

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
3.2	Implémentation et Validation	9
4	Loi de descente (Open Descent)	12
4.1	Stratégie de commande modale	12
4.2	Observateur d'état	13
4.3	Choix des pôles	14
4.4	Gestion de l'erreur statique : Ajout d'un intégrateur	14
4.5	Modèle non linéaire	16
4.6	Résultats de simulation	16
5	Lois de maintien du niveau et de la vitesse	18
5.1	Linéarisation	18
5.2	Synthèse par commande optimale (LQT)	18
5.3	Réglage des matrices de pondération	18
5.4	Analyse du couplage	19
6	Simulation du scénario complet	20
6.1	Gestion de la transition	20
6.2	Analyse des résultats globaux	20
7	Conclusion	21

Table des figures

1.1	Comportement de la loi de descente [1, Fig 4.1]	6
3.1	Schémas Simulink de la boucle de tenue d'assiette	10
3.2	Comparaison des réponses indicielles (Linéaire vs Non-linéaire)	10
4.1	Schéma Simulink de la commande avec préfiltre H	13
4.2	Réponse de la commande modale à une consigne en échelon d'amplitude 3	14
4.3	Schéma Simulink de la boucle de vitesse avec intégrateur sur le système linéaire	15
4.4	Réponse de la commande modale à une consigne en échelon d'amplitude 3	15
4.5	Schéma de la commande modale appliquée au modèle non linéaire	16
4.6	Réponse du système non linéaire à un échelon de consigne d'amplitude 2 kts	17

Liste des tableaux

2.1	Paramètres caractéristiques de l'A320 (Groupe Golf) [1, Tab 2.1]	7
2.2	Analyse modale du modèle longitudinal linéarisé au trim	8

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 :

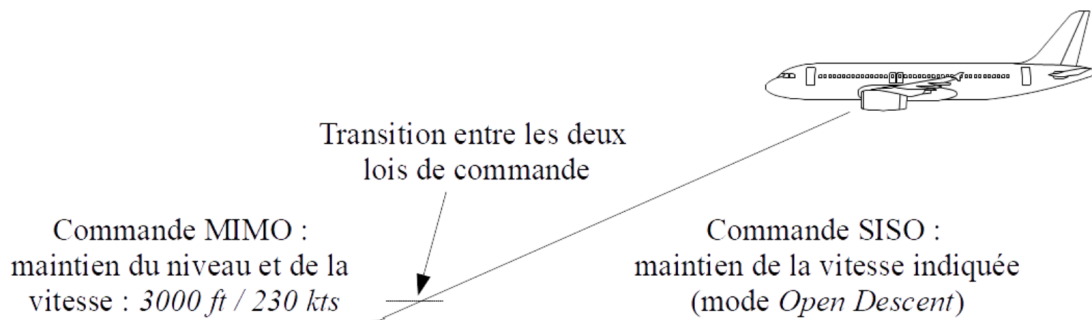


FIGURE 1.1 – Comportement de la loi de descente [1, Fig 4.1]

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_{THR} : 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 : $\delta_{THR} = 0.1$ (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 (obtenues via la commande `damp(A4)`) met en évidence les deux modes longitudinaux classiques :

- **Le mode phugoïde** : mode lent caractérisé par une pulsation naturelle $\omega_n = 0,0721$ rad/s et un amortissement faible $\zeta = 0.1267$.
- **L'oscillation d'incidence** : mode rapide caractérisé par une pulsation naturelle $\omega_n = 2,8896$ rad/s et un amortissement plus important $\zeta = 0.5167$.

Ces résultats sont résumés dans le tableau suivant :

TABLE 2.2 – Analyse modale du modèle longitudinal linéarisé au trim

Mode	Amortissement (ζ)	Pulsation (ω_n [rad/s])
Phugoïde	0,1267	0,0721
Oscillation d'incidence	0,5167	2,8896

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, 4.5].

3.1 Synthèse du correcteur

Nous avons retenu une structure de commande classique de type PI-D (Proportionnel, Intégral, Dérivé sur la mesure) associée à un pré-filtre sur la consigne. Les gains ont été calculés via la méthode de Wang, présentée dans le sujet au paragraphe 3.5.1 [1] et dans l'article de *Wang et al.* [2]. La méthode de Wang demande le placement d'un pôle supplémentaire pour calculer les gains du PI-D. Nous avons choisi celui-ci avec un graphique « root-locus » (lieu des racines) en cherchant un pôle qui est plus rapide que les actuels afin de ne pas affecter la dynamique du système. En regardant la figure ci-dessous, on peut constater qu'un pôle à -5.66 satisfait ce critère.

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

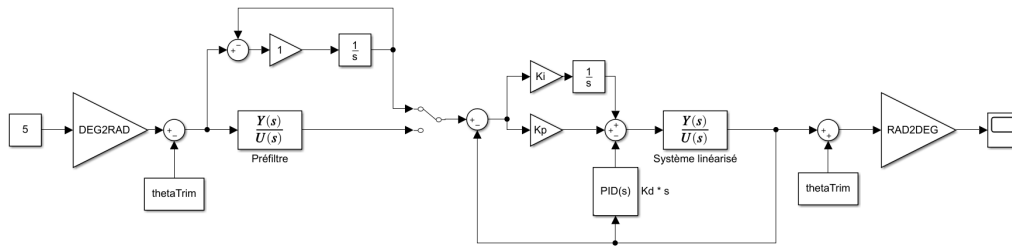
$$t_s = 3 \text{ s} \quad \text{et} \quad D = 5\% \tag{3.1}$$

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.

