



HAL
open science

Méthodes indirectes d'adaptation et de décision pour la sécurisation du vol des drones à volure fixe

Adèle Boche

► **To cite this version:**

Adèle Boche. Méthodes indirectes d'adaptation et de décision pour la sécurisation du vol des drones à volure fixe. Automatique / Robotique. Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), 2018. Français. NNT: . tel-02274972

HAL Id: tel-02274972

<https://hal.science/tel-02274972>

Submitted on 30 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)*

Présentée et soutenue le 18 décembre 2018 par :

ADÈLE BOCHE

Méthodes indirectes d'adaptation et de décision pour la sécurisation du vol des drones à voilure fixe

JURY

ISABELLE FANTONI	Directeur de Recherche	Rapporteur
DIDIER THEILLIOL	Professeur des Universités	Rapporteur
CHARLES	Ingénieur de Recherche	Examineur
POUSSOT-VASSAL	Ingénieur de Recherche	Directeur de thèse
HENRY DE PLINVAL	Ingénieur de Recherche	Directeur de thèse
JEAN-LOUP FARGES	Professeur des Universités	Président
VINCENT COCQUEMPOT		

École doctorale et spécialité :

EDSYS : Automatique 4200046

Unité de Recherche :

ONERA - The French Aerospace Lab - Département de Traitement de l'Information et Systèmes

Directeur(s) de Thèse :

JEAN-LOUP FARGES et HENRY DE PLINVAL

Rapporteurs :

ISABELLE FANTONI et DIDIER THEILLIOL

Remerciements

En premier lieu, je souhaite remercier mes encadrants, Henry et Jean-Loup, d'avoir cru en moi et de m'avoir donné l'opportunité de faire cette thèse. Je les remercie pour leur soutien et leur implication tout au long de ce doctorat. Cette thèse est aussi le fruit de leurs connaissances et expériences. Jean-Loup, j'ai beaucoup appris de ta rigueur scientifique ainsi que de tes grandes connaissances. Merci aussi pour tes relectures très sérieuses, que ce soit pour les articles ou pour le manuscrit. Henry, je te remercie d'avoir réussi à te rendre disponible, malgré ton emploi du temps déjà bien chargé, lorsque j'en avais besoin. Merci aussi pour ta bonne humeur permanente. Je tiens également à remercier Isabelle Fantoni et Didier Theilliol pour avoir accepté d'être les rapporteurs de ma thèse et pour leurs retours pertinents sur le manuscrit, Vincent Cocquempot pour avoir présidé le jury et Charles pour avoir accepté de faire partie de mon jury. J'ai été honoré de présenter mon travail devant ces membres du jury, qui ont su exprimer leur intérêt, tout en introduisant des remarques et conseils avisés.

Ces trois années ont aussi été marquées par la présence de mes collègues du DTIS, qu'ils soient doctorants, post-doctorants ou permanents, dont l'importance a été indéniable durant cette thèse. Merci à tous ceux qui m'ont accueilli et aidé, merci à tous ceux qui ont apporté la bonne ambiance et merci à tous ceux qui m'ont accompagné lors d'after-works. Tout d'abord je vais remercier ceux avec qui j'ai partagé le bureau et qui ont dû me supporter. Pauline, c'était un plaisir de parler de tout sauf du boulot et de partager notre passion pour les ateliers créatifs et pour nos chats. Mathieu, un grand merci pour ton aide Latex et pour avoir si bien su ordonner mon bureau. Guillaume et Matteo, merci pour les pauses Pokémon Go. Gustav, un grand merci pour ton aide Simulink et pour ta lenteur en tout point. Manu, merci pour ton optimisme. Même s'ils n'ont pas partagé mon bureau, je remercie aussi les autres doctorants : Seb, Valentin, Ejagen et Oktay, merci pour les pauses café, Vincent, merci pour ces matchs de volley endiablés et Mehdi, merci d'être toi tout simplement ... Bonne chance à tous pour la fin de la thèse. Je n'oublie pas ceux qui sont déjà docteur et parti de l'Onera : Jérémy, Hélène, Alvaro et Fabrice. Je souhaite aussi remercier Cédric pour tes connaissances sur le filtre de Kalman mais surtout pour les moments potins, Charles pour cette soirée à l'esprit chupitos, Mario je ne te remercie pas de n'avoir jamais suivi mes si bons conseils. Merci au reste de l'équipe pour ces trois années. Ça a été un plaisir de travailler parmi vous.

Enfin et plus personnellement, je tiens à remercier mes chatons, Lucie, Manon, Charlotte, Justine, Jeanne, Hannah et Alix, les copines de l'ENSICA, Mathilde, Lisa, Marie, Claire, Sara et Clara mais aussi tous les potes de l'ENSICA, notamment Fred, Landiou et Hoa qui ont assisté à ma soutenance, et mes anciens colocataires Eneko, Pikey, Paul qui m'ont supporté au début de ma thèse. Merci à tous les autres pour vos encouragements, notamment Erlé et Emilie. Finalement, j'ai une pensée pour ma famille qui m'a soutenue tout au long de mes études et sans qui je n'aurais pas pu arriver jusqu'ici aussi facilement, un grand merci à Kévin et Bifidus d'être tout simplement et de me supporter au quotidien.

Table des matières

Notations	ix
Acronymes	xiii
1 Introduction générale	1
1.1 Contexte	1
1.1.1 Les drones	1
1.1.2 La sécurité des drones	3
1.1.3 Les méthodes existantes	4
1.2 Objectif et démarche de la thèse	4
1.3 Organisation de la thèse	6
2 Un problème de reconfiguration d'un mini-drone en panne lors de l'atterrissage	9
2.1 Les aspects spécifiques de la sécurité du vol des drones	11
2.1.1 L'expérience des aéronefs habités et l'adaptation nécessaire des exigences associées	11
2.1.2 Un impact important d'une panne sur la sécurité du contrôle de la trajectoire en vol	12
2.1.3 La variété des types de fautes possibles	14
2.1.4 Une opportunité pour la redondance analytique	19
2.2 Le drone Altimum 4	19
2.2.1 Actionneurs	20
2.2.2 Capteurs	21
2.2.3 Modèle de simulation	25
2.3 Scénarios de panne pour le drone Altimum 4	27
2.3.1 La phase d'atterrissage	27
2.3.2 Les types de fautes envisagés	28
2.4 Conclusion	28
3 Etat de l'art	31
3.1 La commande tolérante aux fautes	33
3.1.1 Les méthodes passives	33
3.1.2 Les méthodes actives	35
3.2 Modélisation des systèmes et de leurs fautes	38
3.2.1 Systèmes à état continu	38
3.2.2 Systèmes à état discret	40
3.2.3 Systèmes hybrides	41
3.3 Détection et diagnostic de faute	43
3.3.1 Classification des méthodes	44

3.3.2	Génération de résidus	44
3.3.3	Des variantes du filtre de Kalman	48
3.3.4	Estimation d'état sous contraintes	54
3.3.5	Signaux facilitant l'estimation des paramètres	55
3.3.6	Tests statistiques	55
3.3.7	Méthodes discrètes et hybrides	57
3.4	Les méthodes de reconfiguration pour la FTC indirecte	60
3.4.1	Les méthodes de restructuration	61
3.4.2	Les méthodes de commutation	65
3.5	Conclusion	66
4	Définition d'un cadre et de modèles génériques pour la commande tolérante aux fautes	67
4.1	Structure générale de la méthode proposée combinant les méthodes indirectes de commutation et d'adaptation	69
4.2	Modélisation à états continus	70
4.2.1	Représentation d'état du modèle dynamique sans faute	71
4.2.2	Modélisation des fautes actionneurs	72
4.2.3	Modélisation des fautes capteurs	74
4.2.4	Observabilité théorique du système augmenté avec les paramètres de fautes	75
4.2.5	Représentation d'état du modèle dynamique avec fautes	78
4.2.6	Application au drone Altium 4	78
4.3	Modélisation à états discrets	81
4.3.1	États, Actions et Récompenses	81
4.3.2	Probabilités de transition	82
4.3.3	Observations et probabilités d'observation	83
4.3.4	Application au drone Altium 4	85
4.4	Conclusion	86
5	Développement d'une méthode active de détection et de diagnostic des fautes	87
5.1	Développement d'un filtre de Kalman adaptatif avec réouverture de la covariance	89
5.1.1	La méthode d'estimation retenue : le filtre de Kalman adaptatif	89
5.1.2	Utilisation de signatures de pannes pour isoler la faute	90
5.1.3	Adaptation de la covariance du filtre	93
5.1.4	Application de contraintes	95
5.1.5	Réglage du filtre de Kalman	96
5.2	Développement d'une méthode active	98
5.2.1	Excitation des commandes redondantes pour l'observabilité	98
5.2.2	Excitation des commandes pour l'Altium 4	99
5.3	Utilisation des données de l'estimation pour la détection et le diagnostic des fautes	100

5.4	Conclusion	101
6	Validation de la méthode active de détection et de diagnostic des fautes	103
6.1	Démonstration expérimentale de la pertinence du modèle de fautes	105
6.1.1	Identifiabilité des paramètres b_0 des actionneurs redondants	105
6.1.2	Identifiabilité des paramètres b_1 des actionneurs redondants	105
6.2	Critères d'évaluation	106
6.2.1	Critère d'évaluation quantitative pour l'estimation	106
6.2.2	Critère d'évaluation quantitative pour la détection et l'identification	109
6.3	Validation sur mesures générées à partir du modèle linéaire	109
6.3.1	Cas où l'on considère uniquement les pannes actionneurs	110
6.3.2	Cas où l'on considère uniquement les pannes capteurs	125
6.3.3	Cas où l'on considère les pannes actionneurs et capteurs	134
6.4	Validation sur mesures générées à partir du modèle non linéaire	137
6.4.1	Cas où l'on considère uniquement les pannes actionneurs	137
6.4.2	Cas où l'on considère uniquement les pannes capteurs	143
6.5	Conclusion	143
7	Amélioration de la performance par reconfiguration du système de commande	145
7.1	Principe du système de reconfiguration	147
7.2	Définition de l'indicateur de performance	147
7.3	Conception des contrôleurs	148
7.3.1	Conception du contrôleur nominal	148
7.3.2	Conception d'un contrôleur adaptatif pour les fautes d'actionneurs redondants	149
7.3.3	Conception d'un contrôleur adaptatif pour les fautes d'actionneurs non redondants	152
7.4	Utilisation de méthodes d'optimisation pour sélectionner le contrôleur	153
7.4.1	Hypothèses du processus de prise de décision	153
7.4.2	Utilisation du formalisme POMDP	155
7.4.3	Utilisation du formalisme MPC	160
7.4.4	Utilisation du formalisme SMPC	160
8	Évaluation de la capacité de la méthode à suivre une trajectoire d'atterrissage en cas de pannes	163
8.1	Critère d'évaluation	165
8.1.1	Critère d'évaluation quantitative pour la reconfiguration	165
8.1.2	Critère d'évaluation quantitative pour le suivi de trajectoire	165
8.2	Étude comparative entre la méthode FTC développée et une loi de commande classique	165
8.2.1	Comportement en mode nominal	166

