

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET
POPULAIRE.



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE.



UNIVERSITÉ MOULOUD MAMMERI, TIZI-OUZOU.

FACULTÉ DES SCIENCES

Département des Mathématiques

SPÉCIALITÉ : MATHÉMATIQUES

OPTION : Recherche Opérationnelle

MÉMOIRE DE MASTER

Présenté par :

MELABA Lamia

Sujet :

**Méthodes de contrôle optimal pour l'estimation des paramètres
dynamiques d'un aéronef**

Devant le jury d'examen composé de :

Mr Mellah Omar	MCA à l'UMMTO	Président
Mr Ibaouene Youcef	MCB à l'UMMTO	Directeur
Mme Bedouhene Fazia	Professeure à l'UMMTO	Co-directrice
Mr Yousfi Smail	MCB à l'UMMTO	Examineur
Mme Mokhtari Hanifa	MCB à l'UMMTO	Examinatrice

Promotion : 2024-2025.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à mon directeur de mémoire, Monsieur Youcef IBAOUENE pour son accompagnement précieux tout au long de ce mémoire. Ses conseils avisés, sa disponibilité et ses remarques constructives ont été d'une aide indispensable pour mener à bien ce travail. Je le remercie également pour sa patience, son encouragement et son expertise qui ont grandement enrichi ma réflexion et ma démarche académique. Cette collaboration a été pour moi une réelle opportunité d'apprentissage et de progression. J'exprime aussi ma gratitude à ma co-directrice Madame Fazia BEDOUHENE pour son précieux soutien et ses suggestions pertinentes. Son regard complémentaire, ses remarques ciblées et son encouragement ont significativement contribué à l'aboutissement de ce projet.

Aussi, j'adresse mes plus vifs remerciements à Monsieur Omar MELLAH, Monsieur Smail YOUSFI et Madame Hanifa MOKHTARI pour l'honneur qu'ils m'ont fait en acceptant d'être les membres de mon jury ainsi que pour les remarques et les corrections apportées à mon travail.

Je tiens à exprimer ma profonde gratitude à l'ensemble des enseignants qui m'ont accompagnée durant tout mon parcours universitaire. Leurs connaissances, leur passion et leur engagement pédagogique ont construit les fondations de ma formation. Chaque cours, chaque conseil et chaque discussion ont contribué à forger ma pensée et à nourrir ma curiosité intellectuelle.

Je tiens également à remercier du fond du cœur mes chers parents, mon cher frère, ma chère sœur, pour leur soutien inconditionnel tout au long de ce parcours. Vos encouragements, votre patience et votre présence ont été une source de force et de motivation pour moi. Que ce soit dans les moments de doute ou de réussite, vous avez toujours été là pour me porter, et je vous en suis infiniment reconnaissante. Ce diplôme est aussi le vôtre, car sans votre amour et vos sacrifices, je n'y serais pas arrivée. Je veux aussi dire un énorme merci à mes chères copines pour leur soutien, leur bonne humeur et leur présence précieuse tout au long de cette aventure.

NOTATIONS

- \mathbb{R} : corps des nombres réels
- t_0 : temps initial
- Δt : variation de t
- \mathbf{v}^\top : transposé de vecteur \mathbf{v}
- \mathbf{v} : vecteur des résidus
- $Cov(\mathbf{v})$: matrice de covariance de \mathbf{v}
- θ : vecteur de paramètres
- $\hat{\theta}$: valeurs optimales des paramètres
- θ_0 : valeurs initiaux des paramètres
- tr : trace
- D : différentielle
- \det : déterminant
- $J(\theta, \mathbf{R})$: fonction de coût original
- $\tilde{J}(\theta)$: fonction de coût reformulé
- $\mathbb{L}(Z|\theta)$ fonction de vraisemblance
- $N + 1$: nombre d'échantillons
- \mathbf{R} : matrice de covariance des résidus
- $\mathcal{M}_{n \times m}$: espace de matrices de m lignes et n colonnes
- $p(z_k|\theta)$: probabilité de z_k sachant θ
- s_i : facteur d'échelle pour la i -ème sortie
- $\arg \max$: argument de maximum
- $\arg \min$: argument de minimum
- $C_{\text{eq}}(x(t), u(t), \theta)$: contrainte d'égalité
- $C_{\text{eq}}(x(t), u(t), \theta)$: contrainte d'inégalité
- s_i : facteur d'échelle pour la i -ème sortie
- x : états du système
- $y(t)$: vecteur de sortie mesurable.
- \mathbf{y} : sorties du système
- X : matrice de régression pour la première estimation des moindres carrés

- Y : variables dépendantes pour la première estimation des moindres carrés
- $f(x, u, \theta)$: dynamique des systèmes
- $g(x, u, \theta)$: équation de sortie
- n : est utilisé pour indiquer la dimension des vecteurs, par exemple n_z est utilisé pour la dimension de z .

TABLE DES MATIÈRES

Remerciement	i
Notations	ii
Table des matières	iv
Introduction générale	1
1 Méthode de l'erreur de sortie	3
1.1 Énoncé du problème	3
1.2 Identification des paramètres	4
1.3 Contrôle optimal	14
1.4 Mise à l'échelle numérique (Scaling)	18
2 Résolution du problème d'identification des paramètres d'un avion	20
2.1 Décomposition en mouvements longitudinal et latéral	20
2.2 Génération des mesures	21
2.3 Estimation initiale des paramètres par moindres carrés ordinaires et filtrage passe-bas avant-arrière	22
3 Résultats attendus	28
3.1 Application de l'algorithme de points intérieurs IPOPT sur le mouvement longitudinal	28
3.2 Résultats de simulation (génération des mesures)	37
Annexes	39
3.3 Annexe A : Estimation des paramètres	39
3.4 Annexe B : Contrôle optimal	42
Bibliographie	48

INTRODUCTION GÉNÉRALE

L'estimation des paramètres dynamiques des aéronefs constitue un enjeu majeur en ingénierie aéronautique, car elle permet de modéliser avec précision le comportement des avions en vol. Ces modèles, généralement décrits par des systèmes d'équations différentielles non linéaires, sont indispensables pour la conception des systèmes de contrôle, la simulation des performances et l'amélioration de la sécurité. Ils contiennent toutefois des paramètres inconnus qu'il faut estimer indirectement à partir de données mesurées, en s'appuyant sur des méthodes telles que le maximum de vraisemblance ou les moindres carrés. L'identification précise de ces paramètres reste cependant un défi de taille, en raison de la complexité des systèmes dynamiques en jeu, des incertitudes sur les mesures et des non-linéarités inhérentes aux équations de mouvement. Par ailleurs, la nécessité de disposer de données expérimentales précises et représentatives des conditions réelles de vol ajoute une difficulté supplémentaire à ce processus.

Ce travail s'inscrit dans ce contexte qui vise à approfondir et à élargir les résultats obtenu d'un article intitulé "Aircraft Parameter Estimation Using Optimal Control Methods" [GGBH16]. Il propose une méthodologie avancée pour l'estimation des paramètres dynamiques des aéronefs, en combinant des techniques d'optimisation, des méthodes de contrôle optimal, ainsi que des approches statistiques telles que le principe du maximum de vraisemblance introduit par Fisher [Fis22]. L'objectif ultime de ce travail est de développer une méthodologie robuste pour l'estimation de ces paramètres, reposant sur l'utilisation de techniques spécifiques qui seront détaillées dans le premier chapitre.

Plusieurs sources offrent des analyses approfondies et des perspectives complémentaires. Parmi les références incontournables, on peut citer Ravindra V. Jategaonkar, dont l'ouvrage "Flight Vehicle System Identification : A Time Domain Methodology" [Jat06] propose une lecture détaillée. De même, les travaux de Vladislav Klein et Eugene A. Morelli [MK16] apportent un éclairage original sur les méthodes d'identification des systèmes dynamiques d'aéronefs : fondements théoriques et applications pratiques. Enfin, des études récentes, telles que [HSH⁺23], [HHH25a], [HHH25b], [SZBH21], [ZYYM20], [LHDH23], [SGH24], [PGVSGDA⁺24], [RJ95], [TMP⁺06], [WCH⁺22], [SB24], [Kle89], [Ili89], [ILI87], [GB18], [MCS⁺99], offrent un aperçu des développements les plus récents dans ce domaine.

Organisation du mémoire

1. Dans le premier chapitre, nous présentons le modèle mathématique du système dynamique dont les paramètres sont à estimer. Ensuite, nous introduisons une approche issue de la théorie des probabilités, appelée **maximum de vraisemblance**, qui possède plusieurs propriétés statistiques souhaitables pour un bon estimateur.

Nous définissons alors une fonction de coût, construite de manière à ce que maximiser la vraisemblance revienne à minimiser cette fonction. Pour cela, une procédure de relaxation est mise en œuvre afin de réduire l'écart entre les variables mesurées (les sorties du système) et les réponses estimées (prédites par le modèle). Cette procédure repose sur l'utilisation d'un algorithme spécifique : **l'algorithme de Gauss-Newton**.

Cependant, cette méthode présente certaines limites, notamment en ce qui concerne la sensibilité au choix des paramètres initiaux. Pour surmonter ces difficultés, nous proposons une approche basée sur le **contrôle optimal**, en recourant à des techniques plus récentes telles que la **méthode de point intérieur**.

2. Le chapitre 2 examine en profondeur la structure mathématique des modèles utilisés pour l'estimation des paramètres dynamiques d'un aéronef. Plus précisément, nous présentons de manière complète les équations différentielles linéaires décrivant les mouvements longitudinal et latéral-directionnel de l'appareil, en explicitant chaque terme ainsi que sa signification physique.
3. Dans le troisième chapitre, nous présentons les résultats attendus de l'application des méthodes de contrôle optimal à l'estimation des paramètres d'un modèle dynamique d'aéronef, dans le cadre du mouvement longitudinal.
4. Deux annexes viennent compléter ce mémoire. La première présente en détail les propriétés statistiques de l'estimation par maximum de vraisemblance. La seconde expose les notions fondamentales de contrôle optimal, qui constituent un socle théorique essentiel tout au long de ce travail.

MÉTHODE DE L'ERREUR DE SORTIE

La méthode de l'erreur de sortie est une technique d'estimation des paramètres pour les modèles dynamiques. Elle joue un rôle crucial en ingénierie aéronautique, notamment pour la conception de systèmes de contrôle et la simulation du comportement en vol. L'objectif principal de cette méthode est de minimiser l'erreur entre les variables mesurées (sorties du système) et les réponses estimées (prédites par le modèle).

Dans ce chapitre, nous nous concentrons sur la formulation du problème par l'estimation des paramètres et la méthodologie de résolution utilisée.

1.1 Énoncé du problème

Le modèle mathématique du système dynamique dont les paramètres doivent être estimés est supposé être décrit par la représentation générale suivante

$$\begin{cases} \dot{x}(t) &= f(x(t), u(t), \theta), \\ y(t) &= g(x(t), u(t), \theta), \end{cases} \quad (1.1)$$

où

- $x(t) \in \mathbb{R}^{n_x}$ est un vecteur d'état.
- $u(t) \in \mathbb{R}^{n_u}$ est un vecteur de commande.
- $y(t) \in \mathbb{R}^{n_y}$ est un vecteur de sortie mesurable.
- $\theta \in \mathbb{R}^{n_\theta}$ sont des paramètres à estimer.
- $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x}$ et $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_y}$ sont des fonctions suffisamment différentiables pour permettre un développement en série de Taylor.

Un grand nombre de problèmes pratiques d'identifications de systèmes peuvent être caractérisés par des mesures à temps discret effectuées sur un système dynamique à temps continu. Dans le chapitre 3 de [MK16], les auteurs ont montré que le problème de la modélisation dynamique des avions appartient à cette catégorie. Donc même si le système réel est considéré comme continu,

la nature de l'échantillonnage des mesures à intervalles fixes Δt , nécessite l'utilisation d'équation de sortie discrétisée

$$y_k = y(t_k) = g(x(t_k), u(t_k), \theta) = g(x_k, u_k, \theta), \quad t_k = t_0 + k\Delta t, \quad \forall k \in \{0, \dots, N\}, \quad (1.2)$$

où t_0 représente le temps initial et Δt l'intervalle d'échantillonnage des valeurs discrètes.

1.2 Identification des paramètres

1.2.1 La fonction de vraisemblance maximale pour l'estimation des paramètres de modèle (1.1)

Dans cette partie, nous allons étudier les propriétés statistiques des estimateurs de modèle dynamique (1.1). Pour un traitement bien détaillé de ces propriétés statistiques, le lecteur pourra se référer à l'annexe A.

Dans le cadre de ce problème d'identification des paramètres, une méthode d'erreur de sortie sera mise en œuvre. La base de cette méthode est un ensemble de $N + 1$ mesures discrètes $\mathbf{Z} = [z_0, \dots, z_N]$ avec $\mathbf{Z} \in \mathbb{R}^{n_z \times (N+1)}$. Ces mesures sont considérées comme la somme de la sortie déterministe d'un modèle paramétré $y_k(\theta)$ et d'un bruit aléatoire additif \mathbf{v}_k , que l'on appelle **résidu**. La formulation du problème présentée précédemment constitue un cas spécifique de la formulation plus générale adoptée par la suite, qui ne se limite pas aux modèles dynamiques. L'équation de sortie peut être formulée comme suit

$$z_k = y_k(\theta) + \mathbf{v}_k, \quad k = 0, \dots, N.$$

Supposons que le résidu \mathbf{v}_k est indépendant et identiquement distribué, la probabilité d'obtenir Z , étant donné les paramètres θ , peut être exprimée par la fonction de vraisemblance définie comme suit

$$\mathbb{L}(\mathbf{Z}|\theta) = p(\mathbf{Z}|\theta) = p([z_0, \dots, z_N]|\theta). \quad (1.3)$$

Comme $Z = [z_0, \dots, z_N]$ est un ensemble des échantillons indépendants et identiquement distribués (c'est-à-dire, suit la même loi) (*i.i.d.*), on a

$$p([z_0, \dots, z_N]|\theta) = p(z_0|\theta) \times \dots \times p(z_N|\theta) = \prod_{k=0}^N p(z_k|\theta). \quad (1.4)$$

D'autre part, $y_k(\theta)$ est une valeur déterministe pour θ fixé, alors la probabilité $p(z_k|\theta)$ ne porte pas sur $y_k(\theta)$ mais sur \mathbf{v}_k , et donc, pour tout $k = 0, \dots, N$, on a

$$\mathbb{L}(\mathbf{Z}|\theta) = \prod_{k=0}^N p(z_k|\theta) = \prod_{k=0}^N p(\mathbf{v}_k|\theta). \quad (1.5)$$

Maintenant, Si la composante aléatoire est distribuée selon une distribution gaussienne multivariée avec une moyenne nulle et une matrice de covariance $Cov[\mathbf{v}] = \mathbf{R}$ (matrice symétrique, définie positive), les vecteurs de bruit \mathbf{v}_k suivent une loi normale multivariée $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$ et ils sont *i.i.d.*, alors la densité de probabilité de \mathbf{v}_k est donnée par

$$p(\mathbf{v}_k|\theta) = \frac{1}{\sqrt{(2\pi)^{n_z} \det(\mathbf{R})}} \exp\left(-\frac{1}{2} \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k\right). \quad (1.6)$$

Par conséquent, la fonction de vraisemblance $\mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R})$ est le produit des densités gaussiennes individuelles devient

$$\mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R}) = \prod_{k=0}^N \frac{1}{\sqrt{(2\pi)^{n_z} \det(\mathbf{R})}} \exp\left(-\frac{1}{2}(\mathbf{z}_k - \mathbf{y}_k(\boldsymbol{\theta}))^\top \mathbf{R}^{-1}(\mathbf{z}_k - \mathbf{y}_k(\boldsymbol{\theta}))\right) \quad (1.7)$$

Dans le cas général où \mathbf{R} est inconnu et doit être considéré comme un paramètre supplémentaire, l'estimateur du maximum de vraisemblance $\hat{\boldsymbol{\theta}}$ correspond à l'ensemble des paramètres qui maximisent la probabilité d'observer les mesures Z . Ainsi, le problème d'estimation des paramètres se ramène à un problème d'optimisation de la forme suivante

$$\max_{\boldsymbol{\theta}, \mathbf{R}} \mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R}). \quad (1.8)$$

Dans de nombreuses situations, il est plus pratique de travailler avec le logarithme de la fonction de vraisemblance, que l'on désigne sous le nom de fonction **log-vraisemblance**. Cela s'explique notamment par le fait que l'objectif est souvent de déterminer le maximum de vraisemblance et comme la fonction logarithme est strictement croissante, le maximum de la vraisemblance et celui de la log-vraisemblance sont atteints au même point, c'est-à-dire,

$$\arg \max_{\boldsymbol{\theta}, \mathbf{R}} \mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R}) = \arg \max_{\boldsymbol{\theta}, \mathbf{R}} \ln(\mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R})). \quad (1.9)$$

Par ailleurs, l'utilisation de la log-vraisemblance simplifie généralement le calcul des dérivées nécessaires à la recherche du maximum de vraisemblance. En effet, dans le cas de plusieurs observations indépendantes, le produit des vraisemblances individuelles se transforme, via le logarithme, en une somme de logarithmes. Or, il est nettement plus facile de dériver une somme que de manipuler un produit. On a alors

$$\ln \mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R}) = \sum_{k=0}^N \ln \left[\frac{1}{\sqrt{(2\pi)^{n_z} \det(\mathbf{R})}} \exp\left(-\frac{1}{2}(\mathbf{z}_k - \mathbf{y}_k(\boldsymbol{\theta}))^\top \mathbf{R}^{-1}(\mathbf{z}_k - \mathbf{y}_k(\boldsymbol{\theta}))\right) \right].$$

En utilisant les propriétés du logarithme ($\ln(ab) = \ln a + \ln b$ et $\ln e^x = x$), on obtient

$$\begin{aligned} \ln \mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R}) &= \sum_{k=0}^N \left[-\frac{n_z}{2} \ln(2\pi) - \frac{1}{2} \ln(\det(\mathbf{R})) - \frac{1}{2} \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k \right] \\ &= \sum_{k=0}^N \left[-\frac{n_z}{2} \ln(2\pi) \right] + \sum_{k=0}^N \left[-\frac{1}{2} \ln(\det(\mathbf{R})) \right] + \sum_{k=0}^N \left[-\frac{1}{2} \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k \right] \\ &= -\frac{n_z}{2} \ln(2\pi) \sum_{k=0}^N 1 - \frac{1}{2} \ln(\det(\mathbf{R})) \sum_{k=0}^N 1 - \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k \\ &= -\frac{(N+1)n_z}{2} \ln(2\pi) - \frac{N+1}{2} \ln(\det(\mathbf{R})) - \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k. \end{aligned}$$

On pose

$$J(\boldsymbol{\theta}, \mathbf{R}) = -\ln(\mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R})) = \frac{(N+1)n_z}{2} \ln(2\pi) + \frac{N+1}{2} \ln(\det(\mathbf{R})) + \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k. \quad (1.10)$$

Alors

$$\arg \max_{\boldsymbol{\theta}, \mathbf{R}} (\ln(\mathbb{L}(\mathbf{Z}|\boldsymbol{\theta}, \mathbf{R}))) = \arg \max_{\boldsymbol{\theta}, \mathbf{R}} (-J(\boldsymbol{\theta}, \mathbf{R})) = \arg \min_{\boldsymbol{\theta}, \mathbf{R}} J(\boldsymbol{\theta}, \mathbf{R}). \quad (1.11)$$

Par conséquent, le problème d'optimisation (1.9) consiste alors à minimiser la fonction $J(\theta, \mathbf{R})$. La forte dépendance mutuelle entre θ et \mathbf{R} rend la résolution directe du problème (1.11) entraîne des difficultés de convergence. Afin de surmonter cette complexité, une approche en deux phases, appelée méthode de relaxation, est adoptée. Elle consiste à décomposer l'optimisation de la fonction de vraisemblance en deux étapes successives, ce qui permet de traiter séparément l'estimation des paramètres du système et celle de la matrice de covariance du bruit.