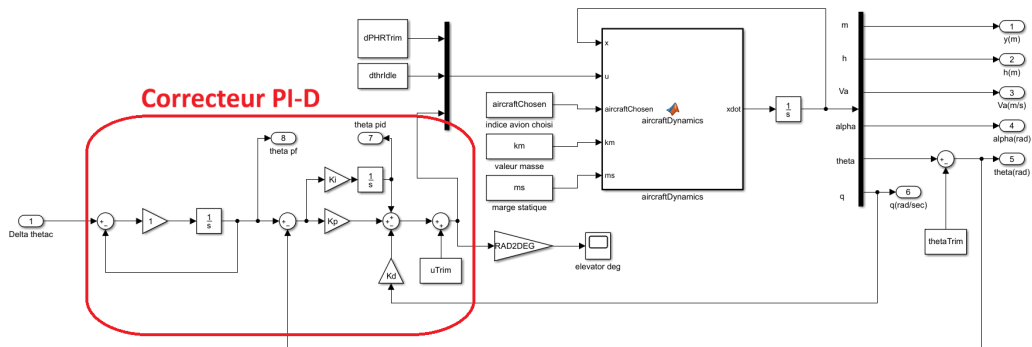
3.2 Implémentation et Validation

Les structures de commande ont été implémentées sous Simulink pour le modèle linéarisé (servant à la synthèse) et le modèle non-linéaire complet (servant à la validation).

Afin de valider la synthèse, nous avons superposé la réponse indicielle des deux modèles pour une consigne de 5° .



(a) Système linéarisé



(b) Système non-linéaire (Avion naturel)

FIGURE 3.1 – Schémas Simulink de la boucle de tenue d'assiette

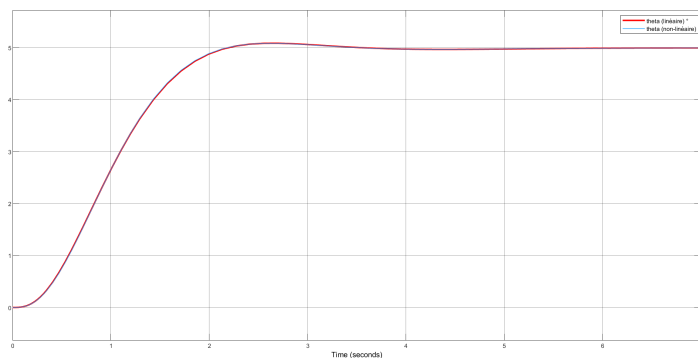


FIGURE 3.2 – Comparaison des réponses indicielles (Linéaire vs Non-linéaire)

L'analyse de la figure 3.2 suggère :

1. **Spécifications** : Temps de réponse ≈ 3 s, dépassement $\approx 5\%$ (pic à $\approx 5,25^\circ$).
2. **Validité** : Superposition parfaite des courbes rouge et bleue. La linéarisation est valide pour les faibles amplitudes (5°).

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, 4.6].

4.1 Stratégie de commande modale

La linéarisation est effectuée sur le modèle Simulink `acDynModelGolfTenueClassique`, qui inclut déjà la boucle interne d'assiette. Le système linéarisé A_6 est d'ordre 6 et correspond au vecteur d'état suivant :

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

Où x_{pid} et x_{pf} sont les états internes du correcteur PI-D et du pré-filtre de la boucle interne. La commandabilité de ce système augmenté a été vérifiée via le rang de la matrice de commandabilité (fonction MatLab `ctrb`).

La commande modale consiste à choisir les pôles du système bouclé pour influencer le comportement de ses modes. La fonction MatLab `place` permet de calculer le gain du retour d'état qui sera utilisé pour commander le système (on note ce gain K_x).

Le retour d'état seul ne permet pas de commander le système, on ajoute une précommande pour adapter la commande au retour d'état. La précommande est calculée ainsi [1, eq. 3.50] :

$$H = -(C \cdot (A - BK)^{-1} \cdot B)^{-1} \quad (4.2)$$

L'architecture de commande du système est illustrée ci-dessous :

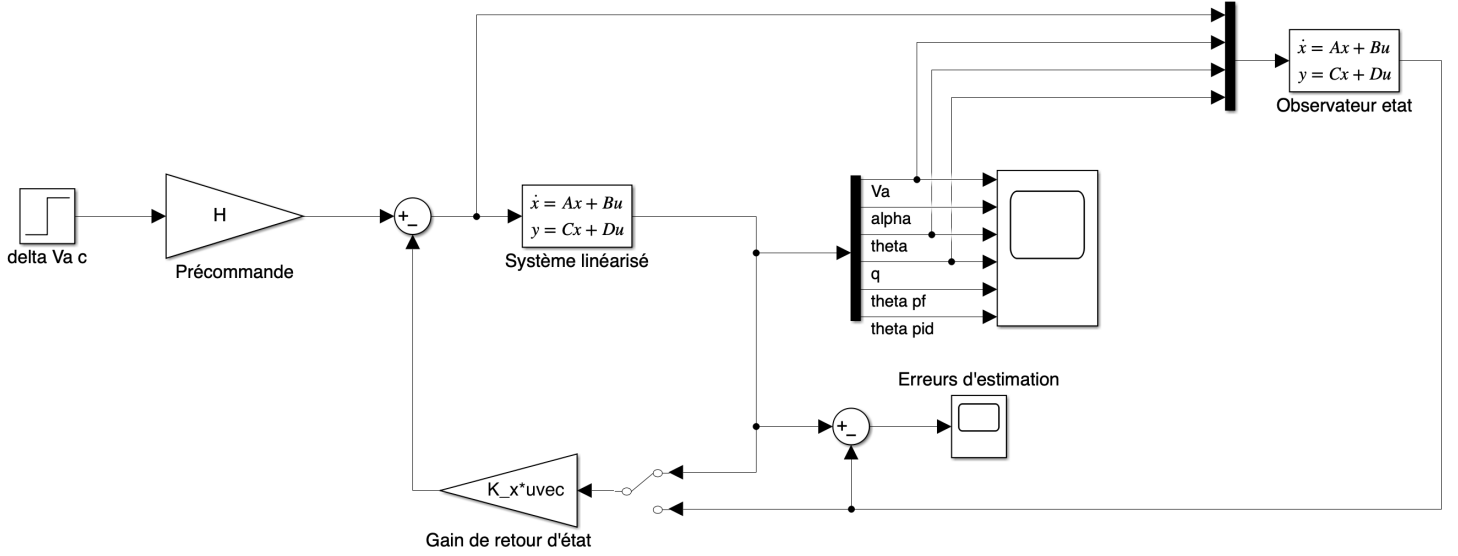


FIGURE 4.1 – Schéma Simulink de la commande avec préfiltre H

4.2 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.