8.2.2	Comportement en cas de pannes actionneurs	167
8.2.3	Comportement en cas de pannes capteurs	171
8.3	Impact d'un système de commande basé sur une méthode linéaire	176
8.3.1	Comportement en cas de pannes élevons	177
8.3.2	Comportement en cas de pannes moteur	177
8.4	Impact d'une mauvaise estimation sur le suivi de trajectoire	178
8.5	Conclusion	182
9	Conclusion	185
9.1	Conclusions	185
9.2	Perspectives	186
9.2.1	Amélioration du modèle	186
9.2.2	Amélioration de la méthode	186
A	Modèle dynamique de l'Altium 4	189
A.1	Équations générales	189
A.1.1	Équations dynamiques et cinématiques	189
A.1.2	Moments et Forces	190
A.1.3	Coefficients aérodynamiques	191
A.1.4	Élevons	191
A.1.5	Moteur et hélice	191
A.2	Modèle complet	191
A.3	Modèle longitudinal	192
B	Représentations d'état	195
B.1	Modèles	196
B.1.1	Représentation d'état des systèmes non linéaires	196
B.1.2	Représentation d'état des systèmes linéaires	196
B.2	Stabilité, observabilité et contrôlabilité d'un système	197
B.2.1	Stabilité	197
B.2.2	Définitions de la contrôlabilité et de l'observabilité	197
B.2.3	Contrôlabilité et observabilité des systèmes discrets linéaire à paramètres invariants	198
B.2.4	Contrôlabilité et observabilité des systèmes discrets linéaire à paramètres variants	198
B.2.5	Contrôlabilité et observabilité des systèmes discrets non-linéaires	199
C	Systèmes à événements discrets	201
C.1	Les automates	201
C.2	Les modèles de Markov	201
C.3	Les processus de décision Markovien	202
C.3.1	Processus MDP	202
C.3.2	Processus POMDP	203

D	Filtre de Kalman	205
D.1	Le filtre de Kalman pour les systèmes linéaires	205
D.1.1	L'approche classique	205
D.1.2	Une approche séquentielle pour le traitement des observations	206
D.2	Le filtre de Kalman pour les systèmes non-linéaires	207
D.2.1	Le filtre de Kalman étendu	207
D.2.2	Le filtre de Kalman non parfumé	208
E	Loi normale et tests associés	209
E.1	Loi normale	209
E.2	Loi normale multivariée	209
E.3	Le test de Student et le test du χ^2	210
F	Méthode LQR	211
F.1	Théorème LQR	211
F.2	Théorème LQR réduit	212
F.3	Théorème LQR réduit avec terme additionnel	213
G	Exemple de POMDP à deux dimensions	217
G.1	Paramètre du problème	217
G.2	Cas avec l'espace de recherche simplifié	218
G.3	Cas avec l'espace de recherche complet	219
H	Comparaison entre le filtre TSKF avec et sans contrainte	223
H.1	Pannes moteur	223
H.2	Pannes élevons	228
	Publications	231
	Bibliographie	233

Notations

Symboles

A	matrices d'état d'une représentation d'état
\vec{a}	vecteur accélération
a_x	accélération longitudinale dans le repère avion
a_y	accélération latérale dans le repère avion
a_z	accélération verticale dans le repère avion
B	matrice de commande d'une représentation d'état
\mathbf{b}	vecteur des paramètres de fautes
\mathbf{b}_0	vecteur des biais des fautes actionneurs
\mathbf{b}_1	vecteur d'efficacité des fautes actionneurs
C	matrice de sortie d'une représentation d'état
\mathbf{c}_0	vecteur des biais des fautes capteurs
\mathbf{c}_1	vecteur d'efficacité des fautes capteurs
E	état du moteur
g	constante de gravitation
H	horizon de commande prédictive
K	gain du contrôleur par retour d'état
LA	état de l'élevon gauche
RA	état de l'élevon droit
m	nombre de sorties du système
m_a	masse de l'avion
n	nombre d'états du système
\vec{n}	vecteur facteur de charge
n_b	nombre de paramètres de fautes du système
$n_{\text{contrôleur}}$	nombre de contrôleurs
n_{mode}	nombre de modes
n_x	facteur de charge longitudinale dans le repère avion
n_y	facteur de charge latérale dans le repère avion
n_z	facteur de charge verticale dans le repère avion
l	nombre de commandes du système
p	vitesse de lacet
q	vitesse de tangage
Q	matrices d'erreurs de modélisation associé au filtre de Kalman
Q_p	matrice Q de la commande LQ
r	vitesse de roulis
R	matrices de bruits de mesures associé au filtre de Kalman
R_p	matrice R de la commande LQ
R_{rot}	matrice de rotation entre le repère terrestre et le repère avion
u	vitesse horizontale dans le repère avion

\mathbf{u}	vecteur du signal de commande d'un système
$\overline{\mathbf{u}}$	vecteur des valeurs maximales du signal de commande d'un système
$\underline{\mathbf{u}}$	vecteur des valeurs minimales du signal de commande d'un système
\mathbf{u}^f	vecteur des commandes effectives étant donné les fautes d'un système
v	vitesse latérale dans le repère avion
\mathbf{v}	vecteur représentant les bruits de mesures
V_a	vitesse air
V_x	vitesse horizontal dans le repère terrestre
V_y	vitesse latéral dans le repère terrestre
V_z	vitesse vertical dans le repère terrestre
w	vitesse verticale dans le repère avion
\mathbf{w}^x	vecteur représentant les erreurs de modélisation sur l'état
\mathbf{w}^b	vecteur représentant les erreurs de modélisation sur les paramètres de fautes
W_x	vent horizontal
W_y	vent latéral
W_z	vent vertical
x	1 ^{ère} composante de la position de l'avion dans le repère terrestre
\mathbf{x}	vecteur d'état d'un système
y	2 ^{ème} composante de la position de l'avion dans le repère terrestre
z	3 ^{ème} composante de la position de l'avion dans le repère terrestre
\mathbf{z}	vecteur de sorties réelles d'un système
\mathbf{z}^f	vecteur de sorties mesurées étant donné les fautes d'un système
α	angle d'incidence
β	angle de dérapage
δ_a	ailerons virtuels
δ_e	profondeur virtuelle
δ_l^i	$i^{\text{ème}}$ composant de l'élevon gauche
δ_r^i	$i^{\text{ème}}$ composant de l'élevon droit
δ_x	moteur
ϕ	angle de roulis
θ	angle de tangage
ψ	angle de lacet

Opérateurs

\hat{a}	estimation du paramètre a
a^i	$i^{\text{ème}}$ ligne du vecteur a
A_i	$i^{\text{ème}}$ ligne de la matrice A
$A_{.i}$	$i^{\text{ème}}$ colonne de la matrice A
A_{ij}	scalaire de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne de la matrice A
A^T	matrice transposée de A

A^{-1}	matrice inverse de A
\mathbb{I}_n	matrice identité de taille n
\mathbb{O}_{nm}	matrice nulle de taille $n \times m$
\mathbb{R}	ensemble des réels

Acronymes

AKF	de l'anglais Augmented Kalman Filter, Filtre de Kalman augmenté
EKF	de l'anglais Extended Kalman Filter, Filtre de Kalman étendu
DES	de l'anglais Discrete Event Systems, Systèmes à événements discrets
DoD	de l'anglais United State Department of Defense, Département de la Défense des États-Unis
DX	de l'anglais Diagnosis, Diagnostic
CA	de l'anglais Control Allocation, Allocation de commande
CUSUM	de l'anglais CUmulative SUM, Algorithme de somme cumulative
EA	de l'anglais Eigenstrucure Assignment, Placement de la structure propre
FAA	de l'anglais Federal Aviation Administration
FDI	de l'anglais Fault Detection and Isolation, Détection et isolation de fautes
FDIR	de l'anglais Fault Detection Identification and Reconfiguration, Détection, identification de fautes et reconfiguration
FDD	de l'anglais Fault Detection and Diagnosis, Détection et diagnostic de fautes
FTC	de l'anglais Fault Tolerant Control, Commande tolérante aux fautes
GLR	de l'anglais Generalized Likelihood Ratio, Rapport de vraisemblance généralisée
GPS	de l'anglais Global Positioning System, Système de positionnement par satellite
ICAO	de l'anglais International Civil Aviation Organization, Organisation de l'Aviation Civile Internationale
IMM	de l'anglais Interacting Multiple Model, Interaction multi-modèle
IMU	de l'anglais Inertial Measurement Unit, Centrale inertielle
LQ	Commande linéaire quadratique
MLD	de l'anglais Mixed Logic Dynamical, Système mixte à dynamique continue et logique
MMAC	de l'anglais Multiple Model Adaptive Control, Commande adaptative multi-modèles
MMAE	de l'anglais Multiple Model Adaptive Estimation, Estimation adaptative multi-modèles
MMST	de l'anglais Multiple Model Switching and Tuning, Commutation et Adaptation multi-modèles
MPC	de l'anglais Model Predictive Control, Commande prédictive
POMDP	de l'anglais Partially Observable Markov Decision Process, Processus de décision markovien partiellement observable
SMC	de l'anglais Sliding Mode Control, Commande par mode glissant

SMPC	de l'anglais Stochastic Model Predictive Control, Commande prédictive stochastique
SPRT	de l'anglais Sequential Probability Ratio Test, Test séquentiel du rapport de probabilité
TSEKF	de l'anglais Two-Stage Extended Kalman Filter, Filtre de Kalman à deux étages étendu
TSKF	de l'anglais Two-Stage Kalman Filter, Filtre de Kalman à deux étages
UAV	de l'anglais Unmanned Aerial Vehicle, Aéronef sans humain à bord
UKF	de l'anglais Unscented Kalman Filter, Filtre de Kalman non parfumé

Introduction générale

Sommaire

1.1	Contexte	1
1.1.1	Les drones	1
1.1.2	La sécurité des drones	3
1.1.3	Les méthodes existantes	4
1.2	Objectif et démarche de la thèse	4
1.3	Organisation de la thèse	6

1.1 Contexte

1.1.1 Les drones

Les aéronefs sans pilote humain à bord, en anglais *Unmanned Aerial Vehicles* (UAVs), également appelés robots aériens ou drones, sont capables de mener à bien des missions en étant pilotés à distance ou de manière autonome. Depuis l'apparition des drones dans l'espace aérien, ceux-ci sont utilisés dans de nombreux domaines militaires ou civils. On peut notamment citer la photographie (cf. Figure 1.1a), la surveillance de sites, les opérations de sauvetage (cf. Figure 1.1b), ou encore les opérations d'exploration et de détection. Par exemple, après un tremblement de terre, un drone peut être envoyé dans un bâtiment avant les humains pour vérifier si celui-ci risque de s'effondrer ou non. Contrairement à un robot terrestre, le drone peut survoler des débris.