1.2.2 Optimisation de la fonction de coût

La condition nécessaire pour obtenir un extremum est l'annulation du gradient de $J(\theta, \mathbf{R})$. Ce gradient peut être exprimé comme suit

$$\frac{\partial J(\theta, \mathbf{R})}{\partial(\theta, \mathbf{R})} = \begin{bmatrix} \frac{\partial J(\theta, \mathbf{R})}{\partial \theta} \\ \frac{\partial J(\theta, \mathbf{R})}{\partial \mathbf{R}} \end{bmatrix} = \mathbf{0}. \quad (1.12)$$

Pour estimer correctement les deux parties de l'équation (1.12) c'est compliqué directement. Alors elles seront résolues séparément, tout en conservant l'autre partie constante. Autrement dit, la dérivée par rapport à θ est mise à zéro, tout en gardant \mathbf{R} constant, et inversement. En ce qui concerne le second élément, il est possible de le déterminer de manière analytique, ce qui permet d'évaluer la matrice de covariance du bruit pour un θ donné, et d'en déduire l'estimateur du maximum de vraisemblance.

Dans ce travail, nous adoptons une procédure itérative proposée par [Jat06], afin de traiter le problème (1.11), que nous exploitons par la suite. Cette méthode nécessite, dans un premier temps, de résoudre le second terme de l'équation (1.12), pour toute valeur donnée de vecteur de paramètre θ , c'est-à-dire, de trouver la forme analytique de \mathbf{R} pour un θ donné.

Résolution de l'équation de dérivée partielle par rapport à \mathbf{R}

Pour un θ donné, on a

$$\begin{aligned} \frac{\partial J(\theta, \mathbf{R})}{\partial \mathbf{R}} = 0 &\Leftrightarrow \frac{\partial}{\partial \mathbf{R}} \left(\frac{(N+1)n_z}{2} \ln(2\pi) + \frac{N+1}{2} \ln(\det(\mathbf{R})) + \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k \right) = 0 \\ &\Leftrightarrow \frac{N+1}{2} \frac{\partial}{\partial \mathbf{R}} (\ln(\det(\mathbf{R})) + \frac{1}{2} \frac{\partial}{\partial \mathbf{R}} \left(\sum_{k=0}^N \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k \right)) = 0. \end{aligned} \quad (1.13)$$

Premièrement, nous calculons $\frac{\partial}{\partial \mathbf{R}} (\ln(\det(\mathbf{R})))$. Comme \mathbf{R} est définie positive, on a

$$\det(\exp(\mathbf{R})) = \exp(\text{tr}(\mathbf{R})).$$

Alors

$$\det(\mathbf{R}) = \det(e^{\ln(\mathbf{R})}) = e^{\text{tr}(\ln(\mathbf{R}))},$$

et donc

$$\ln(\det(\mathbf{R})) = \ln(e^{\text{tr}(\ln(\mathbf{R}))}) = \text{tr}(\ln(\mathbf{R})).$$

Par conséquent

$$D(\ln(\det(\mathbf{R}))) = D(\text{tr}(\ln(\mathbf{R}))) = \text{tr}(D(\ln(\mathbf{R}))).$$

D'autre part, on a

$$D(\ln(\mathbf{R})) = \mathbf{R}^{-1}D\mathbf{R}.$$

Alors

$$D(\ln(\det(\mathbf{R}))) = \text{tr}(\mathbf{R}^{-1}D\mathbf{R}).$$

Ce qui prouve que

$$\frac{\partial \ln(\det(\mathbf{R}))}{\partial \mathbf{R}} = \frac{D(\ln(\det(\mathbf{R})))}{D\mathbf{R}} = \mathbf{R}^{-1}.$$

Maintenant, nous calculons $\frac{\partial}{\partial \mathbf{R}}(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k)$. On a

$$D(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k) = \mathbf{v}_k^\top (D\mathbf{R}^{-1}) \mathbf{v}_k.$$

D'autre part, on a

$$D(\mathbf{R}\mathbf{R}^{-1}) = D(\mathbf{I}) = 0 \quad (\text{où } \mathbf{I} \text{ désigne la matrice identité}).$$

Alors

$$D(\mathbf{R}\mathbf{R}^{-1}) = \mathbf{R}(D\mathbf{R}^{-1}) + (D\mathbf{R})\mathbf{R}^{-1} = 0.$$

Ce qui implique

$$D\mathbf{R}^{-1} = -\mathbf{R}^{-1}(D\mathbf{R})\mathbf{R}^{-1}.$$

On a donc

$$D(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k) = -\mathbf{v}_k^\top \mathbf{R}^{-1}(D\mathbf{R})\mathbf{R}^{-1} \mathbf{v}_k.$$

Comme $(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k)$ est un scalaire, on a

$$D(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k) = \text{tr}\left(D(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k)\right) = -\text{tr}\left(\mathbf{v}_k^\top \mathbf{R}^{-1}(D\mathbf{R})\mathbf{R}^{-1} \mathbf{v}_k\right).$$

D'après la propriété de la trace de produit des matrices, on aura

$$\text{tr}\left(\mathbf{v}_k^\top \mathbf{R}^{-1}(D\mathbf{R})\mathbf{R}^{-1} \mathbf{v}_k\right) = \text{tr}\left(\mathbf{R}^{-1}(D\mathbf{R})\mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top\right) = \text{tr}\left(\mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1}(D\mathbf{R})\right).$$

Donc

$$D(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k) = -\text{tr}\left(\mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1}(D\mathbf{R})\right) = -\mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1}(D\mathbf{R}).$$

Par conséquent

$$\frac{\partial}{\partial \mathbf{R}}(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k) = \frac{D(\mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k)}{D\mathbf{R}} = -\mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1}.$$

Ce qui prouve que

$$\begin{aligned} \frac{\partial}{\partial \mathbf{R}} \left(\frac{(N+1)n_z}{2} \ln(2\pi) + \frac{N+1}{2} \ln(\det(\mathbf{R})) + \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top \mathbf{R}^{-1} \mathbf{v}_k \right) \\ = -\frac{1}{2} \sum_{k=0}^N \mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1} + \frac{N+1}{2} \mathbf{R}^{-1}. \end{aligned}$$

Finalement, le problème (1.13) devient

$$\frac{\partial J(\theta, \mathbf{R})}{\partial \mathbf{R}} = 0 \Leftrightarrow -\frac{1}{2} \sum_{k=0}^N \mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1} + \frac{N+1}{2} \mathbf{R}^{-1} = 0. \quad (1.14)$$

On multiplie chaque terme de l'équation (1.14) à gauche par \mathbf{R} , on aura

$$\mathbf{R} \left(-\frac{1}{2} \mathbf{R}^{-1} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1} \right) + \mathbf{R} \cdot \left(\frac{N+1}{2} \mathbf{R}^{-1} \right) = 0,$$

et donc

$$-\frac{1}{2} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1} + \frac{N+1}{2} \mathbf{I} = 0.$$

On multiplie maintenant à droite les termes de l'équation ci-dessus par \mathbf{R} , on aura

$$\left(-\frac{1}{2} \mathbf{v}_k \mathbf{v}_k^\top \mathbf{R}^{-1} \right) \mathbf{R} + \left(\frac{N+1}{2} \mathbf{I} \right) \mathbf{R} = -\frac{1}{2} \sum_{k=0}^N \mathbf{v}_k \mathbf{v}_k^\top + \frac{N+1}{2} \mathbf{R} = 0.$$

Le but de ces deux étapes est d'éliminer les inverses matricielles \mathbf{R}^{-1} et obtenir une forme explicite de \mathbf{R} (une solution analytique pour \mathbf{R}). Alors, nous déduisons que

$$\hat{\mathbf{R}}(\theta) = \frac{1}{N+1} \sum_{k=0}^N (z_k - y_k(\theta))(z_k - y_k(\theta))^\top = \frac{1}{N+1} \sum_{k=0}^N \mathbf{v}_k \mathbf{v}_k^\top \quad (1.15)$$

est la solution du problème (1.13).

Procédure de relaxation

Une fois \mathbf{R} estimé, on remplace son expression donnée par l'équation (1.15) dans l'équation (1.10), ce qui conduit à

$$J(\theta, \hat{\mathbf{R}}) = \frac{(N+1)n_z}{2} \ln(2\pi) + \frac{N+1}{2} \ln \left(\det(\hat{\mathbf{R}}) \right) + \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top \hat{\mathbf{R}}^{-1} \mathbf{v}_k.$$

Par conséquent, le premier et dernier terme du membre de droite de l'équation ci-dessus étant des constantes, ils peuvent être négligés sans incidence sur le résultat de la minimisation. Donc, minimiser la fonction de coût revient à minimiser la fonction

$$J(\theta, \hat{\mathbf{R}}) = \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top \hat{\mathbf{R}}^{-1} \mathbf{v}_k. \quad (1.16)$$

D'une autre manière, l'insertion de l'estimateur de la matrice de covariance $\hat{\mathbf{R}}$, défini par (1.15) dans l'équation (1.10) permet de reformuler la fonction de coût comme suit

$$J(\theta) = \frac{(N+1)n_z}{2} \ln(2\pi) + \frac{N+1}{2} \ln \left(\det(\hat{\mathbf{R}}(\theta)) \right) + \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k^\top (\hat{\mathbf{R}}(\theta))^{-1} \mathbf{v}_k. \quad (1.17)$$

Or

$$\sum_{k=0}^N \mathbf{v}_k^\top (\hat{\mathbf{R}}(\theta))^{-1} \mathbf{v}_k = \sum_{k=0}^N \text{tr} \left((\hat{\mathbf{R}}(\theta))^{-1} \mathbf{v}_k \mathbf{v}_k^\top \right) = \text{tr} \left((\hat{\mathbf{R}}(\theta))^{-1} \sum_{k=0}^N \mathbf{v}_k \mathbf{v}_k^\top \right). \quad (1.18)$$

D'autre part, d'après (1.15), on a

$$\sum_{k=0}^N \mathbf{v}_k \mathbf{v}_k^\top = (N+1) \hat{\mathbf{R}}(\theta).$$

En remplaçant le terme ci-dessus dans l'équation (1.18), on obtient

$$\begin{aligned} \sum_{k=0}^N \mathbf{v}_k^\top (\hat{\mathbf{R}}(\theta))^{-1} \mathbf{v}_k &= \text{tr} \left((\hat{\mathbf{R}})^{-1}(\theta) (N+1) \hat{\mathbf{R}}(\theta) \right) \\ &= (N+1) \text{tr} \left((\hat{\mathbf{R}}(\theta))^{-1} \hat{\mathbf{R}}(\theta) \right) = (N+1) \text{tr}(\mathbf{I}) = (N+1)n_z, \end{aligned} \quad (1.19)$$

où \mathbf{I} est une matrice identité de $\mathcal{M}_{n_z \times n_z}(\mathbb{R})$. En substituant l'équation (1.19) dans la fonction de coût (1.17), on aura

$$J(\theta) = \frac{(N+1)n_z}{2} \ln(2\pi) + \frac{N+1}{2} \ln \left(\det(\hat{\mathbf{R}}(\theta)) \right) + \frac{1}{2}(N+1)n_z. \quad (1.20)$$

Comme N et n_z sont fixés, le premier et troisième terme du membre de droite de l'équation (1.20) sont également constants. ils peuvent donc être négligés sans affecter les résultats de la minimisation de la fonction de coût donné par l'équation (1.20). Cette fonction se réduit alors à

$$J(\theta) = \frac{N+1}{2} \ln \left(\det(\hat{\mathbf{R}}(\theta)) \right). \quad (1.21)$$

On peut ainsi résumer la procédure de relaxation de la manière suivante

1. Choisir une valeur initiale θ_0 .
2. Intégrer $\dot{x}(t) = f(x(t), u(t), \theta_0)$ sur $[t_0, t_N]$ pour obtenir $x(t_k)$.
3. Simuler les sorties $y(t_k, \theta_0) = g(x(t_k), u(t_k), \theta_0)$ et calculer les résidus $\mathbf{v}_k = z_k - y(t_k, \theta_0)$.
4. Estimer $\hat{\mathbf{R}}$ à partir des résidus obtenus.
5. Calculer $J(\theta, \hat{\mathbf{R}})$ donné par l'équation (1.16) (ou $J(\theta)$ donné par l'équation (1.21)).
6. Minimiser $J(\theta, \hat{\mathbf{R}})$ (ou minimiser $J(\theta)$) par une méthode d'optimisation non linéaire.
7. Répéter jusqu'à convergence.

Remarque 1.2.1 ([Jat06, P.87]) Il n'existe pas de démonstration générale de la convergence, mais de nombreux exemples portant sur des véhicules aériens de complexité variée, rapportés dans la littérature, montrent clairement qu'il s'agit de la seule approche pragmatique pour l'optimisation non linéaire.

Dans notre cas, la sortie de $y_k(\theta)$ est non linéaire en θ , donc minimiser la fonction de coût J donnée par (1.16) avec une méthode classique n'est pas évident. Pour surmonter cette difficulté, nous utiliserons un algorithme itératif spécialisé, qui est nommé dans la littérature **Algorithme de Gauss-Newton**.

Algorithme de Gauss-Newton

La condition nécessaire pour la minimisation de la fonction de coût par rapport à θ est donnée par

$$\frac{\partial J}{\partial \theta}(\theta, \hat{\mathbf{R}}) = 0. \quad (1.22)$$

Le développement en série de Taylor de $\frac{\partial J}{\partial \theta}(\theta, \hat{\mathbf{R}})$ autour de la i -ème valeur de paramètres θ est donné par

$$\frac{\partial J}{\partial \theta}(\theta_{i+1}, \hat{\mathbf{R}}) = \frac{\partial J}{\partial \theta}(\theta_i, \hat{\mathbf{R}}) + \frac{\partial^2 J}{\partial \theta^2}(\theta_i, \hat{\mathbf{R}}) \Delta \theta, \quad (1.23)$$

où $\Delta \theta = \theta_{i+1} - \theta_i$ le changement du paramètre et $\frac{\partial^2 J}{\partial \theta^2}(\theta_i, \hat{\mathbf{R}})$ est le second gradient (appelé **matrice Hessienne**) de la fonction de coût par rapport à θ à l'itération i . En utilisant la

condition nécessaire de l'équation (1.22) pour le minimum, on annule le membre de droite de l'équation (1.23) et on résout pour $\Delta\theta$, on obtient

$$\Delta\theta = - \left(\frac{\partial^2 J}{\partial \theta^2}(\theta_i, \hat{\mathbf{R}}) \right)^{-1} \left(\frac{\partial J}{\partial \theta}(\theta_i, \hat{\mathbf{R}}) \right). \quad (1.24)$$

En partant d'une valeur initiale supposée θ_0 , on a

$$\theta_{i+1} = \theta_i + \Delta\theta,$$

ce qui fournit une solution itérative pour trouver le minimum. Un tel algorithme est communément appelé la **méthode de Newton-Raphson**.

Cette méthode nécessite donc de calculer le premier et le second gradient de la fonction de coût, et ainsi que les sensibilités des sorties

1. **Gradient de la fonction de coût** : D'après les équations (1.16) et (1.21), deux approches différentes peuvent être utilisées pour évaluer le gradient de la fonction de coût.

Premièrement, on utilise la fonction de coût donnée par l'équation (1.16). On a

$$\frac{\partial J(\theta, \hat{\mathbf{R}})}{\partial \theta} = \frac{1}{2} \sum_{k=0}^N \frac{\partial}{\partial \theta} \left(\mathbf{v}_k^\top \hat{\mathbf{R}}^{-1} \mathbf{v}_k \right). \quad (1.25)$$

Notons $f_k(\theta) = \mathbf{v}_k^\top(\theta) \hat{\mathbf{R}}^{-1} \mathbf{v}_k(\theta)$. Pour $\mathbf{v}_k = (v_1, \dots, v_{n_z})^\top$, on a alors

$$f_k(\theta) = \mathbf{v}_k^\top(\theta) \hat{\mathbf{R}}^{-1} \mathbf{v}_k(\theta) = \sum_{i=1}^{n_z} \sum_{j=1}^{n_z} v_i(\theta) \left(\hat{\mathbf{R}}^{-1} \right)_{ij} v_j(\theta). \quad (1.26)$$

En utilisant les propriétés des dérivées d'un produit, on obtient

$$\begin{aligned} \frac{\partial f_k(\theta)}{\partial \theta} &= \sum_{i=1}^{n_z} \sum_{j=1}^{n_z} \frac{\partial}{\partial \theta} \left(v_i(\theta) \left(\hat{\mathbf{R}}^{-1} \right)_{ij} v_j(\theta) \right) \\ &= \sum_{i=1}^{n_z} \sum_{j=1}^{n_z} \left(\frac{\partial v_i(\theta)}{\partial \theta} \left(\hat{\mathbf{R}}^{-1} \right)_{ij} v_j(\theta) + v_i(\theta) \left(\hat{\mathbf{R}}^{-1} \right)_{ij} \frac{\partial v_j(\theta)}{\partial \theta} \right). \end{aligned} \quad (1.27)$$

Comme $\hat{\mathbf{R}}^{-1}$ de taille $n_z \times n_z$ est symétrique (*i.e.*, $(\hat{\mathbf{R}}^{-1})_{ij} = (\hat{\mathbf{R}}^{-1})_{ji}$), l'équation (1.27) devient

$$\frac{\partial f_k(\theta)}{\partial \theta} = 2 \sum_{i=1}^{n_z} \sum_{j=1}^{n_z} \left(\frac{\partial v_i(\theta)}{\partial \theta} \right)^\top \left(\hat{\mathbf{R}}^{-1} \right)_{ij} v_j(\theta).$$

L'écriture matricielle associée s'écrit sous la forme

$$\frac{\partial f_k(\theta)}{\partial \theta} = 2 \left(\frac{\partial \mathbf{v}_k(\theta)}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1} \mathbf{v}_k(\theta).$$

D'autre part, comme $\mathbf{v}_k = z_k - y_k(\theta)$, on a

$$\frac{\partial \mathbf{v}_k(\theta)}{\partial \theta} = - \frac{\partial y_k(\theta)}{\partial \theta},$$

et donc

$$\frac{\partial f_k(\theta)}{\partial \theta} = 2 \left(- \frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1} \mathbf{v}_k = -2 \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1} \mathbf{v}_k.$$