Pour reconstruire le vecteur d'état complet, nous utilisons un observateur de Luenberger défini par :

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y_{mes} - C\hat{x}) \quad (4.3)$$

Où \hat{x} est l'état estimé et L la matrice de gain de l'observateur. Cet observateur est synthétisé en utilisant les trois sorties mesurées disponibles définies dans la matrice C_{obs} :

$$y_{mes} = [V_a, \theta, q]^T \quad (4.4)$$

L'observabilité de la paire (A_6, C_{obs}) a été validée (rang plein).

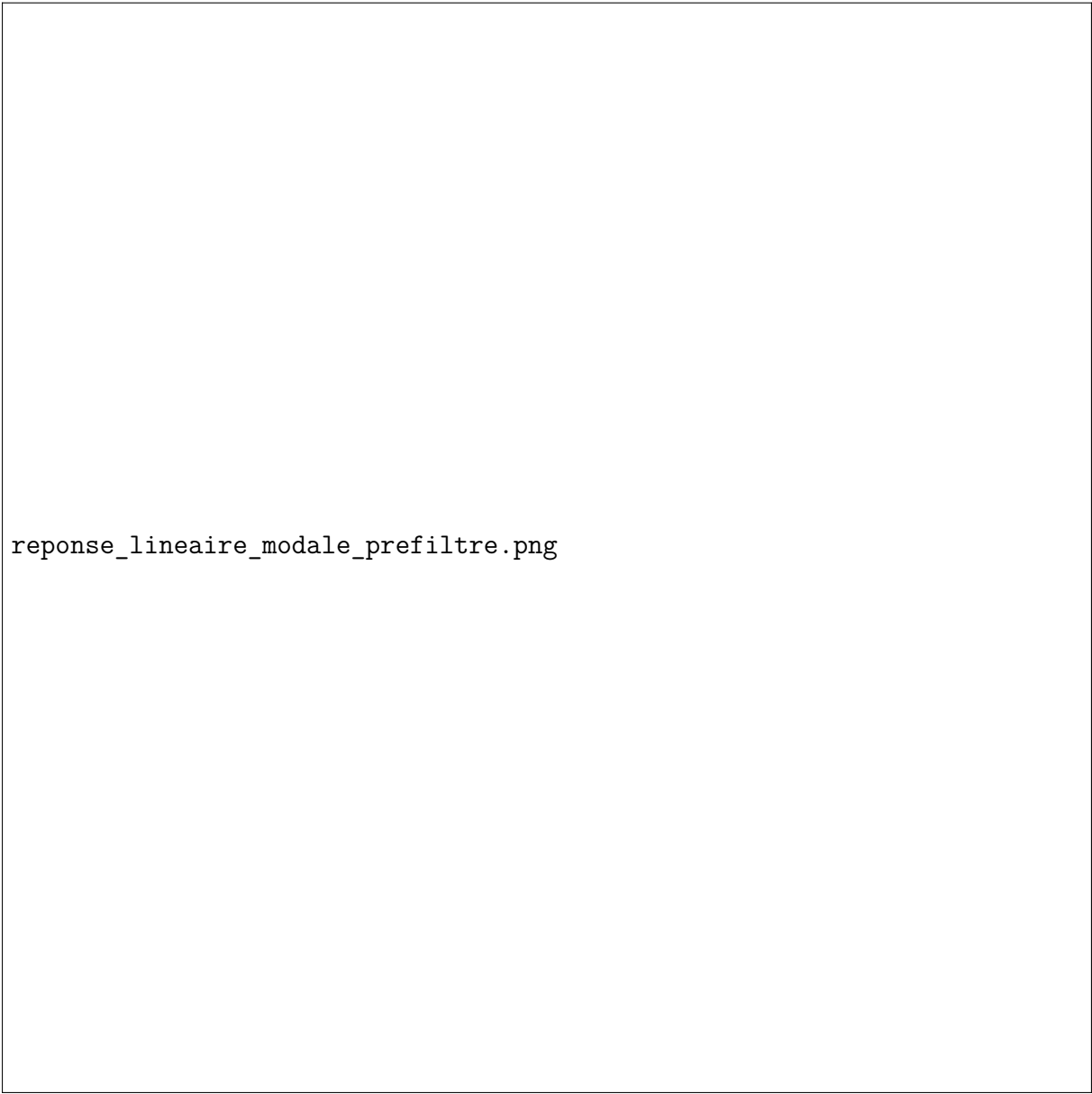
Pour le calcul du gain L , bien que la théorie suggère généralement une dynamique d'erreur d'estimation plus rapide que celle de la commande, nous avons choisi de placer les pôles de l'observateur sur les mêmes dynamiques que celles du système en boucle fermée (vecteur `poles_desires(1:6)`). Ce choix vise à limiter l'amplification du bruit de mesure sur les états estimés, notamment sur l'incidence α . D'après les tests effectués, ce réglage s'est avéré satisfaisant pour notre application et une dynamique plus rapide n'a pas semblé nécessaire.

4.3 Choix des pôles

Pour calculer le gain du retour d'état, nous avons choisi les pôles suivants :

- **Dynamique dominante** (dp) : Conservée de la méthode de Wang (boucle interne).
- **Modes rapides** : Pôles complexes conjugués conservés vers $-3.4 \pm 0.15i$ (observés dans la matrice A_6 via la fonction MatLab `eig`). Nous avons choisi de ne pas modifier ces modes naturels pour éviter de solliciter inutilement les actionneurs.
- **Pôles réels de réglage** : Fixés à -3 et -4 .

La simulation de la commande modale donne le résultat suivant :



reponse_lineaire_modale_prefiltre.png

FIGURE 4.2 – Réponse de la commande modale à une consigne en échelon d'amplitude 3

La réponse de cette commande n'est pas du tout satisfaisante : l'erreur statique est de l'ordre de 2.42.