(a) Photographie



(b) Opération de sauvetage

FIGURE 1.1 – Exemples d'utilisation de drones

Les drones offrent une grande variété de types et peuvent être classés selon leur masse, leur altitude de fonctionnement ou encore leur endurance (Section I

[Valavanis 2014]). Concernant les drones longue endurance, on distingue les drones de haute altitude et longue endurance, tel que le RQ-4 Global Hawk (cf. Figure 1.2a) qui sont des drones stratégiques, destinés à des missions de reconnaissance et les drones de moyenne altitude et longue endurance, tel que le MQ-9 Reaper (cf. Figure 1.2b) qui sont des drones tactiques.



(a) RQ-4 Global Hawk



(b) MQ-9 Reaper

FIGURE 1.2 – Exemples de drones de longue endurance

Les drones dits de moyenne et courte portée sont des drones tactiques, destinés à des missions de surveillance et fortement utilisés dans le milieu civil. Ce type de drone est en très fort développement avec par exemple le Sperwer (cf. Figure 1.3a) ou le Thales Watchkeeper WK450 (cf. Figure 1.3b).



(a) Sperwer



(b) Thales Watchkeeper WK450

FIGURE 1.3 – Exemples de drones de courte et moyenne portée

Finalement, on retrouve les mini drones et micro drones qui se classent respectivement dans la catégorie des $5kg - 30kg$ et des moins de $5kg$. Les mini-drones sont, comme leur nom l'indique des drones de petite taille, de l'ordre du mètre, caractérisés par une autonomie de quelques heures avec par exemple le DT26M (cf. Figure 1.4a) ou le Boreal (cf. Figure 1.4b). C'est la catégorie qui donne le plus souvent lieu au développement de plateformes expérimentales. Les mini-drones sont populaires pour illustrer des travaux de recherche.

Différentes configurations existent pour les drones, notamment les voilures fixes, les voilures tournantes et les convertibles. Les voilures fixes permettent d'accomplir principalement des missions d'observation. En effet, la durée de vol d'un drone à voilure fixe est en général plus importante que celle d'un drone à voilure tournante. Les voilures tournantes sont plus adaptées aux vols stationnaires et à basse vitesse. Les convertibles, peu utilisés à ce jour, tentent de combiner les avantages des deux

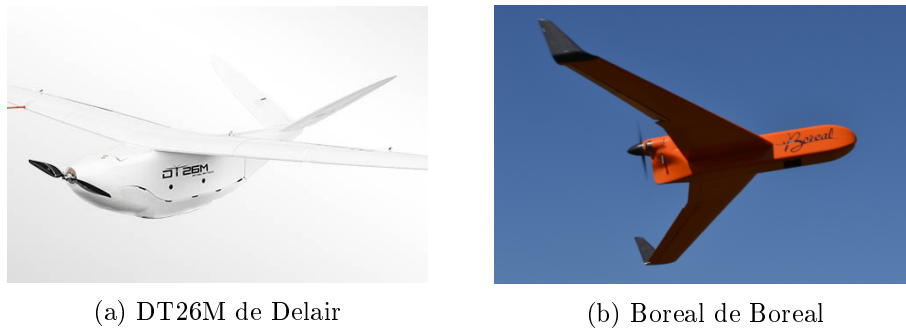


FIGURE 1.4 – Exemples de mini drones

configurations précédentes. Plusieurs études ont été réalisées afin de développer des véhicules au comportement équivalent à un hélicoptère à basse vitesse et à celui d'un avion à haute vitesse. Cependant, la conception de tels engins est complexe d'un point de vue mécanique et la phase de transition entre le mode hélicoptère et le mode avion s'avère difficile à gérer.

1.1.2 La sécurité des drones

Avant toute chose, il est important de distinguer sécurité de sureté. La sécurité concerne la capacité d'un système à ne causer aucun danger sur les humains, les équipements ou l'environnement. En aéronautique, la sureté concerne la lutte contre la malveillance. Néanmoins, le terme sureté de fonctionnement englobe la fiabilité, la maintenabilité, la disponibilité et la sécurité. Le contexte de cette thèse est celui de la sécurité des drones.

De par la multiplicité de leurs applications, le nombre de drones dans l'espace aérien a considérablement augmenté (comme le montre la Figure 1.5¹), entraînant une progression nette du nombre d'incidents. Selon l'Agence européenne de la sécurité aérienne, plus de 1400 incidents ont été recensés en 2016 contre seulement 606 sur la période 2011-2015.

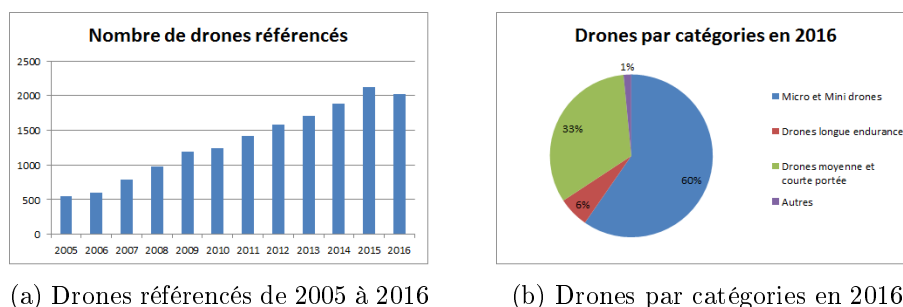


FIGURE 1.5 – Nombre de drones référencés sur la période 2005-2016

Des problèmes majeurs de sécurité voient le jour et en conséquence la sécurisa-

1. <https://rps-info.com/>

tion du vol des drones devient de plus en plus importante. La commande tolérante aux fautes, en anglais *Fault Tolerant Control* (FTC), pourrait alors contribuer à l'obtention, à moindre coût, d'un niveau de sécurité acceptable pour le vol des drones.

1.1.3 Les méthodes existantes

- Deux communautés s'intéressent aux problèmes de sécurisation :
- La communauté Automatique qui utilise des modèles quantitatifs pour représenter le comportement du système. Elle se base principalement sur la représentation de systèmes à l'aide de modèles représentant des évolutions continues. Ceux-ci décrivent l'évolution du système avec des équations différentielles dans le cas du temps continu ou avec des équations récurrentes dans le cas du temps discret (cf. Annexe B).
 - La communauté Intelligence Artificielle qui utilise des modèles qualitatifs pour décrire le système. Elle se base principalement sur la représentation de systèmes discrets ou logiques. Les approches à base de modèles discrets décrivent l'évolution du système avec des événements observables ou non observables.

Alors que les ingénieurs en commande négligent souvent les phénomènes discrets pouvant se produire, par exemple un changement brutal de comportement, les informaticiens travaillent sur des modèles prenant en compte ces types d'évènements mais simplifiant parfois excessivement la modélisation des évolutions continues des états.

On peut alors distinguer deux types d'approches ; la FTC appliquée à des systèmes à événements discrets et la FTC appliquée à des systèmes continus. La première approche exploite les méthodes d'Intelligence Artificielle alors que la deuxième utilise les techniques d'Automatique classique. Même si la distinction entre les deux peut parfois être difficile à déterminer, on peut résumer leurs différences de la façon suivante. Pour les techniques d'automatique classique, une loi de commande est définie, dont le comportement varie par nature selon les signaux observés par les différents capteurs embarqués, soit par retour direct de ces signaux, soit par synthèse de cette loi après une étape de reconstitution d'un modèle représentatif de l'appareil. Pour les méthodes d'Intelligence Artificielle, des décisions sont posées par l'ordinateur embarqué, en résolvant un problème d'optimisation permettant de prendre en compte les effets des différentes décisions possibles étant données des incertitudes sur ces effets.

1.2 Objectif et démarche de la thèse

Au cours de ces dernières années, de nombreuses recherches ont été faites sur ce sujet, cependant, la FTC est toujours un domaine de recherche actif. On propose de développer une méthode générique permettant un recouvrement du comportement du drone en cas de pannes capteurs et actionneurs en s'appuyant d'une part sur les techniques de commande adaptative et d'autre part sur les techniques de prise de

décision dans l'incertain. On ambitionne de coupler véritablement ces deux types de techniques ce qui s'avère complexe compte tenu des cadres de modélisation très différents permettant de les appliquer. Cette méthode doit être évaluée pour des scénarios précis de pannes capteurs et de pannes actionneurs. Les objectifs de la méthode ainsi développée sont d'une part de permettre une reconfiguration et une adaptation en cas de panne, et d'autre part de ne pas dégrader la performance en l'absence de panne.

Étant données les problématiques principales introduites précédemment, les trois objectifs de la thèse sont les suivants :

- développer une méthode tolérante aux fautes qui se base sur la représentation de systèmes continus et discrets rendant compte au mieux du comportement réel d'un drone, permettant le recouvrement en cas de pannes capteurs et actionneurs,
- appliquer cette méthode pour l'atterrissage d'un drone à voilure fixe et
- valider cette méthode.