Par conséquent

$$\frac{\partial J(\theta, \hat{\mathbf{R}})}{\partial \theta} = \frac{1}{2} \sum_{k=0}^N \frac{\partial f_k(\theta)}{\partial \theta} = - \sum_{k=0}^N \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1} (z_k - y_k(\theta)). \quad (1.28)$$

On peut aussi utiliser l'équation (1.21) pour estimer la fonction de coût. Notons

$$\tilde{J}(\theta) = \det(\hat{\mathbf{R}}(\theta)).$$

Alors on a

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{N+1}{2} \left(\frac{\partial \ln(\tilde{J})}{\partial \tilde{J}} \times \frac{\partial \tilde{J}(\theta)}{\partial \theta} \right) = \frac{N+1}{2} \left(\frac{1}{\tilde{J}(\theta)} \times \frac{\partial \tilde{J}(\theta)}{\partial \theta} \right). \quad (1.29)$$

Nous calculons d'abord $\frac{\partial \tilde{J}(\theta)}{\partial \theta}$. En utilisant la formule des chaînes, on obtient

$$\frac{\partial \tilde{J}(\theta)}{\partial \theta} = \sum_{k=0}^N \left(\frac{\partial y_k}{\partial \theta} \right)^\top \frac{\partial \tilde{J}(\theta)}{\partial y_k}, \quad (1.30)$$

tels que $\frac{\partial y_k}{\partial \theta}$ sont les sensibilités du modèle et $\frac{\partial \tilde{J}(\theta)}{\partial y_k}$ est le gradient par rapport à la sortie. Donc dans ce cas, nous avons besoin de déterminer le gradient de \tilde{J} par rapport à la sortie y_k . Nous rappelons que dans le cas général, pour une matrice M et un scalaire a , on a

$$\frac{\partial \det(M)}{\partial a} = \det(M) \cdot \text{tr} \left(M^{-1} \frac{\partial M}{\partial a} \right). \quad (1.31)$$

Alors la dérivée partielle de la fonction $\tilde{J}(\theta)$ par rapport à la j -ème ($j \in 1 \dots, n_z$) élément de la sortie à l'instant k est

$$\frac{\partial \tilde{J}(\theta)}{\partial y_{kj}} = \det(\hat{\mathbf{R}}(\theta)) \cdot \text{tr} \left(\hat{\mathbf{R}}(\theta)^{-1} \frac{\partial \hat{\mathbf{R}}(\theta)}{\partial y_{kj}} \right). \quad (1.32)$$

Nous calculons d'abord la dérivée partielle par rapport à la j -ème élément de la sortie à l'instant k . On a

$$\hat{\mathbf{R}}(\theta) = \frac{1}{N+1} \sum_{k=0}^N (z_k - y_k(\theta))(z_k - y_k(\theta))^\top.$$

Alors

$$\frac{\partial \hat{\mathbf{R}}}{\partial y_{kj}} = \frac{1}{N+1} \left[e_j (y_k - z_k)^\top + (y_k - z_k) e_j^\top \right],$$

où e_j est le vecteur unitaire dans la direction j . En substituant $\frac{\partial \hat{\mathbf{R}}}{\partial y_{kj}}$ dans l'équation (1.32), on aura

$$\begin{aligned} \frac{\partial \tilde{J}(\theta)}{\partial y_{kj}} &= \frac{1}{N+1} \det(\hat{\mathbf{R}}) \text{tr} \left(\hat{\mathbf{R}}^{-1} \left[e_j (y_k - z_k)^\top + (y_k - z_k) e_j^\top \right] \right) \\ &= \frac{1}{N+1} \det(\hat{\mathbf{R}}) \left(\text{tr} \left(\hat{\mathbf{R}}^{-1} \left[e_j (y_k - z_k)^\top \right] \right) + \text{tr} \left(\hat{\mathbf{R}}^{-1} \left[(y_k - z_k) e_j^\top \right] \right) \right). \end{aligned} \quad (1.33)$$

Comme $\hat{\mathbf{R}}^{-1}$ est symétrique, on déduit que

$$\text{tr} \left(\hat{\mathbf{R}}^{-1} \left[e_j (y_k - z_k)^\top + (y_k - z_k) e_j^\top \right] \right) = 2 \text{tr} \left(\hat{\mathbf{R}}^{-1} e_j (y_k - z_k)^\top \right).$$

D'autre part, la matrice $\hat{\mathbf{R}}^{-1}$ peut être décomposée selon

$$\hat{\mathbf{R}}^{-1} = [r_1, \dots, r_{n_z}], \quad \text{où } r_j = [r_{j1}, \dots, r_{jn_z}]^\top, \quad \forall j \in \{1, \dots, n_z\}.$$

Par conséquent

$$\begin{aligned} \text{tr} \left(\hat{\mathbf{R}}(\theta)^{-1} \left[e_j(y_k - z_k)^\top + (y_k - z_k)e_j^\top \right] \right) &= 2\text{tr} \left([r_1, \dots, r_{n_z}] e_j(y_k - z_k)^\top \right) \\ &= 2\text{tr} \left(r_j(y_k - z_k)^\top \right) = 2r_j^\top (y_k - z_k). \end{aligned}$$

En remplaçant cette dernière dans (1.33), on obtient

$$\frac{\partial \tilde{J}(\theta)}{\partial y_{kj}} = \frac{2}{N+1} \det(\hat{\mathbf{R}}(\theta)) \left(2r_j^\top (y_k - z_k) \right). \quad (1.34)$$

Après avoir rassemblé les composantes $\frac{\partial \tilde{J}(\theta)}{\partial y_{kj}}$ (pour tous les indices j) dans un vecteur colonne et en combinant les équations précédentes, il en résulte

$$\frac{\partial \tilde{J}(\theta)}{\partial y_k} = \frac{2}{N+1} \det(\hat{\mathbf{R}}(\theta)) \cdot \hat{\mathbf{R}}(\theta)^{-1} (y_k - z_k). \quad (1.35)$$

En substituant l'expression de ci-dessus dans l'équation (1.30), on aura

$$\frac{\partial \tilde{J}}{\partial \theta} = \frac{2}{N+1} \det(\hat{\mathbf{R}}(\theta)) \sum_{k=0}^N \left(\frac{\partial y_k}{\partial \theta} \right)^\top \left(\hat{\mathbf{R}}^{-1}(\theta) (y_k - z_k) \right). \quad (1.36)$$

L'insertion de (1.36) dans l'équation (1.29), conduit à

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{k=0}^N \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) (y_k(\theta) - z_k) = - \sum_{k=0}^N \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) (z_k - y_k(\theta)). \quad (1.37)$$

Conclusion : Les deux approches conduisent au même gradient.

2. **Gradient second de la fonction de coût :** On a

$$\begin{aligned} \frac{\partial^2 J(\theta, \hat{\mathbf{R}}(\theta))}{\partial \theta^2} &= - \sum_{k=0}^N \left[\frac{\partial}{\partial \theta} \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) (z_k - y_k) + \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) \left(\frac{\partial}{\partial \theta} (z_k - y_k) \right) \right] \\ &= \sum_{k=0}^N \left[\left(\frac{\partial^2 y_k}{\partial \theta^2} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) (z_k - y_k) + \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) \left(\frac{\partial y_k}{\partial \theta} \right) \right]. \end{aligned}$$

Par conséquent

$$\frac{\partial^2 J(\theta, \hat{\mathbf{R}}(\theta))}{\partial \theta^2} = \sum_{k=0}^N \left[\left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) \left(\frac{\partial y_k}{\partial \theta} \right) + \frac{\partial^2 y_k}{\partial \theta^2} \hat{\mathbf{R}}^{-1}(\theta) (z_k - y_k(\theta)) \right]. \quad (1.38)$$

Le calcul du second gradient $\frac{\partial^2 J}{\partial \theta^2}$, donné par l'équation (1.38), est plus complexe et demande davantage de temps, car il est nécessaire le calcul du second gradient $\frac{\partial^2 y_k}{\partial \theta^2}$. Toutefois, le second terme du membre de droite de l'équation (1.38) contient le terme $(z_k - y_k)$, et il a été constaté que cette contribution tend vers zéro à mesure que le processus converge. Dans un cas idéal, les résidus $(z_k - y_k)$ devraient simplement correspondre aux erreurs de

mesure aléatoires à chaque instant t_k . En se basant sur notre hypothèse de bruit à moyenne nulle, ce second terme tend à s'annuler lorsqu'il est sommé sur une quantité suffisante de points de données.

En tenant compte de cette considération pratique, nous négligeons le second terme de l'équation (1.38) et nous approchons le second gradient $\frac{\partial^2 J}{\partial \theta^2}$ par

$$\frac{\partial^2 J(\theta, \hat{\mathbf{R}}(\theta))}{\partial \theta^2} \approx \sum_{k=0}^N \left(\frac{\partial y_k}{\partial \theta} \right)^\top \hat{\mathbf{R}}^{-1}(\theta) \left(\frac{\partial y_k}{\partial \theta} \right). \quad (1.39)$$

Cette approche s'avère efficace dans la majorité des applications pratiques. Dans la littérature, elle est également désignée sous les appellations **Algorithme de Gauss-Newton** (pour plus de détail, voir [Jat06, P.90]).

Remarque 1.2.2 l'approximation est validée lorsque les résidus $(z_k - y_k)$ sont très petits près de la solution et donc le terme ignoré $\frac{\partial^2 y_k}{\partial \theta^2}$ est négligeable.

3. **Sensibilités des coefficients** : La mise à jour itérative du vecteur de paramètres θ par la méthode de Gauss-Newton nécessite le calcul des gradients de la réponse $\frac{\partial y_k}{\partial \theta}$. L'intérêt de cette étape est d'évaluer comment des variations des paramètres θ_i affectent la sortie (il a été démontré que l'augmentation de la sensibilité se traduit par un impact plus significatif du paramètre sur la sortie) (voir [MK16, P.197]).

Nous déduisons du théorème de dérivation des fonctions composées que

$$\frac{\partial y}{\partial \theta} = \frac{\partial}{\partial \theta} g(x, u, \theta) = \frac{\partial g}{\partial x} \frac{\partial x}{\partial \theta} + \frac{\partial g}{\partial \theta}. \quad (1.40)$$

La sensibilité d'état $\frac{\partial x}{\partial \theta}$ est calculée en résolvant l'équation de sensibilité d'état, qui est obtenue en différenciant l'équation d'état (1.1) par rapport aux paramètres inconnus,

$$\frac{D}{Dt} \left(\frac{\partial x}{\partial \theta} \right) = \frac{\partial}{\partial \theta} \left(\frac{Dx}{Dt} \right) = \frac{\partial}{\partial \theta} \dot{x} = \frac{\partial}{\partial \theta} f(x, u, \theta) = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \theta} + \frac{\partial f}{\partial \theta} \quad \text{et} \quad \frac{\partial x(t_0)}{\partial \theta} = 0.$$

Comme le modèle dynamique (1.1) est non linéaire, les dérivées analytiques de f et g sont souvent complexes et fortement dépendantes de la structure propre à chaque problème. C'est pourquoi l'approche numérique est préférée, qui est souvent plus rapide et moins coûteuse. Par exemple, on peut mentionner

- (i) **Méthode des différences finies** : on perturbe les paramètres légèrement et on observe les variations de la sortie.
- (ii) **La méthode de surface locale** : elle construit une petite approximation autour du point d'intérêt et utilise sa pente pour estimer les sensibilités.

Remarque 1.2.3 ([Jat06, p.95]) La méthode de **Gauss-Newton** non contrainte effectue la mise à jour complète des paramètres à chaque itération. En général, cette méthode donne de très bons résultats. Cependant, la mise à jour repose sur l'hypothèse de linéarité locale et de fonction de coût quadratique. De plus, une approximation du second gradient est utilisée pour alléger les calculs numériques, et les gradients de réponse sont eux-mêmes approximés numériquement. En raison de ces limitations, la méthode de **Gauss-Newton** peut, dans certains cas, mal se comporter par exemple lorsque les valeurs initiales sont éloignées de l'optimum ou en présence d'erreurs numériques dues à l'approximation par différences finies des gradients à proximité de

l'optimum. Elle peut alors présenter une divergence locale temporaire ou un blocage. Dans de telles situations, il devient nécessaire de contrôler soigneusement la taille du pas, une fois la direction du vecteur de correction établie.

Remarque 1.2.4 Le problème général est reformulé comme un problème d'optimisation sous contraintes dynamiques suivant

$$\hat{\theta} = \arg \min_{\theta} J(\theta) \quad \text{sous} \quad \begin{cases} \dot{x}(t) = f(x(t), u(t), \theta), \\ y_k(\theta) = g(x(t_k), u(t_k), \theta), \\ \hat{\mathbf{R}}(\theta) = \frac{1}{N+1} \sum_{k=0}^N v_k v_k^{\top}. \end{cases}$$

Remarque 1.2.5 Si $y_k(\theta)$ est linéaire en θ alors, on peut écrire

$$\mathbf{y}_k(\theta) = \mathbf{M}_k \theta,$$

où \mathbf{M}_k est une matrice de taille $n_y \times n_{\theta}$. Alors la sensibilité de coefficient est définie par

$$\frac{\partial \mathbf{y}_k}{\partial \theta} = \mathbf{M}_k,$$

et donc l'équation $\frac{\partial J(\theta, \mathbf{R})}{\partial \theta} = 0$ devient un système linéaire donné par

$$\sum_{k=0}^N \mathbf{M}_k^{\top} \mathbf{R}^{-1} (z_k - \mathbf{M}_k \theta) = 0.$$

Par conséquent

$$\theta = \left(\sum_{k=0}^N \mathbf{M}_k^{\top} \mathbf{R}^{-1} \mathbf{M}_k \right)^{-1} \left(\sum_{k=0}^N \mathbf{M}_k^{\top} \mathbf{R}^{-1} z_k \right).$$

Dans ce cas, il est facile d'utiliser la procédure des itération proposée par [Jat06].

1.3 Contrôle optimal

Dans la section précédente, comme indiqué dans la remarque 1.2.3, l'algorithme de **Gauss-Newton** peut être sensible aux valeurs initiales et aux bruits de mesures, qui nous ramène parfois à la divergence. Dans de telles situations, le problème d'identification de paramètres énoncé précédemment peut se reformuler en un problème de contrôle optimal avec des commandes connues, mais les paramètres du modèle et les états sont inconnus. La tâche consiste alors à déterminer les paramètres $\hat{\theta} \in \mathbb{R}^{n_{\theta}}$ ainsi que la trajectoire d'état correspondante $\hat{x}(t) \in \mathbb{R}^{n_x}$ de manière optimale qui minimisent la fonction de coût $J(x(t), u(t), \theta)$, sous les contraintes dynamiques énoncées dans la remarque 1.2.4, c'est-à-dire, les contrainte d'égalité

$$C_{\text{eq}}(x(t), u(t), \theta) : \begin{cases} \dot{x}(t) = f(x(t), u(t), \theta), \\ y_k(\theta) = g(x(t_k), u(t_k), \theta), \\ \hat{\mathbf{R}}(\theta) = \frac{1}{N+1} \sum_{k=0}^N v_k v_k^{\top}, \end{cases}$$

ainsi que les contraintes d'inégalité suivantes

$$C_{\text{ineq}}(x(t), u(t), \theta) : \begin{cases} \mathbf{x}_{\min} \leq \mathbf{x}_k \leq \mathbf{x}_{\max}, & k = 0, \dots, N \\ \theta_{\min} \leq \theta \leq \theta_{\max}. \end{cases}$$

En conséquence, le problème d'estimation de paramètre annoncé dans le premier chapitre peut être transformé en un problème de contrôle optimal, donné sous la forme

$$\min_{\theta} J(x(t), u(t), \theta) \quad \text{sous contraintes :} \begin{cases} C_{\text{eq}}(x(t), u(t), \theta) = 0, \\ C_{\text{ineq}}(x(t), u(t), \theta) \leq 0. \end{cases} \quad (1.41)$$

Ce problème est non linéaire ; par conséquent, il n'est pas toujours évident de trouver une solution analytique. Dans ce cas, le recours à des méthodes numériques s'avère indispensable pour calculer la solution. On distingue alors trois types de ces méthodes pour la résolution

- **Méthode directe** : Elle consiste à discrétiser le problème continu et à le formuler sous la forme d'un problème d'optimisation non linéaire (**NLP**).
- **Méthode indirecte** : Elle s'appuie sur le principe du maximum de Pontryagin pour formuler un problème aux valeurs limites, résolu numériquement par des techniques de tir simple ou multiple.
- **Méthode de schémas de collocation trapézoïdal** : Une méthode de discrétisation utilisée pour résoudre des problèmes de contrôle optimal en transformant les équations différentielles en contraintes algébriques.

Pour une description plus détaillée sur ces méthodes, le lecteur pourra se référer à l'annexe B.

Afin de résoudre le problème (1.41), nous utilisons la méthode directe et la méthode de collocation trapézoïdal, et nous utilisons un solveur d'optimisation **IPOPI** (Algorithme de point intérieur) pour la résolution numérique.

1.3.1 Principe fondamental de la méthode directe

le concept de base de la méthode directe consiste à suivre les étapes suivantes :

1. **Discrétisation du temps et paramétrisation des états** : Nous divisons l'intervalle de temps en N intervalle $[t_k, t_{k+1}]$, avec $0 \leq k \leq N - 1$, et nous approximons la fonction d'état $x(t)$ par des variables discrètes (*i.e.*, $x_k \approx x(t_k)$).
2. **Approximation trapézoïdale** : L'objectif de cette approximation est de remplacer l'équation différentielle par une contrainte algébrique.

Pour un pas $[t_k, t_{k+1}]$, l'intégration donne

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} f(x(t), u(t), \theta) dt.$$

Dans le cas général, pour une fonction h définie sur $[a, b]$, en effectuant l'approximation de l'intégrale par la règle du trapèze, on aura

$$\int_a^b h(t) dt \approx \frac{b-a}{2} (h(a) + h(b)).$$

Lorsque cela est appliqué à notre cas, on obtient

$$x_{k+1} \approx x_k + \frac{\Delta t}{2} (f(x_k, u_k, \theta) + f(x_{k+1}, u_{k+1}, \theta)).$$

3. **Formulation du problème NLP** : Nous transformons le problème de contrôle optimal (1.41) en un problème d'optimisation non linéaire suivant

$$\min_{\theta} \tilde{J}(\mathbf{X}, \mathbf{U}, \theta) \quad \text{sous contrainte} \quad (1.42)$$

$$C_{\text{eq}}(x_k, u_k, \theta) : \begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\Delta t}{2} (f(\mathbf{x}_k, \mathbf{u}_k, \theta) + f(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, \theta)), & k = 0, \dots, N-1, \\ y_k(\theta) = g(x_k, u_k, \theta), \\ \hat{\mathbf{R}}(\theta) = \frac{1}{N+1} \sum_{k=0}^N v_k v_k^{\top}. \end{cases}$$

$$C_{\text{ineq}}(x_k, u_k, \theta) : \begin{cases} \mathbf{x}_{\min} - \mathbf{x}_k \leq 0 & k = 0, \dots, N, \\ \mathbf{x}_k - \mathbf{x}_{\max} \leq 0 & k = 0, \dots, N, \\ \theta_{\min} - \theta \leq 0, \\ \theta - \theta_{\max} \leq 0, \end{cases}$$

où $\mathbf{X} = (x_0, \dots, x_N)^{\top}$ et $\mathbf{U} = (u_0, \dots, u_N)^{\top}$.