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

Pour pallier les limitations de la précommande seule, le sujet suggère l'ajout d'un intégrateur pour annuler l'erreur statique de vitesse [1, 3.6.2]. Cette approche est indispensable car le modèle linéaire varie fortement au cours de la descente. Nous avons donc augmenté le système avec un état intégrateur ξ défini par :

$$\dot{\xi} = V_{EAS,cons} - V_{EAS,mes} \quad (4.5)$$

On choisit pour l'intégrateur un pôle à -10.33 , ce qui assure une élimination rapide de l'erreur statique, et vérifie la contrainte de stabilité imposée ($\text{Re} < -1.3$).

Le schéma de la commande modale avec intégrateur appliquée au système linéaire est illustré ci-dessous :

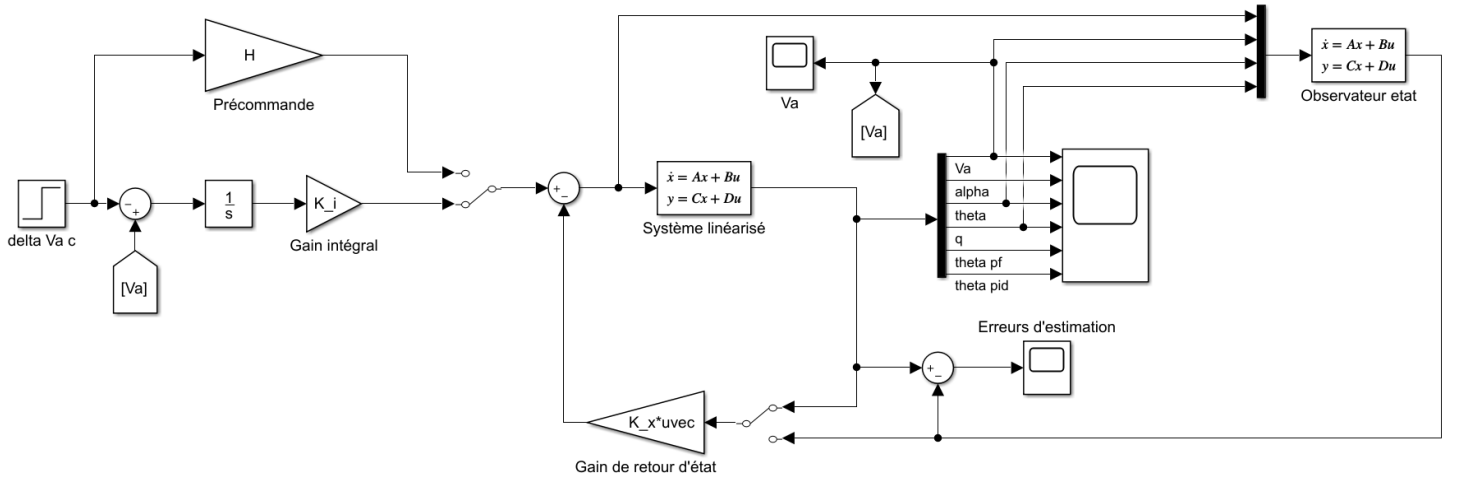


FIGURE 4.3 – Schéma Simulink de la boucle de vitesse avec intégrateur sur le système linéaire

La simulation de la commande modale avec intégrateur donne le résultat suivant :

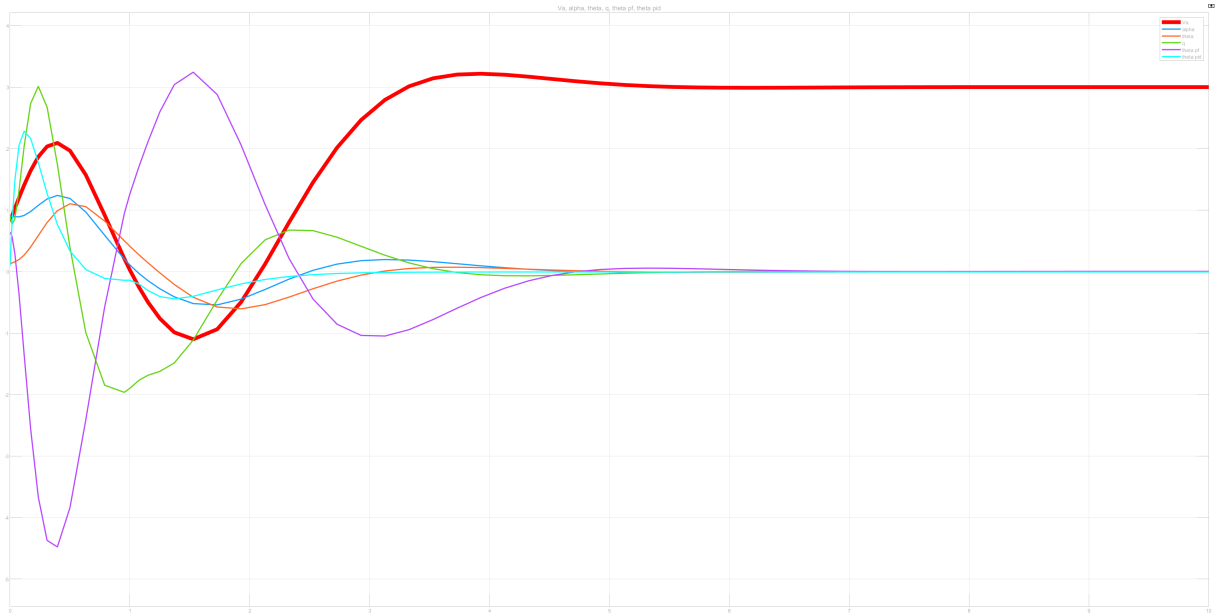


FIGURE 4.4 – Réponse de la commande modale à une consigne en échelon d’amplitude 3

Cette réponse est satisfaisante : le système répond en environ 3,5 s et avec un dépassement de l’ordre de 6 %. Nous avons donc implémenté cette commande sur le modèle non linéaire.