Afin de répondre à ces objectifs, le développement de la méthode s'articule autour de trois grands axes.

Dans un premier temps, la démarche consiste à développer une méthode active de commande tolérante aux fautes actionneurs et capteurs en combinant les cadres de modélisation continus et discrets. L'objectif est de bénéficier des avantages des approches utilisant les techniques d'automatique classique et de celles exploitant les méthodes d'intelligence artificielle. Pour cela, l'idée consiste à combiner une modélisation continue permettant d'estimer l'état et les paramètres de fautes et une modélisation discrète permettant de prendre une décision en ligne quant au contrôleur à utiliser. L'estimation continue permet d'avoir plus d'informations sur la faute qu'avec une modélisation discrète, alors que celle-ci prend en compte de nombreuses techniques d'optimisation qui sont plus adaptées à la tâche de décision. Un second objectif consiste à développer une architecture FTC qui combine les méthodes indirectes de commutation et d'adaptation afin de limiter les désavantages de ces méthodes.

La méthode développée étant indirecte, l'étape suivante consiste à développer et valider une méthode permettant de détecter et de diagnostiquer la faute. Le filtre de Kalman, qui combine efficacité et facilité d'implémentation, permet d'estimer non seulement l'état du système mais aussi des paramètres de faute permettant d'avoir différentes informations sur la faute. Néanmoins, il n'est pas adapté aux pannes prenant la forme d'un saut dans la valeur des variables concernées. L'idée est donc de développer un filtre de Kalman sensible aux sauts induits par les pannes, que ce soit pour les actionneurs ou les capteurs. La détection de fautes basées sur les résidus, informations disponible grâce au filtre, engendre en général des fausses alarmes. Pour minimiser le nombre de fausses alarmes, l'idée est d'utiliser les données de l'estimation de façon probabiliste pour la détection et le diagnostic de faute, en anglais *Fault Detection and Diagnosis* (FDD).

Une fois la faute détectée et identifiée, le système de commande doit réagir pour pouvoir compenser cette faute. L'étape suivante consiste donc à améliorer le suivi

de la trajectoire par reconfiguration du système de commande. L'objectif est de limiter le nombre de contrôleurs en utilisant des contrôleurs adaptatifs, tout en ayant des contrôleurs faciles à concevoir. Un critère est défini de façon à concevoir des contrôleurs, un contrôleur nominal pour le mode nominal et des contrôleurs adaptatifs pour les modes dégradés. Une fois les contrôleurs conçus, il faut développer un outil permettant de prendre une décision quant au choix du contrôleur en ligne. L'idée est d'utiliser des techniques d'optimisation permettant de minimiser le coût, en se basant sur la probabilité de faute. Pour cela, trois approches différentes sont envisagées ; la première se base uniquement sur le cadre discret, alors que la seconde et la troisième se basent sur les cadres discret et continu.

Le deuxième objectif de la thèse consiste à appliquer la méthode développée à un drone à voilure fixe. Cette application peut se faire progressivement, en considérant au fur et à mesure des hypothèses de fautes de plus en plus complexes :

- un seul actionneur peut tomber en panne,
- deux actionneurs redondants peuvent tomber en panne,
- tous les actionneurs peuvent tomber en panne et
- tous les actionneurs peuvent tomber en panne et certains capteurs peuvent tomber en panne.

L'intégration de fautes supplémentaires peut entraîner des améliorations sur la méthode de détection et de diagnostic tout au long de la thèse.

Finalement, pour le troisième objectif, la méthode de FTC doit être vérifiée avant de pouvoir être implémentée sur un drone. Pour les besoins de simulation de la méthode, il est nécessaire de générer des mesures capteurs à partir d'un modèle de drone. Deux choix sont alors possibles : générer les mesures à partir du modèle linéarisé ou générer les mesures à partir du modèle non linéaire. L'étape suivante consiste à évaluer la capacité de la méthode à suivre une trajectoire d'atterrissage en cas de pannes grâce à un modèle de drone. Pour cela, une banque de scénarios est établie pour pouvoir tester la méthode. L'idée est de vérifier si la méthode FTC permet bien d'avoir de meilleures performances qu'une méthode prenant en compte une loi de commande classique, en cas de panne lors d'un atterrissage. Finalement, la vérification de la méthode étant faite sur un modèle, il est possible d'avoir des erreurs de modélisation. Or, la robustesse d'une méthode de ce type à des erreurs de modélisation est un attribut essentiel : elle est donc étudiée.

1.3 Organisation de la thèse

Ce manuscrit est organisé en neuf parties permettant de distinguer :

- l'analyse du problème, ainsi que l'état de l'art associé à la FTC (Chapitre 2 et Chapitre 3),
- la méthode FTC développée (les principes généraux dans le Chapitre 4, la méthode FDD dans le Chapitre 5 et le mécanisme de reconfiguration dans le Chapitre 7),
- l'application et l'évaluation de la méthode sur un drone (l'évaluation de la

méthode FDD dans le Chapitre 6 et l'évaluation du suivi de trajectoire dans le Chapitre 8) et,

— l'analyse critique de notre méthode, ainsi que les perspectives (Chapitre 9).

Le deuxième chapitre vise à exposer les caractéristiques du problème de sécurité du vol des drones, ainsi qu'à introduire un drone à voilure fixe, L'Altimum 4, instance concrète de ce problème. Après avoir évoqué les différences entre les exigences de sécurité des avions habités, et celles des drones, ce chapitre permet de présenter les types de fautes pouvant intervenir à bord des drones et les conséquences que cela peut engendrer. Finalement, ce chapitre termine par une synthèse permettant d'établir notamment les fautes que l'on souhaite traiter.

Le troisième chapitre est consacré à l'état de l'art de la FTC. Il permet de mettre en perspective et d'expliquer certains choix relatifs à la démarche de cette thèse, par exemple le choix de s'orienter vers des méthodes actives indirectes et non vers des méthodes passives, ou encore le choix d'utiliser des méthodes basées modèle.

Le quatrième chapitre présente l'architecture, ainsi que le cadre de modélisation générique de la méthode active de commande tolérante aux fautes développée au cours de cette thèse. Des modélisations discrètes et continues y sont présentées.

Le cinquième chapitre est dédié au développement d'une méthode permettant de détecter et de diagnostiquer la faute. Ce chapitre peut être décomposé en deux parties ; le développement d'un estimateur sensible aux sauts de pannes à partir de la modélisation continue, la mise en place d'une détection de faute robuste aux fausses alarmes à partir de la modélisation discrète en utilisant les informations fournies par l'estimateur. L'évaluation de la méthode est l'objectif du sixième chapitre. La détection et le diagnostic de la faute étant essentiels pour obtenir de bonnes performances, le filtre développé doit être suffisamment précis et fiable. La méthode développée est donc évaluée quantitativement et qualitativement.

Le septième chapitre est dédié au développement du système de commande actif permettant de compenser la faute détectée. Pour les capteurs, la faute est compensée par la méthode de détection et de diagnostic. La reconfiguration actionneur se décompose en deux parties ; la conception de contrôleurs, c'est-à-dire du contrôleur nominal et des contrôleurs adaptatifs pour les modes dégradés, et le développement d'un outil permettant de prendre une décision en ligne quant au choix du contrôleur. Trois techniques ont été utilisées pour la tâche de décision, ainsi ce chapitre est conclu avec une étude comparative de façon à mettre en avant les avantages et les désavantages des différentes méthodes.

Le huitième chapitre vise à évaluer la méthode développée. A partir d'une banque de scénarios, on évalue grâce à un modèle de drone la capacité de la méthode à suivre une trajectoire d'atterrissage en cas de pannes. De plus, la robustesse de la méthode face à des cas de pannes non envisagés par le modèle et face aux incertitudes liées à la modélisation du drone est évaluée.

Enfin, le neuvième chapitre vise à conclure et à avoir une analyse critique sur le travail effectué au cours de cette thèse. De plus, des perspectives y sont présentées.

Un problème de reconfiguration d'un mini-drone en panne lors de l'atterrissage

Sommaire

2.1	Les aspects spécifiques de la sécurité du vol des drones . . .	11
2.1.1	L'expérience des aéronefs habités et l'adaptation nécessaire des exigences associées	11
2.1.2	Un impact important d'une panne sur la sécurité du contrôle de la trajectoire en vol	12
2.1.3	La variété des types de fautes possibles	14
2.1.4	Une opportunité pour la redondance analytique	19
2.2	Le drone Altimum 4	19
2.2.1	Actionneurs	20
2.2.2	Capteurs	21
2.2.3	Modèle de simulation	25
2.3	Scénarios de panne pour le drone Altimum 4	27
2.3.1	La phase d'atterrissage	27
2.3.2	Les types de fautes envisagés	28
2.4	Conclusion	28

Résumé

CE chapitre vise à présenter le problème lié à cette thèse, qui s'inscrit dans le cadre de la sécurité des drones à voilure fixe. Dans un premier temps, les différences entre ce qui est mis en œuvre au niveau des aéronefs habités et ce qui peut être mis en œuvre au niveau des drones sont analysées. Suit une description des différentes fautes pouvant impacter le vol. Puis, afin de poser ces problèmes de sécurité du vol dans un contexte concret, le drone Altimum 4 de l'Avion Jaune ainsi qu'un modèle de simulation de ce drone pouvant permettre le développement et l'évaluation de méthodes FTC sont présentés.

2.1 Les aspects spécifiques de la sécurité du vol des drones

2.1.1 L'expérience des aéronefs habités et l'adaptation nécessaire des exigences associées

L'étude de la sécurité du vol des drones suit une méthodologie similaire à celle des avions habités. Il y a d'une part une définition des objectifs de sécurité et d'autre part une analyse des fautes pouvant survenir au niveau des différentes fonctions et des différents composants du véhicule. Ces deux aspects se rejoignent par la construction d'arbres de fautes liant celles-ci aux cas de pannes classés en fonction des objectifs.

En ce qui concerne la définition des objectifs de sécurité, la *Federal Aviation Administration* (FAA) distingue cinq catégories de risques : catastrophiques, dangereux, majeurs, mineurs et sans effets. La Figure 2.1 montre une classification des défaillances en fonction d'une part des catégories de risques et d'autre part de la probabilité d'occurrence des fautes.

