

PROJET NO 3RÉGULATEUR ET OBSERVATEUR POUR AVIONModèle à utiliser pour tout le projet

- Modèle non linéaire : modèle AVION LONGITUDINAL
- Modèle linéaire : modèle obtenu avec TRIM + LINMOD dans le projet no 1.
 - TRIM pour vol horizontal à $V_T = 80 \text{ m/s}$
- Pour simuler une perturbation dans le modèle non linéaire, que le régulateur/observateur devra contrôler/estimer, on fera des simulations de validation avec un état initial de 85 m/s (état ini = $[85, \alpha_e, \theta_e, q_e]$) plutôt que 80 m/s (les autres variables d'état restent à leur valeur d'équilibre).
- Pour le régulateur, on suppose que la boucle est fermée sur l'actionneur de l'élévateur (angle delta), donc la matrice B du régulateur devient :

$$B_{\text{delta}} = B(:, 1) \rightarrow \text{1}^{\text{ère}} \text{ colonne pour delta}$$

La propulsion reste à l'équilibre α_e
- Pour l'observateur, on suppose que l'unique sortie mesurée est le "pitch rate" q , mesuré avec un gyro bruité. La déviation standard du bruit sera

$$\sigma = 1 \text{ mrad/s} \quad (\text{env. } 0.05 \frac{\text{deg}}{\text{s}})$$

La matrice de mesure est donc : $C_{\text{MES}} = [0, 0, 0, 1]$
 On mesure en radians/s.

PARTIE A : DESIGN DU RÉGULATEUR (PLACEMENT DE PÔLES)

- Critère de performances On veut:

→ temps de stabilisation $t_s \leq 8s$
 → facteur d'amortissement dominant de $\sqrt{2}/2 \approx 0.707$.

(P2-1) Calculer et justifier la position des pôles désirés P .

(P2-2) Déterminer la commandabilité du système avec B_{delta} en utilisant toutes les ~~X~~³ méthodes discutées dans les notes de cours. Donner les résultats par "copie-collage" de la fenêtre ~~"LOG" de "MATRIX-X"~~ (sur WORD)
"Command Window" de MATLAB

~~(P2-3) Faire le design du régulateur (calcul de K) avec la méthode "longue" des notes de cours (algorithme à la page 5-22).~~ Optionnel pour l'année 2014...

(P2-4) Calculer la matrice de gain K du régulateur avec ~~"poleplace".~~ ~~Vérifier que le résultat est le même qu'en (P2-3).~~ Vérifier que $\text{eig}(A - B_{\text{delta}}K) = P$
Sur MATLAB, la fonction est "place".

(P2-5) Valider le design du régulateur par simulation sur le modèle nonlinéaire AVION_LONGITUDINAL. Renommer le modèle (et son fichier) à AVION_CONTROL pour éviter confusion (~~utiliser edit - rename~~).

Vous pouvez aussi faire une simulation avec le modèle original AVION_LONGITUDINAL et une autre avec AVION_CONTROL et les comparer sur les mêmes graphiques.

Dupliquer le modèle nonlinéaire

→ asservir le premier avec le régulateur et mettre ses sorties de 1 à 16
 → laisser le deuxième comme tel et mettre ses sorties de 17 à 32 → il servira de référence par rapport au système régulé.

(IMP) Voir page 5-17 pour s'assurer que le régulateur ne reçoive que les Δx .

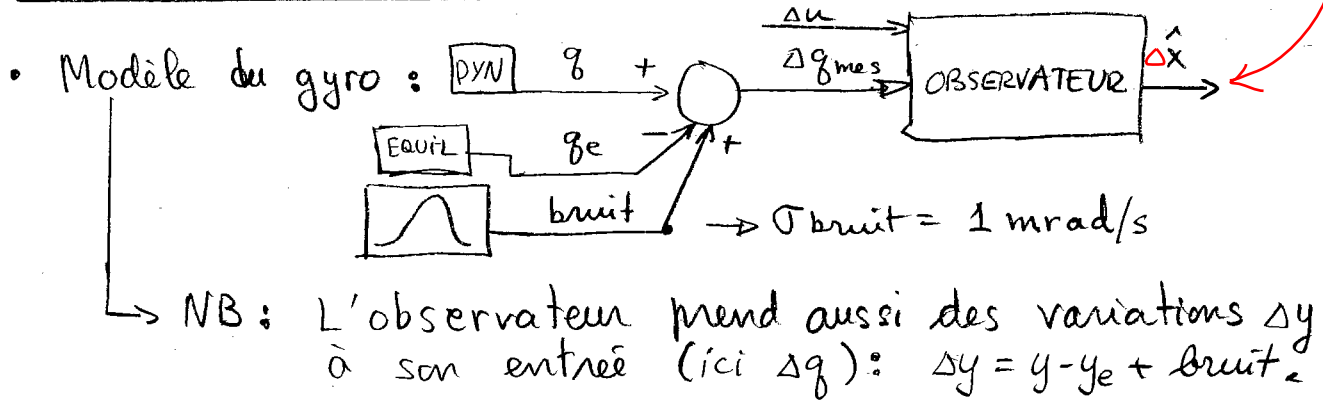
Comparer les sorties pour voir la performance du régulateur, quand $\text{etat_ini} = [\underline{85}, \alpha_e, \theta_e, q_e]$.

