

Projet de synthèse AVI

Marius Michaud

`marius.michaud@alumni.enac.fr`

Guillaume Claudon

`guillaume.claudon@alumni.enac.fr`

Ulysse Bourdic-Girard

`ulysse.bourdic-girard@alumni.enac.fr`

École Nationale de l'Aviation Civile (ENAC)

Janvier 2026



Table des matières

1	Introduction	6
2	Modélisation et Linéarisation	7
2.1	Caractéristiques de l'avion	7
2.2	Point de fonctionnement (Trim)	7
2.3	Linéarisation	7
3	Commande de l'assiette (Boucle Interne)	9
3.1	Synthèse du correcteur	9
4	Loi de descente (Open Descent)	10
4.1	Stratégie de commande modale	10
4.2	Gestion de l'erreur statique : Ajout d'un intégrateur	10
4.3	Placement des pôles	10
4.4	Observateur d'état	11
4.5	Résultats de simulation	11
5	Lois de maintien du niveau et de la vitesse	12
5.1	Commande Optimale (LQT)	12
6	Conclusion	13
7	Annexes	14

Table des figures

1.1	Comportement de la loi de descente [1, Fig 4.1]	6
3.1	Schéma Simulink de la boucle de tenue d'assiette (Commande Classique) .	9
4.1	Schéma Simulink de la boucle de vitesse avec Intégrateur (Commande Modale)	11
4.2	Évolution de la Vitesse Vraie (TAS) lors de la descente à EAS constante .	11

Liste des tableaux

2.1	Paramètres caractéristiques de l'A320 (Groupe Golf) [1, Tab 2.1]	7
-----	--	---

Bibliographie

- [1] Thierry MIQUEL, Florian SANSOU, Amélie CARRIEROU, Jim SHARPLES, Jean-Philippe CONDOMINES et Antoine DROUIN : Commande linéaire des systèmes dynamiques — projet de synthèse. *ENAC*, décembre 2025.
- [2] Qing-Guo WANG, Zhiping ZHANG, Karl Johan ÅSTRÖM et Lee See CHEK : Guaranteed dominant pole placement with PID controllers. *Journal of Process Control*, 19(2):349–355, 2009.

1 Introduction

Le but de ce projet est d'implémenter une loi de descente pour un avion de ligne. Cet avion est en vol stabilisé à un niveau de vol (FL) et une vitesse équivalente (*EAS*) qui dépendent de chaque groupe. Il est autorisé par le contrôle aérien à descendre jusqu'au niveau 30, où il se stabilise à une vitesse de 230 kts [1, 4.1]. Le comportement de cette loi est synthétisé dans le schéma suivant :



FIGURE 1.1 – Comportement de la loi de descente [1, Fig 4.1]

Dans le cadre de ce projet, on ne considère que l'axe longitudinal de l'avion, on néglige l'influence des commandes de roulis et de lacet. Le modèle de l'avion est donc commandé par trois entrées [1, 4.3] :

- δ_{PHR} : le braquage du plan horizontal réglable (PHR)
- δ_{el} : le braquage de la gouverne de profondeur
- δ_{th} : la position de la manette de poussée

Pour remplir ce cahier des charges, nous allons découper la commande de l'avion en trois sous-commandes distinctes :

- **La commande de l'assiette** pour laquelle on implémentera une commande classique ;
- **La loi de descente** pour laquelle on implémentera une commande modale ;
- **Les lois de maintien du niveau et de la vitesse** pour lesquelles on implémentera une commande optimale.

2 Modélisation et Linéarisation

Avant de procéder à la synthèse des lois de commande, il est nécessaire de définir le modèle de l'avion et son point de fonctionnement.

2.1 Caractéristiques de l'avion

L'avion étudié est un Airbus A320 (groupe Golf). Il est caractérisé par sa marge statique (ms) et son coefficient de masse (km). Le tableau suivant récapitule les paramètres de simulation :

TABLE 2.1 – Paramètres caractéristiques de l'A320 (Groupe Golf) [1, Tab 2.1]

Paramètre	Valeur
Marge statique (ms)	0.5
Coefficient de masse (km)	0.3
Niveau de vol initial (FL)	150
Vitesse équivalente (EAS)	260 kts

2.2 Point de fonctionnement (Trim)

Conformément au scénario *Open Descent* décrit dans le sujet [1], la poussée est maintenue fixe sur la position *Idle*. Le calcul du point d'équilibre (*trim*) a été réalisé via la fonction `utComputeTrimIdle`. Pour l'avion assigné à l'équipe Golf, le point d'équilibre obtenu est :

- Altitude : 15 000 ft (4572 m)
- Vitesse vraie (V_a) : ≈ 139 m/s (pour 260 kts EAS)
- Poussée : $\delta_{th} = 0.1$ (Idle)
- Assiette (θ_{trim}) et Incidence (α_{trim}) : calculées pour assurer la portance.