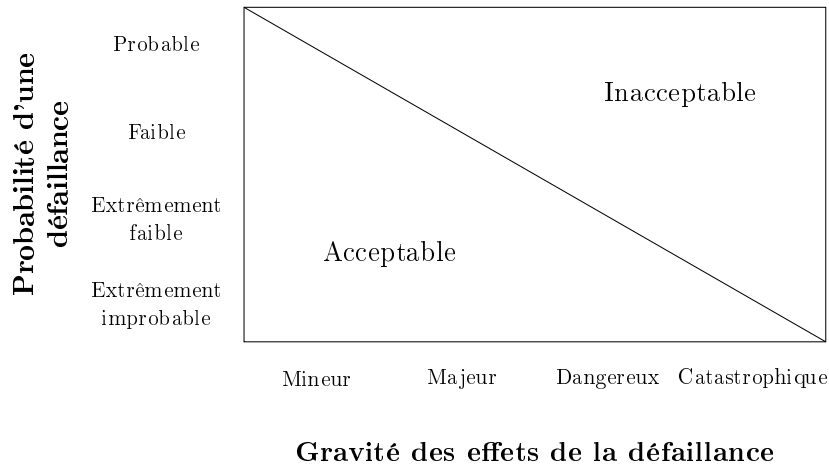


FIGURE 2.1 – Classification des défaillances en fonction de la catégorie de risque et de la probabilité de faute [Hayhurst 2007]

Les catégories de risques pour les drones ont initialement été définies par Hayhurst en se basant sur la définition des catégories de risques pour les avions [Hayhurst 2007]. Il existe néanmoins d'autres définitions similaires (Chapitre 92 [Valavanis 2014]). L'Organisation de l'Aviation Civile Internationale, en anglais *International Civil Aviation Organization* (ICAO), a proposé une définition en 2009 et le département de la Défense des États-Unis, en anglais *United States Department of Defense* (DoD), a proposé une définition en 2010. Le Tableau 2.1 permet de comparer ces définitions.

Il faut noter que le cas de panne catastrophique ne se résume pas à tout type de crash qui dans le cas d'un avion habité entraîne le décès des passagers, mais les crashes entraînant le décès de personnes au sol ou la perte totale du drone. Les exigences de sécurité posées sur le système de drone sont donc à adapter par rapport à celles des avions habités.

Hayhurst [Hayhurst 2007]	ICAO [Valavanis 2014]	DoD [Valavanis 2014]
<i>Catastrophique</i> : défaillances qui peuvent engendrer un ou plusieurs décès ou des blessures graves à des personnes, ou une perte de contrôle prolongée entraînant la perte du drone	<i>Catastrophique</i> : équipement détruit ; plusieurs morts	<i>Catastrophique</i> : peut aboutir à un ou plusieurs effets suivants ; mort, invalidité totale permanente, impact environnemental important et irréversible ou perte dépassant \$10M
<i>Dangereux</i> : défaillances qui réduisent la capacité opérationnelle du drone ou la capacité du pilote à s'en sortir dans des conditions opérationnelles défavorables. Par exemple, une réduction très importante des marges de sécurité ou des capacités fonctionnelles ; une détresse physique ou une charge de travail trop importante pour que le pilote puisse mener les tâches de façon complète et précise ; un inconfort physique de personnes, possiblement des blessures	<i>Dangereux</i> : réduction très importante des marges de sécurité, détresse physique, ou charge de travail telle que l'opérateur ne peut pas s'appuyer sur le système pour réussir ses tâches précisément et totalement. Blessure sérieuse et dommage matériel important	<i>Critique</i> : peut aboutir à un ou plusieurs effets suivants ; invalidité partielle permanente, blessures ou maladie professionnelle qui peut entraîner une hospitalisation d'au moins trois personnes, impact environnemental important et réversible, ou perte dépassant \$1M mais inférieure à \$10M
<i>Majeur</i> : défaillances qui réduisent la capacité opérationnelle du drone ou la capacité du pilote à s'en sortir dans des conditions opérationnelles défavorables. Par exemple, une réduction importante des marges de sécurité ou des capacités fonctionnelles ; une augmentation importante de la charge de travail du pilote ou des conditions nuisant à l'efficacité du pilote ; un inconfort pour le pilote - possiblement des blessures - ou possiblement un inconfort physique de personnes.	<i>Majeur</i> : réduction importante des marges de sécurité, une réduction de la capacité des opérateurs à gérer des conditions opérationnelles défavorables suite à une augmentation de la charge de travail, ou suite à des conditions nuisant à leur efficacité. Incident sérieux et blessure à des personnes	<i>Minime</i> : peut aboutir à un ou plusieurs effets suivants ; blessures ou maladie professionnelle entraînant dix ou plus jours d'arrêt de travail, impact environnemental modéré et réversible, ou perte dépassant \$100K mais inférieure à \$1M
<i>Mineur</i> : défaillances qui ne réduisent pas significativement la sécurité du drone et qui impliquent des actions faisables par le pilote. Par exemple, une légère réduction des marges de sécurité ou des capacités fonctionnelles ; ou une légère augmentation de la charge de travail du pilote, comme un changement au niveau du plan de vol.	<i>Mineur</i> : nuisance ; limitations opérationnelles ; utilisation des procédures d'urgence ; incident mineur	<i>Négligeable</i> : peut aboutir à un ou plusieurs effets suivants ; blessures ou maladie entraînant moins de dix jours d'arrêt de travail, impact environnemental minime, ou perte inférieure à \$100K
<i>Sans effet</i> : défaillances qui n'ont pas d'effets sur la sécurité du drone, c'est à dire, qui n'affectent pas la capacité opérationnelle du drone ou qui n'augmentent pas la charge de travail du pilote.	<i>Négligeable</i> : peu de conséquence	

Tableau 2.1 – Exemples de définitions pour les catégories de risques

2.1.2 Un impact important d'une panne sur la sécurité du contrôle de la trajectoire en vol

En ce qui concerne l'analyse des fautes, des études ont permis d'identifier les aléas les plus fréquents ainsi que leur impact sur la sécurité du vol. Par exemple, une étude montre l'impact des risques fonctionnels des drones dans la perspective

de leur insertion en milieu aérien non ségrégué, en cherchant à analyser l'impact des aléas rencontrés par les drones sur le trafic général [Hayhurst 2007]. Basée sur la décomposition fonctionnelle d'un avion, une décomposition fonctionnelle d'un drone générique est effectuée (cf. Figure 2.2).

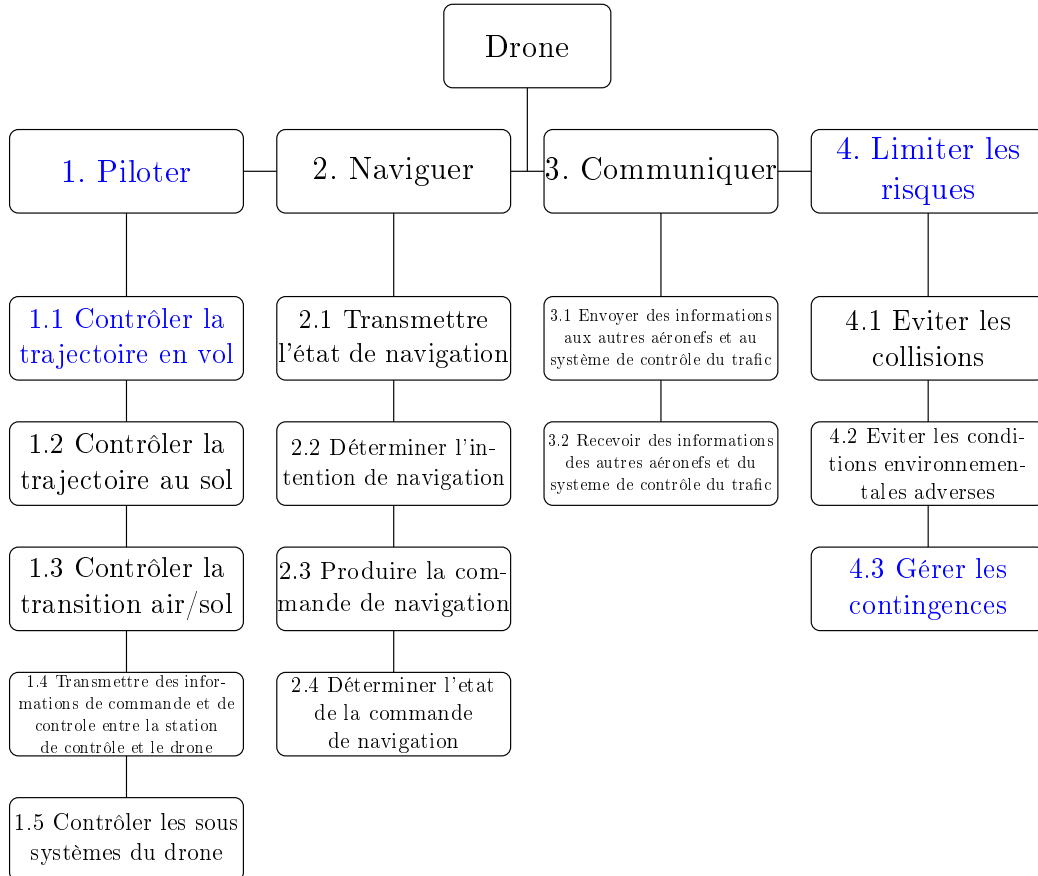


FIGURE 2.2 – Niveau haut de la décomposition fonctionnelle [Hayhurst 2007]

Pour chaque groupe de fonctions, une évaluation des dangers fonctionnels est faite. Cette analyse n'est pas complète mais permet d'avoir une vue préliminaire. Sur un total de 132 cas de pannes, on a :

- 49 pannes pour le groupe de fonctions “Piloter”, dont 15 catastrophiques et 9 dangereuses, dont
 - 19 pour contrôler la trajectoire en vol
 - 5 pour contrôler la trajectoire au sol
 - 11 pour contrôler la transition air / sol
 - 5 pour transmettre des informations
 - 9 pour contrôler des sous-systèmes du drone
- 15 pour le groupe de fonctions “Naviguer”
- 16 pour le groupe de fonctions “Communiquer”
- 52 pour le groupe de fonctions “Limiter les risques”, dont 9 catastrophiques