4.5 Modèle non linéaire

Le schéma suivant illustre l’intégration de la commande modale sur le système non linéaire. Nous avons conservé la possibilité d’alterner entre la précommande H et l’intégrateur, même si l’intégrateur se révèle être supérieur. Il est aussi important de noter que l’intégrateur agit sur l’erreur d’EAS (et non de TAS), car c’est la grandeur pilotée.

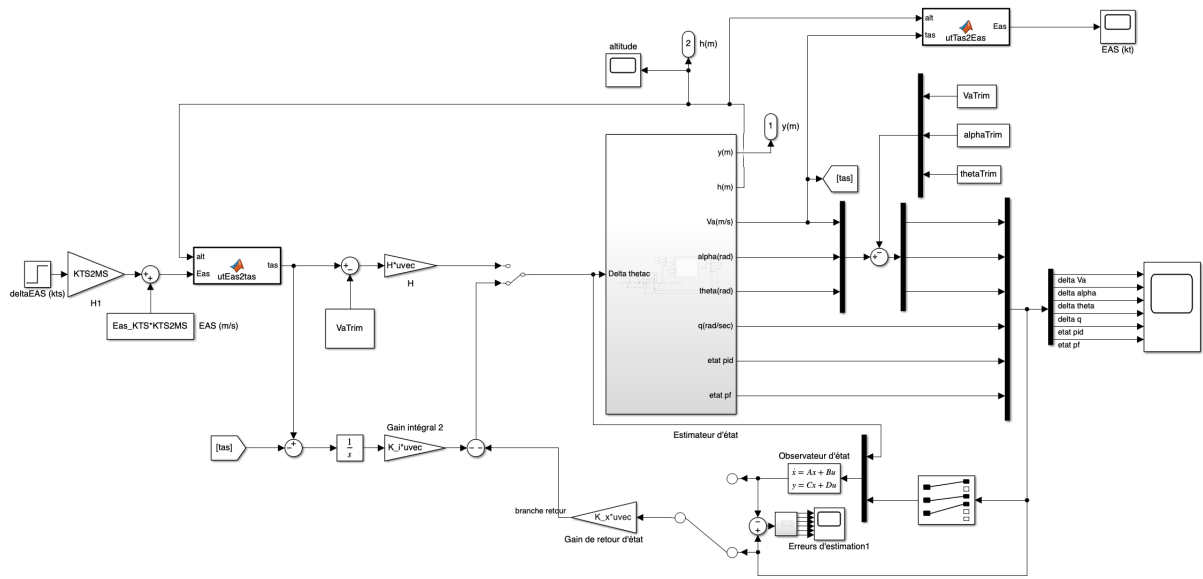


FIGURE 4.5 – Schéma de la commande modale appliquée au modèle non linéaire

4.6 Résultats de simulation

Le graphique suivant présente la réponse du modèle non linéaire à un échelon d'amplitude 2 kts sur une durée de 60 s. Le dépassement de l'ordre de 0,2 % et le temps de réponse de 6 s sont adéquats. L'erreur statique relative est de l'ordre de 0,1 %, ce qui est suffisant pour un contrôle de la vitesse au noeud près. On peut donc conclure que le comportement de la loi de maintien de la vitesse sur le modèle non linéaire est satisfaisant.

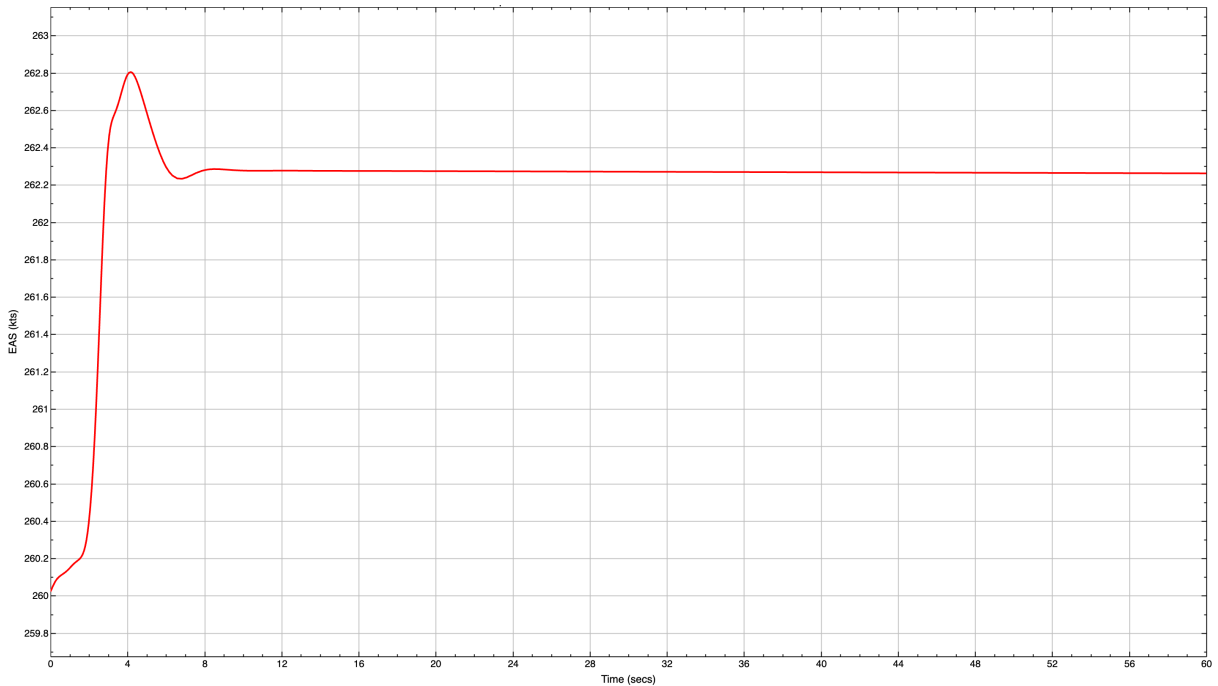


FIGURE 4.6 – Réponse du système non linéaire à un échelon de consigne d'amplitude 2 kts

5 Lois de maintien du niveau et de la vitesse

Pour la phase de stabilisation au niveau 30, l'avion doit maintenir simultanément son altitude et sa vitesse aérodynamique de 230 kts. Il s'agit d'un problème multi-variable (MIMO) où la poussée δ_{thr} et la profondeur δ_{el} sont utilisées conjointement.