La résolution de ce problème se fait par un algorithme **IPOPT** (Interior Point OPTimizer), un solveur numérique open-source conçu pour résoudre des problèmes d'optimisation non linéaire à grande échelle. Il est particulièrement efficace pour les problèmes d'optimisation continue, où les variables peuvent prendre n'importe quelle valeur réelle dans un certain intervalle, et où les fonctions objectif et de contraintes sont lisses (différentiables).

1.3.2 Algorithme de point intérieur (IPOPT)

L'algorithme de point intérieur transforme le problème (1.42) en

$$\min_{\theta} \left(\tilde{J}(\mathbf{X}, \mathbf{U}, \theta) - \mu \sum_{k=0}^N \log(-C_{\text{ineq}}(x_k, u_k, \theta)) \right) \quad \text{sous contraintes d'égalité} \quad (1.43)$$

$$C_{\text{eq}}(x_k, u_k, \theta) : \begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\Delta t}{2} (f(\mathbf{x}_k, \mathbf{u}_k, \theta) + f(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, \theta)) & k = 0, \dots, N-1 \\ y_k(\theta) = g(x_k, u_k, \theta), \\ \hat{\mathbf{R}}(\theta) = \frac{1}{N+1} \sum_{k=0}^N v_k v_k^{\top}, \end{cases}$$

avec $\mu > 0$ le paramètre de barrière. Ce problème est résolu par des méthodes classiques d'optimisation avec contraintes d'égalités.

L'application de cet algorithme se fait en suivant les étapes suivantes :

1. Discrétisation de la dynamique en transformant le problème (1.42) en un problème (1.43). (déjà fait)
2. Formuler la fonction de coût avec barrière
3. Résoudre le problème (1.43) à l'aide de l'**algorithme de Newton**.
4. Le processus est répété jusqu'à ce que la convergence soit atteinte.

1.3.3 Algorithme de Newton pour optimisation avec contraintes d'égalité

Les étapes de l'algorithme de Newton sont les suivantes

1. **Formulation de la Lagrangienne** : L'objectif principal de la Lagrangienne est de transformer le problème (1.43) en un système équations à résoudre, en combinant la fonction $\left(\tilde{J}(\mathbf{X}, \mathbf{U}, \theta) - \mu \sum_{k=0}^N \log(-C_{\text{ineq}}(x_k, u_k, \theta))\right)$ et la contrainte d'égalité $C_{\text{eq}}(x_k, u_k, \theta)$. Notons $\xi = (\theta, \mathbf{X})^\top$ le vecteur des inconnus. La Lagrangienne de notre cas est donnée

$$\mathcal{L}(\xi, \lambda) = \left(\tilde{J}(\xi) - \mu \sum_{k=0}^N \log(-C_{\text{ineq}}(\xi)) \right) + \lambda^\top C_{\text{eq}}(\xi), \quad (1.44)$$

où λ est le vecteur des multiplicateurs de Lagrange associés aux contraintes d'égalité.

2. **Condition de Karush-Kuhn-Tucher (KKT)** : L'objectif est de trouver (ξ, λ) tel que

$$\begin{cases} \nabla_{\xi} \mathcal{L}(\xi, \lambda) = 0, \\ C_{\text{eq}}(\xi) = 0. \end{cases}$$

Ce système peut s'écrire sous forme matricielle, désigné sous le terme de **système des conditions KKT** :

$$\begin{bmatrix} \nabla_{\xi}^2 \mathcal{L}(\xi, \lambda) & \nabla_{\xi} C_{\text{eq}}(\xi)^\top \\ \nabla_{\xi} C_{\text{eq}}(\xi) & 0 \end{bmatrix} \begin{bmatrix} \Delta \xi \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla_{\xi} \mathcal{L}(\xi, \lambda) \\ C_{\text{eq}}(\xi) \end{bmatrix},$$

où

$$\nabla_{\xi} \mathcal{L}(\xi, \lambda) = \nabla_{\xi} \tilde{J}(\xi) + \mu \sum_{k=0}^N \frac{1}{C_{\text{ineq}}(\xi)} \nabla_{\xi} C_{\text{ineq}}(\xi) + \nabla_{\xi} C_{\text{eq}}(\xi)^\top \lambda,$$

et

$$\begin{aligned} \nabla_{\xi}^2 \mathcal{L}(\xi, \lambda) = & \nabla_{\xi}^2 \tilde{J}(\xi) + \mu \sum_{k=0}^N \left(\frac{1}{(C_{\text{ineq}}(\xi))^2} \nabla_{\xi} C_{\text{ineq}}(\xi) \nabla_{\xi} C_{\text{ineq}}(\xi)^\top - \frac{1}{C_{\text{ineq}}(\xi)} \nabla_{\xi}^2 C_{\text{ineq}}(\xi) \right) \\ & + \nabla_{\xi} \left(\nabla_{\xi} C_{\text{eq}}(\xi)^\top \lambda \right). \end{aligned}$$

L'application de cet algorithme suit les étapes suivantes :

1. **Initialisation** : En utilisant la méthode d'**estimations par les moindres carrés ordinaires (OLS)** ainsi que la méthode de **filtrage avant/arrière** nous obtenons les paramètres initiaux θ_0 et les états initiaux \mathbf{X}^0 admissibles, c'est-à-dire vérifiant toutes les inégalités $C_{\text{ineq}} < 0$. Nous choisissons λ^0 et le paramètre de barrière $\mu > 0$ et fixons le seuil de convergence.
2. **Évaluation à l'itération k** : Pour résoudre le problème (1.43) nous avons besoin de
 - évaluer $\nabla_{\xi} \mathcal{L}(\xi^{(k)}, \lambda^{(k)})$, $\nabla_{\xi}^2 \mathcal{L}(\xi^{(k)}, \lambda^{(k)})$, $\nabla_{\xi} C_{\text{eq}}(\xi^{(k)})$ et $C_{\text{eq}}(\xi^{(k)})$,
 - Résoudre le problème

$$\begin{bmatrix} \nabla_{\xi}^2 \mathcal{L}(\xi^{(k)}, \lambda^{(k)}) & \nabla_{\xi} C_{\text{eq}}(\xi^{(k)})^\top \\ \nabla_{\xi} C_{\text{eq}}(\xi^{(k)}) & 0 \end{bmatrix} \begin{bmatrix} \Delta \xi^{(k)} \\ \Delta \lambda^{(k)} \end{bmatrix} = - \begin{bmatrix} \nabla_{\xi} \mathcal{L}(\xi^{(k)}, \lambda^{(k)}) \\ C_{\text{eq}}(\xi^{(k)}) \end{bmatrix},$$

- **Mettre à jour :** Une fois que la direction de descente $(\Delta\xi^{(k)}, \Delta\lambda^{(k)})$ est déterminée, la solution est réévaluée

$$\begin{cases} \xi^{(k+1)} = \xi^{(k)} + \alpha^{(k)} \Delta\xi^{(k)} \\ \lambda^{(k+1)} = \lambda^{(k)} + \alpha^{(k)} \Delta\lambda^{(k)}, \end{cases}$$

où $\alpha^{(k)}$ est un réel choisi pour que $C_{\text{ineq}}(\xi^{(k+1)}) < 1$.

3. Réduire μ et répéter jusqu'à ce que la convergence soit atteinte.

1.4 Mise à l'échelle numérique (Scaling)

Dans le cadre de l'utilisation d'outils d'optimisation numérique, il est impératif d'appliquer une mise à l'échelle numérique à l'ensemble des valeurs pertinentes afin de les ramener à un ordre de grandeur raisonnable. Une démarche fréquente consiste à normaliser l'ensemble des valeurs dans l'intervalle $[0,1]$. Cette démarche s'applique aux éléments suivants : les paramètres d'optimisation, la fonction coût et les contraintes. La mise à l'échelle des paramètres et des contraintes peut être réalisée de manière efficace par une transformation linéaire appropriée. Cette opération est effectuée automatiquement dans le code de contrôle optimal utilisé dans le cadre de cette étude. Cependant, la normalisation de la fonction de coût s'avère plus complexe.

Les fonctions de coût J et \tilde{J} comparent les valeurs simulées aux mesures obtenues en réalité, tel que

$$\mathbf{v}_k = \begin{bmatrix} (v_k)_1 \\ (v_k)_2 \\ \vdots \\ (v_k)_{n_y} \end{bmatrix} = \mathbf{z}_k - \mathbf{y}_k(\theta), \quad k = 0, \dots, N.$$

Lorsque les résidus de la fonction de coût présentent des ordres de grandeur différents, un ou plusieurs termes peuvent dominer la fonction, tandis que les autres valeurs y contribuent à peine. Dans ce cas, une optimisation équilibrée devient impossible et la solution optimale obtenue risque d'être non pertinente. Par conséquent, aucune solution optimale raisonnable ne peut être obtenue. De plus, en raison de la nature de \mathbf{R} dans l'équation (1.13), une faible amplitude des résidus $(\mathbf{v}_k)_i$ ainsi qu'un nombre d'échantillons N peuvent entraîner la fin prématurée de l'algorithme d'optimisation : Au fur et à mesure que les résidus diminuent jusqu'à $|(\mathbf{v}_k)_i| < 1$, ils conduisent la fonction de coût $J = \det(\mathbf{R})$ à un point proche de la précision de la machine où la procédure d'optimisation numérique échoue.

Afin de résoudre ces problèmes, il est proposé de mettre à l'échelle les résidus, ce qui permettrait éventuellement de mettre à l'échelle la fonction de coût globale. Dans le cadre de cette étude, les résidus ont été mis à l'échelle. Les résidus échelonnés sont les suivants

$$(\mathbf{v}_k^*)_i = s_i \cdot (\mathbf{v}_k)_i, \quad i = 1, \dots, n_y$$

avec s_i , $i \in [1, \dots, n_y]$ donné par

$$s_i = \sqrt{(N+1) \left(\sum_{k=0}^N (\mathbf{v}_k)_i^2 \right)^{-1}}$$

étant des facteurs d'échelle constants qui sont calculés en utilisant la solution initiale θ_0 et maintenue constante au cours du processus.

Soit $J(\theta)^*$ la fonction de coût après le scaling , tel que

$$J(\theta_0)^* = \det \mathbf{R}^*(\theta_0) = \left| \frac{1}{N+1} \sum_{k=0}^N \mathbf{v}_k^* \mathbf{v}_k^{*\top} \right|,$$

et \mathbf{R}^* est donnée sous la forme matricielle suivante

$$\mathbf{R}^* = \frac{1}{N+1} \det \begin{bmatrix} \sum_{k=0}^N s_1^2 (\mathbf{v}_k)_1^2 & \cdots & \sum_{k=0}^N s_1 s_{n_y} (\mathbf{v}_k)_1 (\mathbf{v}_k)_{n_y} \\ \vdots & \ddots & \vdots \\ \sum_{k=0}^N s_1 s_{n_y} (\mathbf{v}_k)_1 (\mathbf{v}_k)_{n_y} & \cdots & \sum_{k=0}^N s_{n_y}^2 (\mathbf{v}_k)_{n_y}^2 \end{bmatrix},$$

supposons que les termes hors-diagonale (covariances croisées) sont négligeables alors

$$\mathbf{R}^* \approx \frac{1}{N+1} \text{diag} \left(\sum_{k=0}^N s_1^2 (\mathbf{v}_k)_1^2, \dots, \sum_{k=0}^N s_{n_y}^2 (\mathbf{v}_k)_{n_y}^2 \right),$$

et donc, le déterminant de la matrice \mathbf{R}^* devient le produit des éléments diagonaux

$$\det(\mathbf{R}^*) \approx \frac{1}{N+1} \prod_{i=1}^{n_y} \left(s_i^2 \sum_{k=0}^N (\mathbf{v}_k)_i^2 \right).$$

En remplaçant s_i par sa valeur, on obtient

$$\begin{aligned} \det(\mathbf{R}^*) &\approx \frac{1}{N+1} \prod_{i=1}^{n_y} \left(\frac{N+1}{\sum_{j=0}^N (v_j)_i^2} \cdot \sum_{k=0}^N (v_k)_i^2 \right) \\ &\approx \frac{1}{N+1} \prod_{i=1}^{n_y} \sum_{k=0}^N \left[(N+1) \left(\sum_{j=0}^N (v_j)_i^2 \right)^{-1} (v_k)_i^2 \right] \\ &= \prod_{i=1}^{n_y} \sum_{k=0}^N \frac{(v_k)_i^2}{\sum_{j=0}^N (v_j)_i^2} = 1. \end{aligned}$$

RÉSOLUTION DU PROBLÈME D'IDENTIFICATION DES PARAMÈTRES D'UN AVION

Dans ce chapitre, nous allons détailler la structure mathématique des modèles utilisés pour l'estimation des paramètres dynamiques de l'aéronef donné. Notre objectif est de présenter de manière exhaustive les équations différentielles linéaires qui gouvernent les mouvements longitudinal et latéral-directionnel de l'appareil, en explicitant chaque terme et son interprétation physique.

2.1 Décomposition en mouvements longitudinal et latéral

Un modèle dynamique linéaire de l'avion est considéré, conformément à l'approche décrite notamment dans [GGBH16]. Ce modèle global se décompose en deux sous-modèles distincts : l'un dédié à la dynamique longitudinale, l'autre à la dynamique latérale.

2.1.1 Mouvement longitudinal

Le comportement longitudinal de l'appareil est généralement représenté à l'aide d'un système d'équations différentielles linéaires, dont la forme standard est donnée par

$$\begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} X_V & -g \cos \gamma_0 & X_\alpha & X_q \\ -Z_V & \frac{g}{V_0} \sin \gamma_0 & -Z_\alpha & -Z_q \\ Z_V & -\frac{g}{V_0} \sin \gamma_0 & Z_\alpha & Z_q + 1 \\ M_V & 0 & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} V \\ \gamma \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} X_\eta & X_{\delta_T} \\ -Z_\eta & -Z_{\delta_T} \\ Z_\eta & Z_{\delta_T} \\ M_\eta & M_{\delta_T} \end{bmatrix} \begin{bmatrix} \eta \\ \delta_T \end{bmatrix}, \quad (2.1)$$

où

- La vitesse cinématique V , l'angle de trajectoire γ , l'angle d'attaque α et la vitesse angulaire de tangage q sont les états.

- La déflexion de l'élévateur η et la commande de poussée δ_T sont des entrées.

Le vecteur des paramètres aérodynamiques à estimer pour la dynamique longitudinale s'écrit comme suit

$$\theta_{\text{long}} = [X_V \quad X_\alpha \quad X_q \quad Z_V \quad Z_\alpha \quad Z_q \quad M_V \quad M_\alpha \quad M_q \quad X_\eta \quad X_{\delta_T} \quad Z_\eta \quad Z_{\delta_T} \quad M_\eta \quad M_{\delta_T}]^\top.$$

2.1.2 Mouvement latéral

La dynamique latérale de l'avion peut être modélisée à l'aide du système d'équations suivant

$$\begin{bmatrix} \dot{r} \\ \dot{\beta} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} N_r & N_\beta & N_p & 0 \\ Y_r - \cos \alpha_0 & Y_\beta & Y_p + \sin \alpha_0 & \frac{g}{V_0} \cos \alpha_0 \\ L_r & L_\beta & L_p & 0 \\ \tan \phi_0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \beta \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} N_\xi & N_\zeta \\ -Y_\xi & -Y_\zeta \\ L_\xi & L_\zeta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix}, \quad (2.2)$$

où

- les variables d'état sont le taux de lacet r , angle de dérapage β , le taux de roulis p et l'angle d'inclinaison ϕ .
- Les entrées sont les ailerons ξ et les gouvernes de direction ζ .

Le vecteur des paramètres à estimer est donné par

$$\theta_{\text{lat}} = [N_r \quad N_\beta \quad N_p \quad Y_r \quad Y_\beta \quad Y_p \quad L_r \quad L_\beta \quad L_p \quad N_\xi \quad N_\zeta \quad Y_\xi \quad Y_\zeta \quad L_\xi \quad L_\zeta]^\top.$$

Remarque 2.1.1 Dans les deux types de mouvements présentés ci-dessus, les composantes du vecteur de paramètres correspondent aux dérivées des forces et des moments agissant sur le système. Les notations X , Y et Z représentent les forces selon les axes longitudinaux, latéraux et verticaux respectivement, tandis que L , M et N désignent les moments de roulis, de tangage et de lacet. Les indices associés précisent la variable dont dépend chaque dérivée. Par exemple $L_r = \frac{\partial L}{\partial r}$.

2.2 Génération des mesures

Dans cette section, nous détaillons la méthodologie de simulation non linéaire d'un aéronef utilisée pour générer des données de mesure artificielles, essentielles pour valider notre approche d'estimation paramétrique. Ces données synthétiques reproduisent les conditions réelles d'un vol instrumenté, tout en permettant un contrôle précis des sources de bruit et des excitations dynamiques.

Les mesures sont générées à partir d'un modèle non linéaire décrit par

$$\begin{cases} \dot{x}_{\text{nl}}(t) = f_{\text{nl}}(x_{\text{nl}}(t), u(t)), \\ y_{\text{nl}}(t) = g_{\text{nl}}(x_{\text{nl}}(t), u(t)), \\ x_{\text{nl}}(t_0) = x_{\text{nl},0}, \end{cases} \quad (2.3)$$

où

- $x_{\text{nl}}(t)$ est le vecteur d'état non linéaire.

- $u(t)$ est le vecteur des commandes.
- f_{nl}, g_{nl} sont deux fonctions non linéaires modélisant la dynamique et les sorties du système.

Remarque 2.2.1 Pour garantir une identification fiable, les entrées $u(t)$ sont conçues pour exciter spécifiquement les dynamiques longitudinales et latérales.

Le modèle (2.3) est implémenté dans **MATLAB**, où l'ont décrit comment les entrées de commande sont générées, et comment les mesures bruitées sont produites pour l'estimation des paramètres.

Les états simulés, parfaitement connus, sont ensuite corrompus par un bruit blanc Gaussien avec un rapport signal-sur-bruit de 0.1, donné par

$$z_k = x_{nl}(t_k) + v_k,$$

où

- z_k : Mesure bruitée à l'instant t_k ,
- $x_{nl}(t_k)$: Valeur vraie (simulée),
- v_k : Bruit Gaussien centré (moyenne nulle) avec écart-type valant 10% de l'amplitude du signal.

Reproduisant ainsi les incertitudes de mesure des capteurs réels tout en conservant un référentiel de vérité terrain. Cette approche offre un double avantage : elle permet une vérification aisée des résultats tout en maintenant une proximité avec les conditions réelles de vol. Bien que limitée à des données synthétiques, cette méthodologie pose les bases pour une future extension à des données expérimentales, dont l'application est en cours d'investigation.

2.3 Estimation initiale des paramètres par moindres carrés ordinaires et filtrage passe-bas avant-arrière

Dans cette section, nous allons décrire une méthode pour générer une estimation initiale des paramètres inconnus θ et des états du modèle dynamique de l'avion. Cette étape est cruciale pour garantir la convergence de l'algorithme d'optimisation utilisé dans la méthode de contrôle optimal. La stratégie adoptée repose sur une approche par moindres carrés ordinaires OLS, permettant d'extraire une première approximation des paramètres à partir des données expérimentales pour estimer les paramètres initiaux à partir des données mesurées, tout en traitant le bruit présent dans les mesures à l'aide d'un filtre passe-bas.

