlab®* (voir : <https://fr.mathworks.com/help/matlab/projects.html>).

Ces projets permettent la gestion de plusieurs fichiers Matlab® / Simulink® avec l'exécution de certains scripts au démarrage. Ils facilitent également la travail collaboratif avec par exemple

une intégration github et des outils pour la gestions des dépendances entre différents fichiers.

## 2.2 Commandes Matlab®

Les sections suivantes présentent des listes non exhaustives de commandes Matlab®, classées par catégories.

### 2.2.1 Organisation des fichiers

- *clear* : Efface les variables du "workspace" Matlab®
- *close* : Ferme les fenêtres de figures
- *bdclose* : Ferme les fenêtres Simulink®
- *clc* : Nettoie la "command window" Matlab®
- *%* : Au début d'une ligne de commentaire
- *%%* : Au début d'une nouvelle section
- *save* : Sauvegarde des données
- *load* : Charge des données
- <nom\_modele\_simulink> : Ouvre et lance le modèle Simulink® ou le fichier Matlab® correspondant

### 2.2.2 Commandes pour l'affichage de données

- *figure* : Ouvre une nouvelle fenêtre de figure
- *plot* : Affiche des données dans un graphe en 2D
- *subplot* : Affiche différents graphes sur la même fenêtre
- *hold* (on or off) : Affiche plusieurs courbes sur le même graphe
- *grid* (on or off) : Affiche ou enlève une grille en fond du graphe
- *xlim* (*ylim*) : Limite d'affichage de l'axe *x* (ou *y*)
- *title* : Affiche le titre d'un graphe
- *suptitle* : Affiche un titre global pour une fenêtre comprenant plusieurs graphes
- *legend* : Ajoute une légende à un graphe
- *xlabel* (*ylabel*) : Ajoute un titre à l'axe *x* (ou *y*)
- *plot3* : Affiche des données en 3D (les commandes précédentes peuvent être utilisées, celles par axes sont également valables pour un axe *z*)

### 2.2.3 Commandes utiles pour l'automatique

- *eye* : Matrice identité
- *zeros* : Matrice ou vecteur de 0
- *ones* : Matrice ou vecteur de 1
- *find* : Renvoie l'indice des éléments non nuls
- *sort* : Tire dans un ordre croissant ou décroissant
- *rank* : Renvoie le rang d'une matrice
- *eig* : Retourne les vecteurs propres et valeurs propres
- *poly* : Converti des racines en polynôme
- *roots* : Trouve les racine d'un polynôme
- *polyval* : Évalue un polynôme
- *linmod* : Renvoie un modèle d'espace d'état associé à un système d'équations différentielles ordinaires

- *damp* : Retourne la fréquence naturelle, l'amortissement et le gain d'un système
- *tf* : Construit une fonction de transfert à partir de coefficients
- *pole* : Renvoie les pôles d'un système linéaire
- *zero* : Retourne les zéros et le gain d'un système linéaire
- *zpk* : Construit un système sous la forme zéros, pôles, gain
- *tf2zp* : Renvoie le système zéro, pôles, gain associé à une fonction de transfert
- *minreal* : Donne la réalisation minimale d'un système avec les compensations pôles zéros
- *ss* : Construit un système sous forme d'espace d'état
- *ss2tf* : Renvoie la fonction de transfert associée à un système sous forme d'espace d'état
- *series* : Connection en série de deux systèmes dynamiques
- *feedback* : Connection par boucle de retour de deux systèmes dynamiques
- *rlocus* : Renvoie le lieu des racines d'Evans associé à un système dynamique
- *obsv* : Calcule la matrice d'observabilité
- *ctrb* : Calcule la matrice de commandabilité
- *care* : Résout des équations de Riccati en temps continu
- *lqr* : Design d'un gain de retour par la méthode LQR pour un système sous forme d'espace d'état
- *step* : Réponse indicielle d'un système dynamique
- *stepinfo* : Renvoie les caractéristiques de la réponse indicielle d'un système dynamique

## 2.3 Blocs Simulink®

Lors du travail sur des fichiers Simulink®, il est important de garder en mémoire que le "workspace" Matlab® est un espace de travail partagé et que des variables du "workspace" peuvent être directement utilisées dans les bloc Simulink®. Les blocs sont disponibles dans la librairie Simulink® ou en double cliquant sur le fond et en entrant le nom du bloc désiré.

### 2.3.1 Debug et organisation de fichiers Simulink®

- Affichage des dimensions des signaux (debug/informations overlay/signals)
- Nommer les blocs et signaux
- *Subsystem* : Regroupe les blocs et signaux en un sous-système
- *In* et *Out* : Entrées et sorties d'un sous-système ou d'un modèle
- *Goto* et *From* : Envoie les signaux d'un bloc *Goto* au bloc *From* correspondant
- *From Workspace* : Charge des données du "workspace" Matlab®
- *To Workspace* : Envoie les données d'un modèle Simulink® au "workspace" Matlab® (le format *Timeseries* est préférable)

### 2.3.2 Blocs Simulink® pour l'automatique

- *Step* : Source signal échelon
- *Ramp* : Source signal rampe
- *Sine Wave* : Source de signal sinusoïdale
- *Constant* : Bloc source d'un signal constant, permet également de passer une valeur du "Workspace" Matlab® au fichier Simulink®
- *Scope* : Bloc permettant (entre autre) l'affichage de signaux
- *Sum* : Somme ou différence de signaux
- *Gain* : Gain multiplicatif d'un signal

## 10 CHAPITRE 2. LISTE - NON EXHAUSTIVE - DE COMMANDES MATLAB® POUVANT ÊTRE UTILES A

- *Switch* : Bloc à trois entrées, passe de l'entrée 1 à l'entrée 3 en fonction d'une condition sur l'entrée 2
- *Manual Switch* : Bloc à deux entrées, permet de passer manuellement entre l'entrée 1 et l'entrée 2
- *Integrator* : Calcule l'intégrale d'un signal
- *Derivative* : Calcule la dérivée d'un signal
- *Mux* : Multiplexeur
- *Demux* : Démultiplexeur
- *Transfer Fcn* : Bloc permettant la simulation d'un système sous forme de fonction de transfert
- *State-Space* : Bloc permettant la simulation d'un système sous forme d'espace d'état
- *Matlab Function* : Bloc permettant l'ajout de code Matlab® dans des fichiers Simulink®

## Chapitre 3

# Projet Tutoré : commande du micro-drone DonuT

### 3.1 Introduction

Le micro-drone DonuT a été développé par l'ISAE-SUPAERO<sup>1</sup>. Il est composé de deux rotors contra-rotatifs à pas fixes qui peuvent être inclinés indépendamment sur deux axes, comme présenté sur la Figure 3.1.



FIGURE 3.1 – Micro-drone DonuT

### 3.2 Modélisation

La modélisation d'un micro-drone de type DonuT peut par exemple être trouvée dans Zhao & al.<sup>2</sup>.

1. S. Prothin & JM. Moschetta, A Vectoring Thrust Coaxial Rotor for Micro Air Vehicle : Modeling, Design and Analysis, 21ème Congrès Français de Mécanique Bordeaux, 26 au 30 août 2013

2. C. Zhao, H. Lin, L. Feng, S. Bai, F. Hu and L. Zhao, Backstepping-based Tracking Control for a Coaxial Rotor UAV, 2022 China Automation Congress (CAC), Xiamen, China, 2022, pp. 5182-5187, doi : 10.1109/CAC57257.2022.10055474

### 3.2.1 Relations cinématiques

Soit  $\underline{\eta} = [\phi \ \theta \ \psi]^T$  le vecteur des angles d'Euler permettant de définir l'orientation du drone vis-à-vis du repère inertiel. La matrice de rotation  $\mathbf{R}_i^b(\underline{\eta})$  du repère inertiel vers le repère du drone (*body*), paramétrée par les angles d'Euler, a l'expression suivante :

$$\mathbf{R}_i^b(\underline{\eta}) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ (s_\phi s_\theta c_\psi - c_\phi s_\psi) & (s_\phi s_\theta s_\psi + c_\phi c_\psi) & s_\phi c_\theta \\ (c_\phi s_\theta c_\psi + s_\phi s_\psi) & (c_\phi s_\theta s_\psi - s_\phi c_\psi) & c_\phi c_\theta \end{bmatrix} \quad (3.1)$$

où  $c_x$  est utilisé pour désigner  $\cos(x)$  et  $s_x$  pour désigner  $\sin(x)$ .

En désignant par  $\underline{\xi} = [x \ y \ z]^T$  le vecteur représentant la position du centre de gravité du drone dans le repère inertiel et par  $\underline{v}^b = [u \ v \ w]^T$  le vecteur représentant la vitesse de translation du centre de gravité du drone dans le repère lié au drone, nous avons :

$$\dot{\underline{\xi}} = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = (\mathbf{R}_i^b(\underline{\eta}))^{-1} \underline{v}^b = (\mathbf{R}_i^b(\underline{\eta}))^T \underline{v}^b \quad (3.2)$$

Le vecteur vitesse de rotation dans le repère inertiel, noté  $\dot{\underline{\eta}} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ , et le vecteur vitesse de rotation dans le repère lié au drone, noté  $\underline{\omega} = [p \ q \ r]^T$ , sont reliées comme suit :

$$\dot{\underline{\eta}} = \mathbf{W}(\underline{\eta})^{-1} \underline{\omega} \quad (3.3)$$

où :

$$\mathbf{W}(\underline{\eta})^{-1} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \quad (3.4)$$

Notons que :

$$\mathbf{W}(0)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbb{I} \quad (3.5)$$

### 3.2.2 Expression des forces et des moments

Soit  $\mathbf{R}_j^b(\underline{u}_j)$  la matrice de rotation du rotor  $j$ , où  $j$  vaut soit 1 (rotor bas, ou *lower rotor*) soit 2 (rotor haut, ou *upper rotor*), vers le repère du drone (*body*), paramétrée par les angles  $\delta_{x_j}$  et  $\delta_{y_j}$  de braquage des axes du rotor  $j$  :

$$\underline{u}_j = \begin{bmatrix} \delta_{x_j} \\ \delta_{y_j} \end{bmatrix} \Rightarrow \mathbf{R}_j^b(\underline{u}_j) = \begin{bmatrix} \cos(\delta_{x_j}) & \sin(\delta_{x_j}) \sin(\delta_{y_j}) & -\sin(\delta_{x_j}) \cos(\delta_{y_j}) \\ 0 & \cos(\delta_{y_j}) & \sin(\delta_{y_j}) \\ \sin(\delta_{x_j}) & -\cos(\delta_{x_j}) \sin(\delta_{y_j}) & \cos(\delta_{x_j}) \cos(\delta_{y_j}) \end{bmatrix} \quad (3.6)$$

Le force  $\underline{f}^j$  générée par le rotor  $j$  dans le repère lié au rotor à l'expression suivante, où  $C_l$  est le coefficient de portance (positif) de l'hélice et  $\omega_j$  la vitesse de rotation du rotor  $j$  :

$$\underline{f}^j = \begin{bmatrix} 0 \\ 0 \\ -C_l \omega_j^2 \end{bmatrix} \quad (3.7)$$

L'expression de la résultante  $\underline{f}_a^b$  des forces aérodynamiques exprimées dans le repère du drone est la suivante :

$$\begin{aligned} \underline{f}_a^b &= \mathbf{R}_1^b(\underline{u}_1) \underline{f}^1 + \mathbf{R}_2^b(\underline{u}_2) \underline{f}^2 \\ &= \begin{bmatrix} C_l s_{\delta_{x_1}} c_{\delta_{y_1}} \omega_1^2 + C_l s_{\delta_{x_2}} c_{\delta_{y_2}} \omega_2^2 \\ -C_l s_{\delta_{y_1}} \omega_1^2 - C_l s_{\delta_{y_2}} \omega_2^2 \\ -C_l c_{\delta_{x_1}} c_{\delta_{y_1}} \omega_1^2 - C_l c_{\delta_{x_2}} c_{\delta_{y_2}} \omega_2^2 \end{bmatrix} \end{aligned} \quad (3.8)$$

En désignant par  $\times$  le produit vectoriel, par  $d$  la distance selon l'axe vertical du centre de gravité du drone au centre de gravité du rotor et par  $C_d$  le coefficient du moment de traînée (positif) de l'hélice, le moment aérodynamique  $\underline{\tau}_j^b$  du rotor  $j$  exprimé dans le repère lié au drone s'exprime comme la somme du couple due à la force de traction de l'hélice et du couple qui s'oppose à la rotation de celle-ci :

$$\begin{aligned} \underline{\tau}_j^b &= \mathbf{S}(d) \mathbf{R}_j^b(\underline{u}_j) \underline{f}^j + \mathbf{R}_j^b(\underline{u}_j) \begin{bmatrix} 0 \\ 0 \\ C_d \omega_j^2 \end{bmatrix} \\ &= \begin{bmatrix} \left( d C_l s_{\delta_{y_j}} - C_d s_{\delta_{x_j}} c_{\delta_{y_j}} \right) \omega_j^2 \\ \left( d C_l s_{\delta_{x_j}} c_{\delta_{y_j}} + C_d s_{\delta_{y_j}} \right) \omega_j^2 \\ C_d c_{\delta_{x_j}} c_{\delta_{y_j}} \omega_j^2 \end{bmatrix} \end{aligned} \quad (3.9)$$