5.1 Linéarisation

Pour mettre en œuvre la commande, on linéarise le modèle avion au nouveau point de trim : 230 kts et 3000 ft. Cette linéarisation est réalisée par `utAcDynamicsFunction.m` et donne le point de fonctionnement suivant :

$$\text{trimValLF} = \begin{pmatrix} \delta_{\text{PHR}} \\ \alpha_{\text{trim}} \\ \delta_{\text{THR}} \end{pmatrix} = \begin{pmatrix} -10,0^\circ \\ 3,6^\circ \\ 0,25\% \end{pmatrix} \quad (5.1)$$

5.2 Synthèse par commande optimale (LQT)

Nous avons implémenté une structure *Linear Quadratic Tracker* (LQT) pour minimiser l'erreur de suivi entre la sortie du système y et un modèle de référence y_r . Le critère à minimiser est défini par :

$$J = \frac{1}{2} \int_0^\infty (e^T Q e + \tilde{u}^T R \tilde{u}) dt \quad (5.2)$$

Où Q et R sont les matrices de pondération permettant de régler le compromis entre précision du suivi et énergie de commande.

5.3 Réglage des matrices de pondération

Pour respecter les spécifications (temps de réponse < 20 s pour V_a et < 2 s pour la pente, sans dépassement), nous avons choisi :

- **Matrice Q** : Une pondération forte a été placée sur l'erreur de pente pour garantir la réactivité de la boucle d'altitude.
- **Matrice R** : Les coefficients ont été ajustés pour éviter des sollicitations brutales des moteurs (limitation de la fatigue) et des gouvernes.

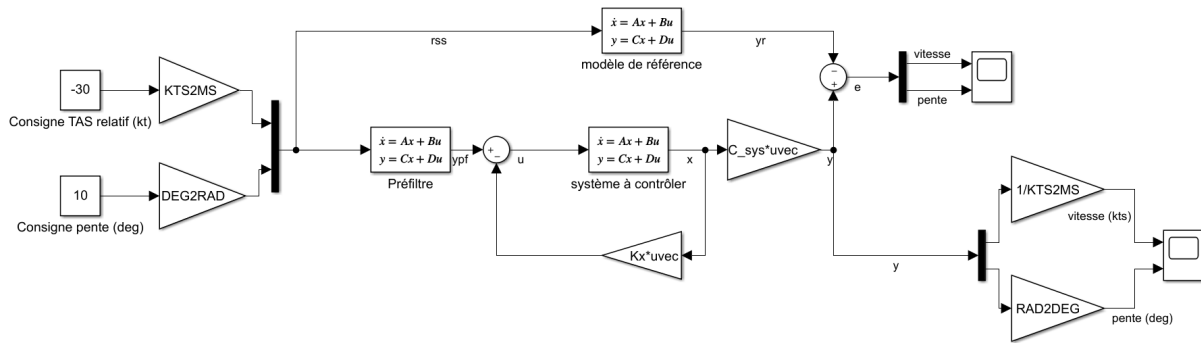


FIGURE 5.1 – Schéma de la commande optimale appliquée au modèle linéaire

5.4 Analyse du couplage

Contrairement aux lois SISO précédentes, la commande optimale LQT permet un **découplage** partiel des axes. Une variation de consigne de vitesse est compensée par une action combinée sur la poussée (action directe) et sur la profondeur (pour contrer le moment cabreur induit par les moteurs). Le graphique suivant illustre comment répond le modèle linéaire grâce à ce découplage.

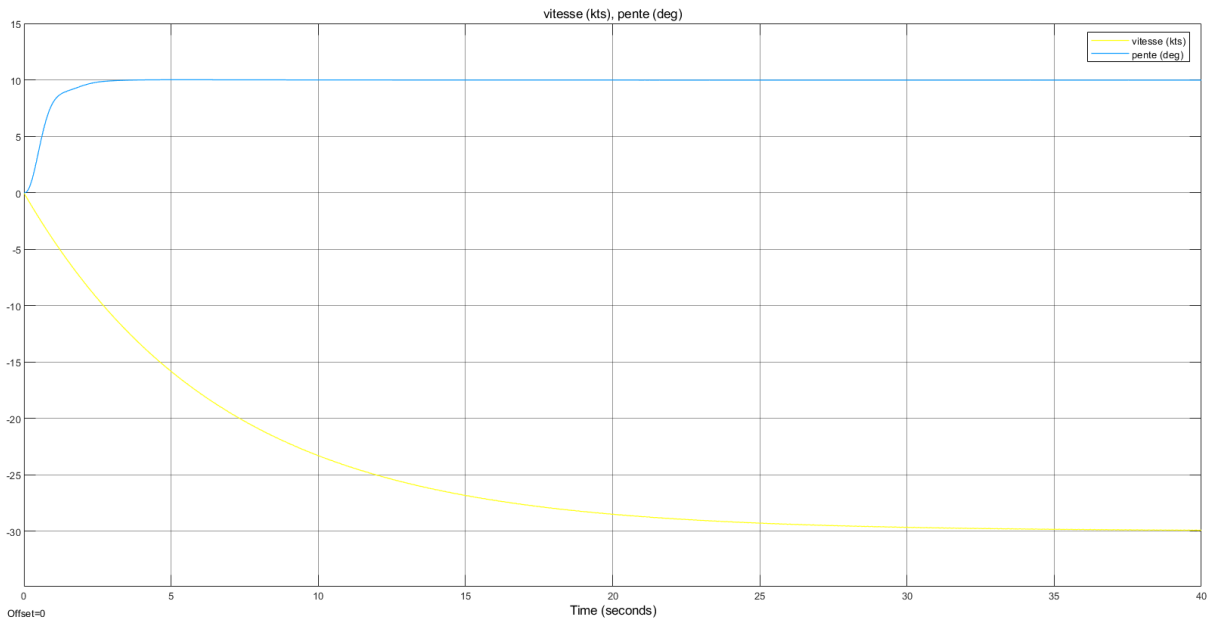


FIGURE 5.2 – Réponse de la commande optimale appliquée au modèle linéaire à des échelons de pente de 10° et de vitesse de 230 kts