2.3.1 Formulation du problème

Forme matricielle du modèle linéaire du mouvement longitudinal et latéral

Les modèles linéaires du mouvement longitudinal (2.1) et latéral (2.2) peuvent être réécrits sous forme d'équations d'état linéaires

$$\dot{x} = Ax + Bu, \tag{2.4}$$

où :

- $x \in \mathbb{R}^4$ est le vecteur d'état,
- $u \in \mathbb{R}^2$ est le vecteur de commande,

- $A \in \mathcal{M}_4(\mathbb{R})$ est la matrice d'état,
- $B \in \mathcal{M}_{4,2}(\mathbb{R})$ est la matrice de commande.

Les matrices A et B peuvent être interprétées ligne par ligne, permettant de traiter chaque équation d'état de manière indépendante pour l'estimation par (OLS). Ainsi, pour la j -ième équation d'état, nous avons

$$\dot{x}_j = \mathbf{a}_j^\top \mathbf{x} + \mathbf{b}_j^\top \mathbf{u}, \quad j = 1, \dots, 4.$$

Séparation des termes connus et inconnus

Pour chaque équation d'état, on sépare les termes fixes (connus) notés par $\bar{\mathbf{a}}_j$ et $\bar{\mathbf{b}}_j$ et les termes à estimer notés par $\hat{\mathbf{a}}_j$ et $\hat{\mathbf{b}}_j$. On obtient alors une nouvelle équation pour chaque ligne de l'équation (2.4) donnée par

$$\dot{x}_j = (\hat{\mathbf{a}}_j^\top + \bar{\mathbf{a}}_j^\top) \mathbf{x} + (\hat{\mathbf{b}}_j^\top + \bar{\mathbf{b}}_j^\top) \mathbf{u}, \quad j = 1, \dots, 4$$

et on déduit que

$$\dot{x}_j - \bar{\mathbf{a}}_j^\top \mathbf{x} - \bar{\mathbf{b}}_j^\top \mathbf{u} = [\mathbf{x}^\top \quad \mathbf{u}^\top] \begin{bmatrix} \hat{\mathbf{a}}_j \\ \hat{\mathbf{b}}_j \end{bmatrix}$$

Cette équation constitue la forme fondamentale pour l'application des (OLS). Dans le cadre de cette étude, le terme « gauche » est considéré comme la variable dépendante. En revanche, le terme « droite » représente le produit de la matrice des variables indépendantes, et du vecteur des paramètres inconnus à estimer.

En combinant tous les $N + 1$ équations, on aura le système suivant

$$\begin{bmatrix} (\dot{x}_j)_0 \\ \vdots \\ (\dot{x}_j)_N \end{bmatrix} - \begin{bmatrix} x_0^\top & u_0^\top \\ \vdots & \vdots \\ x_N^\top & u_N^\top \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{a}}_j \\ \tilde{\mathbf{b}}_j \end{bmatrix} = \begin{bmatrix} x_0^\top & u_0^\top \\ \vdots & \vdots \\ x_N^\top & u_N^\top \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}}_j \\ \hat{\mathbf{b}}_j \end{bmatrix}, \quad (2.5)$$

Notons par

- Y_j le vecteur qui regroupe les valeurs de la dérivée d'état \dot{x}_j à chaque pas de temps, ajustées par les contributions des termes connus. Il représente la variable que le modèle tente d'expliquer.
- X_j la matrice qui est construite à partir des états x_k et des entrées u_k mesurés ou simulés à chaque pas de temps. Chaque ligne de X_j correspond aux valeurs des variables indépendantes à un instant donné.

Alors le système (2.5) s'écrit sous forme

$$Y_j = X_j \begin{bmatrix} \hat{\mathbf{a}}_j \\ \hat{\mathbf{b}}_j \end{bmatrix},$$

où $\begin{bmatrix} \hat{\mathbf{a}}_j \\ \hat{\mathbf{b}}_j \end{bmatrix}$ le vecteur des coefficients que nous cherchons à estimer. Les estimations des éléments de $\begin{bmatrix} \hat{\mathbf{a}}_j^\top & \hat{\mathbf{b}}_j^\top \end{bmatrix}$ peuvent maintenant être calculées via le **pseudo-inverse de Moore-Penrose** de

X_j , tel que

$$\begin{bmatrix} \hat{\mathbf{a}}_j \\ \hat{\mathbf{b}}_j \end{bmatrix} = \left(X_j^\top X_j \right)^{-1} X_j^\top \cdot Y_j.$$

Le nombre d'éléments dans $\begin{bmatrix} \hat{\mathbf{a}}_j^\top & \hat{\mathbf{b}}_j^\top \end{bmatrix}$ varie en fonction de la ligne du système d'espace d'états considérée. En effet, les éléments zéro sont supprimés de $\begin{bmatrix} \hat{\mathbf{a}}_j^\top & \hat{\mathbf{b}}_j^\top \end{bmatrix}$ et X_j respectivement. Dans le cadre de cette étude, en combinant les résultats pour toutes les lignes $j = 1, \dots, 4$, on obtient une estimation initiale θ_0 du vecteur de paramètres θ .

Par exemple pour $\dot{\beta}$, le problème d'estimation des paramètres est donné par

$$\begin{bmatrix} \dot{\beta}_0 \\ \vdots \\ \dot{\beta}_N \end{bmatrix} - \begin{bmatrix} r_0 & \beta_0 & p_0 & \xi_0 & \zeta_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_N & \beta_N & p_N & \xi_N & \zeta_N \end{bmatrix} \begin{bmatrix} -\cos \alpha_0 \\ 0 \\ \sin \alpha_0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_0 & \beta_0 & p_0 & \xi_0 & \zeta_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_N & \beta_N & p_N & \xi_N & \zeta_N \end{bmatrix} \begin{bmatrix} Y_r \\ Y_\beta \\ Y_p \\ -Y_\xi \\ -Y_\zeta \end{bmatrix}$$

$$\begin{bmatrix} \hat{Y}_r \\ \hat{Y}_\beta \\ \hat{Y}_p \\ -\hat{Y}_\xi \\ -\hat{Y}_\zeta \end{bmatrix} = \left(X_\beta^\top X_\beta \right)^{-1} X_\beta^\top Y_\beta,$$

2.3.2 Application des Moindres Carrés Ordinaires OLS

L'estimation par moindres carrés ordinaires OLS s'appuie sur la minimisation de la somme des carrés des résidus entre les mesures et les prédictions du modèle dans le cadre de la résolution des équations d'état. Pour chaque équation d'état \dot{x}_j , on cherche à résoudre un système linéaire surdéterminé de la forme

$$Y_j = X_j \theta_j + \varepsilon,$$

où

- $\mathbf{Y}_j \in \mathbb{R}^{N+1}$: vecteur des dérivées estimées $\dot{x}_j(t_k) - \hat{\mathbf{a}}_j^\top \mathbf{x}(t_k) - \hat{\mathbf{b}}_j^\top \mathbf{u}(t_k)$,
- $\mathbf{X}_j \in \mathcal{M}_{(N+1),2}(\mathbb{R})$: matrice des mesures d'état et de commande,
- $\boldsymbol{\theta}_j \in \mathbb{R}^2$: paramètres à estimer,
- $\varepsilon \in \mathbb{R}^{N+1}$: bruit résiduel.

Construction des Matrices OLS

La construction des matrices des moindres carrés se fait en deux étapes principales

1. Assemblage de X_j et Y_j , c'est-à-dire pour chaque instant t_k ($k \in \{0, \dots, N\}$), on a

- La ligne k de X_j est

$$(x_1(t_k), \dots, x_{n_x}(t_k), u_1(t_k), \dots, u_{n_u}(t_k)),$$

- L'élément k de Y_j est

$$\dot{x}_j(t_k) = \sum_{i=1}^{n_x} \bar{\mathbf{a}}_{ji} x_i(t_k) - \sum_{i=1}^{n_u} \bar{\mathbf{b}}_{ji} u_i(t_k).$$

Par exemple, dans le mouvement latéral, pour $\dot{\beta}$, on a

$$\mathbf{Y}_\beta = \begin{bmatrix} \dot{\beta}(t_0) + r(t_0) \cos(\alpha_0) - p(t_0) \sin(\alpha_0) \\ \vdots \\ \dot{\beta}(t_N) + r(t_N) \cos(\alpha_0) - p(t_N) \sin(\alpha_0) \end{bmatrix}, \quad \mathbf{X}_\beta = \begin{bmatrix} r(t_0) & \beta(t_0) & p(t_0) & \xi(t_0) & \zeta(t_0) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r(t_N) & \beta(t_N) & p(t_N) & \xi(t_N) & \zeta(t_N) \end{bmatrix},$$

2. Vérification du rang de X_j , c'est-à-dire, pour que $(X_j^\top \cdot X_j)$ soit inversible, X_j doit être de rang plein, c'est-à-dire que ses colonnes soient linéairement indépendantes.

Remarque 2.3.1 Dans le cadre de la collecte de données, il est essentiel que les mesures enregistrées, typiquement liées aux entrées et aux sorties du système, fournissent des informations suffisantes. Cette exigence est primordiale pour garantir la capacité à observer, identifier ou estimer avec fiabilité les dynamiques notables du système.

Résolution par moindres carrés ordinaires OLS

Par exemple pour $\dot{\beta}$, le problème d'estimation des paramètres est donné par

$$\begin{bmatrix} \dot{\beta}_0 \\ \vdots \\ \dot{\beta}_N \end{bmatrix} - \begin{bmatrix} r_0 & \beta_0 & p_0 & \xi_0 & \zeta_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_N & \beta_N & p_N & \xi_N & \zeta_N \end{bmatrix} \begin{bmatrix} -\cos \alpha_0 \\ 0 \\ \sin \alpha_0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_0 & \beta_0 & p_0 & \xi_0 & \zeta_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_N & \beta_N & p_N & \xi_N & \zeta_N \end{bmatrix} \begin{bmatrix} Y_r \\ Y_\beta \\ Y_p \\ -Y_\xi \\ -Y_\zeta \end{bmatrix}$$

$$\begin{bmatrix} \hat{Y}_r \\ \hat{Y}_\beta \\ \hat{Y}_p \\ -\hat{Y}_\xi \\ -\hat{Y}_\zeta \end{bmatrix} = \left(X_\beta^\top X_\beta \right)^{-1} X_\beta^\top Y_\beta, \quad (2.6)$$

tels que les produits matriciels $X_\beta X_\beta^\top$ et $X_\beta^\top Y_\beta$ sont donnés successivement par

$$X_\beta^\top X_\beta = \sum_{k=0}^N \begin{bmatrix} r_k^2 & \beta_k r_k & \cdots & -\xi_k r_k & -\zeta_k r_k \\ \beta_k r_k & \beta_k^2 & \cdots & -\xi_k \beta_k & -\zeta_k \beta_k \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\xi_k r_k & -\xi_k \beta_k & \cdots & \xi_k^2 & \zeta_k \xi_k \\ -\zeta_k r_k & -\zeta_k \beta_k & \cdots & \zeta_k \xi_k & \zeta_k^2 \end{bmatrix},$$

et

$$X_\beta^\top Y_\beta = \sum_{k=0}^N \begin{bmatrix} r_k(\dot{\beta}_k + \cos(\alpha_0)r_k - \sin(\alpha_0)p_k) \\ \beta_k(\dot{\beta}_k + \cos(\alpha_0)r_k - \sin(\alpha_0)p_k) \\ \vdots \\ -\zeta_k(\dot{\beta}_k + \cos(\alpha_0)r_k - \sin(\alpha_0)p_k) \end{bmatrix}.$$

Si les données sont excitantes, c'est-à-dire, la matrice $X_\beta X_\beta^\top$ est symétrique et définie positive, la solution du système linéaire (2.6) peut être donnée par la décomposition QR , telle que la matrice X_β peut être factorisée par

$$X_\beta = QR,$$

où

- Q : matrice orthogonale ou matrice inverse ($Q^\top Q = I$),
- R : matrice triangulaire supérieure.

Alors

$$X_\beta^\top X_\beta = (QR)^\top (QR) = R^\top Q^\top QR = R^\top R \quad (\text{car } Q^\top Q = I)$$

et son inverse est donnée par

$$\left(X_\beta^\top X_\beta \right)^{-1} = \left(R^\top R \right)^{-1} = R^{-1} \left(R^\top \right)^{-1}.$$

Donc la solution est donnée par

$$\begin{bmatrix} \hat{Y}_r \\ \hat{Y}_\beta \\ \hat{Y}_p \\ -\hat{Y}_\xi \\ -\hat{Y}_\zeta \end{bmatrix} = R^{-1} (R^\top)^{-1} (QR)^\top Y_\beta = R^{-1} (R^\top)^{-1} R^\top Q^\top Y_\beta = R^{-1} Q^\top Y_\beta.$$

On peut aussi trouver la solution par la décomposition **SVD**, c'est-à-dire, la décomposition en **valeurs singulières**

$$X = U \Sigma V^\top,$$

où

- U et V sont des matrices orthogonales, dont les colonnes sont respectivement les vecteurs singuliers à gauche et à droite de la matrice. Plus précisément, les colonnes de U sont les vecteurs singuliers à gauche, celles de V à droite,
- Σ est la matrice diagonale des σ_i (valeurs singulières).

2.3.3 Méthode de Filtrage pour l'estimation des états et de leurs dérivées

Dans le cadre de l'estimation des paramètres d'un modèle dynamique d'aéronef, une étape cruciale consiste à obtenir des valeurs numériques précises pour les dérivées des états $(\dot{x}_j)_k$. Ces dérivées sont essentielles pour construire la matrice Y_j , qui permet de calculer une estimation initiale des paramètres θ_0 via la méthode des moindres carrés ordinaires (OLS). Cependant, les mesures réelles sont souvent bruitées, et l'utilisation de différences finies pour estimer les

dérivées à partir de ces mesures peut amplifier le bruit, conduisant à des résultats peu fiables. Pour contourner ce problème, on propose une approche basée sur un filtre passe-bas appliqué de manière bidirectionnelle (avant et arrière dans le temps).

Problème des différences finies

La dérivée $\dot{x}(t)$ peut être approximée numériquement à l'aide de la méthode des différences finies. Pour un signal échantillonné à intervalles réguliers de pas Δt , la dérivée au temps t_k s'exprime par

$$\dot{x}(t_k) \approx \frac{x(t_{k+1}) - x(t_k)}{\Delta t}.$$

Cette approximation repose sur un développement de Taylor au premier ordre et constitue une méthode simple pour estimer les variations instantanées d'un signal discret.

Si les mesures $x(t_k)$ sont bruitées, le calcul des différences finies amplifie les hautes fréquences (bruit), car

$$\text{Bruit dans } \dot{x} \approx \frac{\text{Bruit dans } x}{\Delta t}.$$

Plus Δt est petit, plus l'erreur est grande. Par conséquent, les dérivées estimées deviennent trop bruitées pour être utilisées dans l'estimation des paramètres. Pour atténuer le bruit tout en conservant les dynamiques utiles, on utilise un filtre passe-bas du premier ordre, appliqué en deux passes

- **Passé avant** : Lissage des données dans le sens normal du temps.
- **Passé arrière** : Correction du déphasage induit par la première passe.

1. Filtrage passe-bas (première passe) : Le filtre est défini par l'équation différentielle

$$\begin{cases} \dot{\hat{x}}_j^f(t) &= -\frac{1}{T}\hat{x}_j^f(t) + \frac{1}{T}x_j(t), \\ \hat{x}_j^f(t=0) &= x_j(t=0), \end{cases}$$

où T est le temps du filtre.

2. Correction du déphasage (seconde passe) : La première passe introduit un décalage temporel. Pour le compenser, on applique le même filtre en temps inverse

$$\dot{\hat{x}}_j^b(t_b) = -\frac{1}{T}\hat{x}_j^b(t_b) + \frac{1}{T}\hat{x}_j^f(t_b), \quad t_b = t_f - t$$

avec

$$\hat{x}_j^b(t_b = 0) = \hat{x}_j^f(t = t_f).$$

À cette étape, les états filtrés $\hat{x}_j^b(t)$ sont synchronisés avec les entrées non filtrées u_k et on élimine le déphasage tout en conservant le lissage.

Au cours de la deuxième exécution du filtre, des informations supplémentaires sur les dérivées de l'état aux instants d'échantillonnage t_k peuvent être extraites

$$\begin{bmatrix} (\hat{x}_j^s)_k \\ -(\dot{\hat{x}}_j^s)_k \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{T} \end{bmatrix} \hat{x}_j^s(t_k) + \begin{bmatrix} 0 \\ \frac{1}{T} \end{bmatrix} \hat{x}_j^f(t_k),$$

avec $(\hat{x}_j^s)_k$ est l'estimation lissée de l'état et $(\dot{\hat{x}}_j^s)_k$ est l'estimation lissée de la dérivée sont utilisés pour calculer Y_j pour obtenir une estimation initiale de θ_0 et les états lissés $(\hat{x}_j^s)_k$ peuvent être utilisés comme valeurs de départ pour la trajectoire de l'état à reconstruire.

RÉSULTATS ATTENDUS

Dans ce chapitre, nous présentons les résultats attendus de l'application des méthodes de contrôle optimal à l'estimation des paramètres d'un modèle dynamique d'aéronef pour le mouvement longitudinal. Ces résultats sont divisés en plusieurs sections, correspondant aux différents aspects analysés dans notre travail. Pour chaque résultat, une interprétation détaillée est fournie, mettant en lumière les implications pratiques et théoriques.

3.1 Application de l'algorithme de points intérieurs IPOPT sur le mouvement longitudinal

Pour faire une étude à notre cas, on utilise les données simulées pour le mouvement longitudinal et le mouvement latéral.