La matrice  $\mathbf{S}(d)$  est une matrice antisymétrique qui permet de remplacer le produit vectoriel, noté  $\times$ , par un produit matriciel :

$$\begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \times := \mathbf{S}(d) = -\mathbf{S}(d)^T = \begin{bmatrix} 0 & -d & 0 \\ d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.10)$$

En tenant compte du fait que les 2 rotors tournent dans des sens de rotation opposés, le moment aérodynamique total  $\underline{\tau}_a^b$  dans le repère lié au drone s'obtient en soustrayant le moment aérodynamique de chaque rotor :

$$\begin{aligned} \underline{\tau}_a^b &= \underline{\tau}_1^b - \underline{\tau}_2^b \\ &= \begin{bmatrix} \left( d C_l s_{\delta_{y_1}} - C_d s_{\delta_{x_1}} c_{\delta_{y_1}} \right) \omega_1^2 - \left( d C_l s_{\delta_{y_2}} - C_d s_{\delta_{x_2}} c_{\delta_{y_2}} \right) \omega_2^2 \\ \left( d C_l s_{\delta_{x_1}} c_{\delta_{y_1}} + C_d s_{\delta_{y_1}} \right) \omega_1^2 - \left( d C_l s_{\delta_{x_2}} c_{\delta_{y_2}} + C_d s_{\delta_{y_2}} \right) \omega_2^2 \\ C_d c_{\delta_{x_1}} c_{\delta_{y_1}} \omega_1^2 - C_d c_{\delta_{x_2}} c_{\delta_{y_2}} \omega_2^2 \end{bmatrix} \end{aligned} \quad (3.11)$$

### 3.2.3 Relation entre les commandes du drone et les forces et moments appliqués

La relation reliant les commandes  $\{\omega_j, \delta_{x_j}, \delta_{y_j}\}$ ,  $j = 1, 2$  du drone aux forces aérodynamiques  $\underline{f}_a^b := [\underline{f}_x \ \underline{f}_y \ \underline{f}_z]^T$  et moments  $\underline{\tau}_a^b := [\underline{\tau}_x \ \underline{\tau}_y \ \underline{\tau}_z]^T$  qui lui sont appliqués peut être obtenue en inversant les relations (3.8) et (3.11).

Remarquons tout d'abord que :

$$\begin{bmatrix} 0 \\ 0 \\ C_d \omega_j^2 \end{bmatrix} = -\frac{C_d}{C_l} \begin{bmatrix} 0 \\ 0 \\ -C_l \omega_j^2 \end{bmatrix} = -\frac{C_d}{C_l} \underline{f}^j \quad (3.12)$$

Par conséquent les relations (3.8) et (3.11) se ré-écrivent comme suit :

$$\begin{cases} \underline{f}_a^b = \mathbf{R}_1^b(\underline{u}_1) \underline{f}^1 + \mathbf{R}_2^b(\underline{u}_2) \underline{f}^2 \\ \underline{\tau}_a^b = \mathbf{S}(d) \mathbf{R}_1^b(\underline{u}_1) \underline{f}^1 - \frac{C_d}{C_l} \mathbf{R}_1^b(\underline{u}_1) \underline{f}^1 - \mathbf{S}(d) \mathbf{R}_2^b(\underline{u}_2) \underline{f}^2 + \frac{C_d}{C_l} \mathbf{R}_2^b(\underline{u}_2) \underline{f}^2 \end{cases} \quad (3.13)$$

Soit, sous forme vectorielle :

$$\begin{bmatrix} \mathbb{I} \\ \mathbf{S}(d) - \frac{C_d}{C_l} \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{I} \\ -(\mathbf{S}(d) - \frac{C_d}{C_l} \mathbb{I}) \end{bmatrix} \begin{bmatrix} \mathbf{R}_1^b(\underline{u}_1) \underline{f}^1 \\ \mathbf{R}_2^b(\underline{u}_2) \underline{f}^2 \end{bmatrix} = \begin{bmatrix} f_a^b \\ \underline{\tau}_a^b \end{bmatrix} \quad (3.14)$$

Il vient après inversion de cette relation :

$$\begin{bmatrix} \mathbf{R}_1^b(\underline{u}_1) \underline{f}^1 \\ \mathbf{R}_2^b(\underline{u}_2) \underline{f}^2 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbb{I} \\ \mathbf{S}(d) - \frac{C_d}{C_l} \mathbb{I} & -(\mathbf{S}(d) - \frac{C_d}{C_l} \mathbb{I}) \end{bmatrix}^{-1} \begin{bmatrix} f_a^b \\ \underline{\tau}_a^b \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} \mathbb{I} & \mathbf{Y} \\ \mathbb{I} & -\mathbf{Y} \end{bmatrix} \begin{bmatrix} \underline{f}^b \\ \underline{\tau}_a^b \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} \underline{f}^b + \mathbf{Y} \underline{\tau}_a^b \\ \underline{f}_a^b - \mathbf{Y} \underline{\tau}_a^b \end{bmatrix} \quad (3.15)$$

où :

$$\mathbf{Y} = \begin{bmatrix} -C_d/C_l & \frac{d}{(C_d/C_l)^2+d^2} & 0 \\ \frac{(C_d/C_l)^2+d^2}{(C_d/C_l)^2+d^2} & -C_d/C_l & 0 \\ \frac{(C_d/C_l)^2+d^2}{(C_d/C_l)^2+d^2} & 0 & -\frac{C_l}{C_d} \end{bmatrix} \quad (3.16)$$

En notant  $\begin{bmatrix} f_x^j & f_y^j & f_z^j \end{bmatrix}^T$  les 3 composantes de  $\mathbf{R}_j^b(\underline{u}_j) \underline{f}^j$ , les expressions de  $\delta_{x_j}$ ,  $\delta_{y_j}$  et  $\omega_j^2$  sont obtenues comme suit :

$$\begin{aligned} \mathbf{R}_j^b(\underline{u}_j) \underline{f}^j &= \begin{bmatrix} C_l s_{\delta_{x_j}} c_{\delta_{y_j}} \omega_j^2 \\ -C_l s_{\delta_{y_j}} \omega_j^2 \\ -C_l c_{\delta_{x_j}} c_{\delta_{y_j}} \omega_j^2 \end{bmatrix} := \begin{bmatrix} f_x^j \\ f_y^j \\ f_z^j \end{bmatrix} \\ &\Rightarrow \begin{bmatrix} \omega_j^2 \\ \delta_{x_j} \\ \delta_{y_j} \end{bmatrix} = \begin{bmatrix} \frac{1}{C_l} \sqrt{(f_x^j)^2 + (f_y^j)^2 + (f_z^j)^2} \\ \arctan\left(\frac{f_x^j}{-f_z^j}\right) \\ \arctan\left(\frac{-f_y^j \cos(\arctan(\frac{f_x^j}{-f_z^j}))}{-f_z^j}\right) \end{bmatrix}, j = 1, 2 \quad (3.17) \end{aligned}$$

### 3.2.4 Équations de Newton-Euler

Soient  $m$  la masse du drone,  $\mathbf{I}_b$  la matrice d'inertie du drone,  $g$  l'accélération de la pesanteur et  $\underline{e}_3$  le vecteur suivant :

$$\underline{e}_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.18)$$

La résultante des forces extérieures  $\underline{f}^b$  exprimée dans le repère du drone a l'expression suivante (le signe + devant le terme où  $m g$  apparaît indique que l'axe vertical est orienté vers le bas) :

$$\begin{aligned} \underline{f}^b &= \underline{f}_a^b + \mathbf{R}_i^b(\eta) m g \underline{e}_3 \\ &= \sum_{j=1}^2 \mathbf{R}_j^b(\underline{u}_j) \underline{f}^j + \mathbf{R}_i^b(\eta) m g \underline{e}_3 \\ &= \begin{bmatrix} C_l s_{\delta_{x_1}} c_{\delta_{y_1}} \omega_1^2 + C_l s_{\delta_{x_2}} c_{\delta_{y_2}} \omega_2^2 - m g s_\theta \\ -C_l s_{\delta_{y_1}} \omega_1^2 - C_l s_{\delta_{y_2}} \omega_2^2 + m g s_\phi c_\theta \\ -C_l c_{\delta_{x_1}} c_{\delta_{y_1}} \omega_1^2 - C_l c_{\delta_{x_2}} c_{\delta_{y_2}} \omega_2^2 + m g c_\phi c_\theta \end{bmatrix} \quad (3.19) \end{aligned}$$

Les équations de Newton-Euler décrivant la dynamique du centre de gravité du drone dans le repère lié au drone s'exprime sous la forme suivante :

$$\begin{bmatrix} m\mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b \end{bmatrix} \begin{bmatrix} \dot{\underline{v}}^b \\ \dot{\underline{\nu}} \end{bmatrix} + \begin{bmatrix} m\mathbf{S}(\underline{\nu}) & \mathbf{0} \\ \mathbf{0} & \mathbf{S}(\underline{\nu})\mathbf{I}_b \end{bmatrix} \begin{bmatrix} \underline{v}^b \\ \underline{\nu} \end{bmatrix} = \begin{bmatrix} f_a^b + \mathbf{R}_i^b(\underline{\eta})m g e_3 \\ \underline{\tau}_a^b \end{bmatrix} \quad (3.20)$$

Où :

$$\underline{\nu} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \Rightarrow \mathbf{S}(\underline{\nu}) = -\mathbf{S}(\underline{\nu})^T = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (3.21)$$

Après inversion de (3.20) nous obtenons les relations suivantes :

$$\begin{bmatrix} \dot{\underline{v}}^b \\ \dot{\underline{\nu}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i^b(\underline{\eta})g e_3 - \mathbf{S}(\underline{\nu})\underline{v}^b \\ -\mathbf{I}_b^{-1}\mathbf{S}(\underline{\nu})\mathbf{I}_b\underline{\nu} \end{bmatrix} + \begin{bmatrix} \mathbb{I}/m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b^{-1} \end{bmatrix} \begin{bmatrix} f_a^b \\ \underline{\tau}_a^b \end{bmatrix} \quad (3.22)$$

Soient  $\underline{x}$  et  $\tilde{\underline{u}}$  les vecteurs suivants :

$$\underline{x} = \begin{bmatrix} \underline{v}^b \\ \underline{\nu} \end{bmatrix} \quad \text{et} \quad \tilde{\underline{u}} = \begin{bmatrix} f_a^b \\ \underline{\tau}_a^b \end{bmatrix} \quad (3.23)$$

Les équations de Newton-Euler (3.22) s'écrivent alors sous la forme suivante :

$$\dot{\underline{x}} = f_{ne}(\underline{x}, \underline{\eta}) + \mathbf{B}_{ne}\tilde{\underline{u}} \quad (3.24)$$

où :

$$\begin{cases} f_{ne}(\underline{x}, \underline{\eta}) = \begin{bmatrix} \mathbf{R}_i^b(\underline{\eta})g e_3 - \mathbf{S}(\underline{\nu})\underline{v}^b \\ -\mathbf{I}_b^{-1}\mathbf{S}(\underline{\nu})\mathbf{I}_b\underline{\nu} \end{bmatrix} \\ \mathbf{B}_{ne} = \begin{bmatrix} \mathbb{I}/m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b^{-1} \end{bmatrix} \end{cases} \quad (3.25)$$

La relation reliant le vecteur des commandes physiques du drone, noté  $\underline{u}$ , aux forces et moments générés par les rotors, noté  $\tilde{\underline{u}}$ , est obtenue par la relation (3.17) :

$$\underline{u} = [\omega_1^2 \quad \delta_{x_1} \quad \delta_{y_1} \quad \omega_2^2 \quad \delta_{x_2} \quad \delta_{y_2}]^T \Rightarrow \underline{u} = \Phi(\tilde{\underline{u}}) \quad (3.26)$$

### 3.2.5 Représentation d'état

Soient  $\underline{x}$  le vecteur d'état et  $\tilde{\underline{u}}$  le vecteur de commande du drone :

$$\underline{x} = \begin{bmatrix} \underline{v}^b \\ \underline{\nu} \\ \underline{\eta} \\ \underline{\xi} \end{bmatrix} \quad \text{et} \quad \tilde{\underline{u}} = \begin{bmatrix} f_a^b \\ \underline{\tau}_a^b \end{bmatrix} \quad (3.27)$$

En regroupant (3.2), (3.3) et (3.24) nous obtenons la représentation d'état suivante :

$$\begin{bmatrix} \dot{\underline{v}}^b \\ \dot{\underline{\nu}} \\ \dot{\underline{\eta}} \\ \dot{\underline{\xi}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i^b(\underline{\eta})g e_3 - \mathbf{S}(\underline{\nu})\underline{v}^b \\ -\mathbf{I}_b^{-1}\mathbf{S}(\underline{\nu})\mathbf{I}_b\underline{\nu} \\ \mathbf{W}(\underline{\eta})^{-1}\underline{\nu} \\ (\mathbf{R}_i^b(\underline{\eta}))^T\underline{v}^b \end{bmatrix} + \begin{bmatrix} \mathbb{I}/m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b^{-1} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} f_a^b \\ \underline{\tau}_a^b \end{bmatrix} \quad (3.28)$$

Soit sous forme compacte :

$$\dot{\underline{x}} = f(\underline{x}) + \mathbf{B}\tilde{\underline{u}} \quad (3.29)$$

où :

$$\begin{cases} f(\underline{x}) = \begin{bmatrix} \mathbf{R}_i^b(\underline{\eta}) g e_3 - \mathbf{S}(\underline{\nu}) \underline{v}^b \\ -\mathbf{I}_b^{-1} \mathbf{S}(\underline{\nu}) \mathbf{I}_b \underline{\nu} \\ \mathbf{W}(\underline{\eta})^{-1} \underline{\nu} \\ (\mathbf{R}_i^b(\underline{\eta}))^T \underline{v}^b \end{bmatrix} \\ \mathbf{B} = \begin{bmatrix} \mathbb{I}/m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b^{-1} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \end{cases} \quad (3.30)$$

### 3.3 Point d'équilibre

Calculez les valeurs des forces et des moments s'appliquant sur le drone dans le cas d'un vol stationnaire.