2.3 Linéarisation

La linéarisation du modèle non-linéaire autour de ce point de trim a été effectuée à l'aide de la commande `linmod` de Matlab. Cela nous a permis d'extraire le modèle d'état longitudinal réduit (matrices A_4 , B_4) décrivant la dynamique des états $[V_a, \alpha, \theta, q]^T$ sous l'action de la gouverne de profondeur δ_{el} .

L'analyse des valeurs propres de la matrice A_4 met en évidence les modes naturels de l'avion :

- **Le mode phugoïde** : mode lent, peu amorti, caractérisé par un échange d'énergie cinétique et potentielle.
- **L'oscillation d'incidence** : mode rapide et bien amorti.

3 Commande de l'assiette (Boucle Interne)

Ce chapitre détaille la conception de la boucle interne de pilotage, dont l'objectif est d'asservir l'assiette longitudinale θ de l'avion via la gouverne de profondeur δ_{el} [1].

3.1 Synthèse du correcteur

Nous avons retenu une structure de commande classique de type PI-D (Proportionnel, Intégral, Dérivée sur la mesure) associée à un pré-filtre sur la consigne. Les gains ont été calculés via la méthode de Wang [1][2].

Conformément au cahier des charges, nous avons conservé les spécifications initiales pour la méthode de Wang, à savoir :

$$t_s = 3 \text{ s} \quad \text{et} \quad D = 5\% \quad (3.1)$$

Ce réglage permet de placer les pôles dominants de la boucle fermée interne afin de respecter les critères de performance temporelle et de dépassement demandés.

La structure finale implémentée sous Simulink est présentée ci-dessous :

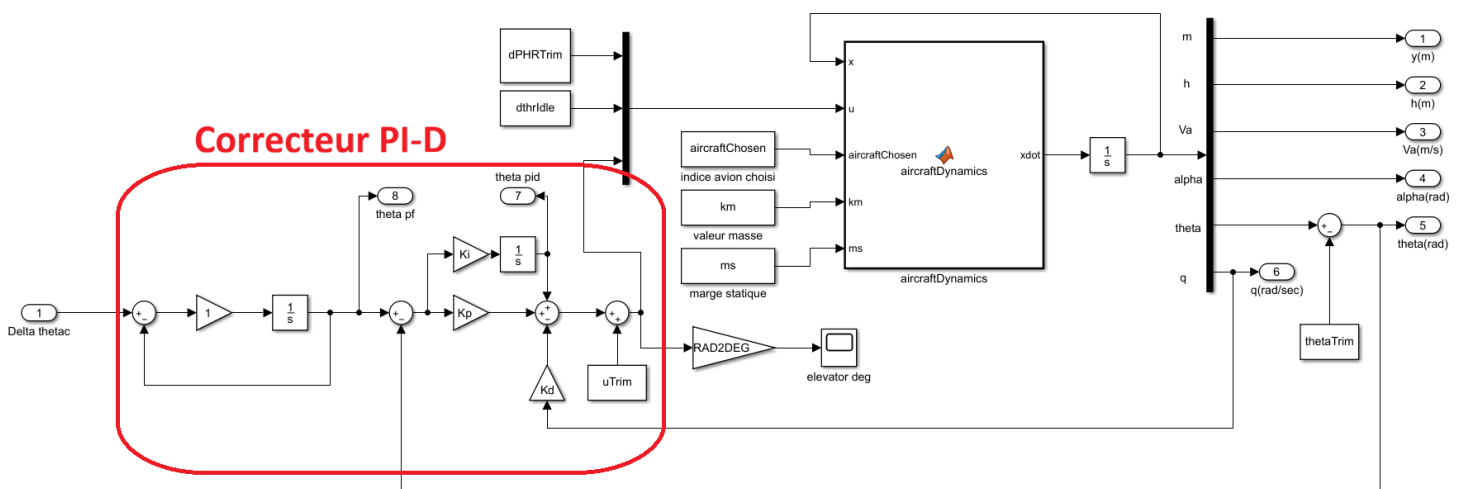


FIGURE 3.1 – Schéma Simulink de la boucle de tenue d'assiette (Commande Classique)

4 Loi de descente (Open Descent)

L'objectif de la loi de descente est de maintenir une vitesse équivalente (*EAS*) constante en agissant sur la consigne d'assiette θ_c , la poussée restant figée à *Idle* [1].