6 Simulation du scénario complet

6.1 Gestion de la transition

Le passage de la loi *Open Descent* (vitesse tenue par l'assiette) aux lois de maintien (vitesse tenue par la poussée) constitue une phase critique du vol. La transition entre les lois de commande est faite par un bloc **Switch** qui bascule lorsque l'altitude de l'avion passe sous la barre de 914 m (3000 ft). Ceci implique que l'avion se stabilise environ 50 ft sous son altitude cible. On peut corriger cela naïvement en réglant le seuil du bloc **Switch** (un seuil à 17 m fonctionne), ou en implémentant une loi qui gère plus finement la capture de l'altitude.

6.2 Intégration des différentes lois de commandes

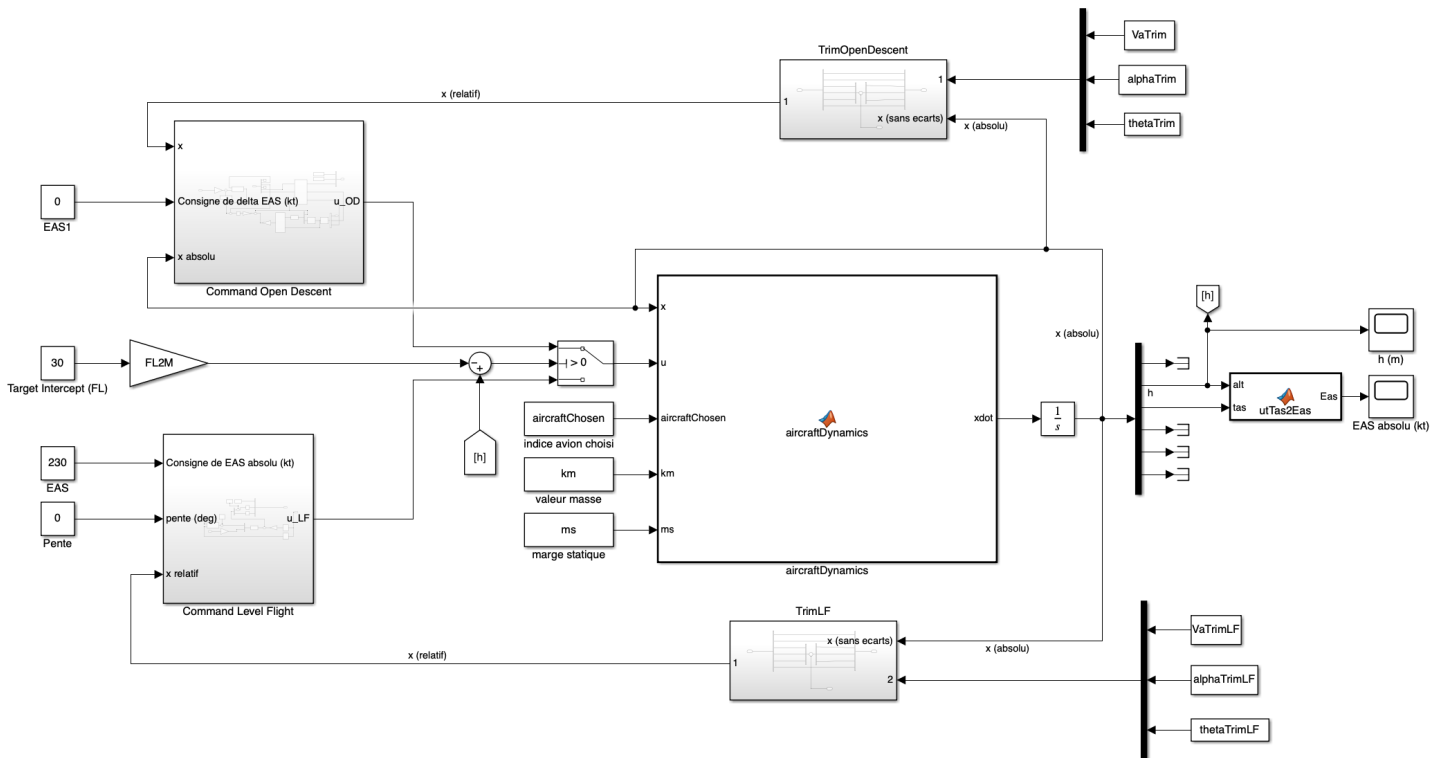
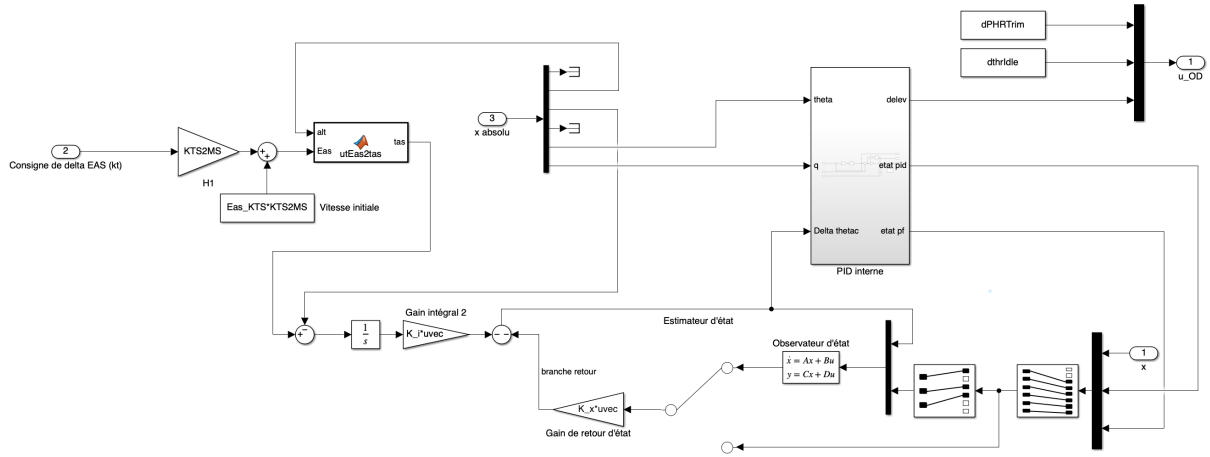
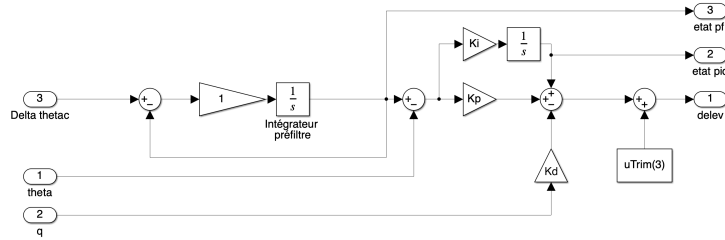