#### 3.3.1 Indications

Pour un vol stationnaire, nous avons :

$$\begin{cases} \underline{v}^b = \underline{0} \\ \underline{\nu} = \underline{0} \Rightarrow \mathbf{S}(\underline{\nu}) = \mathbf{0} \\ \underline{\eta} = \underline{0} \Rightarrow \mathbf{R}_i^b(\underline{\eta}) = \mathbb{I} \end{cases} \Rightarrow f(\underline{x}) = \begin{bmatrix} g e_3 \\ \underline{0} \\ \underline{0} \\ \underline{0} \end{bmatrix} \quad (3.31)$$

La représentation d'état (3.28) devient donc dans le cas d'un vol stationnaire :

$$\underline{0} = \begin{bmatrix} g e_3 \\ \underline{0} \\ \underline{0} \\ \underline{0} \end{bmatrix} + \begin{bmatrix} \mathbb{I}/m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b^{-1} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{f}^b \\ \underline{\tau}_a^b \end{bmatrix} \Rightarrow \underline{0} = \begin{bmatrix} g e_3 + \frac{\underline{f}_a^b}{m} \\ \mathbf{I}_b^{-1} \underline{\tau}_a^b \end{bmatrix} \Rightarrow \begin{cases} \underline{f}_a^b = -m g e_3 \\ \underline{\tau}_a^b = \underline{0} \end{cases} \quad (3.32)$$

### 3.4 Linéarisation

Linéarisez le modèle du drone autour d'un point de vol stationnaire.

#### 3.4.1 Indications

La linéarisation de la dynamique au voisinage du vol stationnaire a l'expression suivante :

$$\begin{bmatrix} \dot{\underline{v}}^b \\ \dot{\underline{\nu}} \\ \dot{\underline{\eta}} \\ \dot{\underline{\xi}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{A}_{13} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbb{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{v}^b \\ \underline{\nu} \\ \underline{\eta} \\ \underline{\xi} \end{bmatrix} + \begin{bmatrix} \mathbb{I}/m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b^{-1} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{f}^b \\ \underline{\tau}_a^b \end{bmatrix} \quad (3.33)$$

où :

$$\mathbf{A}_{13} = \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.34)$$

## 3.5 Commande classique

Concevez un correcteur PI-D et un pré-filtre permettant de commander l'assiette latérale  $\phi$  du drone.

Les spécifications sont les suivantes :

- Erreur de position nulle ;
- Temps de réponse  $t_s$  en boucle fermée inférieur à 0.5 sec ;
- Dépassement  $D$  de la réponse indicelle de l'ordre de 5%.

Testez votre loi de commande avec de *petites* valeurs de consigne dans un premier temps, par exemple  $\phi_c = 5$  degrés.

La représentation d'état (3.33) permet d'exprimer la dynamique linéarisée du vecteur  $\underline{\eta}$  des angles d'Euler :

$$\begin{cases} \dot{\underline{\eta}} = \underline{\nu} \\ \dot{\underline{\nu}} = \mathbf{I}_b^{-1} \underline{\tau}_a^b \end{cases} \Rightarrow \ddot{\underline{\eta}} = \mathbf{I}_b^{-1} \underline{\tau}_a^b \quad (3.35)$$

En notant  $\mathbf{F}(s)$  la matrice de transfert du système, nous avons :

$$\underline{\eta}(s) = \mathbf{F}(s) \underline{\tau}_a^b(s) \quad \text{où} \quad \mathbf{F}(s) = \frac{1}{s^2} \mathbf{I}_b^{-1} \quad (3.36)$$

Comme  $\mathbf{I}_b^{-1}$  est une matrice diagonale, nous obtenons 3 équations découplées.

En posant  $\mathbf{I}_b = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}$  et  $\underline{\tau}_a^b = \begin{bmatrix} \tau_\phi^b \\ \tau_\theta^b \\ \tau_\psi^b \end{bmatrix}$ , il vient pour l'assiette latérale  $\phi$  du drone :

$$\begin{cases} \underline{\tau}_a^b = \begin{bmatrix} \tau_\phi^b \\ \tau_\theta^b \\ \tau_\psi^b \end{bmatrix} \\ \mathbf{I}_b = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \end{cases} \Rightarrow \phi(s) = \frac{1}{I_x s^2} \tau_\phi^b(s) \quad (3.37)$$

### 3.5.1 Méthode de Wang & all

En désignant par  $\{\lambda_{dp}, \bar{\lambda}_{dp}\} = -m\omega_0 \pm j\omega_0\sqrt{1-m^2}$  les poles dominants souhaités en boucle fermée, il vient à partir des spécifications :

$$\begin{cases} t_s = 0.5 \text{ sec} \\ D = 5\% \end{cases} \Rightarrow \begin{cases} m = \frac{-\ln(D)}{\sqrt{\pi^2 + \ln^2(D)}} \approx 0.8261 \\ w_n \approx \frac{4}{mt_s} \approx 11.6 \text{ rad/sec} \end{cases} \Rightarrow \{\lambda_{dp}, \bar{\lambda}_{dp}\} = -m\omega_0 \pm j\omega_0\sqrt{1-m^2} = -8 \pm j8.39$$

Les paramètres  $K_p$ ,  $K_i$  et  $K_d$  du correcteur PI-D sont alors obtenus par la méthode de Wang & all<sup>3</sup> :

- Soit  $\{\lambda_{dp}, \bar{\lambda}_{dp}\}$  la paire des poles complexes conjugués, que l'on souhaite *dominants* :

$$\begin{cases} \lambda_{dp} = a + jb \\ \bar{\lambda}_{dp} = a - jb \end{cases} \Leftrightarrow \begin{cases} a = \operatorname{Re}(\lambda_{dp}) \\ b = \operatorname{Im}(\lambda_{dp}) \end{cases} \quad (3.38)$$

3. Guaranteed dominant pole placement with PID controllers, Qing-Guo Wang, Zhiping Zhang, Karl Johan Astrom, Lee See Chek, Journal of Process Control 19 (2009) 349–35

— Calculez  $X_1$  et  $X_2$  définis comme suit :

$$\begin{cases} X_1 = \frac{1}{2b} \operatorname{Im} \left( \frac{-1}{F(\lambda_{dp})} \right) - \frac{1}{2a} \operatorname{Re} \left( \frac{-1}{F(\lambda_{dp})} \right) \\ X_2 = \frac{1}{2b} \operatorname{Im} \left( \frac{-1}{F(\bar{\lambda}_{dp})} \right) + \frac{1}{2a} \operatorname{Re} \left( \frac{-1}{F(\bar{\lambda}_{dp})} \right) \end{cases} \quad (3.39)$$

— Soit  $F(s) = \frac{N(s)}{D(s)}$  où  $N(s)$  est le numérateur de  $F(s)$  et  $D(s)$  son dénominateur. Tracez le lieu des racines de la fonction de transfert *virtuelle* suivante :

$$L(s) = \frac{-N(s)(s^2 - 2as + (a^2 + b^2))}{2asD(s) + N(s)(2aX_2s^2 - 2a(a^2 + b^2)X_1)} \quad (3.40)$$

- Choisissez sur le lieu des racines de  $L(s)$  la valeur de  $K_p$  assurant que la paire  $\{\lambda_{dp}, \bar{\lambda}_{dp}\}$  constitue la paire de poles *dominants* du bouclage :
- Une fois  $K_p$  choisi, les paramètres  $K_i$  et  $K_d$  sont obtenus comme suit :

$$\begin{cases} K_i = -(a^2 + b^2) \left( \frac{K_p}{2a} + X_1 \right) \\ K_d = -\frac{K_p}{2a} + X_2 \end{cases} \quad (3.41)$$

Une fois que le correcteur PI-D est obtenu, le pré-filtre  $C_{pf}(s)$  doit être calculé pour que le dépassement obtenu soit conforme aux spécifications.

### 3.6 Commande modale

Concevez un correcteur par retour d'état (vitesse verticale dans le repère du drone et altitude du drone) permettant de commander l'altitude du drone, puis ajoutez un observateur d'état permettant d'estimer la vitesse verticale et l'altitude du drone à partir de la mesure de l'altitude de celui-ci.

Les spécifications sont les suivantes :

- Erreur de position nulle ;
- Temps de réponse  $t_s$  en boucle fermée inférieur à 0.5 sec ;
- Dépassemement  $D$  de la réponse indicielle de l'ordre de 5%.

N'oubliez pas de rajouter le terme de commande à *l'équilibre* et testez votre loi de commande avec de *petites* valeurs de consigne dans un premier temps, par exemple  $z_c = 0.1$  mètres.

La représentation d'état (3.33) permet d'exprimer la dynamique linéarisée du vecteur  $\underline{\xi}$  de la position du centre de gravité du drone :

$$\begin{cases} \dot{\underline{\xi}} = \underline{v}^b \\ \dot{\underline{v}}^b = \mathbf{A}_{13}\underline{\nu} + \frac{1}{m}\underline{f}_a^b \end{cases} \quad (3.42)$$

En posant  $\underline{f}_a^b = [ f_x^b \ f_y^b \ f_z^b ]^T$  et  $\underline{v}^b = [ u \ v \ w ]^T$ , et en tenant compte de la structure de la matrice  $\mathbf{A}_{13}$ , il vient pour la dynamique de l'altitude  $z$  du drone :

$$\begin{cases} \underline{f}_a^b = [ f_x^b \ f_y^b \ f_z^b ]^T \\ \underline{v}^b = [ u \ v \ w ]^T \\ \mathbf{A}_{13} = \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases} \Rightarrow \begin{bmatrix} \dot{z} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} f_z^b \quad (3.43)$$

Soit :

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}f_z^b \quad \text{où} \quad \underline{x} = \begin{bmatrix} z \\ w \end{bmatrix} \quad (3.44)$$

La commande par retour d'état avec une pré-commande  $\mathbf{H}$  s'écrit alors comme suit, où  $z_c$  désigne l'altitude de consigne :

$$f_z^b = -\mathbf{K}\underline{x} + \mathbf{H}z_c \quad (3.45)$$

Une fois la commande par retour d'état appliquée, la dynamique du système bouclé est la suivante :

$$\dot{\underline{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\underline{x} + \mathbf{B}\mathbf{H}z_c \quad (3.46)$$

### 3.6.1 Calcul de la pré-commande

En régime stationnaire, nous avons :

$$\underline{0} = (\mathbf{A} - \mathbf{B}\mathbf{K})\underline{x}_e + \mathbf{B}\mathbf{H}z_c \Rightarrow \underline{x}_e = -(\mathbf{A} - \mathbf{B}\mathbf{K})^{-1}\mathbf{B}\mathbf{H}z_c \quad (3.47)$$

Nous avons de plus :

$$\underline{x}_e = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} z_e \\ w_e \end{bmatrix} = \mathbf{C}\underline{x}_e = -\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{K})^{-1}\mathbf{B}\mathbf{H}z_c \quad (3.48)$$

Par conséquent l'erreur de position sera nulle, i.e.  $z_e = z_c$ , dès lors que :

$$-\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{K})^{-1}\mathbf{B}\mathbf{H} = \mathbb{I} \quad (3.49)$$

La pré-commande  $\mathbf{H}$  a donc l'expression suivante :

$$\boxed{\mathbf{H} = -\left(\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{K})^{-1}\mathbf{B}\right)^{-1}} \quad (3.50)$$

### 3.6.2 Ajout d'un intégrateur

L'ajout d'un intégrateur se fait de la manière suivante : à partir de la représentation d'état :

$$\begin{cases} \dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} \\ \underline{y} = \mathbf{C}\underline{x} \end{cases} \quad (3.51)$$

nous rajoutons le terme intégral suivant, où  $\underline{y}_c$  est le signal de consigne pour la sortie  $\underline{y}$  :

$$\underline{e}_I = \int_0^t \left( \underline{y}_c(\tau) - \underline{y}(\tau) \right) d\tau \quad (3.52)$$

La relation  $\underline{y} = \mathbf{C}\underline{x}$  entraîne :

$$\underline{y} = \mathbf{C}\underline{x} \Rightarrow \dot{\underline{e}}_I = \underline{y}_c - \underline{y} = \underline{y}_c - \mathbf{C}\underline{x} \quad (3.53)$$

Cela conduit à la représentation d'état suivante du système augmenté de l'intégrateur :

$$\begin{cases} \begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{e}}_I \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{e}_I \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \underline{u} + \begin{bmatrix} \mathbf{0} \\ \mathbb{I} \end{bmatrix} \underline{y}_c \\ \underline{y} = [\mathbf{C} \quad \mathbf{0}] \begin{bmatrix} \underline{x} \\ \underline{e}_I \end{bmatrix} \end{cases} \quad (3.54)$$

Soit  $\underline{x}_a$  le vecteur d'état du système augmenté de l'intégrateur :

$$\underline{x}_a = \begin{bmatrix} \underline{x} \\ e_I \end{bmatrix} \quad (3.55)$$

La représentation d'état (3.54) s'écrit alors :

$$\begin{cases} \dot{\underline{x}}_a = \mathbf{A}_a \underline{x}_a + \mathbf{B}_a \underline{u} + \mathbf{E} \underline{y}_c \\ \underline{y} = \mathbf{C}_a \underline{x}_a \end{cases} \quad (3.56)$$

La commande par retour d'état  $\underline{u}$  a alors l'expression suivante, où  $\mathbf{K}_p$  représente le terme *proportionnel* de la commande et  $\mathbf{K}_i$  le terme *integral* :

$$\underline{u} = -\mathbf{K}_a \underline{x}_a = -[\mathbf{K}_p \quad \mathbf{K}_i] \begin{bmatrix} \underline{x} \\ e_I \end{bmatrix} = -\mathbf{K}_p \underline{x} - \mathbf{K}_i e_I \quad (3.57)$$

Notez que l'utilisation d'un intégrateur permet d'éviter utilisation du terme de pré-commande  $\mathbf{H}$ . Le schéma bloc correspondant est représenté sur le Figure 3.2.