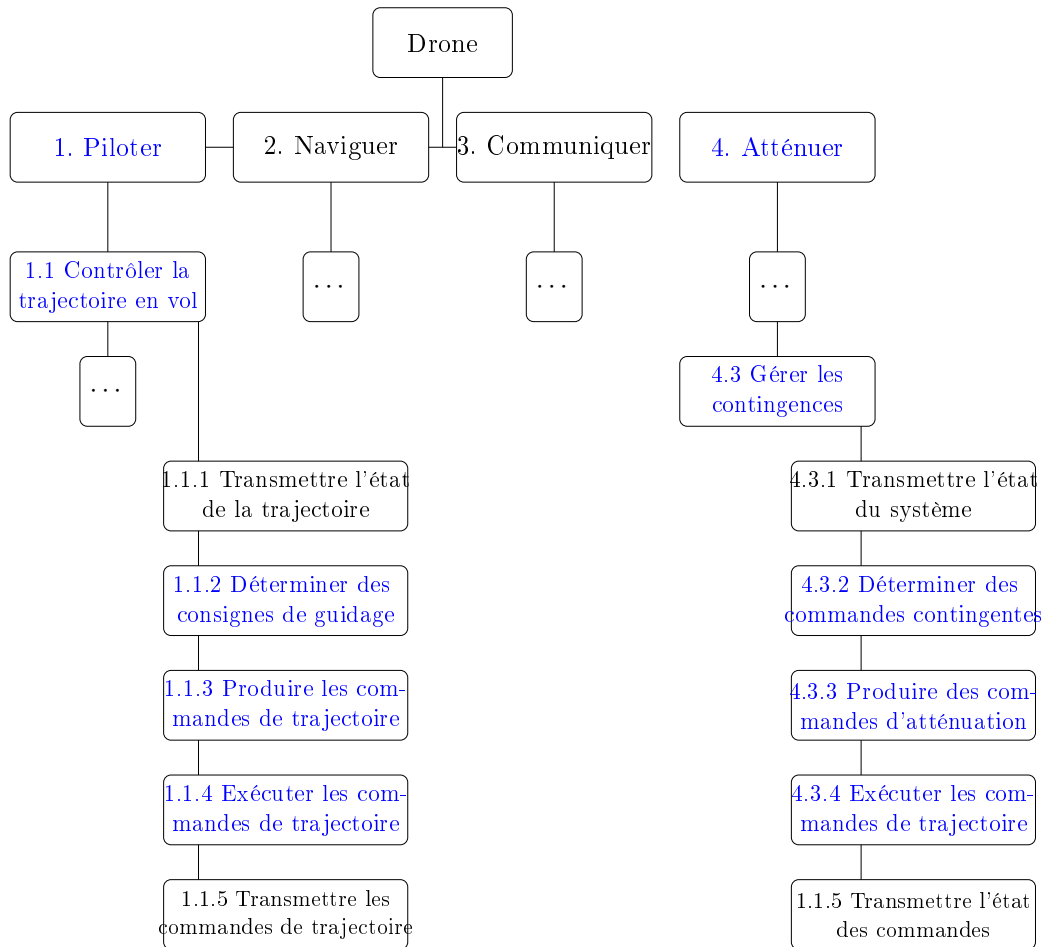


FIGURE 2.3 – Niveau bas de la décomposition fonctionnelle [Hayhurst 2007]

et 9 dangereuses, dont

- 16 pour éviter la collision
- 5 pour éviter les conditions environnementales adverses
- 19 pour gérer les contingences

On s'intéresse plus particulièrement aux sous-groupes de fonctions inscrites en bleu sur la Figure 2.2. Ces sous-groupes de fonctions se décomposent en différentes parties de plus bas niveau (cf. Figure 2.3).

2.1.3 La variété des types de fautes possibles

On peut distinguer quatre types de fautes : les fautes actionneurs, les fautes capteurs, les fautes du système de commande et les fautes composants. Ces fautes interviennent à différents endroits dans la boucle implémentant les fonctions de navigation, guidage et pilotage comme le montre la Figure 2.4. Ces fautes peuvent intervenir soit de façon permanente, soit de façon intermittente.

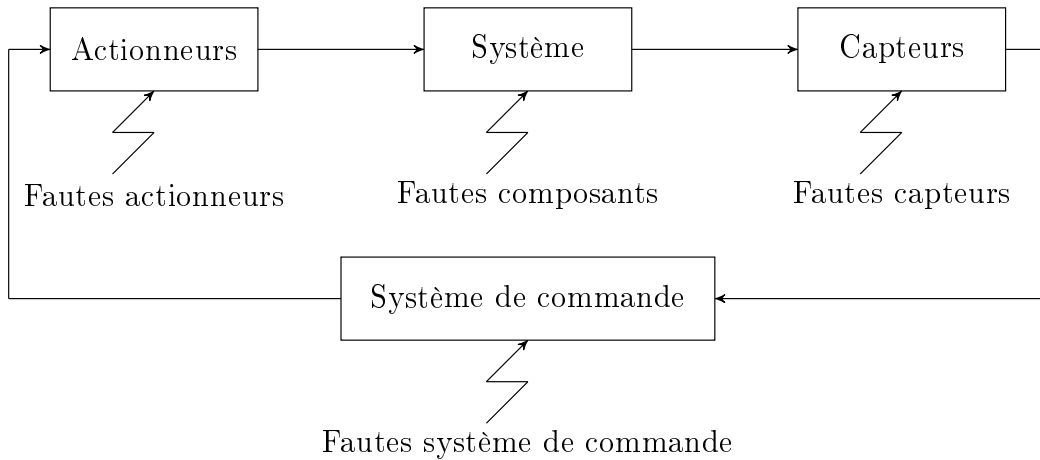


FIGURE 2.4 – Différents types de fautes

2.1.3.1 Fautes actionneurs

Les actionneurs sont indispensables, ils permettent au système de commande d’agir sur le système. De manière générale un actionneur est un système mécanique qui transforme l’énergie qui lui est fournie en une action physique qui a un effet sur le système sur lequel il s’applique.

La panne d’un actionneur peut venir d’un problème mécanique au niveau du servomoteur, par exemple si le servomoteur est bloqué dans une position, ou d’un problème électrique en amont, par exemple si le servomoteur est déconnecté de la batterie. En condition nominale, l’actionneur va se positionner à la valeur envoyée par le système de commande. Quand une faute apparaît au niveau de l’actionneur, l’action physique ne sera pas complètement effectuée. Nous sommes donc amenés à distinguer l’action physique de la commande, $u^f(t)$, de la valeur envoyée par le système de commande $u(t)$.

Généralement, on distingue différents types de fautes actionneurs [Ducard 2009].

Les pannes solides qui substituent le signal nominal :

- Actionneur bloqué : dans ce cas, l’actionneur se bloque à une certaine valeur et ne répond plus aux commandes, $u^f(t) = a$ où a est une valeur constante (cf. Figure 2.5 (b), courbe bleu pointillée).
- Actionneur bloqué à une limite : dans ce cas, l’actionneur se déplace jusqu’à la position limite inférieure ou supérieure sans prendre en compte le signal de commande, $u^f(t) = \bar{u}$ ou $u^f(t) = \underline{u}$ (cf. Figure 2.5 (b), courbe bleu).
- Actionneur flottant : dans ce cas, l’actionneur se déplace indépendamment des ordres qui lui sont envoyés (cf. Figure 2.5 (a)).

Les pannes liquides qui gardent la dynamique du système, c’est à dire que le signal nominal est multiplié par un facteur ou un signal lui est ajouté :

- Actionneur en perte d’efficacité : dans ce cas, l’actionneur perd partiellement de son efficacité par rapport au cas nominal, $u^f(t) = ku(t)$ où k est le facteur d’efficacité $0 < k < 1$ (cf. Figure 2.5 (c)).

- Actionneur avec un biais : dans ce cas, l'actionneur a un terme additionnel constant par rapport au cas nominal, $u^f(t) = u(t) + a$ où a est le biais (cf. Figure 2.5 (d)).

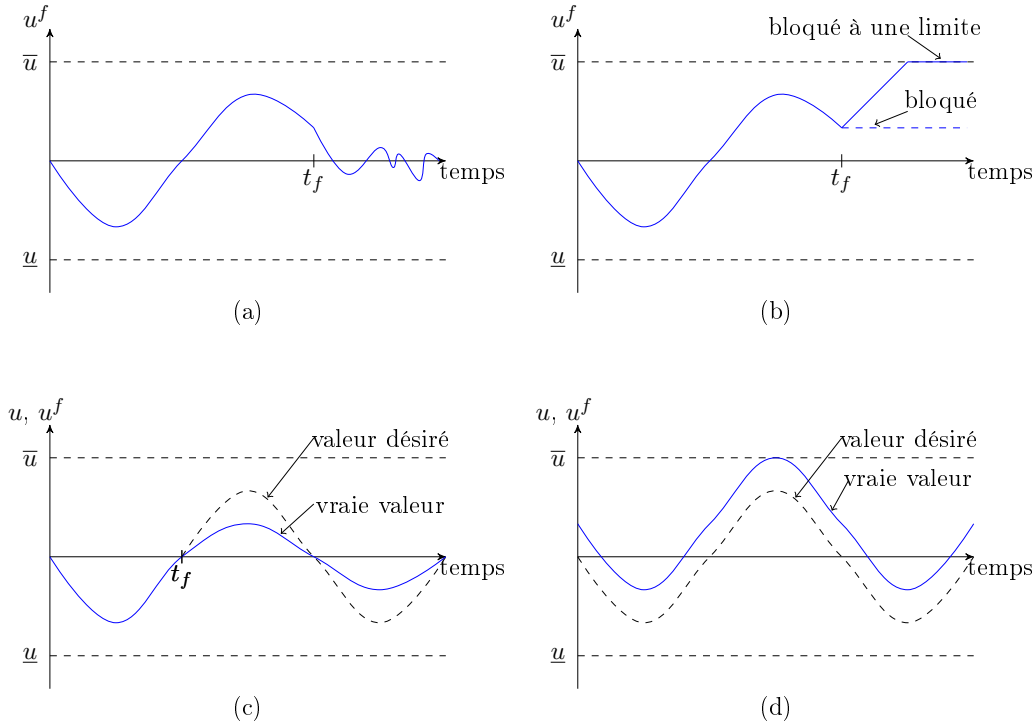


FIGURE 2.5 – Différents types de fautes actionneurs [Ducard 2009]

On retrouve aussi la panne oscillatoire liquide ou solide [Goupil 2010] :

- Actionneur en panne oscillatoire solide : dans ce cas, l'actionneur se déplace de façon oscillatoire, $u^f(t) = f_{sol}(t)$ où f_{sol} est un signal oscillatoire.
- Actionneur en panne oscillatoire liquide : dans ce cas, un terme oscillatoire est ajouté au signal de commande $u^f(t) = u(t) + f_{liq}(t)$ où f_{liq} est un signal oscillatoire.