Temps (s)	V (m/s)	γ (rad)	α (rad)	q (rad/s)	η (rad)	δ_T (°)
0.0	100.0	0.01	0.05	0.0	0.0	80
0.1	100.2	0.012	0.052	0.001	0.01	80
0.2	100.1	0.011	0.051	0.002	0.02	80

Table 4.1 – Données simulées du mouvement longitudinal

3.1.1 Application de la méthode de filtrage avant-arrière (Forward-Backward Filtering) pour le mouvement longitudinal

Pour le mouvement longitudinal, nous allons appliquer la méthode de **filtrage avant-arrière**, dont l'équation d'état (2.1) peut s'écrire comme suit

$$\begin{cases} \dot{V} = X_V V + X_\alpha \alpha + X_q q + X_\eta \eta + X_{\delta_T} \delta_T, \\ \dot{\gamma} = -Z_V V + \frac{g \sin \gamma_0}{V_0} \gamma - Z_\alpha \alpha - Z_q q - Z_\eta \eta - Z_{\delta_T} \delta_T, \\ \dot{\alpha} = Z_V V - \frac{g \sin \gamma_0}{V_0} \gamma + Z_\alpha \alpha + (Z_q + 1)q + Z_\eta \eta + Z_{\delta_T} \delta_T, \\ \dot{q} = M_V V + M_\alpha \alpha + M_q q + M_\eta \eta + M_{\delta_T} \delta_T, \end{cases} \quad (3.1)$$

Pour l'équation \dot{V}

- Nous utilisons **différence finie centrée** pour estimer \dot{V} aux points intermédiaires, on trouve

$$\dot{V}_1 \approx \frac{V_2 - V_0}{2\Delta t} = \frac{100.1 - 100.0}{0.2} = 0.5 \text{ m/s}^2,$$

Pour les points extrêmes, nous utilisons des **différences avant/arrière**, on trouve

$$\dot{V}_0 = \frac{V_1 - V_0}{\Delta t} = \frac{100.2 - 100.0}{0.1} = 2.0 \text{ m/s}^2 \quad (\text{pour } t = 0.0 \text{ s}),$$

$$\dot{V}_2 = \frac{V_2 - V_1}{\Delta t} = \frac{100.1 - 100.2}{0.1} = -1.0 \text{ m/s}^2 \quad (\text{pour } t = 0.2 \text{ s}),$$

- Nous appliquons un filtre passe-bas (lissage des données) avec un temps de retard T choisi empiriquement (par exemple, $T = 0.1 \text{ s}$) la formule de filtre avant est donnée par

$$\dot{\hat{V}}^f(t) = -\frac{1}{T} \hat{V}^f(t) + \frac{1}{T} V(t), \quad \hat{V}^f(0) = V_0$$

Résolution numérique par la **méthode d'Euler explicite** avec $\Delta t = 0,1 \text{ s}$, on a

$$\hat{V}_{k+1}^f = \hat{V}_k^f + \Delta t \left(-\frac{1}{T} \hat{V}_k^f + \frac{1}{T} V_k \right),$$

- À $t = 0.0 \text{ s}$: $\hat{V}_f(0) = V(0) = 100.0 \text{ V}$,
- A $t = 0.1 \text{ s}$:

$$\hat{V}_1^f = \hat{V}_0^f + \Delta t \left(-\frac{\hat{V}_0^f}{T} + \frac{V_0}{T} \right) = 100.0 + 0.1 \left(-\frac{100.0}{0.1} + \frac{100.0}{0.1} \right) = 100.0 \text{ m/s},$$

- A $t = 0.2 \text{ s}$:

$$\hat{V}_2^f = 100.0 + 0.1 \left(-\frac{100.0}{0.1} + \frac{100.2}{0.1} \right) = 100.0 + 0.1 \times 2.0 = 100.2 \text{ m/s}.$$

- Nous appliquons un filtrage arrière, l'équation de filtre est donnée par

$$\dot{\hat{V}}^b(t_b) = -\frac{1}{T} \hat{V}^b(t_b) + \frac{1}{T} \hat{V}^f(t_b), \quad t_b = 0.2 - t, \quad \hat{V}^b(0) = \hat{V}^f(0.2) = 100.2 \text{ m/s},$$

- À $t_b = 0.1 \text{ s}$ ($t = 0.1 \text{ s}$), on a

$$\hat{V}_1^b = 100.2 + 0.1 \left(-\frac{100.2}{0.1} + \frac{100.0}{0.1} \right) = 100.2 + 0.1 \times (-2.0) = 100.0 \text{ m/s},$$

- À $t_b = 0.2\text{ s}$ ($t = 0.0\text{ s}$), on a

$$\hat{V}_0^b = 100.0 + 0.1 \left(-\frac{100.0}{0.1} + \frac{100.0}{0.1} \right) = 100.0 \text{ m/s},$$

- Nous appliquons une estimation des dérivées lissées donnée par la formule

$$\dot{V}_k^s = \frac{1}{T} (V_k - \hat{V}_k^b),$$

on trouve

- À $t = 0.0\text{ s}$:

$$\dot{V}_0^s = \frac{1}{0.1} (100.0 - 100.0) = 0 \text{ m/s}^2,$$

- À $t = 0.1\text{ s}$:

$$\dot{V}_1^s = \frac{1}{0.1} (100.2 - 100.0) = 2.0 \text{ m/s}^2,$$

- À $t = 0.2\text{ s}$:

$$\dot{V}_2^s = \frac{1}{0.1} (100.1 - 100.2) = -1.0 \text{ m/s}^2.$$

Les résultats finaux sont présentés dans le tableau suivant

Temps t (s)	V (m/s)	\hat{V}^b (m/s)	\hat{V}^s (m/s ²)
0.0	100.0	100.0	0.0
0.1	100.2	100.0	2.0
0.2	100.1	100.2	-1.0

TABLE 3.1 – Résultats des estimations de vitesse et d'accélération

Pour γ , α et q , nous avons effectué une programmation Matlab, et les résultats sont présentés dans la table et la figure suivantes

t (s)	γ (rad)	$\dot{\gamma}$ (rad/s)	α (rad)	$\dot{\alpha}$ (rad/s)	q (rad/s)	\dot{q} (rad/s ²)
0,0	0,010 000	0,000 000	0,050 000	0,000 000	0,000 000	0,000 000
0,1	0,010 000	0,020 000	0,050 000	0,020 000	0,000 000	0,010 000
0,2	0,011 000	-0,010 000	0,051 000	-0,010 000	0,001 000	0,010 000

TABLE 3.2 – Résultats numériques du filtrage pour γ , α et q

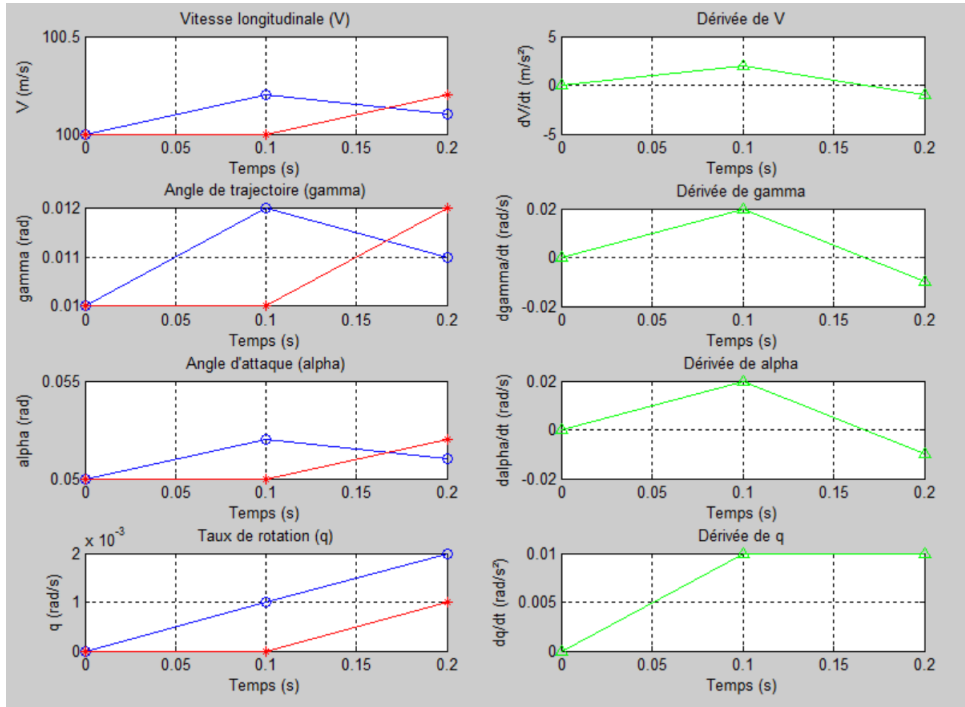


FIGURE 3.1 – Résultats de la méthode de filtrage avant-arrière pour le mouvement longitudinal

3.1.2 Initialisation par Moindres Carrés Ordinaires OLS

L'initialisation par OLS est une étape cruciale pour fournir une première estimation des paramètres avant d'utiliser des méthodes plus sophistiquées comme le maximum de vraisemblance. Nous allons donc estimer les paramètres θ du modèle d'état

$$\dot{x} = A(\theta)x + B(\theta)u,$$

du mouvement longitudinal, Pour chaque équation d'état, nous calculons les paramètres initiaux associés.

1. Calcul des paramètres $X_V, X_\alpha, X_q, X_\eta, X_{\delta_T}$ de l'équation \dot{V} .

Le problème d'estimation par les moindres carrés est donné par

$$\dot{V}_k = X_V V_k + X_\alpha \alpha_k + X_q q_k + X_\eta \eta_k + X_{\delta_T} \delta_{T,k} + \varepsilon_k,$$

avec

$$Y = \begin{bmatrix} \dot{V}_0 \\ \dot{V}_1 \\ \vdots \\ \dot{V}_N \end{bmatrix}, \quad X = \begin{bmatrix} V_0 & \alpha_0 & q_0 & \eta_0 & \delta_{T,0} \\ V_1 & \alpha_1 & q_1 & \eta_1 & \delta_{T,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ V_N & \alpha_N & q_N & \eta_N & \delta_{T,N} \end{bmatrix}.$$

Nous cherchons à estimer le vecteur de paramètres $\theta = [X_V, X_\gamma, X_\alpha, X_q, X_\eta, X_{\delta_T}]^\top$ qui minimise l'erreur quadratique entre les valeurs prédites et observées de \dot{V} . La solution théorique des moindres carrés est donnée par

$$\theta_V = (X^\top X)^{-1} X^\top Y.$$

Pour trois points de mesures (instants) (t_0, t_1, t_2) , on a la matrice des observations X et le vecteur des observations Y donnés comme suit

$$X = \begin{bmatrix} V_0 & \alpha_0 & q_0 & \eta_0 & \delta_{T,0} \\ V_1 & \alpha_1 & q_1 & \eta_1 & \delta_{T,1} \\ V_2 & \alpha_2 & q_2 & \eta_2 & \delta_{T,2} \end{bmatrix} = \begin{bmatrix} 100.0 & 0.05 & 0.0 & 0.0 & 1.396 \\ 100.2 & 0.052 & 0.001 & 0.01 & 1.396 \\ 100.1 & 0.051 & 0.002 & 0.02 & 1.396 \end{bmatrix},$$

et

$$Y = \begin{bmatrix} \dot{V}_0 \\ \dot{V}_1 \\ \dot{V}_2 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 2.0 \\ -1.0 \end{bmatrix}.$$

- Premièrement, nous calculons de $X^\top Y$

$$X^\top Y = \begin{bmatrix} 100.0 \cdot 0.0 + 100.2 \cdot 2.0 + 100.1 \cdot (-1.0) \\ 0.05 \cdot 0.0 + 0.052 \cdot 2.0 + 0.051 \cdot (-1.0) \\ 0.0 \cdot 0.0 + 0.001 \cdot 2.0 + 0.002 \cdot (-1.0) \\ 0.0 \cdot 0.0 + 0.01 \cdot 2.0 + 0.02 \cdot (-1.0) \\ 1.396 \cdot 0.0 + 1.396 \cdot 2.0 + 1.396 \cdot (-1.0) \end{bmatrix} = \begin{bmatrix} 100.3 \\ 0.053 \\ 0.0 \\ 0.0 \\ 1.396 \end{bmatrix}.$$

- Par la suite, nous calculons $X^\top X$

$$X^\top X = \begin{bmatrix} 100.0 & 100.2 & 100.1 \\ 0.05 & 0.052 & 0.051 \\ 0.0 & 0.001 & 0.002 \\ 0.0 & 0.01 & 0.02 \\ 1.396 & 1.396 & 1.396 \end{bmatrix} \begin{bmatrix} 100.0 & 0.05 & 0.0 & 0.0 & 1.396 \\ 100.2 & 0.052 & 0.001 & 0.01 & 1.396 \\ 100.1 & 0.051 & 0.002 & 0.02 & 1.396 \end{bmatrix}$$

$$= \begin{bmatrix} 30060 & 15.316 & 0.3004 & 3.004 & 419.22 \\ 15.316 & 0.007805 & 0.000154 & 0.00154 & 0.213559 \\ 0.3004 & 0.000154 & 0.000005 & 0.00005 & 0.004188 \\ 3.004 & 0.00154 & 0.00005 & 0.0005 & 0.04188 \\ 419.22 & 0.21359 & 0.004188 & 0.04188 & 5.8464 \end{bmatrix}$$

- Finalement, nous calculons $(X^\top X)^{-1}$. La matrice $X^\top X$ est singulière ($\det(X^\top X) \approx 0$), ce qui rend son inversion numérique instable. Nous utilisons alors une **décomposition SVD** pour calculer sa pseudo-inverse (pour plus de détails sur cette méthode le lecteur peut se référer à l'annexe B)

$$X^\top X = U \Sigma V^\top,$$

U , V sont des matrices orthogonales (leurs colonnes sont des vecteurs orthonormés), et Σ est la matrice diagonale contenant les valeurs singulières $\sigma_i \geq 0$, triées par ordre décroissant. Pour calculer les valeurs de ces matrices, on calcule d'abord les valeurs propres λ_i et les vecteurs propres v_i associés à la matrice $(X^T X)$. On a donc

$$\det(X^T X - \lambda I) = 0.$$

La matrice $X^T X$ est 5×5 , alors par un calcul numérique (Matlab), on trouve

$$\lambda_1 \approx 30065.854594, \quad \lambda_2 \approx 0.000203, \quad \lambda_3 \approx 0.000004, \quad \lambda_4 \approx \lambda_5 \approx 0.$$

et la matrice des vecteurs propres associés donnée par

$$V \approx \begin{bmatrix} -0.9999 & 0.0005 & 0.0130 & 0.0006 & 1.3177 \\ -0.0005 & 0.0566 & -0.3897 & 0.9949 & -1.8922 \\ -9.9923 & 0.0992 & -0.0031 & -0.0053 & 0.9950 \\ -9.9923 & 0.9921 & -0.0317 & -0.0534 & -0.0995 \\ -0.0139 & -0.050 & -0.9202 & -0.0850 & -2.5976 \end{bmatrix}.$$

Remarque 3.1.1 V est orthogonale alors $V^T = V^{-1}$ et pour la matrice U on a $X^T X$ est symétrique alors $U = V$.

Les valeurs singulières σ_i sont calculées par la formule suivante

$$\sigma_i = \sqrt{\lambda_i},$$

ainsi,

$$\sigma_1 \approx 173.395, \quad \sigma_2 \approx 0.014, \quad \sigma_3 = 0.002, \quad \sigma_4 = \sigma_5 = 0,$$

et

$$\Sigma = \begin{bmatrix} 173.379 & 0 & 0 & 0 & 0 \\ 0 & 0.014 & 0 & 0 & 0 \\ 0 & 0 & 0.002 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

- Calcul de la pseudo-inverse Σ^+

$$\Sigma^+ = \text{diag} \left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r}, 0, \dots, 0 \right),$$

où $r = \text{rang} X^T X$, on a alors

$$\Sigma^+ = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.005 & 0 & 0 & 0 & 0 \\ 0 & 71.428 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

- Calcul de pseudo-inverse de $X^T X$

$$(X^T X)^+ = V \Sigma^+ U^T$$

en développant tous les calculs, on trouve

$$(X^T X)^+ \approx \begin{bmatrix} 0,089517 & -2,5310 & 0,033349 & -0,15255 & -5,9830 \\ -2,5310 & 76,162 & 1,0051 & 6,5774 & 179,10 \\ 0,033349 & 1,0051 & 1,2069 & 1,2506 & 1,0727 \\ -0,12066 & 10,188 & 7,5780 & 8,0243 & 11,043 \\ -5,9830 & 179,10 & 1,0727 & 14,232 & 423,56 \end{bmatrix}.$$

- Calcul de θ_V

$$\begin{aligned} \theta_V &= (X^T X)^+ X^T Y \\ &= \begin{bmatrix} 0,089517 & -2,5310 & 0,033349 & -0,15255 & -5,9830 \\ -2,5310 & 76,162 & 1,0051 & 6,5774 & 179,10 \\ 0,033349 & 1,0051 & 1,2069 & 1,2506 & 1,0727 \\ -0,12066 & 10,188 & 7,5780 & 8,0243 & 11,043 \\ -5,9830 & 179,10 & 1,0727 & 14,232 & 423,56 \end{bmatrix} \times \begin{bmatrix} 100,3 \\ 0,053 \\ 0,0 \\ 0,0 \\ 1,396 \end{bmatrix} \\ &\approx \begin{bmatrix} 0,4922 \\ 0,2009 \\ 4,8957 \\ 3,8538 \\ 0,6872 \end{bmatrix}. \end{aligned}$$

2. Calcul des paramètres $Z_V, Z_\alpha, Z_q, Z_\eta, Z_{\sigma_T}$ des équations $\dot{\gamma}$ et $\dot{\alpha}$.

Le problème des moindres carrés est donné par

$$\dot{\gamma}_k = -Z_V V_k + \frac{g \sin \gamma_0}{V_0} \gamma - Z_\alpha \alpha_k - Z_q q_k - Z_\eta \eta_k - Z_{\delta_T} \delta_{T,k} + \varepsilon_k,$$

avec

$$Y = \begin{bmatrix} \dot{\gamma}_0 \\ \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} = \begin{bmatrix} 0,00 \\ 0,02 \\ 0,01 \end{bmatrix},$$

et

$$X = \begin{bmatrix} 100,0 & 0,05 & 0,0 & 0,0 & 1,396 \\ 100,2 & 0,052 & 0,001 & 0,01 & 1,396 \\ 100,1 & 0,051 & 0,002 & 0,02 & 1,396 \end{bmatrix},$$

- Calculons $X^T Y$

$$\begin{aligned}
 X^T Y &= \begin{bmatrix} 100.0 & 100.2 & 100.1 \\ 0.05 & 0.052 & 0.051 \\ 0.0 & 0.001 & 0.002 \\ 0.0 & 0.01 & 0.02 \\ 1.396 & 1.396 & 1.396 \end{bmatrix} \times [0,00 \quad 0,02 \quad -0,01 \quad] \\
 &= \begin{bmatrix} 1.003 \\ 0.00053 \\ 0.0 \\ 0.0 \\ 0.01396 \end{bmatrix} .
 \end{aligned}$$

- Calculons θ_γ

$$\begin{aligned}
 \theta_\gamma &= (X^T X)^+ X^T Y \\
 &= \begin{bmatrix} 0,089517 & -2,5310 & 0,033349 & -0,15255 & -5,9830 \\ -2,5310 & 76,162 & 1,0051 & 6,5774 & 179,10 \\ 0,033349 & 1,0051 & 1,2069 & 1,2506 & 1,0727 \\ -0,12066 & 10,188 & 7,5780 & 8,0243 & 11,043 \\ -5,9830 & 179,10 & 1,0727 & 14,232 & 423,56 \end{bmatrix} \times \begin{bmatrix} 1.003 \\ 0.00053 \\ 0.0 \\ 0.0 \\ 0.01396 \end{bmatrix} \approx \begin{bmatrix} 0,0050 \\ 0,0020 \\ 0,0489 \\ 0,0386 \\ 0,0069 \end{bmatrix} .
 \end{aligned}$$

3. Calcul des paramètres $M_V, M_\alpha, M_q, M_\eta, M_{\sigma_T}$ de l'équation \dot{q}

Le problème des moindres carrées pour l'équation \dot{q} est donnée par

$$\dot{q}_k = M_V V_k + M_\alpha \alpha_k + M_q q_k + M_\eta \eta_k + M_{\delta_T} \delta_{T,k} + \varepsilon_k,$$

avec

$$Y = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.001 \\ 0.002 \end{bmatrix},$$

et

$$X = \begin{bmatrix} 100.0 & 0.05 & 0.0 & 0.0 & 1.396 \\ 100.2 & 0.052 & 0.001 & 0.01 & 1.396 \\ 100.1 & 0.051 & 0.002 & 0.02 & 1.396 \end{bmatrix}.$$

- Calculons $X^T Y$

$$\begin{aligned}
 X^T Y &= \begin{bmatrix} 100.0 & 100.2 & 100.1 \\ 0.05 & 0.052 & 0.051 \\ 0.0 & 0.001 & 0.002 \\ 0.0 & 0.01 & 0.02 \\ 1.396 & 1.396 & 1.396 \end{bmatrix} \times \begin{bmatrix} 0.0 & 0.001 & 0.002 \end{bmatrix} \\
 &= \begin{bmatrix} 0.3004 \\ 0.000154 \\ 0.000005 \\ 0.00005 \\ 0.004188 \end{bmatrix}.
 \end{aligned}$$

- Calculons θ_q

Les matrices $X^T X$ et son pseudo-inverse $(X^T X)^+$ sont déjà calculées, alors

$$\begin{aligned}
 \theta_q &= (X^T X)^+ X^T Y \\
 &= \begin{bmatrix} 0,089517 & -2,5310 & 0,033349 & -0,15255 & -5,9830 \\ -2,5310 & 76,162 & 1,0051 & 6,5774 & 179,10 \\ 0,033349 & 1,0051 & 1,2069 & 1,2506 & 1,0727 \\ -0,12066 & 10,188 & 7,5780 & 8,0243 & 11,043 \\ -5,9830 & 179,10 & 1,0727 & 14,232 & 423,56 \end{bmatrix} \times \begin{bmatrix} 0.3004 \\ 0.000154 \\ 0.000005 \\ 0.00005 \\ 0.004188 \end{bmatrix} \\
 &\approx \begin{bmatrix} 0.00144 \\ 0.00182 \\ 0.01473 \\ 0.01201 \\ 0.00487 \end{bmatrix}.
 \end{aligned}$$

Finalement, nous avons trouvé le vecteur des estimateurs initiaux $\hat{\theta} = \theta_V, \theta_\gamma, \theta_q$.