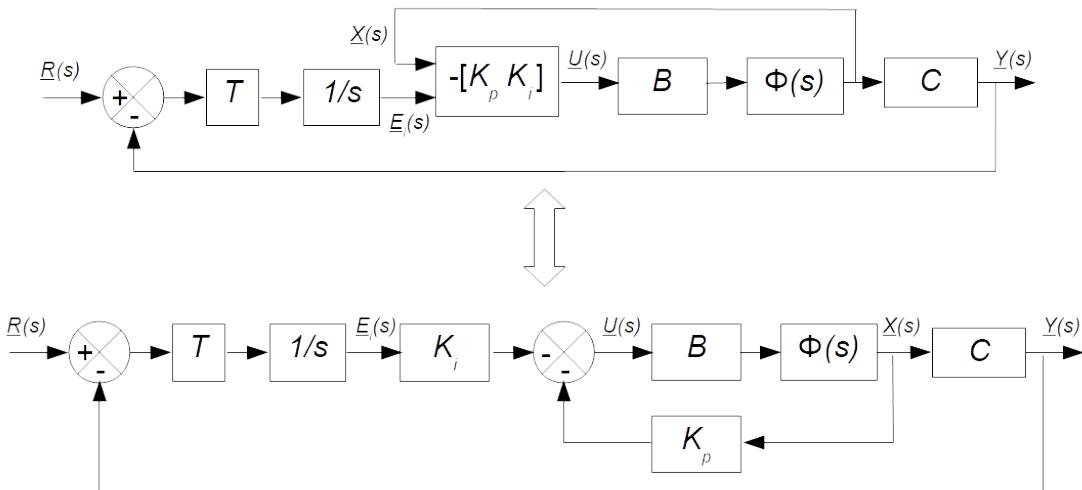


FIGURE 3.2 – Retour d'état avec utilisation d'une intégrateur

### 3.6.3 Observateur d'état

L'observateur d'état, qui a pour entrée  $z$  et pour sortie  $\hat{\underline{x}}$ , a la représentation d'état suivante :

$$\dot{\hat{\underline{x}}} = \mathbf{A} \hat{\underline{x}} + \mathbf{B} f_z^b + \mathbf{L} (z - \hat{z}) \quad \text{où} \quad \begin{cases} \hat{\underline{x}} = \begin{bmatrix} \hat{z} \\ \hat{w} \end{bmatrix} \\ \hat{z} = \mathbf{C} \hat{\underline{x}} \end{cases} \quad (3.58)$$

Lorsque l'observateur d'état est utilisé, la loi de commande devient :

$$f_z^b = -\mathbf{K} \hat{\underline{x}} + \mathbf{H} z_c \quad (3.59)$$

La représentation d'état du système bouclé est alors la suivante :

$$\begin{bmatrix} \dot{\underline{x}} \\ \ddot{\underline{x}} \end{bmatrix} = \mathbf{A}_{cl} \begin{bmatrix} \underline{x} \\ \widehat{\underline{x}} \end{bmatrix} + \begin{bmatrix} \mathbf{B}\mathbf{H} \\ \mathbf{B}\mathbf{H} \end{bmatrix} z_c \quad \text{où} \quad \mathbf{A}_{cl} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{BK} - \mathbf{LC} \end{bmatrix} \quad (3.60)$$

En notant  $\underline{e} = \underline{x} - \widehat{\underline{x}}$  l'erreur d'estimation, nous avons :

$$\underline{e} = \underline{x} - \widehat{\underline{x}} \Rightarrow \dot{\underline{e}} = (\mathbf{A} - \mathbf{LC}) \underline{e} \quad (3.61)$$

Notons que pour un observateur d'état, ce sont les valeurs propres de  $\mathbf{A} - \mathbf{LC}$  qu'il s'agit de placer. Comme les valeurs propres d'une matrice  $\mathbf{X}$  sont celle de sa transposée  $\mathbf{X}^T$ , les valeurs propres de  $\mathbf{A} - \mathbf{LC}$  sont celles de  $(\mathbf{A} - \mathbf{LC})^T = \mathbf{A}^T - \mathbf{C}^T \mathbf{L}^T$ . On utilise alors le résultat précédent en identifiant  $\mathbf{A} - \mathbf{BK}$  à  $\mathbf{A}^T - \mathbf{C}^T \mathbf{L}^T$  :

$$\begin{cases} \mathbf{A} \rightarrow \mathbf{A}^T \\ \mathbf{B} \rightarrow \mathbf{C}^T \\ \mathbf{K} \rightarrow \mathbf{L}^T \end{cases} \quad (3.62)$$

Soit  $\mathbf{P}_n$  la matrice de changement de base suivante :

$$\mathbf{P}_n = \mathbf{P}_n^{-1} = \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ \mathbb{I} & -\mathbb{I} \end{bmatrix} \quad (3.63)$$

En utilisant le fait que les valeurs propres de la matrice d'état ne dépendent pas du choix du vecteur d'état, le polynôme caractéristique du système bouclé peut être calculé comme suit :

$$\begin{aligned} \chi_{A_{cl}}(s) &= \det(s\mathbb{I} - \mathbf{A}_{cl}) \\ &= \det(\mathbf{P}_n^{-1}(s\mathbb{I} - \mathbf{A}_{cl})\mathbf{P}_n) \\ &= \det(s\mathbb{I} - \mathbf{P}_n^{-1}\mathbf{A}_{cl}\mathbf{P}_n) \\ &= \det\left(\begin{bmatrix} s\mathbb{I} - \mathbf{A} + \mathbf{BK} & -\mathbf{BK} \\ \mathbf{0} & s\mathbb{I} - \mathbf{A} + \mathbf{LC} \end{bmatrix}\right) \\ &= \det(s\mathbb{I} - \mathbf{A} + \mathbf{BK}) \det(s\mathbb{I} - \mathbf{A} + \mathbf{LC}) \end{aligned} \quad (3.64)$$

Par conséquent, les valeurs propres du système en boucle fermée sont obtenues comme l'union des valeurs propres de la matrice  $\mathbf{A} - \mathbf{BK}$ , c'est-à-dire de la matrice d'état du système en boucle fermée sans l'observateur de gain  $\mathbf{L}$ , et des valeurs propres de la matrice  $\mathbf{A} - \mathbf{LC}$ , c'est-à-dire la matrice d'état du système en boucle fermée sans le retour d'état de gain  $\mathbf{K}$ . Ce résultat est connu sous le nom de *principe de séparation*.

### 3.7 Commande optimale (cas MIMO)

Concevez un correcteur par retour d'état en utilisant la commande optimale pour commander la position selon les axes  $x$  et  $y$  du plan horizontal du repère inertiel. Vous ajouterez un intégrateur dans la commande par retour d'état afin de diminuer la sensibilité de la loi de commande vis-à-vis de la vitesse angulaire  $\nu$ .

Les spécifications sont les suivantes :

- Erreur de position nulle ;
- Temps de réponse  $t_s$  en boucle fermée inférieur à 0.5 sec ;
- Dépassement  $D$  de la réponse indicielle de l'ordre de 5%.

La représentation d'état (3.33) permet d'exprimer la dynamique linéarisée du vecteur  $\underline{\xi}$  de la position du centre de gravité du drone :

$$\begin{cases} \dot{\underline{\xi}} = \underline{v}^b \\ \dot{\underline{v}}^b = \mathbf{A}_{13}\underline{\nu} + \frac{1}{m}\underline{f}_a^b \end{cases} \quad (3.65)$$

En posant  $\underline{f}_a^b = [ f_x^b \ f_y^b \ f_z^b ]^T$  et  $\underline{v}^b = [ u \ v \ w ]^T$ , et en tenant compte de la structure de la matrice  $\mathbf{A}_{13}$ , il vient pour la dynamique des positions  $x$  et  $y$  du drone dans le plan horizontal :

$$\begin{cases} \underline{f}_a^b = [ f_x^b \ f_y^b \ f_z^b ]^T \\ \underline{v}^b = [ u \ v \ w ]^T \\ \mathbf{A}_{13} = \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases} \Rightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ u \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \end{bmatrix} \begin{bmatrix} f_x^b \\ f_y^b \end{bmatrix} + \underline{n}$$
(3.66)

où  $\underline{n}$  sera considéré comme une *perturbation* : il s'agit du terme de la dynamique dépendant des composantes de la vitesse angulaire  $\underline{\nu}$ . Le vecteur  $\underline{n}$  a l'expressin suivante :

$$\underline{n} = g \begin{bmatrix} 0 \\ 0 \\ -q \\ p \end{bmatrix} \quad (3.67)$$

Il vient :

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B} \begin{bmatrix} f_x^b \\ f_y^b \end{bmatrix} + \underline{n} \quad \text{où} \quad \underline{x} = \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix} \quad (3.68)$$

### 3.7.1 Méthode *Linear Quadratic Tracker* (LQT)

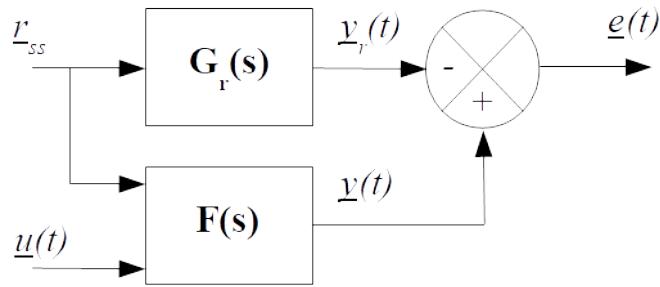


FIGURE 3.3 – Problème LQT (*Linear Quadratic Tracker*)

Intéressons nous à la Figure 3.3 et au problème consistant à concevoir une commande  $u(t)$  minimisant l'erreur  $e(t)$  entre la sortie  $y_r(t)$  d'un modèle de référence représenté par  $\mathbf{G}_r(s)$  et la sortie  $y(t)$  du système à contrôler représenter par  $\mathbf{F}(s)$ .

Les réalisations de  $\mathbf{G}_r(s)$  et de  $\mathbf{F}(s)$  sont les suivantes, où  $\underline{r}_{ss}$  représente un signal de référence constant :

$$\begin{cases} \mathbf{F}(s) : \begin{cases} \dot{\underline{x}}(t) = \mathbf{A}\underline{x}(t) + \mathbf{B}\underline{u}(t) \\ \underline{y}(t) = \mathbf{C}\underline{x}(t) \end{cases} \\ \mathbf{G}_r(s) : \begin{cases} \dot{\underline{x}}_r(t) = \mathbf{A}_r\underline{x}_r(t) + \mathbf{B}_r\underline{r}_{ss} \\ \underline{y}_r(t) = \mathbf{C}_r\underline{x}_r(t) \end{cases} \quad \text{où } \dot{\underline{r}}_{ss} = \frac{d\underline{r}_{ss}}{dt} = 0 \end{cases} \quad (3.69)$$

L'erreur de suivi de trajectoire  $\underline{e}(t)$  s'écrit :

$$\underline{e}(t) := \underline{y}(t) - \underline{y}_r(t) = \mathbf{C}\underline{x}(t) - \mathbf{C}_r\underline{x}_r(t) \quad (3.70)$$

Afin de résoudre ce problème, calculons d'abord les valeurs à l'équilibre où  $\underline{y}_{ss} = \underline{y}_{r_{ss}}$ . A partir de (3.69) il vient :

$$\begin{cases} \underline{0} = \mathbf{A}\underline{x}_{ss} + \mathbf{B}\underline{u}_{ss} \\ \underline{0} = \mathbf{A}_r\underline{x}_{r_{ss}} + \mathbf{B}_r\underline{r}_{ss} \\ \underline{y}_{ss} = \underline{y}_{r_{ss}} \Leftrightarrow \mathbf{C}\underline{x}_{ss} = \mathbf{C}_r\underline{x}_{r_{ss}} \end{cases} \Leftrightarrow \begin{bmatrix} \mathbf{0} \\ -\mathbf{B}_r \\ \mathbf{0} \end{bmatrix} \underline{r}_{ss} = \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{A}_r & \mathbf{0} \\ \mathbf{C} & -\mathbf{C}_r & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{x}_{ss} \\ \underline{x}_{r_{ss}} \\ \underline{u}_{ss} \end{bmatrix} \quad (3.71)$$