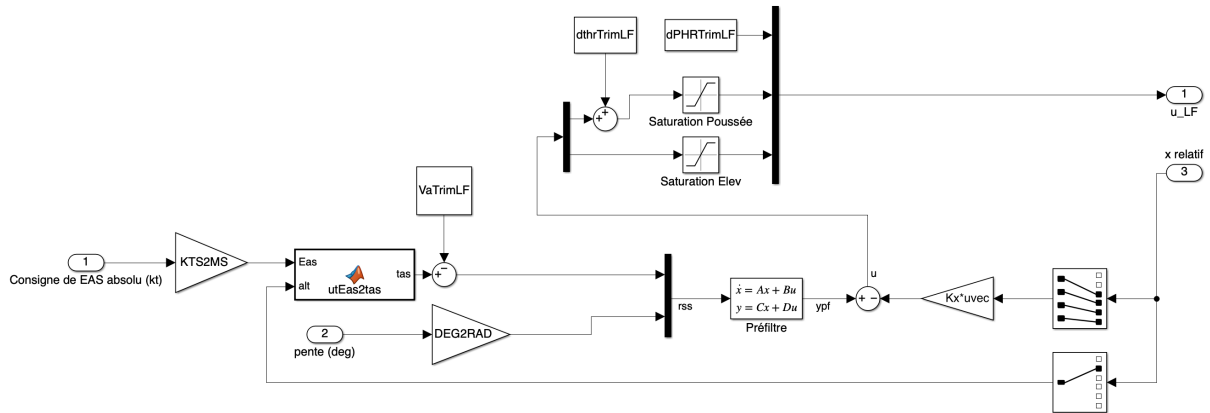
FIGURE 6.1 – Schéma de l'intégration de toutes les lois de commande



(a) Schéma de la commande Open Descent



(b) Schéma du contrôleur PID



(c) Schéma de la commande Level Flight

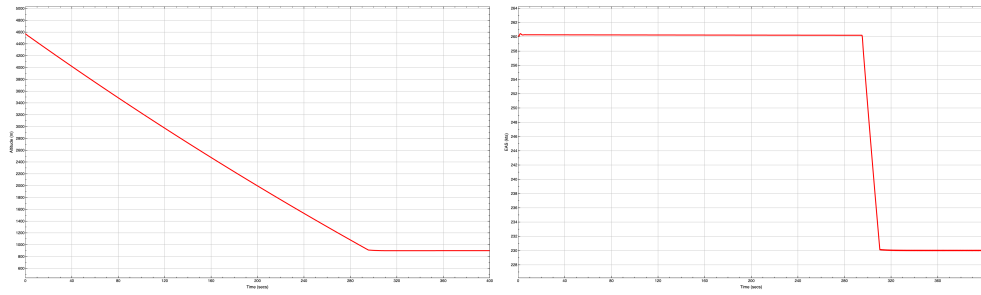
FIGURE 6.2 – Schémas des différentes commandes internes

6.3 Analyse des résultats globaux

La simulation complète montre une descente fluide à 260 kts EAS. Lors de l'approche du FL30 :

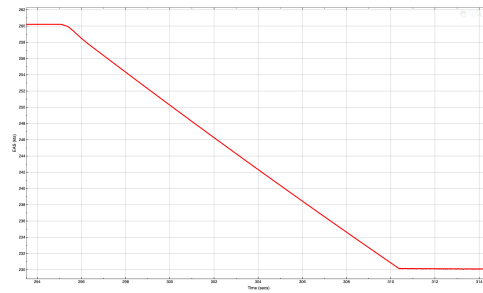
1. La loi de capture d'altitude réduit la pente.
2. La manette des gaz sort de la position *Idle* pour maintenir la vitesse.
3. L'avion se stabilise aux nouvelles consignes sans dépassement excessif, validant la

robustesse des correcteurs sur le modèle non-linéaire.



(a) Courbe de suivi de l'altitude

(b) Courbe de suivi de la vitesse.



(c) Courbe de suivi de la vitesse à la capture du niveau FL30

FIGURE 6.3 – Courbes d'altitude et de vitesse pendant la descente au niveau FL30

7 Conclusion

Ce projet a permis de valider une architecture de pilotage automatique complète. Nous avons pu observer que :

- La linéarisation est un outil puissant mais nécessite l'ajout d'intégrateurs pour compenser les dérives du modèle non-linéaire en cours de descente.
- La commande optimale MIMO offre une bien meilleure gestion des interactions entre les axes que la simple juxtaposition de boucles SISO.

Limites et perspectives Pour améliorer le réalisme, il serait nécessaire d'intégrer des saturations sur les vitesses de déflexion des gouvernes ($15^\circ/\text{s}$ pour la profondeur) et de prendre en compte la dynamique plus lente du PHR par rapport à la gouverne de profondeur.