Remettre (au moins) la comparaison des sorties :

- (a) vitesse (asservie ① et non-asservie ⑰)
- (b) pitch rate q (" ④ " " " ⑳).
- (c) Vérifier que le temps de stabilisation est bien $t_s \leq 8$ s sur les graphiques

L'observateur comme le régulateur ne prennent que des variations autour de l'équilibre (leur design étant basé sur un système linéarisé).

PARTIE B : DESIGN DE L'OBSERVATEUR



(P2-6) Donner/justifier le choix des pôles désinés \underline{P}_e

(P2-7) Déterminer l'observabilité du système avec :

la fonction MATLAB "obsv(A, C)"

(a) ~~observable(system(A, B, C), C_mes, 0)~~

la fonction MATLAB "ctrb(???,??)"

(b) ~~controllable(system(???,??))~~

Utiliser la dualité de obsv et ctrb pour trouver ce qui doit remplacer les ???.

(OPT) Vérifier (pour vous-même) que le système est toujours observable, peu importe la variable d'état (unique) que l'on mesure $C_{MES} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$ vitesse } etc.
 $C_{MES} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$ alpha

Un seul capteur donne les 4 états. Merveilleux n'est-ce pas!

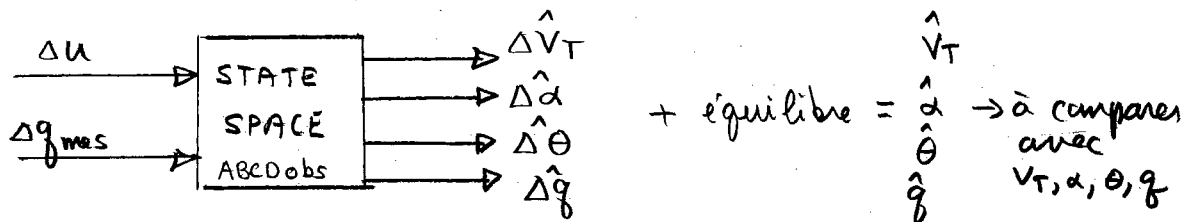
Un seul capteur permet de reconstruire les 4 états du système.

(P2-8) Faire le design de l'observateur (calcul de $\underline{K_e}$) avec ~~place~~ la fonction MATLAB "place".

Vérifier que $\text{eig}(\underline{A} - \underline{K_e} * \underline{C_{MES}}) = \underline{P_e}$

(P2-9) Valider le design de l'observateur par simulation sur le modèle non-linéaire AVION- LONGITUDINAL.

- Renommer le modèle (et son fichier) à AVION-OBSERV.
- Pour l'observateur, utiliser un bloc "STATE-SPACE":



- Le modèle de l'observateur à insérer dans le bloc est:

~~Aobs, Bobs, Cobs, Dobs,~~

$$\underline{ABCDobs} = [\underline{Aobs}, \underline{Bobs}; \underline{Cobs}, \underline{Dobs}]$$

↳ ~~à mettre dans la fenêtre % variable:~~

- \underline{Aobs} : matrice d'état de l'observ. calculée avec $\underline{K_e}$ (à construire)
- \underline{Bobs} : noter que l'observateur a 2 entrées (à construire)
 - (1) l'entrée Δu (qui est zéro pour l'instant)
 - (2) l'entrée mesure Δq_{MES} (en radian)

On doit donc construire \underline{Bobs} (fichier ~~m~~)
- \underline{Cobs} : On suppose que les états sont estimés en radians (pour $\Delta \hat{\alpha}$, $\Delta \hat{\theta}$ et $\Delta \hat{q}$). Cela permet l'addition directe de l'état d'équilibre $\underline{x_e}$ (aussi en radians) pour obtenir l'état estimé $\hat{v}_T, \hat{\alpha}, \hat{\theta}, \hat{q}$. (à construire)
- Donner les équations des 4 matrices de $\underline{ABCDobs}$.
- Mettre les sorties $\hat{v}_T, \hat{\alpha}, \hat{\theta}, \hat{q}$ de 17 à 20

Simulink

- Simuler sur ~~MATRIX-X~~ pour démontrer que l'observateur reconstruit bien les 4 états.

- Partir le système non linéaire aux conditions initiales:

$$\text{etat_ini} = [\underline{85}, \alpha_e, \theta_e, q_e]$$

et partir l'estimateur d'état aux conditions initiales nulles (i.e. initialement, l'estimateur d'état n'a aucune idée de l'état du système).

- Remettre (au moins) la comparaison des états:

(a) vitesse : \hat{V}_T avec V_T

(b) tangage : $\hat{\theta}$ avec θ (comparer radian avec radian ou degré avec degré)

ATTENTION

- NOTE : Plus les pôles P_e sont loins à gauche

→ plus l'observateur converge rapidement

→ plus le bruit du gyro est amplifié

↓
COMPROMIS À FAIRE
en pratique

PARTIE C : RÉGULATEUR + OBSERVATEUR

- (P2-10)
- Relire AVION_CONTROL
 - Le renommer : AVION-OBSCTL (et son fichier)
 - Ajouter l'observateur et le relier au régulateur
- $$\hat{\Delta x} \rightarrow \boxed{K} \xrightarrow{-\Delta u}$$
- Refaire une simulation de validation. (mêmes conditions initiales). Remettre graphique $V_T, \theta, q,$