Soit  $\mathbf{N}$  la matrice suivante :

$$\mathbf{N} := \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{A}_r & \mathbf{0} \\ \mathbf{C} & -\mathbf{C}_r & \mathbf{0} \end{bmatrix} \quad (3.72)$$

En supposant que  $\mathbf{N}$  soit inversible, nous avons :

$$\begin{bmatrix} \underline{x}_{ss} \\ \underline{x}_{r_{ss}} \\ \underline{u}_{ss} \end{bmatrix} = \mathbf{N}^{-1} \begin{bmatrix} \mathbf{0} \\ -\mathbf{B}_r \\ \mathbf{0} \end{bmatrix} \underline{r}_{ss} \quad (3.73)$$

Soit :

$$\mathbf{N}^{-1} \begin{bmatrix} \mathbf{0} \\ -\mathbf{B}_r \\ \mathbf{0} \end{bmatrix} := \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \\ \mathbf{N}_3 \end{bmatrix} \quad (3.74)$$

Ainsi :

$$\begin{bmatrix} \underline{x}_{ss} \\ \underline{x}_{r_{ss}} \\ \underline{u}_{ss} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \\ \mathbf{N}_3 \end{bmatrix} \underline{r}_{ss} \quad (3.75)$$

Soit  $\tilde{\underline{x}}(t)$  l'écart entre la valeur du vecteur d'état  $\underline{x}(t)$  et sa valeur à l'équilibre  $\underline{x}_{ss}$ ,  $\tilde{\underline{x}}_r(t)$  l'écart entre la valeur du vecteur d'état du modèle de référence  $\underline{x}_r(t)$  et sa valeur à l'équilibre  $\underline{x}_{r_{ss}}$  et  $\tilde{\underline{u}}(t)$  l'écart entre la valeur du vecteur de commande  $\underline{u}(t)$  et sa valeur à l'équilibre  $\underline{u}_{ss}$  :

$$\begin{cases} \tilde{\underline{x}}(t) := \underline{x}(t) - \underline{x}_{ss} \\ \tilde{\underline{x}}_r(t) := \underline{x}_r(t) - \underline{x}_{r_{ss}} \\ \tilde{\underline{u}}(t) := \underline{u}(t) - \underline{u}_{ss} \end{cases} \quad (3.76)$$

En utilisant (3.76), la relation  $\tilde{\underline{x}}(t)$  s'écrit :

$$\begin{aligned}\dot{\tilde{\underline{x}}}(t) &= \dot{\underline{x}}(t) \\ &= \mathbf{A}\underline{x}(t) + \mathbf{B}\underline{u}(t) \\ &= \mathbf{A}(\tilde{\underline{x}}(t) + \underline{x}_{ss}) + \mathbf{B}(\tilde{\underline{u}}(t) + \underline{u}_{ss}) \\ &= \mathbf{A}\tilde{\underline{x}}(t) + \mathbf{B}\tilde{\underline{u}}(t) + \mathbf{A}\underline{x}_{ss} + \mathbf{B}\underline{u}_{ss}\end{aligned}\quad (3.77)$$

De même :

$$\begin{aligned}\dot{\tilde{\underline{x}}}_r(t) &= \dot{\underline{x}}_r(t) \\ &= \mathbf{A}_r\underline{x}_r(t) + \mathbf{B}_r\underline{r}_{ss} \\ &= \mathbf{A}_r(\tilde{\underline{x}}_r(t) + \underline{x}_{r_{ss}}) + \mathbf{B}_r\underline{r}_{ss} \\ &= \mathbf{A}_r\tilde{\underline{x}}_r(t) + \mathbf{A}_r\underline{x}_{r_{ss}} + \mathbf{B}_r\underline{r}_{ss}\end{aligned}\quad (3.78)$$

En utilisant (3.71), il est clair que  $\mathbf{A}\underline{x}_{ss} + \mathbf{B}\underline{u}_{ss} = \underline{0}$  et  $\mathbf{A}_r\underline{x}_{r_{ss}} + \mathbf{B}_r\underline{r}_{ss} = \underline{0}$ . Nous obtenons finalement l'équation d'état suivante :

$$\begin{bmatrix} \dot{\tilde{\underline{x}}}(t) \\ \dot{\tilde{\underline{x}}}_r(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \tilde{\underline{x}}(t) \\ \tilde{\underline{x}}_r(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \tilde{\underline{u}}(t) := \mathbf{A}_a \begin{bmatrix} \tilde{\underline{x}}(t) \\ \tilde{\underline{x}}_r(t) \end{bmatrix} + \mathbf{B}_a \tilde{\underline{u}}(t) \quad (3.79)$$

De plus, l'erreur de suivi définie en (3.70) devient :

$$\begin{aligned}\underline{e}(t) &:= \underline{y}(t) - \underline{y}_r(t) \\ &= \mathbf{C}\underline{x}(t) - \mathbf{C}_r\underline{x}_r(t) \\ &= \mathbf{C}(\tilde{\underline{x}}(t) + \underline{x}_{ss}) - \mathbf{C}_r(\tilde{\underline{x}}_r(t) + \underline{x}_{r_{ss}}) \\ &= \mathbf{C}\tilde{\underline{x}}(t) - \mathbf{C}_r\tilde{\underline{x}}_r(t) + \mathbf{C}\underline{x}_{ss} - \mathbf{C}_r\underline{x}_{r_{ss}}\end{aligned}\quad (3.80)$$

En utilisant la dernière équation de (3.71), nous obtenons finalement l'équation de sortie suivante :

$$\begin{aligned}\mathbf{C}\underline{x}_{ss} = \mathbf{C}_r\underline{x}_{r_{ss}} \Rightarrow \underline{e}(t) &= \mathbf{C}\tilde{\underline{x}}(t) - \mathbf{C}_r\tilde{\underline{x}}_r(t) \\ &= [\mathbf{C} \quad -\mathbf{C}_r] \begin{bmatrix} \tilde{\underline{x}}(t) \\ \tilde{\underline{x}}_r(t) \end{bmatrix} \\ &:= \mathbf{C}_a \begin{bmatrix} \tilde{\underline{x}}(t) \\ \tilde{\underline{x}}_r(t) \end{bmatrix}\end{aligned}\quad (3.81)$$

Considérons alors le problème de la minimisation du critère suivant sous les *contraintes* (3.79) et (3.81) :

$$J(\tilde{\underline{u}}(t)) = \frac{1}{2} \int_0^\infty \underline{e}^T(t) \mathbf{Q} \underline{e}(t) + \tilde{\underline{u}}^T(t) \mathbf{R} \tilde{\underline{u}}(t) dt \quad (3.82)$$

La minimisation de  $J(\tilde{\underline{u}}(t))$  est obtenue en appliquant les résultats classiques de la commande LQR. La commande  $\tilde{\underline{u}}(t)$  s'écrit :

$$\tilde{\underline{u}}(t) = -\mathbf{K}_a \begin{bmatrix} \tilde{\underline{x}}(t) \\ \tilde{\underline{x}}_r(t) \end{bmatrix} \text{ où } \mathbf{K}_a = \mathbf{R}^{-1} \mathbf{B}_a^T \mathbf{P} \quad (3.83)$$

La matrice  $\mathbf{P}$  est la solution symétrique définie positive de l'équation algébrique de Riccati suivante :

$$\boxed{\mathbf{0} = \mathbf{A}_a^T \mathbf{P} + \mathbf{P} \mathbf{A}_a - \mathbf{P} \mathbf{B}_a \mathbf{R}^{-1} \mathbf{B}_a^T \mathbf{P} + \mathbf{C}_a^T \mathbf{Q} \mathbf{C}_a} \quad (3.84)$$

La commande  $\underline{u}(t)$  à appliquer sur le système est finalement obtenue à partir de (3.76) :

$$\begin{aligned}\underline{u}(t) &= \tilde{\underline{u}}(t) + \underline{u}_{ss} \\ &= -\mathbf{K}_a \begin{bmatrix} \tilde{\underline{x}}(t) \\ \tilde{\underline{x}}_r(t) \end{bmatrix} + \underline{u}_{ss} \\ &= -\mathbf{K}_a \begin{bmatrix} \underline{x}(t) \\ \underline{x}_r(t) \end{bmatrix} + \mathbf{K}_a \begin{bmatrix} \underline{x}_{ss} \\ \underline{x}_{rss} \end{bmatrix} + \underline{u}_{ss}\end{aligned}\quad (3.85)$$

En scindant  $\mathbf{K}_a$  sous la forme  $\mathbf{K}_a := [\mathbf{K}_x \quad \mathbf{K}_r]$  et en utilisant (3.75) il vient :

$$\begin{aligned}\underline{u}(t) &= -\mathbf{K}_a \begin{bmatrix} \underline{x}(t) \\ \underline{x}_r(t) \end{bmatrix} + \mathbf{K}_a \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \end{bmatrix} \underline{r}_{ss} + \mathbf{N}_3 \underline{r}_{ss} \\ &:= -[\mathbf{K}_x \quad \mathbf{K}_r] \begin{bmatrix} \underline{x}(t) \\ \underline{x}_r(t) \end{bmatrix} + \mathbf{D}_{pf} \underline{r}_{ss} \\ &:= -\mathbf{K}_x \underline{x}(t) + \left( \underbrace{-\mathbf{K}_r \underline{x}_r(t) + \mathbf{D}_{pf} \underline{r}_{ss}}_{\underline{y}_{pf}(t)} \right)\end{aligned}\quad (3.86)$$

où :

$$\begin{cases} \underline{y}_{pf}(t) = -\mathbf{K}_r \underline{x}_r(t) + \mathbf{D}_{pf} \underline{r}_{ss} \\ \mathbf{D}_{pf} := \mathbf{K}_a \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \end{bmatrix} + \mathbf{N}_3 \end{cases}\quad (3.87)$$

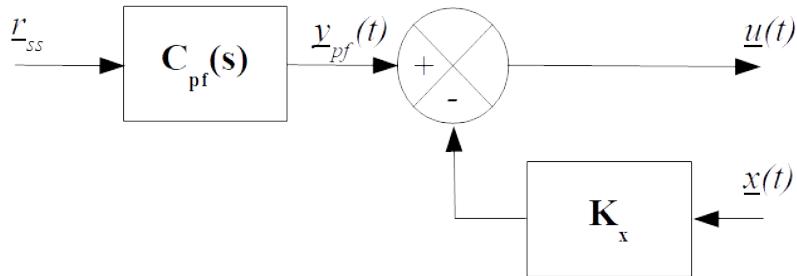


FIGURE 3.4 – structure LQT (*Linear Quadratic Tracker*) avec pré-filtre

Par conséquent la commande optimale  $\underline{u}(t)$ s'obtient comme la somme de deux termes :

- un terme de retour d'état :  $-\mathbf{K}_x \underline{x}(t)$  ;
- un terme  $\underline{y}_{pf}(t)$  représentant la sortie du pré-filtre  $\mathbf{C}_{pf}(s)$  dont la réalisation est la suivante :

$$\mathbf{C}_{pf}(s) : \begin{cases} \dot{\underline{x}}_r(t) = \mathbf{A}_r \underline{x}_r(t) + \mathbf{B}_r \underline{r}_{ss} \\ \underline{y}_{pf}(t) = -\mathbf{K}_r \underline{x}_r(t) + \mathbf{D}_{pf} \underline{r}_{ss} \end{cases}\quad (3.88)$$

Cette structure est illustrée sur la Figure 3.4.



## Chapitre 4

# Projet de Synthèse

### 4.1 Scénario

Le projet consiste en la synthèse de lois de commande pour la gestion de la descente d'un avion de transport.

Le scénario associé à ce projet est représenté sur la Figure 4.1 : un avion de transport commercial stable au niveau de vol et à la vitesse donnés dans la Table 4.1 est autorisé à faire une descente continue (c'est le mode *open descent* qui sera utilisé) jusqu'au niveau de vol 30 où il devra se stabiliser à une vitesse équivalente de 230 noeuds.

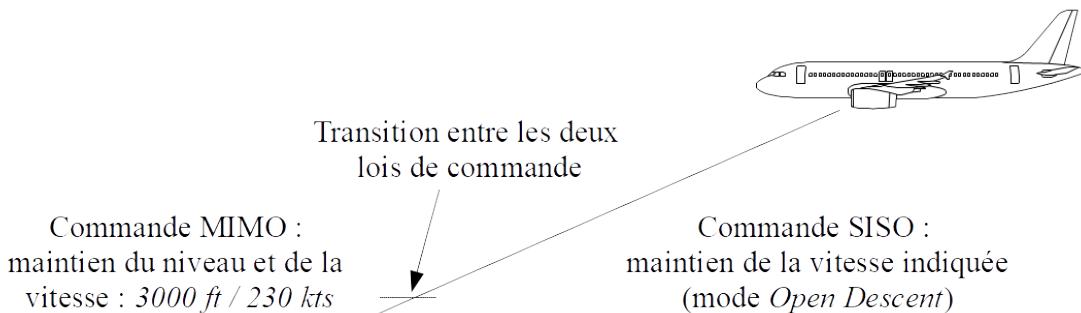


FIGURE 4.1 – Scénario associé au projet

L'avion décélère ensuite et sort les configurations hyper-sustentatrices pour la préparation de l'atterrissage. Cette phase ne fait pas partie du projet et ne sera donc pas simulée.