Cependant ce type de panne est plus rare.

En ce qui concerne les actionneurs des drones, l'ensemble des modes de fautes du drone *Ultra Stick 120 'Ibis'*, un drone à aile fixe électrique, a été étudié [Freeman 2014]. On retrouve des pannes liées à la centrale électrique :

- les pannes liées aux batteries du moteur pouvant entraîner une perte de la poussée, et
- les pannes du régulateur de vitesse électronique pouvant entraîner une vitesse erronée, une perte intermittente ou totale de la poussée.

Parmi les pannes effecteurs, on retrouve également :

- Les pannes liées aux batteries peuvent entraîner une perte de puissance au niveau des actionneurs ou bien un blocage des surfaces à la position courante.
- Les pannes au niveau du multiplexeur peuvent entraîner une perte du signal

envoyé aux actionneurs.

- Les différentes pannes peuvent arriver au niveau des actions de commande des surfaces. Une surface - profondeur, gouverne, aileron ou volet - peut être biaisée, bloquée à une position, flottante, en oscillation, avoir une bande morte augmentée ou avoir une dynamique plus lente.

Remarque 2.1

Les fautes de type blocage sont plus graves que les fautes de type perte d'efficacité ou biais. En effet, une fois un actionneur bloqué, il est possible que le système ne soit plus contrôlable. Il est donc préférable d'avoir des actionneurs redondants dans le cas d'une faute de type blocage. Ainsi une panne blocage au niveau du moteur est plus difficile à gérer qu'une panne blocage au niveau d'un élévon.

2.1.3.2 Fautes capteurs

Un capteur est un système qui mesure des grandeurs physiques. Les capteurs sont entachés de différents types d'erreur [Heredia 2005] [Ducard 2009] [Qi 2013]. On est également amené à distinguer le signal qui devrait être mesuré, $z(t)$, du signal effectivement mesuré, $z^f(t)$.

Ils sont sujets à des pannes solides :

- Capteur en panne totale : dans ce cas, le capteur arrête de fonctionner et donne une sortie constante égale à zéro, $z^f(t) = 0$ (cf. Figure 2.6 (b), courbe bleu). Ce type de panne peut être dû à des problèmes électriques.
- Capteur bloqué à une valeur constante : dans ce cas, la sortie du capteur devient constante $z^f(t) = a$ où a est une valeur constante (cf. Figure 2.6 (b)).
- Capteur en panne oscillatoire solide : dans ce cas, la sortie du capteur est un signal oscillatoire, $z^f(t) = f_{sol}(t)$ où f_{sol} est un signal oscillatoire.

Ils subissent également des pannes liquides :

- Capteur avec faute additive : dans ce cas, l'erreur se présente sous la forme d'un terme additif, un biais $z^f(t) = z(t) + a$ où a une constante (cf. Figure 2.6 (d)). C'est un type de faute commun pour les capteurs analogiques, dû à des changements de températures ou à des problèmes de calibrations.
- Capteur avec dérive : dans ce cas, la sortie du capteur ajoute un terme qui dérive dans le temps, un biais $z^f(t) = z(t) + a(t)$ où $a(t)$ une fonction croissante (cf. Figure 2.6 (a)). C'est un type de faute commun pour les capteurs analogiques, dû à des changements de températures ou à des problèmes de calibrations.
- Capteur avec faute multiplicative : dans ce cas, un facteur multiplicatif k est appliqué à la valeur nominale du capteur $z^f(t) = kz(t)$ où k une constante (cf. Figure 2.6 (c)).
- Capteur plus bruité : dans ce cas, le capteur est moins précis et le bruit de mesure est plus important.

- Capteur en panne oscillatoire liquide : dans ce cas, la sortie du capteur ajoute un terme oscillatoire $z^f(t) = z(t) + f_{liq}(t)$ où f_{liq} est un signal
- Capteur avec valeur aberrante : c'est une panne temporaire qui apparaît par exemple sur les systèmes de positionnement par satellite, en anglais *Global Positioning System* (GPS). Dans ce cas, un point avec un grand écart est donné par le GPS, mais juste après, les mesures suivantes sont correctes, $z^f(t) = z(t) + a\delta(t - t_0)$ où a est une valeur stochastique, t_0 est le temps à laquelle la faute apparaît et $\delta(t)$ est une fonction pulse, valant 1 lorsque $t = 0$, 0 sinon. Ces erreurs peuvent être causées par des pannes au niveau de l'algorithme du GPS ou par un blocage temporaire du signal satellite.

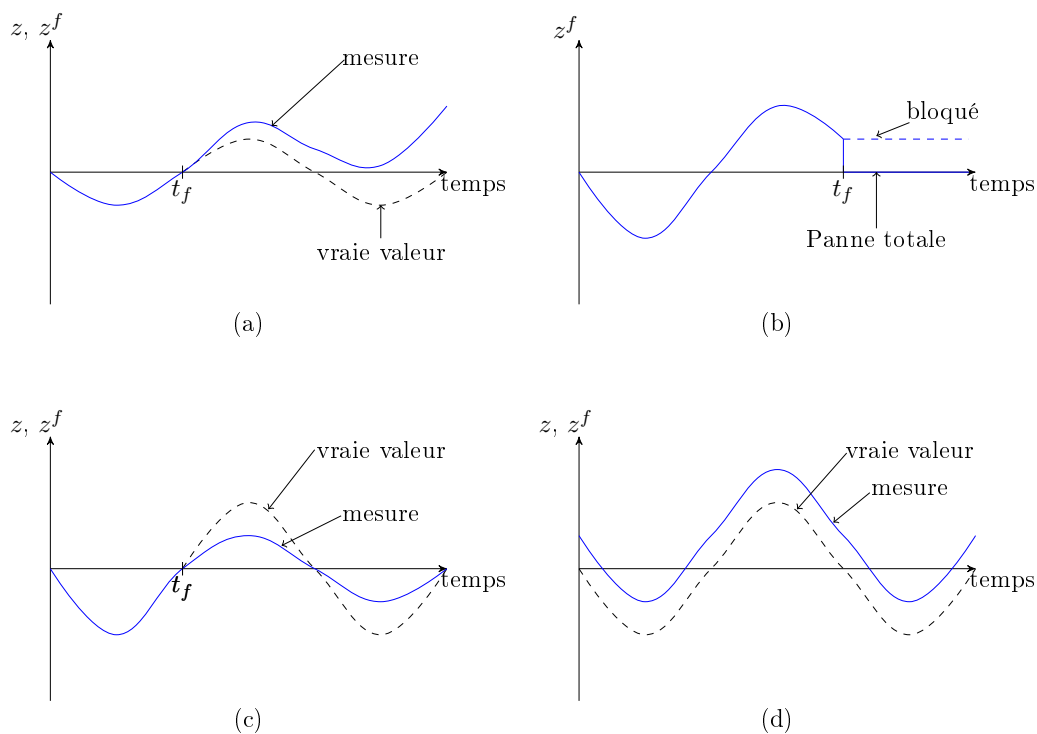


FIGURE 2.6 – Différents types de fautes capteurs [Ducard 2009]

En ce qui concerne le drone *Ultra Stick 120 'Ibis'* on trouve [Freeman 2014] :

- la perte des données capteurs ou du signal de commande pouvant être causée par la déconnexion vis à vis de l'ordinateur de bord ou des batteries, ou par un court-circuit,
- les pannes liées directement aux capteurs pouvant entraîner des biais au niveau des mesures, des données erronées, des blocages ou des pannes totales.

2.1.3.3 Fautes des composants et du système de commande

Les fautes des composants représentent les changements au niveau des paramètres physiques du drone, comme une déformation de la structure de l'aéronef. Ces fautes sont irréversibles et influencent les paramètres du modèle du système.

Par ailleurs le système de commande, y compris le FTC, est implémenté sur une carte électronique. Cette carte peut avoir des erreurs qui par nature peuvent ne pas pouvoir être compensées par le FTC.

2.1.4 Une opportunité pour la redondance analytique

Dans le domaine de l'aviation commerciale et générale, la sûreté du vol d'un engin est assurée par exemple par triplement des matériels assurant les différentes fonctions et par des règles décisionnelles simples. Ainsi, lorsque la ressource nominale est détectée défectueuse, de par le fait que sa sortie est significativement différente de celles des ressources de réserve, une reconfiguration consiste à remplacer la ressource défectueuse par une ressource de réserve [Goupil 2011]. La redondance matérielle peut donc empêcher une perte de contrôle de l'avion. Cependant, la redondance matérielle pour des mini et micro drones ne semble pas adaptée en général à cause du coût, du prix, de la masse et de la consommation.

En effet, mis à part certains drones militaires, le coût d'un drone est sans commune mesure avec le coût d'un avion habité et donc les méthodes permettant de tenir les exigences de sécurité du vol ne peuvent pas être aussi coûteuses que pour les avions habités.

La redondance analytique semble plus adaptée aux drones pour contrer la perte de contrôle. Elle utilise les signaux générés par un modèle mathématique. En particulier, on peut chercher à ce que les systèmes de commande soient performants en situation nominale mais aussi en situation dégradée, par exemple en cas de blocage d'actionneur. Il faut signaler à ce titre que l'absence de précision des capteurs embarqués sur des drones -par comparaison avec les capteurs disponibles en aéronautique civile habitée- rend plus délicate l'utilisation des techniques de redondance analytique développées dans ce dernier cadre. Le but est de pouvoir ramener le drone au sol en toute sécurité malgré la défaillance de composants.