4.1 Stratégie de commande modale

Nous avons utilisé une commande par retour d'état calculée sur le système augmenté. Le système considéré pour cette synthèse est l'avion asservi en assiette (boucle fermée interne). Le vecteur d'état considéré est :

$$x = [V_a, \alpha, \theta, q, x_{pid}, x_{pf}]^T \quad (4.1)$$

Où x_{pid} et x_{pf} sont les états internes du correcteur PI-D et du pré-filtre.

4.2 Gestion de l'erreur statique : Ajout d'un intégrateur

Le sujet suggère l'ajout d'un intégrateur pour annuler l'erreur statique de vitesse [1]. Cette approche est indispensable car le modèle linéaire varie fortement au cours de la descente. Nous avons augmenté le système avec un état intégrateur ξ défini par :

$$\dot{\xi} = \varepsilon = V_{EAS,c} - V_{EAS,m} \quad (4.2)$$

Il est crucial de noter que l'intégrateur agit sur l'erreur d'*EAS* (et non de TAS), car c'est la grandeur pilotée.

4.3 Placement des pôles

La contrainte de conception impose que la partie réelle de tous les pôles en boucle fermée soit inférieure à -1.3 rad/s [1, 4.6]. En utilisant la fonction `place` de Matlab sur le système augmenté, nous avons imposé le vecteur de pôles suivant :

- **Pôles de la dynamique interne (dp)** : conservés depuis la boucle d'assiette. Avec un temps de réponse $t_s = 3 \text{ s}$, ils se situent à environ $-1.33 \pm 1.4i$.
- **Pôle de l'intégrateur** : fixé à -1.33 .
- **Modes rapides de la boucle interne** : conservés vers $-3.4 \pm 0.15i$.

- **Autres pôles** : repoussés vers la gauche à -3 et -4 pour ne pas interférer avec la dynamique dominante.

Le choix du pôle intégrateur à -1.33 permet de satisfaire la spécification de stabilité ($\text{Re} < -1.3$)[1]. On note que la dynamique de l'intégrateur est du même ordre de grandeur que la dynamique de l'assiette ($1.33 \approx 1.33$), ce qui est une limite de cette configuration, mais nécessaire pour respecter les contraintes sans demander une dynamique irréaliste à l'actionneur.

Le schéma complet de la boucle externe est illustré ci-dessous :

FIGURE 4.1 – Schéma Simulink de la boucle de vitesse avec Intégrateur (Commande Modale)

4.4 Observateur d'état

Pour implémenter cette loi de commande par retour d'état, nous devons connaître l'ensemble du vecteur d'état x . Or, certaines grandeurs comme l'incidence α ne sont pas toujours mesurées directement ou sont bruitées. Nous avons donc synthétisé un observateur d'état de Luenberger.

Cet observateur utilise les entrées de commande et les trois sorties mesurées disponibles :

$$y_{mes} = [V_a, \theta, q]^T \quad (4.3)$$

La matrice de gain L de l'observateur a été calculée par placement de pôles (dualité avec la commande) pour assurer une convergence de l'erreur d'estimation vers zéro.

4.5 Résultats de simulation

La validation sur le modèle non-linéaire (sur une durée de 600s) démontre le bon fonctionnement de la loi. On observe sur la figure 4.2 que la Vitesse Vraie (TAS) diminue progressivement au cours de la descente pour maintenir une EAS constante, validant la physique du vol.

FIGURE 4.2 – Évolution de la Vitesse Vraie (TAS) lors de la descente à EAS constante

5 Lois de maintien du niveau et de la vitesse

Pour la phase de croisière et d'approche, l'avion doit maintenir simultanément son altitude et sa vitesse. Il s'agit d'un problème multi-variable (MIMO).

5.1 Commande Optimale (LQT)

Nous avons implémenté une commande optimale de type LQT (*Linear Quadratic Tracker*) [1]. Cette méthode permet de calculer un gain de retour d'état K et une précommande variable assurant le suivi de trajectoire (pente nulle, vitesse constante) tout en minimisant un critère quadratique sur les écarts et l'énergie de commande.

[Insérer ici les résultats de la commande optimale, les courbes de suivi de pente et de vitesse]

6 Conclusion

Ce projet nous a permis de synthétiser et valider une architecture de pilotage automatique complète pour la phase de descente d'un A320.

7 Annexes