Sur les avions de transport modernes, et comme illustré sur la Figure 1.1, le pilote sélectionne la consigne de niveau de vol et de vitesse sur le FCU (Flight Control Unit sur Airbus ou Mode Control Panel sur Boeing) puis engage les lois de capture d'altitude et de maintien de la vitesse en tirant sur la molette. Le Primary Flight Display (PFD) affiche la vitesse indiquée et le niveau de vol courants en vert et la vitesse indiquée et le niveau de vol de consigne en bleu.

Dans la suite de ce projet, nous identifierons la vitesse indiquée de l'avion à sa vitesse équivalente.

## 4.2 Organisation

Les étudiants de chaque groupe de TD sont répartis en équipes. Chaque équipe sera constituée de 2 ou 3 étudiants du même groupe de TD. Chaque équipe étudiera une configuration spécifique donnée par un type d'avion, une marge statique  $ms$  et un coefficient de masse  $km$ , comme indiqué dans la Table 4.1. Chaque équipe étudiera bien évidemment une configuration d'avion différente de celle des autres équipes.

Équipe	Avion	altitude (FL)	<i>EAS</i> (vitesse équivalente, noeuds)	$ms$	$km$
Alpha	Airbus A-320	130	260	0.2	0.5
Bravo	Boeing 737-800	150	260	0.2	0.2
Charlie	Airbus A-319	200	270	0.3	0.7
Delta	Airbus A-321	250	280	0.3	0.9
Echo	Boeing 737-700	300	290	0.4	0.8
Foxtrot	Boeing 737-300	130	260	0.5	0.3
Golf	Airbus A-320	150	260	0.5	0.3
Hotel	Boeing 737-800	200	270	0.4	0.5
India	Airbus A-319	250	280	0.4	0.4
Juliette	Airbus A-321	290	290	0.3	0.7

TABLE 4.1 – Constitution des équipes

Notez que c'est l'équivalent de vitesse, (en anglais, *EAS* : Equivalent AirSpeed), c'est à dire la vitesse de l'aéronef corrigée des effets de la compressibilité à l'altitude donnée, qui est utilisée dans la table 4.1. Jusqu'à environ 200 noeuds *CAS* et 10000 pieds, la différence entre *CAS* et *EAS* est négligeable, mais à des vitesses et des altitudes plus élevées, la *CAS* doit être corrigée de l'erreur de compressibilité afin de déterminer l'*EAS*<sup>1</sup>.

## 4.3 Travail à effectuer

Chaque équipe se verra affecter un avion (dont la dynamique a été étudiée au cours du projet de première année) associé à une marge statique et un coefficient de masse : cf. la Table 4.1.

Le modèle avion est commandé par les trois entrées suivantes :

- Le braquage du plan horizontal réglable, noté  $\delta_{PHR}$  ;
- La position de la commande de poussée, notée  $\delta_{th}$  ;
- Le braquage de la gouverne de profondeur, noté  $\delta_{el}$

Le PHR est utilisé uniquement pour le trim. Les simulations sont réalisées à la valeur de trim du PHR.

Les techniques de commande suivantes qui ont été présentées au cours de ce semestre seront utilisées pour mener à bien le projet :

- Commande classique pour la boucle de commande de l'assiette (*pitch loop*) ;
- Commande modale (par retour d'état) pour la loi de descente en mode *open descent* ;
- Commande optimale (qui est une commande particulière par retour d'état) pour les lois de maintien de niveau et de vitesse.

---

1. [https://mediawiki.ivao.aero/index.php?title=Airspeed\\_definition](https://mediawiki.ivao.aero/index.php?title=Airspeed_definition)

## 4.4 Prise en main du modèle Matlab® / Simulink®

L'objet de cette section est d'obtenir un modèle linéarisé de l'avion étudié et d'identifier les modes lents et à rapides.

Le répertoire *web\_fichiersProjet* qui contient les fichiers Matlab® / Simulink® utiles pour ce projet pourra être chargé sur la page suivante :

<http://recherche.enac.fr/~thierry.mique1/>

Le schéma Simulink® de l'avion naturel est représenté sur la Figure 4.2.

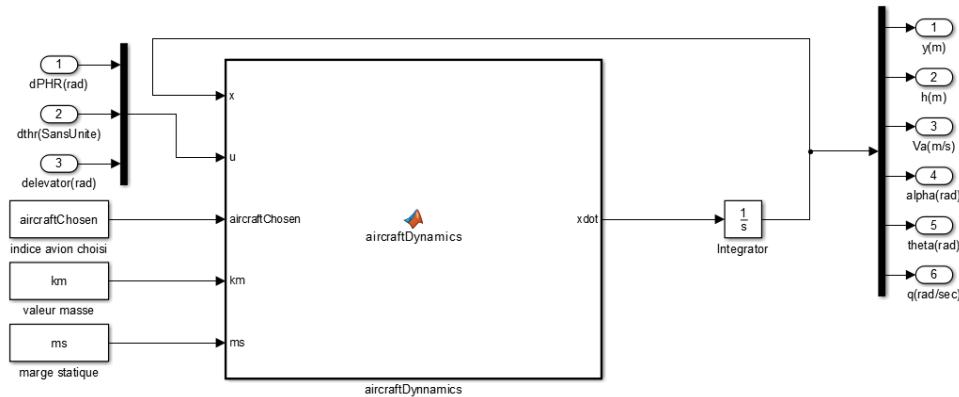


FIGURE 4.2 – Schéma Simulink® de l'avion naturel

### 4.4.1 Point d'équilibre

Trouver les valeurs du vecteur de commande  $u$  et du vecteur d'état  $x$  associées au point de trim de l'avion. A cet effet vous pourrez utiliser la fonction *utComputeTrimIdle.m* située dans le répertoire *web\_fichiersProjet* après avoir analysé son fonctionnement. Vous pourrez aussi valider la solution proposée en utilisant la fonction *utAcDynamicsFunction.m* codant la dynamique longitudinale de l'avion.

### 4.4.2 Linéarisation

Linéariser numériquement le modèle Simulink™ *acDynModel\_ToLinearize.slx* autour du point de trim et obtenir la matrice d'état  $\mathbf{A}$  et la matrice de commande  $\mathbf{B}$  du modèle linéarisé. Pour cela vous pourrez utiliser la fonction *linmod* de Matlab®.

### 4.4.3 Modes

A partir des matrices  $\{\mathbf{A}, \mathbf{B}\}$  extraire les matrices  $\mathbf{A}_4$  et  $\mathbf{B}_4$  associées aux composantes  $[\delta V_a \ \delta \alpha \ \delta \theta \ q]^T$  du vecteur d'état et  $\delta_{el}$  du vecteur d'entrée.

Calculer numériquement les valeurs propres de la matrice  $\mathbf{A}_4$  ainsi que le coefficient d'amortissement et la pulsation naturelle associés à chaque paire de valeurs propres complexes conjuguées. Vous pourrez utiliser les commandes *eig()* et *damp()* de Matlab®. Rappeler le nom des modes associées aux grandes et aux petites échelles de temps.

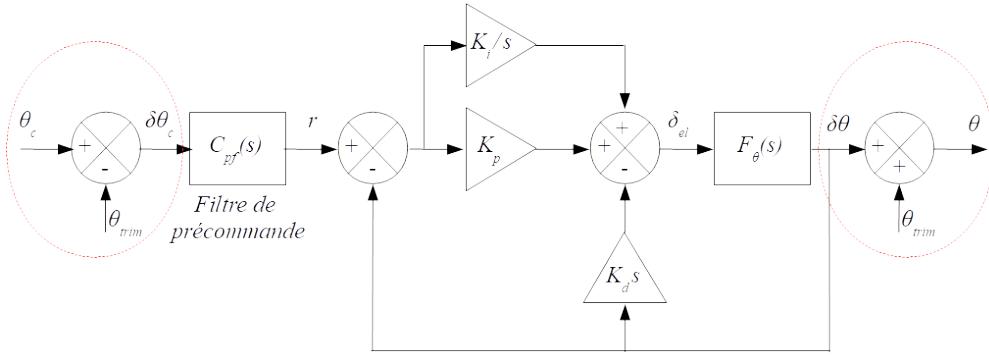


FIGURE 4.3 – Boucle de tenue d'assiette

## 4.5 Commande classique : boucle de tenue d'assiette

L'objet de cette section est la réalisation d'une boucle de tenue de l'assiette de l'avion en utilisant les techniques de la commande classique.

C'est sur cette boucle que s'appuiera la boucle externe de commande de la vitesse équivalente durant la descente de l'avion. Durant la phase de descente la manette de poussée sera sur la position *idle*. Par la suite la poussée *idle* sera associée à la valeur suivante de  $\delta thr$  :

$$\delta thr = 0.1 \quad (4.1)$$

La boucle de tenue d'assiette pour le modèle linéarisé est présentée sur la Figure 4.3 où  $\theta$  désigne l'assiette de l'avion et  $\theta_c$  l'assiette de consigne. Les grandeurs  $\delta\theta$  et  $\delta\theta_c$  représentent les écarts par rapport au point de trim choisi. C'est sur ces écarts que fonctionneront les correcteurs que vous allez réaliser.

La fonction de transfert  $F_\theta(s) = \frac{\delta\theta(s)}{\delta_{el}(s)}$  entre la variation de la position de la gouverne de profondeur  $\delta_{el}(s)$  et la variation d'assiette  $\delta\theta(s)$  est obtenue à partir du modèle avion linéarisé autour du point de trim  $\theta_{trim}$  comme indiqué par les cercles en pointillé sur la Figure 4.3. Pour l'implémentation sous Simulink®, vous devrez compléter le schéma de la Figure 4.4.

Les spécifications sont les suivantes :

- Erreur de position nulle;
- Temps de réponse en boucle fermée inférieur à 3 sec ;
- Dépassemement de la réponse indicielle inférieur à 5%.

Afin de satisfaire les spécifications votre mission est de synthétiser un correcteur PI-D, dont les paramètres sont  $K_p$ ,  $K_i$  et  $K_d$ , ce dernier paramètre étant placé dans la chaîne de retour, ainsi qu'un filtre de précommande, noté  $C_{pf}(s)$  sur la Figure 4.3.

Vous pourrez lancer les simulations sur 10 et 300 secondes afin de constater l'évolution de l'assiette. Visualisez l'amplitude de la commande de votre système. Qu'en pensez vous ?

## 4.6 Commande modale : boucle de tenue de la vitesse équivalente

L'objectif de cette section est de réaliser la régulation de la vitesse équivalente de l'avion grâce aux techniques issues de la commande modale. Vous allez réaliser cette commande en vous appuyant sur la boucle de contrôle de l'assiette réalisée précédemment.

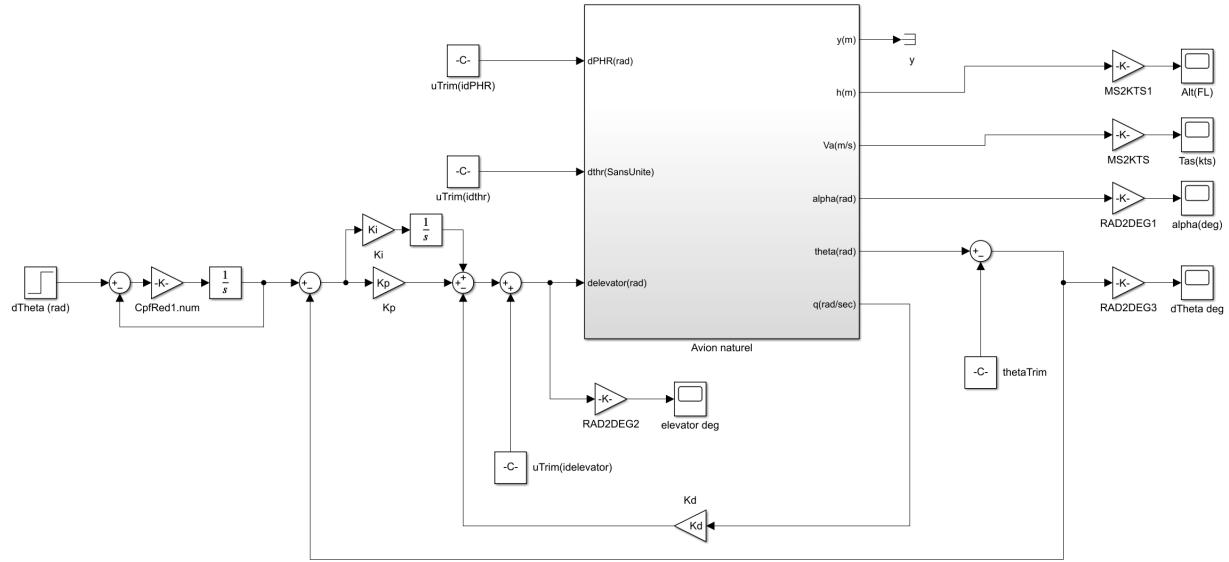


FIGURE 4.4 – Schéma Simulink® de la boucle de tenue d'assiette

La Figure 4.5 représente le bouclage proposé pour réaliser ce mode *Open Descent*. Noter que la vitesse équivalente de consigne est convertie en vitesse aérodynamique de consigne grâce à la fonction Eas2Tas.