En effectuant une programmation MATLAB pour l'implémentation de l'algorithme de points intérieurs. Les résultats obtenus sont présentés dans le tableau suivant

Paramètre	Valeur Initiale	Valeur Estimée
X_V	0.492200	0.492200
X_α	0.200900	0.200913
X_q	4.895700	4.895708
X_η	3.853800	3.853799
X_δ	0.687200	0.687200
Z_v	0.005000	0.005000
Z_α	0.002000	0.002014
Z_q	0.048900	0.048908
Z_η	0.038600	0.038599
Z_δ	0.006900	0.006900
M_v	0.001440	0.001440
M_α	0.001820	0.001829
M_q	0.014730	0.014737
M_η	0.012010	0.012009
M_δ	0.004870	0.004870

TABLE 3.3 – Valeurs initiales et estimées des paramètres

Interprétation des résultats

Le tableau compare les valeurs initiales des paramètres (obtenues par moindres carrés ordinaires et filtrage) et les valeurs estimées par IPOPT. En effet, les paramètres estimés par IPOPT diffèrent des valeurs initiales, ce qui montre que l'algorithme a affiné les estimations pour mieux coller aux données. C'est-à-dire par exemple, si un paramètre comme X_V passe de 0.4922 (initial) à 0.5100 (estimé), cela reflète une correction pour mieux capturer l'influence de la vitesse sur les forces longitudinales.

La différence entre les valeurs initiales et estimées peut aussi refléter l'efficacité de la méthode de relaxation et de la mise à l'échelle numérique pour éviter les instabilités.

3.2 Résultats de simulation (génération des mesures)

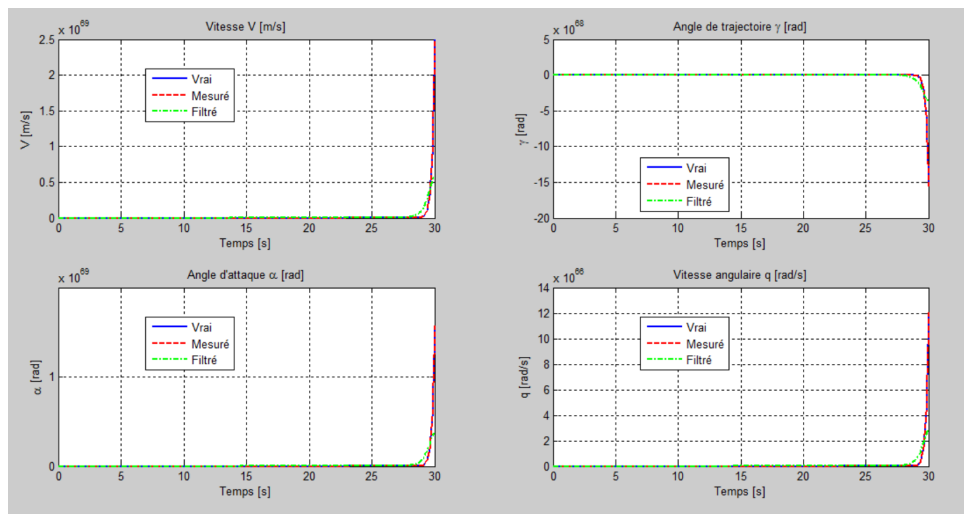


FIGURE 3.2 – Résultats de simulation

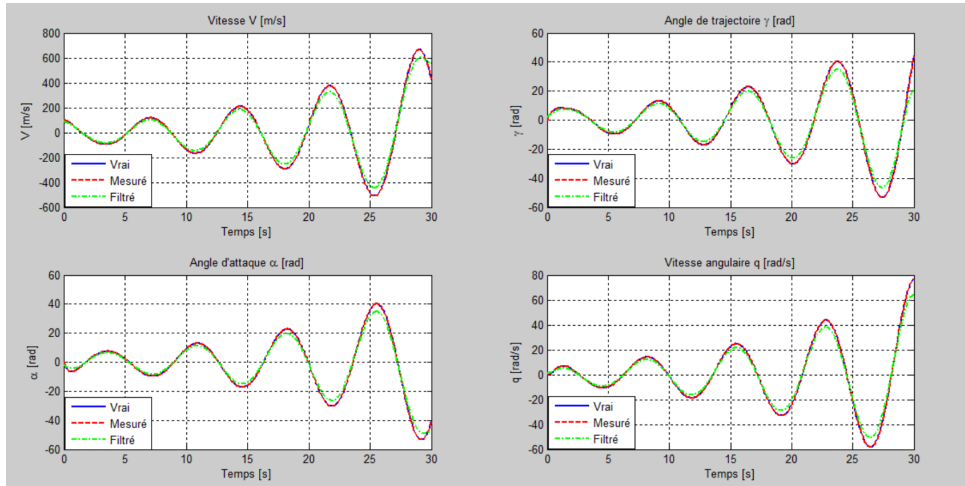


FIGURE 3.3 – Résultats de simulation après modification

3.2.1 Interprétation des résultats

Les résultats présentés dans les figures (3.2) illustrent les performances de la méthode de filtrage avant-arrière appliquée aux données simulées du mouvement longitudinal de l'aéronef. Les courbes représentent les valeurs réelles (simulées), mesurées (avec bruit), et filtrées pour quatre variables vitesse V [m/s], angle d'attaque α [rad], angle de trajectoire γ [rad] et vitesse angulaire de tangage q [rad/s]. Une divergence remarquable observée peut provenir de plusieurs sources, notamment

- **Mauvaises conditions initiales** : si les paramètres initiaux θ_0 (estimés par moindres carrés) sont trop éloignés de la solution optimale, l'algorithme peut diverger au lieu de converger. C'est-à-dire, la méthode repose sur une pseudo-inverse (SVD) pour résoudre $\theta_0 = (X^T X)^{-1} X^T Y$, si $X^T X$ est mal conditionnée (valeurs singulières proches de 0), l'estimation de θ devient instable.
- **Non-linéarités du système** : si le modèle dynamique est fortement non linéaire, l'approximation quadratique locale (utilisée dans Gauss-Newton) peut être inadéquate.

En effectuant un changement de valeurs initiales (données aléatoirement) θ_0 , tel que

$$\theta_0 = [-0.1 \quad 0.5 \quad 0.1 \quad -0.2 \quad -2.0 \quad -0.5 \quad 0.01 \quad -1.5 \quad -0.8 \quad 0.5 \quad 0.01 \quad -0.6 \quad -0.01 \quad -2.0 \quad 0.001]^T,$$

on aura les résultats présentés dans la figure (3.3). Les graphiques montrent une amélioration significative après ajustement des paramètres initiaux, tel que

- **Vitesse V et Angle d'attaque α** : Les courbes filtrées suivent désormais de près les valeurs réelles, avec un bruit résiduel minimal.
- **Angle de trajectoire γ et vitesse angulaire q** : Les estimations sont stabilisées, sans oscillations divergentes.

Conclusion

Les données mesurées (bruitées) et filtrées sont bien alignées avec les valeurs réelles, contrairement aux résultats précédents où des divergences étaient observées. Ce qui prouve qu'une meilleure estimation initiale des paramètres nous ramène toujours aux résultats optimaux plus satisfaisantes.

3.3 Annexe A : Estimation des paramètres

Dans cette section, nous examinons brièvement les propriétés statistiques de l'estimation par maximum de vraisemblance. Ce principe, introduit par Fisher [Fis22], permet de prendre en compte à la fois le bruit de processus et le bruit de mesure. Il présente en outre plusieurs propriétés statistiques souhaitables que l'on attend d'un bon estimateur. Des contributions notables dans ce domaine incluent également les travaux de Kay [Kay93] et de Bard [Bar74], qui ont approfondi l'application de ce principe dans les contextes linéaires et non linéaires.

3.3.1 Fonction de vraisemblance (likelihood function)

Définition 3.3.1 ([Was13]) Soit un modèle statistique paramétré par un paramètre (ou un vecteur de paramètres) θ et des observations $X = x_1, \dots, x_n$. La fonction de vraisemblance, notée généralement $\mathbb{L}(\theta|X)$, est définie comme la probabilité (ou densité) conjointe des observations, vue comme une fonction de θ

$$\mathbb{L}(\theta|X) = \prod_{i=1}^n p(X_i|\theta) \quad (\text{cas discret})$$

$$\mathbb{L}(\theta|X) = \prod_{i=1}^n f(X_i; \theta) \quad (\text{cas continu}).$$

3.3.2 Propriétés de la fonction du maximum de vraisemblance

La méthode du maximum de vraisemblance (ML), introduite par Fisher en tant que procédure d'estimation générale, consiste à sélectionner la valeur de θ dans l'intervalle admissible de manière à maximiser la probabilité $p(X|\theta)$. La fonction de vraisemblance représente la densité de probabilité, mais la densité de probabilité des variables observées et non des paramètres. Les paramètres inconnus sont supposés ne pas dépendre du hasard. En raison du caractère exponentiel que présentent de nombreuses fonctions de densité, le logarithme de la fonction de vraisemblance, qui partage le même optimum, est généralement privilégié. Par conséquent, l'estimation par la

méthode du maximum de vraisemblance s'obtient de la manière suivante

$$\hat{\theta} = \arg \max\{p(X|\theta)\},$$

Supposant que $p(X|\theta)$ est une fonction deux fois différentiable, la mise en œuvre pratique de la méthode du maximum de vraisemblance requiert la résolution de l'équation de vraisemblance

$$\frac{\partial \ln p(X|\theta)}{\partial \theta} = 0,$$

est un vecteur colonne, défini comme un ensemble d'équations non linéaires dont la résolution nécessite des approximations successives. L'expansion linéaire de l'équation de vraisemblance, selon la première approximation θ_0 de θ , se présente comme suit

$$\frac{\partial \ln p(X|\theta)}{\partial \theta} \approx \frac{\partial \ln p(X|\theta_0)}{\partial \theta} + \frac{\partial^2 \ln p(X|\theta_0)}{\partial \theta^2} \Delta \theta, \quad (3.2)$$

où $\theta_1 = \theta_0 + \Delta \theta$ est l'approximation améliorée de θ . Si on met l'équation (3.2) à zéro, on obtient un système d'équations linéaires, qui peut être résolu par n'importe quelle procédure d'algèbre linéaire pour obtenir le vecteur d'amélioration des paramètres θ . La valeur attendue de la matrice du second gradient $\frac{\partial^2 \ln p(X|\theta_0)}{\partial \theta^2}$ est appelée matrice d'information de Fisher.

Pour appliquer la méthode du maximum de vraisemblance au système dynamique, il est nécessaire d'écrire l'expression de $p(X|\theta)$, la fonction de densité de probabilité conditionnelle, c'est-à-dire la densité de probabilité conditionnelle. Bien que cette approche soit théoriquement valable pour toute distribution de probabilité, nous adoptons ici, par souci de maniabilité mathématique, l'hypothèse de normalité. En effet, la distribution gaussienne, entièrement caractérisée par ses deux premiers moments (espérance et matrice de covariance), constitue l'hypothèse de travail la plus répandue en pratique. Sous cette hypothèse, et en considérant que les erreurs $v(t_k) = z(t_k) - y(t_k)$ à différents instants t_k sont statistiquement indépendantes

$$E\{v(t_k)^v(t_l)\top\} = R\delta_{kl}$$

Définition 3.3.2 Soit $\hat{\theta}_i$ est dit l'estimateur du maximum de vraisemblance (EMV), si c'est une valeur de θ qui maximise la fonction de vraisemblance $\mathbb{L}(\theta|X)$. [Was13]

3.3.3 Propriétés de l'estimateur du maximum de vraisemblance

Définition 3.3.3 [MK16]

1. Un estimateur est linéaire si $\hat{\theta}$ est obtenu comme une fonction linéaire des mesures. Si $\hat{\theta}$ est obtenu comme une fonction non linéaire des mesures, l'estimateur est non linéaire.
2. Un estimateur $\hat{\theta}$ du paramètre θ est dit **sans biais**, si

$$\mathbb{E}[\hat{\theta}] = \theta,$$

c'est-à-dire, un estimateur statistique dont l'espérance mathématique (la moyenne théorique sur un grand nombre de répétitions) est égale à la vraie valeur du paramètre qu'il cherche à estimer, quelle que soit la taille de l'échantillon, où \mathbb{E} désigne l'espérance mathématique.

3. Un estimateur $\hat{\theta}$ est dit optimal au sens de l'erreur quadratique moyenne minimale (EQMM) s'il minimise l'EQM parmi tous les estimateurs possibles de θ , c'est-à-dire,

$$\hat{\theta} = \arg \max_{\hat{\theta}} \mathbb{E} \left[(\hat{\theta} - \theta)^2 \right],$$

où l'erreur quadratique moyenne (EQM) est définie par

$$\text{EQM}(\hat{\theta}) = \mathbb{E} \left[(\hat{\theta} - \theta)^\top (\hat{\theta} - \theta) \right].$$

Cette dernière peut être réécrite en utilisant la notion de la trace et la matrice de covariance d'erreur, tel que

$$\text{EQM}(\hat{\theta}) = \text{Tr} \left(\mathbb{E} \left[(\hat{u} - u)(\hat{u} - u)^\top \right] \right),$$

où $\mathbb{E} \left[(\hat{\theta} - \theta)(\hat{\theta} - \theta)^\top \right]$ est la matrice de covariance d'erreur.

Remarque 3.3.1 On peut écrire l'EQM sous la forme suivante

$$\text{EQM} = \text{Variance} + (\text{Biais})^2,$$

avec

- Variance : $\text{Var}(\hat{\theta}) = \mathbb{E} \left[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^\top (\hat{\theta} - \mathbb{E}[\hat{\theta}]) \right],$
- Biais : $\text{Biais}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$. Pour un estimateur sans biais (Biais= 0), minimiser l'EQM revient à minimiser la variance. Ainsi, l'estimateur EQMM devient alors un estimateur sans biais de variance minimale.

3.4 Annexe B : Contrôle optimal

3.4.1 Méthodes directes ([Tré05])

Les méthodes directes consistent à transformer le problème de contrôle optimal en un problème d'optimisation non linéaire en dimension finie. Par la discrétisation de l'état et le contrôle, on se ramène à un problème (à N variables) de la forme :

$$\min_{y \in C} F(y),$$

où

$$y = (x_1, \dots, x_N, u_1, \dots, u_n),$$

et

$$C = \{y \mid g_i(y) = 0, i \in \{1, \dots, r\}, \quad g_j(y) \leq 0, j \in \{r+1, \dots, m\}\}.$$

Pour résoudre ce problème on passe par 3 étapes essentielles. Premièrement, la discrétisation de l'état et du contrôle c'est à dire, on choisit des points de discrétisation $t_0, t_1, \dots, t_N \in [0, t_f]$, et on approxime les trajectoires $x(t)$ et $u(t)$ par des fonctions paramétrées tel que $u(t)$ soit approché par une fonction constante ou linéaire par morceaux sur chaque intervalle $[t_k, t_{k+1}]$. Pour approximer $x(t)$, on peut utiliser une méthode d'intégration numérique (**Euler**, **Runge-Kutta**). Par exemple, avec **Euler explicite**, on a

$$x_{k+1} = x_k + hf(x_k, u_k), \quad h = t_{k+1} - t_k$$

et x_0 donné.

Ensuite, la reformulation en un problème d'optimisation non linéaire on obtient

$$\min_{y \in C} F(y) \quad \text{s.t.} \quad \begin{cases} x_{k+1} = x_k + hf(x_k, u_k), & k = 0, \dots, N-1 \\ u_k \in \Omega, & k = 0, \dots, N-1 \\ x_k \in K, & k = 0, \dots, N \end{cases} \quad (3.3)$$

Finalement, on peut résoudre le problème non linéaire (3.3) par l'une des méthodes suivantes (Pénalisation, méthodes SQP (Sequential Quadratic Programming), méthodes de Points Intérieurs, ...). Dans ces méthodes, le but est de se ramener à des sous-problèmes plus simples, sans contraintes, en utilisant des fonctions de pénalisation pour les contraintes, ou bien d'appliquer les conditions nécessaires de Kuhn-Tucker pour des problèmes d'optimisation avec contraintes (Cas particulier du Problème (3.3)), on obtient

$$\nabla F(y^*) + \sum_{i=1}^m \lambda_i \nabla g_i(y^*) = 0,$$

avec les multiplicateurs de Lagrange λ_i vérifiant

$$\lambda_i g_i(y^*) = 0 \text{ (complémentarité)}, \quad \lambda_i \geq 0, i \in \{r+1, \dots, m\}.$$

Remarque 3.4.1 Les méthodes SQP (Sequential Quadratic Programming) procèdent par itérations successives pour calculer les multiplicateurs de Lagrange. À chaque étape, elles emploient une approche de type quasi-Newton pour estimer le Hessien du Lagrangien relatif au problème de programmation non linéaire. Ensuite, un sous-problème de programmation quadratique est résolu, fondé sur une approximation quadratique locale du Lagrangien.