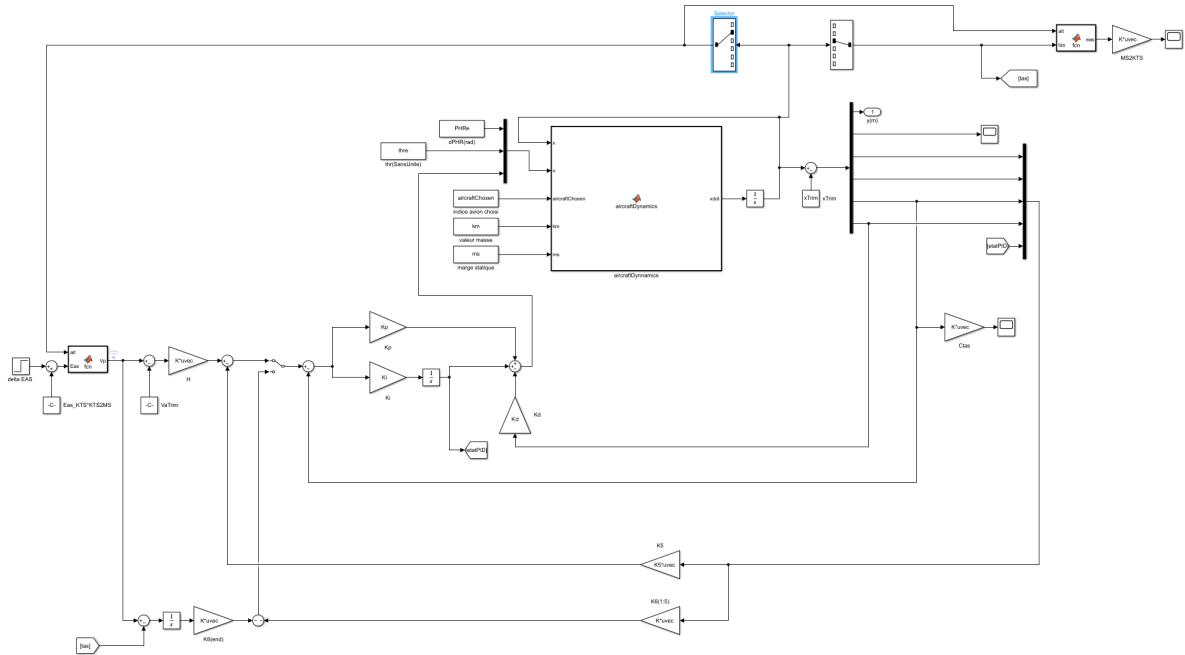


FIGURE 4.5 – Bouclage proposé pour réaliser le mode *Open Descent*

La synthèse de cette loi de commande sera réalisée dans le cas mono-entrée mono-sortie où seule l'assiette de consigne sera utilisée pour commander la vitesse équivalente de l'avion, la manette de poussée étant maintenue sur la position *idle*.

Comme il s'agit ici de réaliser une boucle de *régulation*, les spécifications se focalisent uniquement sur les pôles souhaités en boucle fermée : ceux-ci devront tous se situer à gauche du demi plan complexe  $Re = -1.3 \text{ rad/sec}$ . Même s'il s'agit de faire de la *régulation* autour de la vitesse de trim, n'oubliez pas de synthétiser le filtre de précommande  $H$  (la commande s'écrira donc  $u = -\mathbf{K}x + Hr$ ) : celui ci n'est pas nécessaire avec le modèle linéarisé mais est devient important avec le modèle non-linéaire.

Votre mission est de synthétiser un correcteur par retour d'état basé sur vos connaissances en commande modale (placement de poles) et permettant de satisfaire les spécifications.

Une fois que la commande par retour d'état aura été conçue et testée, vous synthétiserez un observateur d'état en utilisant les trois sorties suivantes :  $V_a$ ,  $\theta$  et  $q$ .

Que constatez vous sur la régulation de la vitesse du modèle de validation (modèle non-linéaire) bouclés par le correcteur que vous venez de concevoir sur une *grande échelle de temps* (600 secondes par exemple) ? Quelle est la cause de l'erreur sur la régulation de la vitesse ? Quelle solution proposez vous pour résoudre ce problème ? Une piste possible est l'ajout d'un intégrateur. Dans ce cas le filtre de précommande  $H$  n'est plus utile. Le bouclage est alors celui de la Figure 4.5 avec le *switch* à gauche du comparateur placé à position basse.

Enfin visualisez l'amplitude de la commande de votre système. Qu'en pensez vous ?

## 4.7 Commande optimale (cas MIMO)

Comme indiqué dans l'introduction, vous supposerez que l'avion se stabilise au niveau de vol 30 où il devra se stabiliser à une vitesse aérodynamique de 230 noeuds. La loi de commande que vous devez synthétiser est indépendante des deux autres lois de commande. Il s'agit d'une loi de commande multi-entrées multi-sorties où la position de la commande de poussée  $\delta_{thr}$  et le braquage de la gouverne de profondeur  $\delta_{el}$  seront utilisés pour commander la vitesse aérodynamique de l'avion et son altitude.

Afin de simplifier le problème, vous synthétiserez dans un premier temps une loi permettant de réaliser une capture de la vitesse aérodynamique et de la *pente*, et non de l'altitude. Les spécifications sont les suivantes :

- Capture de la vitesse aérodynamique : erreur de position nulle ; temps de réponse en boucle fermée inférieur à 20 sec ; pas de dépassement de la réponse indicielle.
- Capture de *pente* : erreur de position nulle, temps de réponse en boucle fermée inférieur à 2 sec ; pas de dépassement de la réponse indicielle.

Cette partie vous permettra de mettre en œuvre la commande LQT (*Linear Quadratic Tracker*).

N'oublier pas de débuter la synthèse avec un modèle avion linéarisé autour d'un vol en palier. Pour cela vous devez renommer la fonction *utComputeTrimIdle.m* qui vous a été fournie en *utComputeTrimLevelFlight.m* par exemple et l'adapter au cas du vol en palier. Indiquez si la synthèse réalisée est découplante. Visualisez l'amplitude des deux commandes de votre système. Qu'en pensez vous ? Enfin, une fois que la loi de capture de pente est réalisée, comment pouvez vous réaliser simplement une loi de capture d'altitude ?

## 4.8 Et après...

Réalisez la simulation complète depuis le début de la descente jusqu'à la stabilisation en niveau en gérant la transition entre la loi *Open Descent* et les lois de maintien en niveau et en vitesse. Commentez vos résultats de simulation et donnez des pistes pour améliorer le réalisme de celles-ci.

## 4.9 Évaluation

Vous réaliserez une implémentation des différentes lois de commande et rédigerez un rapport présentant une synthèse de vos choix de conception ainsi que les résultats de vos simulations en justifiant vos choix.

Pour mémoire, le lien de la page e-campus sur la maîtrise de l'information est le suivant : <https://e-campus.enac.fr/moodle/course/view.php?id=15866>.

Le rapport sera à remettre avant la soutenance orale. Vous présenterez pendant 15 minutes les éléments clefs de votre rapport, puis une séance de questions suivra.

Chaque personne de l'équipe devra prendre la parole durant cet oral. L'équipe sera ensuite interrogée sur l'ensemble des séances. Nous vous demandons de prendre en compte les points suivants :

- Citez vos sources (aussi bien pour les images que le texte) ;
- N'hésitez à grossir la taille de police des légendes, axes et échelles des graphes, ainsi que l'épaisseur des courbes pour que ça soit lisible de loin ;
- Ne pas surcharger les slides en courbes ;
- Ne cherchez pas à *cacher* les maths ;
- Pour les résultats de simulation, indiquez clairement quel est le modèle utilisé (non linéaire ou linéarisé, etc...) ;
- Vous serez interrogés sur le code Matlab<sup>®</sup> / Simulink<sup>®</sup>. Maîtrisez les instructions qu'il contient !

Vous serez évalués sur les points suivants :

- Soutenance orale :
  - Maîtrise du temps
  - Qualité des diapositives
  - Qualité de l'expression
- Séance de questions :
  - Maîtrise du socle, basé sur chacune des techniques de commande
  - Qualité et maîtrise du code Matlab<sup>®</sup> / Simulink<sup>®</sup> réalisé
  - Regard critique, prise de recul, pertinence, exploration et limite de validité des solutions proposées

La grille d'évaluation qui sera utilisée par le jury est présentée ci après.

			Note	Remarques
	Soutenance orale (note sur 5 points)	Critères de notation		
Maitrise du temps /1,5 points		temps < 11 min => 0 point 11 min < temps < 13 min ou > 17 min => 1 point 13 min < temps < 17 min => 1,5 points		
Qualité des diapositives /1,5 points		figures lisibles et avec unités => 0,5 point Slides numérotées => 0,5 point Slides lisibles => 0,5 points		
Qualité de l'expression /2 points		Plan clair et cohérent => 0,5 point Introduction présente et claire => 0,5 point Conclusion présente et claire => 0,5 point Articulation diapo / discours permettant de donner une information claire = tous les items de la slide sont adressés orallement => 0,5 point		
<b>Séance de questions (note sur 15 points)</b>		Socle, basé sur chacune des techniques de commande: Commande classique, sur 3 points Commande modale, sur 3 points Commande optimale, sur 3 points  Code Matlab/Simulink: savoir expliquer ce que font les lignes de code Matlab / des blocs Simulink désignés par le jury: sur 3 points  Regard critique, prise de recul, pertinence, exploration et limite de validité des solutions proposées: sur 3 points		

## Chapitre 5

# Annexe : Modélisation de la dynamique longitudinale d'un avion

Nous reprenons le modèle de la dynamique longitudinale d'un avion de transport commercial que vous avez manipulé en Python en première année. Dans le cadre de ce projet ce modèle a été repris et codé en Matlab® / Simulink®.

### 5.1 Hypothèses et notations

Nous considérons un avion évoluant selon le plan vertical comme représenté sur la figure 5.1.

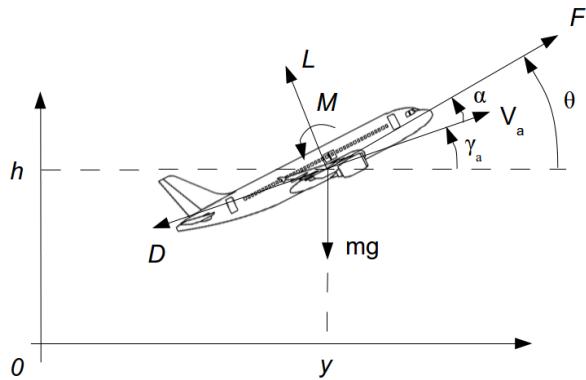


FIGURE 5.1 – Avion évoluant dans le plan vertical

Nous noterons par la suite :

- $y$  la position horizontale de l'avion exprimée en  $m$ ;
- $h$  l'altitude pression exprimée en  $m$ ;
- $m$  la masse de l'avion exprimée en  $kg$  et que nous supposerons constante;
- $g$  l'accélération de la pesanteur :  $g = 9.80665 \text{ m/s}^2$ ;
- $V_a$  la vitesse aérodynamique de l'avion exprimée en  $m/s$ ;

- $L$  (lift) la force de portance, exprimée en  $N$  et perpendiculaire à la vitesse  $V_a$  ;
- $D$  (drag) la force de trainée, exprimée en  $N$  et orientée à l'opposé de la vitesse  $V_a$  ;
- $F$  la force de poussée issue des réacteurs, exprimée en  $N$  ;
- $M$  le moment de tangage, exprimé en  $Nm$
- $\alpha$  l'incidence aérodynamique de l'avion exprimée en  $rad$  ;
- $\theta$  l'assiette de l'avion exprimée en  $rad$  ;
- $q$  la vitesse de tangage de l'avion exprimée en  $rad/sec$  :

$$q = \dot{\theta} \quad (5.1)$$

- $\gamma_a$  la pente de l'avion exprimée en  $rad$  :

$$\gamma_a = \theta - \alpha \quad (5.2)$$

- $\bar{c}$  la corde de référence de l'aile (corde aérodynamique moyenne) exprimée en  $m$  ;

## 5.2 Modèle atmosphérique

Nous utiliserons le modèle d'atmosphère standard (ISA, International Standard Atmosphere) pour lequel la température varie linéairement (loi de Toussaint) avec l'altitude jusqu'à 11 km d'altitude (c'est à dire dans la troposphère). Les notations sont les suivantes :

- $T$  est la température qui dépend de l'altitude  $h$ .

$$\begin{cases} T(h) = T_0 + \beta_T h \\ T_0 = 288.15 \text{ } ^\circ K \\ \beta_T = -0.0065 \text{ } ^\circ K/m \end{cases} \quad (5.3)$$

- $R_s$  est la constante spécifique de l'air qui dépend de son humidité. Pour un air sec nous avons

$$R_s = 287.05 \text{ } m^2/(\text{ } ^\circ K s^2) \quad (5.4)$$

- $\rho_0$  est la masse volumique de l'air au sol exprimée en  $kg/m^3$

$$\rho_0 = 1.225 \text{ } kg/m^3 \quad (5.5)$$

- $\rho(h)$  la masse volumique de l'air exprimée en  $kg/m^3$ . La masse volumique de l'air sec est donnée par la relation suivante

$$\rho(h) = \rho_0 \left( \frac{T(h)}{T_0} \right)^{-\frac{g}{R_s \beta_T} - 1} = \rho_0 \left( 1 + \frac{\beta_T}{T_0} h \right)^{-\frac{g}{R_s \beta_T} - 1} \quad (5.6)$$

Notez la dépendance de la masse volumique  $\rho(h)$  avec la constante spécifique de l'air  $R_s$ , et donc avec son humidité.

- $\kappa$  est le coefficient de compressibilité de l'air

$$\kappa = 1.4 \quad (5.7)$$

- $M_a$  est le Mach de vol. Il est relié à la vitesse aérodynamique  $V_a$  et à la température  $T(h)$  par la relation suivante :

$$M_a = \frac{V_a}{\sqrt{\kappa R_s T(h)}} \quad (5.8)$$

### 5.3 Modèle d'état

Nous considérerons que la Terre est plate et que c'est un référentiel inertiel. Nous y associons un trièdre direct, noté  $\mathcal{R}_i = \{\vec{i}_i, \vec{j}_i, \vec{k}_i\}$ , dans lequel le vecteur unitaire  $\vec{k}_i$  est parallèle à la verticale ascendante du lieu. Dans ce repère, le poids de l'avion a pour expression :

$$\underline{mg} = -mg\vec{k}_i \quad (5.9)$$

Le trièdre aérodynamique, noté  $\mathcal{R}_a = \{\vec{i}_a, \vec{j}_a, \vec{k}_a\}$ , est le trièdre direct lié à la vitesse de l'avion dans lequel le vecteur unitaire  $\vec{i}_a$  est parallèle à la vitesse aérodynamique de l'avion :

$$\underline{V}_a = V_a \vec{i}_a \quad (5.10)$$

Dans ce repère la portance  $L$  et la traînée  $D$  ont les expressions suivantes :

$$\begin{cases} \underline{D} = -D\vec{i}_a \\ \underline{L} = L\vec{j}_a \end{cases} \quad (5.11)$$

Enfin le trièdre avion, noté  $\mathcal{R}_b = \{\vec{i}_b, \vec{j}_b, \vec{k}_b\}$ , est le trièdre direct rigidement lié à l'avion dans lequel le vecteur unitaire  $\vec{i}_b$  est parallèle à la poussée de l'avion :

$$\underline{F} = F\vec{i}_b \quad (5.12)$$

Les relations de passage entre ces trois trièdres sont des matrices de rotations.

Les équations de la dynamique du solide sont établies dans le repère aérodynamique. En utilisant les lois de la cinématique et de la dynamique du solide les équations suivantes peuvent être établies où  $I_{yy}$  est le moment d'inertie de l'avion autour de l'axe de tangage ;

$$\begin{cases} \dot{y} = V_a \cos(\theta - \alpha) \\ \dot{h} = V_a \sin(\theta - \alpha) \\ \dot{V}_a = \frac{F \cos(\alpha) - D}{m} - g \sin(\theta - \alpha) \\ \dot{\alpha} = q - \frac{L + F \sin(\alpha)}{m V_a} + \frac{g}{V_a} \cos(\theta - \alpha) \\ \dot{\theta} = q \\ \dot{q} = \frac{M}{I_{yy}} \end{cases} \quad (5.13)$$

Nous définissons le vecteur d'état  $\underline{x}$  et le vecteur de commande  $\underline{u}$  comme suit :

$$\underline{x} = \begin{bmatrix} y \\ h \\ V_a \\ \alpha \\ \theta \\ q \end{bmatrix} \text{ et } \underline{u} = \begin{bmatrix} \delta_{PHR} \\ \delta_{thr} \\ \delta_{el} \end{bmatrix} \quad (5.14)$$

Où :

- $y$  désigne la position horizontale de l'avion exprimée en  $m$  ;
- $h$  désigne l'altitude pression exprimée en  $m$  ;
- $V_a$  désigne la vitesse aérodynamique de l'avion exprimée en  $m/s$  ;
- $\alpha$  désigne l'incidence aérodynamique de l'avion exprimée en  $rad$  ;
- $\theta$  désigne l'assiette de l'avion exprimée en  $rad$  ;
- $q$  désigne la vitesse de tangage de l'avion exprimée en  $rad/sec$  ;

Et :

- $\delta_{PHR}$  désigne l'angle de braquage de l'empennage horizontal exprimé en *rad* ;
- $\delta_{thr}$  désigne la position de la commande de poussée ;
- $\delta_{el}$  désigne le braquage de la gouverne de profondeur exprimé en *rad*.

Les équations ci-dessus peuvent se ré-écrire à l'aide du champ de vecteur  $f$  :

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad (5.15)$$

## 5.4 Rappel sur la vitesse équivalente

L'équivalent vitesse d'un aéronef, notée *EAS*, est sa vitesse aérodynamique, notée  $V_a$ , corrigée des effets de la compressibilité de l'air à l'altitude où vole l'aéronef. En désignant par  $\rho(h)$  la masse volumique de l'air à l'altitude  $h$ , la relation entre la vitesse équivalente d'un aéronef et sa vitesse aérodynamique est la suivante :

$$\rho(0) EAS^2 = \rho(h) V_a^2 \quad (5.16)$$

## 5.5 Poussée

La force de poussée  $F$  des moteurs est exprimée en  $N$  et est supposée parallèle à l'axe  $\vec{i}_b$  du trièdre avion. Nous utiliserons le modèle proposé par J.Mattingly (*Jack D. Mattingly, William H. Heiser, Daniel H. Daley, Aircraft Engine Design. AIAA Education series, 1987. ISBN : 0-930403-23-1*) pour un réacteur double flux (turbofan) ayant un fort taux de dilution :

$$F = F_0 \left( \frac{\rho(h)}{\rho_0} \right)^{0.6} \left( 0.568 + 0.25 (1.2 - M_a)^3 \right) \delta_{th} \quad (5.17)$$

Où :

- $F_0$  est la poussée maximale au sol et à l'arrêt, exprimée en  $N$
- $\delta_{th}$  est la position de la commande de poussée :  $0 \leq \delta_{th} \leq 1$

## 5.6 Rappel sur le PHR

Pour piloter longitudinalement un avion le pilote dispose du manche et de la manette de poussée.

Le manche provoque un mouvement de tangage grâce à la gouverne de profondeur (*elevator* en anglais) située à l'arrière de l'empennage horizontal. La gouverne de profondeur est utilisée pour modifier la trajectoire de l'avion à très court terme, via une modification de l'assiette et donc de l'incidence et du facteur de charge. En plus de la gouverne de profondeur les avions de transport commercial possèdent le plus souvent un Plan Horizontal Réglable (PHR ou *THS* : *Trimmed Horizontal Stabilizer* en anglais) qui reprend en permanence les efforts en tangage de la gouverne de profondeur lors des phases équilibrées du vol. Cela permet à la gouverne de profondeur de revenir à une déflexion nulle, donc de rester en vol stabilisé sans action sur le manche, et de retrouver la pleine autorité en terme de capacité de modification du moment de tangage. En outre, cela permet de réduire la traînée. Le PHR est déplacé par l'intermédiaire d'une vis sans fin et est actionné soit manuellement par le pilote (via les volants de Trim) soit par le calculateur de commandes du vol.

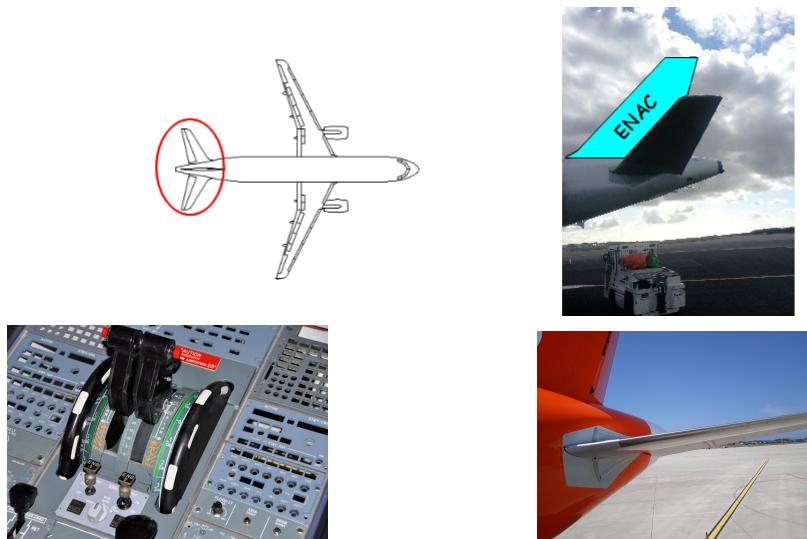


FIGURE 5.2 – Plan Horizontal Réglable (PHR), gouverne de profondeur, volants de trim et manettes de poussée

Sur la Figure 5.2 sont représentés le Plan Horizontal Réglable (PHR), la gouverne de profondeur ainsi que les volants de trim et les manettes de poussée installés dans le poste de pilotage d'un avion de transport commercial (source : [aircockpit.com](http://aircockpit.com)).

Sur la Figure 5.3 est illustré le principe de fonctionnement de l'ensemble {voilure principale ; PHR ; gouverne de profondeur}. Notez la convention : une variation de déflexion de gouverne (PHR ou gouverne de profondeur) dans le sens positif doit générer une variation négative du moment.

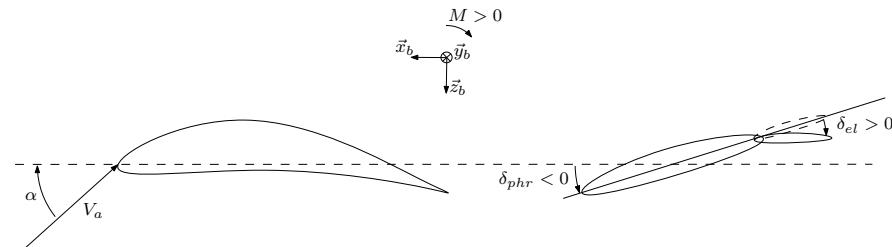


FIGURE 5.3 – Actions du PHR et de la gouverne de profondeur sur le moment aérodynamique

Les ordres de grandeur pour les valeurs de déflexions maximales, ainsi que pour les vitesses de déflexions maximales, sont les suivants :

- La gouverne de profondeur peut se défléchir jusqu'à +/- 30°, avec une vitesse maximale de déflexion de 15°/s ;
- Le PHR peut se défléchir jusqu'à +5° / -15°, avec une vitesse maximale de déflexion de 5°/s.

Si l'on traçait le lieu des racines des fonctions de transfert  $\frac{\delta\theta(s)}{\delta_{el}(s)}$  et  $\frac{\delta\theta(s)}{\delta_{PHR}(s)}$ , en prenant en

compte la dynamique des actionneurs, et avec  $\theta$  l'assiette de l'avion, on verrait les poles de  $\frac{\delta\theta(s)}{\delta_{PHR}(s)}$  plus proches de l'origine du repère que ceux de  $\frac{\delta\theta(s)}{\delta_{el}(s)}$  - car le PHR est beaucoup plus lent que la gouverne de profondeur. Si on augmente le gain pour compenser la lenteur du plan on se retrouve dans la partie instable du lieu des racines. Avec la gouverne de profondeur, beaucoup plus rapide, on n'a pas ce problème. Ce qui corrobore la répartition des rôles entre gouverne de profondeur et PHR mentionnée plus haut.

## 5.7 Paramètres

Les données des tables 5.1, 5.2 et 5.3 seront utilisées. Elles proviennent de l'ouvrage *Elodie Roux, Avions civils à réaction : plan 3 vues et données caractéristiques, 2007. ISBN : 978-2-9529380-2-0*. En ce qui concerne la table 5.3, MTOW (Maximum Takeoff Weight) désigne la masse maximale au décollage et OWE (Operating Empty Weight) la masse à vide en ordre d'exploitation (carburant non inclus!).

La masse  $m$  sera choisie entre la masse à vide en ordre d'exploitation et la masse maximale au décollage. Pour cela nous ferons varier le coefficient  $k_m$  de réglage de la masse entre 0.1 (on met quand même un peu de kérosène dans les réservoirs) et 1 :

$$m = (1 - k_m)\text{OWE} + k_m\text{MTOW} \text{ où } 0.1 \leq k_m \leq 1 \quad (5.18)$$

Équipe	Avion	Moteur	$F_0$ (N)
Alpha	Airbus A-320	CFM 56-5A1	2 x 111205
Bravo	Boeing 737-800	CFM 56-7B24	2 x 106757
Charlie	Airbus A-319	CFM 56-5B5	2 x 97860
Delta	Airbus A-321	CFM 56-5B1	2 x 133446
Echo	Boeing 737-700	CFM 56-7B20	2 x 91633
Foxtrot	Boeing 737-300	CFM 56-3B1	2 x 88694

TABLE 5.1 – Table des motorisations

Avion	$\lambda$	$\lambda_t$	$S$ ( $m^2$ )	$S_t$ ( $m^2$ )	$\bar{c}$ (m)	$L_{fus}$ (m)
Airbus A-320	9.39	5	122.44	31	4.19	37.57
Boeing 737-800	9.45	6.28	124.6	32.8	4.17	38.02
Airbus A-319	9.39	5	122.44	31	4.19	33.84
Airbus A-321	9.13	5	126	31	4.34	44.51
Boeing 737-700	9.44	6.28	124.6	32.8	4.17	32.18
Boeing 737-300	9.16	5.15	91.04	31.31	3.73	32.18

TABLE 5.2 – Table des paramètres géométriques

Avion	$MTOW\ (kg)$	$OWE\ (kg)$
Airbus A-320	73500	39733
Boeing 737-800	70534	41413
Airbus A-319	64000	39358
Airbus A-321	89000	47000
Boeing 737-700	60326	37648
Boeing 737-300	56473	31480

TABLE 5.3 – Table des masses