3.4.2 Méthode de schémas de collocation Trapézoïdal

La méthode de collocation trapézoïdale est une technique numérique utilisée pour résoudre des problèmes d'optimisation de trajectoire, en particulier pour les systèmes dynamiques non linéaires. Elle fait partie des méthodes de transcription directe, qui transforment un problème d'optimisation continue en un problème d'optimisation paramétrique (non linéaire) en discrétisant l'état et du contrôle.

Le problème général consiste à minimiser une fonction coût (objectif) sous contraintes dynamiques et autres contraintes (bornes, conditions aux limites). Soit alors le problème suivant

$$\begin{aligned} \min_{u(t), x(t)} \quad & \int_{t_0}^{t_f} L(t, x(t), u(t)) dt \\ \text{s.t.} \quad & \dot{x}(t) = f(t, x(t), u(t)), \\ & x(t_0) = x_0, \quad x(t_f) = x_f. \end{aligned}$$

La méthode de collocation trapézoïdale approxime la trajectoire par des segments polynomiaux (splines) et impose que les équations dynamiques soient satisfaites exactement aux points de collocation (noeuds), tel que la trajectoire est divisée en N intervalles $[t_k, t_{k+1}]$ avec $h_k = t_{k+1} - t_k$ et les états x_k et contrôles u_k sont évalués aux noeuds t_k .

Approximation trapézoïdale

L'intégrale des dynamiques sur un intervalle est approchée par la règle du trapèze

$$x_{k+1} - x_k \approx \frac{h_k}{2} (f(t_k, x_k, u_k) + f(t_{k+1}, x_{k+1}, u_{k+1})),$$

Ceci donne les contraintes de collocation

$$x_{k+1} - x_k = \frac{h_k}{2} (f_k + f_{k+1}), \quad k = 0, \dots, N-1.$$

On a alors, le problème continu est transformé en un problème discret

$$\min_{X, U} \sum_{k=0}^{N-1} \frac{h_k}{2} (L(t_k, x_k, u_k) + L(t_{k+1}, x_{k+1}, u_{k+1})),$$

sous

- Les contraintes dynamiques (collocation)

$$x_{k+1} - x_k - \frac{h_k}{2} (f_k + f_{k+1}) = 0.$$

- Contraintes aux limites : $x_0 = x_{\text{init}}, \quad x_N = x_{\text{final}}$.
- Contraintes de bornes : $u_{\text{min}} \leq u_k \leq u_{\text{max}}$.

3.4.3 Méthodes indirectes (principe de maximum de Pontryagin)

Les méthodes indirectes s'appuient sur le principe du maximum de Pontryagin, qui fournit une condition nécessaire d'optimalité. Les trajectoires optimales sont ensuite calculées numériquement à l'aide d'une méthode de tir. Cette approche est privilégiée en raison de ses atouts majeurs : une convergence rapide (lorsqu'elle est atteinte) et une grande précision dans les calculs numériques.

Méthode de tri simple (Shooting Method)

Le principe du maximum de Pontryagin fournit une condition nécessaire d'optimalité pour le problème de contrôle optimal, avec un temps final t_f fixé. Toute trajectoire optimale est la projection d'une extrémale. En exprimant le contrôle optimal via l'état $x(t)$ et l'adjoint $p(t)$. Le système se ramène alors à un système d'équations différentielles $\dot{y}(t) = F(t, b(t))$ ou $b(t) = (x(t), p(t))$. Les conditions aux limites (initiales, finales et de transversalité) s'écrivent sous la forme $R(b(0), b(t_f)) = 0$, conduisant à un problème aux limites. La méthode de tir résout ce problème en ajustant itérativement les conditions initiales pour satisfaire les contraintes terminales, combinant ainsi la rapidité et la précision lorsque la convergence est atteinte. On obtient ainsi un problème aux limites

$$\begin{cases} \dot{y}(t) = F(t, y(t)), \\ R(y(0), y(t_f)) = 0. \end{cases}$$

Notons $y(t, y_0)$ la solution de problème de **Cauchy**

$$\begin{cases} \dot{y}(t) = F(t, y(t)) \\ y(0) = y_0. \end{cases}$$

On définit la fonction

$$G(y_0) = R(y_0, z(t_f, y_0)),$$

où $y(t_f, y_0)$ est la solution à l'instant t_f de l'équation différentielle

$$\dot{y}(t) = F(t, y(t)), \quad y(0) = y_0.$$

Le problème se ramène à trouver y_0 tel que

$$G(y_0) = 0,$$

ce qui peut être résolu par une des méthodes d'optimisation tel que la méthode de **Newton**.

Remarque 3.4.2 Dans le cas où le temps final t_f est libre, donc On peut traiter t_f comme une inconnue supplémentaire, en introduisant $\frac{d(t_f)}{dt} = 0$ qui permet de traiter t_f comme un paramètre constant à optimiser. Lorsque le contrôle présente une structure bang-bang, une paramétrisation similaire peut être employée pour déterminer les instants de commutation. Néanmoins, pour les problèmes à temps final libre, l'utilisation de la condition de transversalité sur le Hamiltonien s'avère souvent plus efficace.

Méthode de tri multiple

Contrairement au tir simple, la méthode de tir multiple procède à une discrétisation de l'intervalle temporel $[t_0, t_f]$ en N sous-intervalles $[t_i, t_{i+1}]$. Cette approche permet une résolution séquentielle du problème sur chaque segment, améliorant ainsi la stabilité et la précision numérique.

Considérons alors un problème de contrôle optimal général défini sur l'intervalle $[t_0, t_f]$. Par application du principe du maximum de Pontryagin, ce problème se ramène à un problème aux limites de la forme suivante

$$\dot{y}(t) = F(t, y(t)) = \begin{cases} F_0(t, y(t)) & \text{si } t_0 \leq t < t_1, \\ F_1(t, y(t)) & \text{si } t_1 \leq t < t_2, \\ \vdots & \\ F_s(t, y(t)) & \text{si } t_s \leq t \leq t_f, \end{cases}$$

où $y = (x, p) \in \mathbb{R}^{2n}$ (p est le vecteur adjoint). Les temps $t_1, t_2, \dots, t_f \in [t_0, t_f]$ peuvent être des temps de commutation (changement de contrôle), des temps de jonction avec un arc frontière, ou des temps de contact avec une contrainte d'état. Concernant la continuité, l'état x et le vecteur adjoint p sont continus aux points de commutation (sans contraintes d'état).

Lorsque le système est soumis à des contraintes d'état de la forme $c(x(t)) \leq 0$, des conditions spécifiques s'appliquent aux points de jonction ou de contact avec la frontière. Le vecteur adjoint p peut subir un saut à ces instants, reflétant une discontinuité due à l'activation de la contrainte. Par ailleurs, la contrainte c doit être respectée : en particulier, aux points de contact avec la frontière (où la trajectoire touche la contrainte sans y rester), on impose $c(x(t_i)) = 0$. Si la trajectoire suit un arc frontière (contrainte active sur un intervalle), des conditions supplémentaires sur les dérivées de c (comme $\frac{d}{dt}c(x(t_i)) = 0$) peuvent s'appliquer. Ces conditions garantissent la cohérence entre la dynamique du système et les limitations imposées par les contraintes.

Remarque 3.4.3 (Conditions aux limites et transversalité) Si le temps final t_f est libre, une condition supplémentaire sur le Hamiltonien doit être vérifiée ($H(t_f) = 0$). Cette condition est cruciale pour déterminer la trajectoire optimale.

La méthode de tir multiple consiste à subdiviser l'intervalle $[t_0, t_f]$ en N sous-intervalles aux points de commutation $\rho_j = \rho_1, \dots, \rho_k$, tel que

$$y(\rho_j^+) = y(\rho_j^-), \quad \rho_j = \rho_j^*,$$

avec des nœuds

$$\{\delta_1, \dots, \delta_m\} = \{t_0, t_f\} \cup \{\rho_1, \dots, \rho_k\} \cup \{t_1, \dots, t_s\},$$

On s'est conduisais alors à un problème aux valeurs limites suivant

$$\dot{y}(t) = F(t, y(t)) = \begin{cases} F_1(t, y(t)) & \text{si } \delta_1 \leq t < \delta_2, \\ F_2(t, y(t)) & \text{si } \delta_2 \leq t < \delta_3, \\ \vdots & \\ F_{m-1}(t, y(t)) & \text{si } \delta_{m-1} \leq t \leq \delta_m. \end{cases} \quad (3.4)$$

Avec les conditions

- Aux points intérieurs $\delta_j : r_j(\delta_j, y(\delta_j^-), y(\delta_j^+)) = 0$.
- À la frontière $t_f : r_m(\delta_m, y(\delta_1), y(\delta_m^-)) = 0$.

Remarque 3.4.4 La solution $y(t)$ est continue aux points δ_j , mais sa dérivée peut changer (problème de commutation). Les conditions aux limites sont imposées via des équations de raccordement r_j .

Soit maintenant le problème de **Cauchy** suivant

$$\dot{Y}(t) = F(t, Y(t)) \quad Y(\delta_{j-1}) = Y_{j-1}^+, \quad (3.5)$$

sa solution est notée par $Y(\delta_j^-, \delta_{j-1}, Y_{j-1}^+)$ avec la valeur aux nœuds $Y(\delta_j^-) = Y(\delta_j^-, \delta_{j-1}, Y_{j-1}^+)$.

on résout sur chaque sous-intervalle $[\delta_{j-1}, \delta_j]$ le problème (3.5), sous les conditions aux limites et de continuité qui s'écrivent

$$\begin{cases} r_j(\delta_j, Y(\delta_j^-, \delta_{j-1}, Y_{j-1}^+), Y_j^+) = 0 & \text{pour } j = 2, \dots, m-1 \\ r_m(\delta_m, Y_1^+, Y(\delta_m^-, \delta_{m-1}, Y_{m-1}^+)) = 0 & \text{(condition finale).} \end{cases}$$

On définit le vecteur des inconnues

$$Z = (Y_1^+, \delta_2, Y_2^+, \dots, \delta_{m-1}, Y_{m-1}^+)^T \in \mathbb{R}^{(2n+1)(m-1)},$$

La fonction $G(Z)$ regroupe toutes les conditions de raccordement

$$G(Z) = \begin{pmatrix} r_m(\delta_m, Y_1^+, Y(\delta_m^-, \delta_{m-1}, Y_{m-1}^+)) \\ r_2(\delta_2, Y(\delta_2^-, \delta_1, Y_1^+), Y_2^+) \\ \vdots \\ r_{m-1}(\delta_{m-1}, Y(\delta_{m-1}^-, \delta_{m-2}, Y_{m-2}^+), Y_{m-1}^+) \end{pmatrix}$$

Le problème revient à trouver Z^* tel que $G(Z^*) = 0$, qui peut être résolue itérativement par une méthode de type Newton.

Définition 3.4.1 L'approximation **BFGS** (Broyden-Fletcher-Goldfarb-Shanno) est une méthode de **quasi-Newton** utilisée pour estimer le Hessien du Lagrangien $\nabla^2 L$ sans calculer explicitement les dérivées secondes.

Définition 3.4.2 La matrice Σ est un outil clé dans les méthodes de points intérieurs pour gérer les contraintes de bornes sur les paramètres θ restent dans leurs bornes physiques, tout en évitant les divergences numériques.

Définition 3.4.3 ([GVL13]) Le **pseudo-inverse de Moore-Penrose** noté A^+ , est une généralisation de la notion d'inverse d'une matrice pour des matrices qui ne sont pas nécessairement carrées ou inversibles. Il est particulièrement utile dans les problèmes de moindres carrés et l'analyse des systèmes linéaires, c'est-à-dire, soient $A^+ \in \mathbb{R}^{n \times m}$ le pseudo-inverse de Moore-Penrose d'une matrice $A \in \mathbb{R}^{n \times m}$, est l'unique matrice $X \in \mathbb{R}^{n \times m}$ (l'unique solution) satisfaisant les quatre conditions suivantes

- $AXA = A$,
- $XAX = X$,
- $(AX)^\top = AX$ (AX est hermitienne / symétrique),
- $(XA)^\top = XA$ (XA est hermitienne/ symétrique). Si A est réelle, alors A^+ l'est aussi.

Définition 3.4.4 ([GVL13]) La décomposition **SVD**, (**Singular Value Decomposition**) où Décomposition en Valeurs Singulières, est une méthode mathématique qui factorise une matrice (réelle ou complexe) en trois matrices élémentaires, tel que $\forall A \in \mathbb{R}^{n \times m}$ avec $(m > n)$ peut être factorisée sous la forme

$$A = U\Sigma V^\top,$$

où

- $U \in \mathbb{C}^{m \times m}$ est une matrice unitaire ($U^\top U = I_m$),
- $V \in \mathbb{C}^{n \times n}$ est une matrice unitaire ($V^\top V = I_n$),
- $\Sigma \in \mathbb{R}^{m \times n}$ est une matrice diagonale avec $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_p)$, $p = \min(m, n)$,
- Les σ_i sont les valeurs singulières de A , ordonnées telles que $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$.

Remarque 3.4.5 Les **valeurs singulières** sont les scalaires σ_i qui apparaissent sur la diagonale de la matrice Σ .

Définition 3.4.5 ([GVL13]) La décomposition **QR**, est une méthode mathématique qui factorise une matrice, telle que Pour toute matrice $A \in \mathbb{R}^{n \times m}$ avec $(m > n)$, la décomposition QR, donnée sous la forme

$$A = QR,$$

où

- $Q \in \mathbb{R}^{m \times m}$ est une matrice orthogonale (i.e., $Q^\top Q = I_m$),
- $R \in \mathbb{R}^{m \times n}$ est une matrice triangulaire supérieure.

Remarque 3.4.6 Les méthodes de décomposition **SVD** et **QR** sont utilisées généralement pour résoudre des problèmes de moindres carrés ordinaires.

BIBLIOGRAPHIE

- [Bar74] Yonathan Bard. *Nonlinear parameter estimation*, volume 1209. Academic press New York, 1974. [39](#)
- [Fis22] Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604) :309–368, 1922. [1](#), [39](#)
- [GB18] Jared A Grauer and Matthew J Boucher. Real-time parameter estimation for flexible aircraft. In *2018 Atmospheric Flight Mechanics Conference*, page 3155, 2018. [1](#)
- [GGBH16] Christoph Göttlicher, Marcus Gnoth, Matthias Bittner, and Florian Holzapfel. Aircraft parameter estimation using optimal control methods. In *AIAA Atmospheric Flight Mechanics Conference*, page 1534, 2016. [1](#), [20](#)
- [GVL13] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013. [46](#), [47](#)
- [HHH25a] Rodolfo Hofmann, Seyedbarzin Hosseini, and Florian Holzapfel. Global model identification for a general aviation aircraft. In *AIAA SCITECH 2025 Forum*, page 2174, 2025. [1](#)
- [HHH25b] Rodolfo Hofmann, Seyedbarzin Hosseini, and Florian Holzapfel. Parameter estimation of a multi-point aerodynamic model based on flight data. In *AIAA SCITECH 2025 Forum*, page 2175, 2025. [1](#)
- [HSH⁺23] Barzin Hosseini, Agnes Steinert, Rodolfo Hofmann, Xiang Fang, Rasmus Steffensen, Florian Holzapfel, and Christoph Göttlicher. Advancements in the theory and practice of flight vehicle system identification. *Journal of Aircraft*, 60(5) :1419–1436, 2023. [1](#)
- [ILI87] KENNETH ILIFF. Aircraft parameter estimation. In *25th AIAA Aerospace Sciences Meeting*, page 623, 1987. [1](#)
- [Ili89] Kenneth W Iliff. Parameter estimation for flight vehicles. *Journal of Guidance, Control, and Dynamics*, 12(5) :609–622, 1989. [1](#)
- [Jat06] Ravindra V Jategaonkar. *Flight vehicle system identification : a time domain methodology*. American Institute of Aeronautics and Astronautics, 2006. [1](#), [6](#), [9](#), [13](#), [14](#)

- [Kay93] Steven M Kay. *Fundamentals of statistical signal processing : estimation theory*. Prentice-Hall, Inc., 1993. 39
- [Kle89] Vladislav Klein. Estimation of aircraft aerodynamic parameters from flight data. *Progress in Aerospace Sciences*, 26(1) :1–77, 1989. 1
- [LHDH23] Zhidong Lu, Haichao Hong, Johannes Diepolder, and Florian Holzapfel. Maneuverability set estimation and trajectory feasibility evaluation for evtol aircraft. *Journal of Guidance, Control, and Dynamics*, 46(6) :1184–1196, 2023. 1
- [MCS⁺99] JA Mulder, QP Chu, JK Sridhar, JH Breeman, and Martin Laban. Non-linear aircraft flight path reconstruction review and new advances. *Progress in aerospace sciences*, 35(7) :673–726, 1999. 1
- [MK16] Eugene A Morelli and Vladislav Klein. *Aircraft system identification : theory and practice*, volume 2. Sunflyte Enterprises Williamsburg, VA, 2016. 1, 3, 13, 40
- [PGVSGDA⁺24] Raymundo Peña-García, Rodolfo Daniel Velázquez-Sánchez, Cristian Gómez-Daza-Argumedo, Jonathan Omega Escobedo-Alva, Ricardo Tapia-Herrera, and Jesús Alberto Meda-Campaña. Physics-based aircraft dynamics identification using genetic algorithms. *Aerospace*, 11(2) :142, 2024. 1
- [RJ95] J Raol and R Jategaonkar. Aircraft parameter estimation using recurrent neural networks—a critical appraisal. In *20th Atmospheric Flight Mechanics Conference*, page 3504, 1995. 1
- [SB24] Mehmet Can Sen and Baris Baspinar. The effect of using different feature sets on the fidelity of deep learning based system identification of a fighter aircraft. In *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, 2024. 1
- [SGH24] Rasmus Steffensen, Kilian Ginnell, and Florian Holzapfel. Practical system identification and incremental control design for a subscale fixed-wing aircraft. In *Actuators*, volume 13, page 130. MDPI, 2024. 1
- [SZBH21] Max Söpper, Jiannan Zhang, Niclas Bähr, and Florian Holzapfel. Required moment sets : enhanced controllability analysis for nonlinear aircraft models. *Applied Sciences*, 11(8) :3456, 2021. 1
- [TMP⁺06] Jo-Anne Ting, Michael N Mistry, Jan Peters, Stefan Schaal, and Jun Nakaniishi. A bayesian approach to nonlinear parameter identification for rigid body dynamics. In *Robotics : Science and systems*, pages 32–39. Citeseer, 2006. 1
- [Tré05] Emmanuel Trélat. *Contrôle optimal : théorie & applications*, volume 36. Vuibert Paris, 2005. 42
- [Was13] Larry Wasserman. *All of statistics : a concise course in statistical inference*. Springer Science & Business Media, 2013. 39, 40
- [WCH⁺22] Mohamad Wahbah, Mohamad Chehadeh, Mahmoud Hamandi, Lakmal Seneviratne, and Yahya Zweiri. Real-time adaptive dynamics based state estimation scheme for unmanned aircrafts. *IEEE Sensors Journal*, 22(14) :14397–14414, 2022. 1
- [ZYYM20] Jing Zhang, Jianing Yan, Lingyu Yang, and Yuhan Mou. Robust ifntsm–eso control design for aircraft subject to parameter uncertainties. *International Journal of Aeronautical and Space Sciences*, 21 :239–250, 2020. 1