La nouvelle génération de drones devra donc être capable d'accomplir leur mission avec de meilleures performances mais aussi avec plus de sécurité. Les drones devront être munis d'algorithmes capables de surveiller leur état et de les rendre plus résilients sans pour autant ajouter des capteurs ou des actionneurs.

2.2 Le drone Altimum 4

L'Avion Jaune Altimum 4¹, mini drone à voilure fixe, est disponible à l'ONERA. Un modèle fin, représentatif de sa dynamique, a été établi [Lesprier 2015a]. C'est pourquoi celui-ci permet de poser concrètement le problème présenté dans la section précédente.

Comme le montre la Figure 2.7, l'Altimum 4 est une aile volante de 4 mètres d'envergure dotée d'une motorisation thermique arrière. La plage de vitesse considérée va de $18m/s$ à $28m/s$, celle de l'angle de tangage de -6° à 6° , celle de la masse

1. <http://lavionjaune.fr/>

de $17kg$ à $22kg$ et le centre de gravité peut varier de $\pm 5\%$. Ce drone peut emporter jusqu'à $4kg$ de charge utile et opérer avec un rayon d'action allant jusqu'à $100km$.



FIGURE 2.7 – L'Avion Jaune Altium 4

2.2.1 Actionneurs

Les cinq actionneurs utilisés sont des servomoteurs électriques. Un servomoteur est un système motorisé capable d'atteindre des positions prédéterminées, puis de les maintenir. Ces cinq actionneurs sont ; la commande de poussée δ_x , les commandes des deux élevons gauches δ_l^1 et δ_l^2 et celles des deux élevons droits δ_r^1 et δ_r^2 . Il n'y a pas de retour sur la position des actionneurs. Étant donné qu'il s'agit d'une aile volante sans stabilisateur, le drone dispose d'élevons pour gérer à la fois le roulis et le tangage. Il est possible de définir des braquages pour des ailerons virtuels δ_a et une profondeur virtuelle δ_e . Les relations entre les ailerons, la profondeur et les élevons sont alors les suivantes :

$$\delta_a = \frac{0.4\delta_l^1 + 0.6\delta_l^2 - 0.4\delta_r^1 - 0.6\delta_r^2}{2} \quad \delta_e = -\frac{0.5\delta_l^1 + 0.5\delta_l^2 + 0.5\delta_r^1 + 0.5\delta_r^2}{2} \quad (2.1)$$

La Figure 2.8 montre les actionneurs qui peuvent être disponible sur l'Altium 4, ainsi que leur emplacement sur l'aile gauche.

Lors de cette thèse, seul les gouvernes sont utilisées car les volets et les winglets ne sont pas encore motorisés sur le drone réel.

Remarque 2.2

Pour certaines expérimentations numériques, on suppose que la dynamique longitudinale peut être découplée de la dynamique latérale et donc on s'intéresse uniquement à δ_e . En effet, pour le modèle longitudinal, seules la poussée et la profondeur sont considérées. D'après l'équation 2.1, les élevons gauche δ_l^1 , δ_l^2 et droit δ_r^1 , δ_r^2 ont tous le même effet sur le mouvement longitudinal, en effet le coefficient est le même pour chaque élevon.

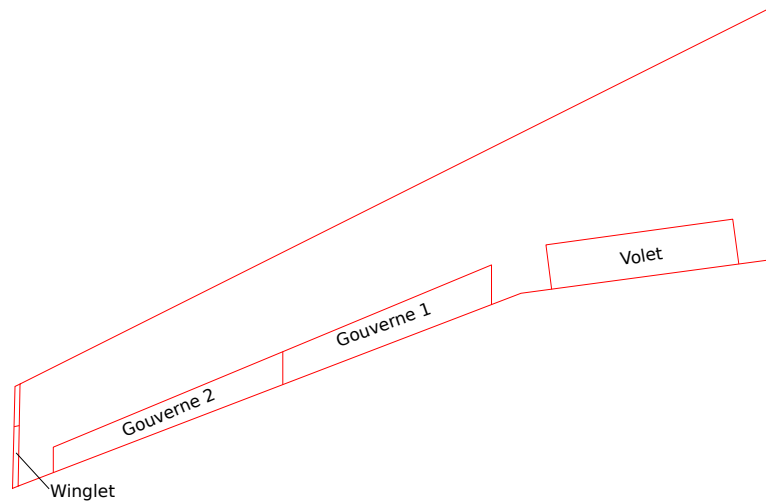


FIGURE 2.8 – Aile gauche de l’Avion Jaune Altimum 4

2.2.2 Capteurs

L’Altimum 4 possède différents capteurs qui permettent de mesurer des grandeurs pertinentes pour sa commande : une centrale inertielle, en anglais *Inertial Measurement Unit* (IMU), un GPS, un baromètre, un magnétomètre, deux inclinomètres, un capteur d’incidence et un capteur de dérapage.

Tous les capteurs utilisés appartiennent à l’une des deux grandes catégories suivantes [Aldon 2001] :

- Les capteurs proprioceptifs qui délivrent leurs mesures par rapport à ce qu’ils perçoivent localement du déplacement du véhicule dans l’espace. Ces mesures ne dépendent pas des conditions environnementales dans lesquelles évolue l’engin.
- Les capteurs extéroceptifs qui prennent leurs mesures par rapport à l’environnement global du mobile. Ces informations correspondent à des observations par rapport à un référentiel absolu attaché à l’environnement.

Tous les capteurs de l’Avion Jaune donnent une information sur le mouvement du véhicule en ce sens ils sont proprioceptifs. Néanmoins, mis à part l’IMU, ils se basent sur des caractéristiques de l’environnement pour déduire cette information. Les capteurs autres que l’IMU peuvent donc être considérés comme extéroceptifs.

Suivant sa technologie, un capteur produisant des informations de type vecteur, fournit ces informations dans un repère donné. On distingue au moins deux repères :

- un repère lié au véhicule, appelé le repère avion et
- un repère lié à la terre, appelé le repère terrestre.

La Figure 2.9 illustre dans le plan vertical le lien entre ces deux repères.

De manière générale, une matrice de rotation R_{rot} (cf. Annexe A) permet de passer des coordonnées exprimées dans le repère terrestre à des coordonnées exprimées

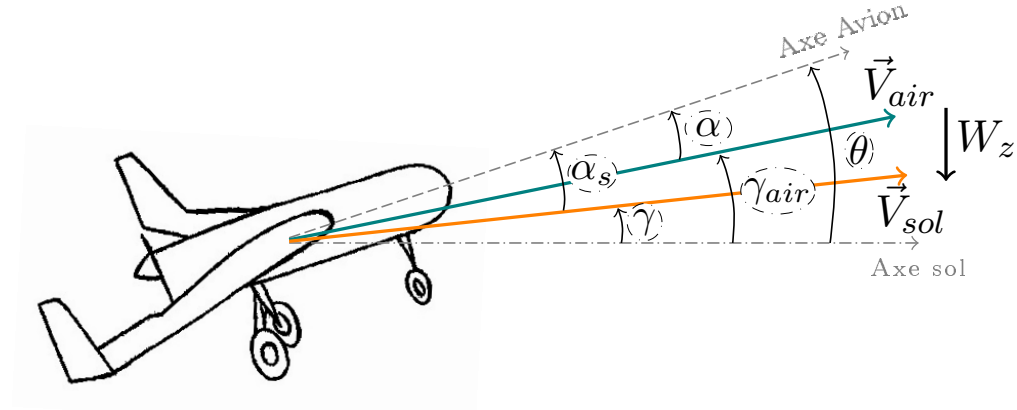


FIGURE 2.9 – Repère avion

dans le repère avion.

2.2.2.1 Centrale inertielle

Par définition, une IMU est composée de trois accéléromètres et de trois gyroscopes permettant de mesurer, dans un repère lié au véhicule, la force spécifique ainsi que sa vitesse de rotation $\{p, q, r\}$. La force spécifique est la somme des forces extérieures autres que gravitationnelles divisée par la masse m_a .

Il est possible de calculer une estimation de l'accélération $\{a_x, a_y, a_z\}$ et du facteur de charge $\{n_x, n_y, n_z\}$ à partir de la force spécifique. En effet, la relation entre l'accélération et la force spécifique est donnée par l'équation 2.2 et la relation entre le facteur de charge et la force spécifique est donnée par l'équation 2.3 :

$$\vec{a} = \vec{g} + \frac{\vec{F}_{ext}}{m_a} \quad (2.2)$$

$$\vec{n} = -\frac{1}{g} \frac{\vec{F}_{ext}}{m_a} \quad (2.3)$$

Où $\frac{\vec{F}_{ext}}{m_a}$ correspond à la force spécifique, $\vec{n} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$ est le facteur de charge,

$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$ est l'accélération, et $\vec{g} = g \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{pmatrix}$ est la constante gravitationnelle exprimée dans le repère avion.

Par intégration de l'accélération et de la vitesse de rotation, l'IMU est capable d'estimer la vitesse linéaire $\{u, v, w\}$ dans le repère avion, la position $\{x, y, z\}$ et l'orientation $\{\phi, \theta, \psi\}$ du mobile relativement à une situation initiale.

Les accéléromètres et les gyromètres sont des capteurs purement proprioceptifs.

Les informations qu'ils fournissent sont importantes pour la commande du drone. Cependant, la précision des informations obtenues par intégration se dégrade continuellement au cours du temps si elles ne sont pas recalées par d'autres informations, ce qui les rend peu fiables et même inutilisables comme unique référence à long terme.

2.2.2.2 Système de positionnement par satellite

Le GPS est l'un des systèmes de positionnement par satellite. Un tel système permet de déterminer la position absolue $\{x, y, z\}$ d'un objet grâce à des signaux émis par une constellation de satellites. Ce système comporte 24 satellites répartis de façon à pouvoir en observer simultanément au moins de 4 à 8 en tout point du globe. Le GPS mesure aussi la vitesse dans le repère terrestre $\{V_x, V_y, V_z\}$ par effet Doppler. A partir de la vitesse actuelle et de celle mesurée au pas précédant, le GPS est capable de déterminer l'accélération du véhicule $\{a_x, a_y, a_z\}$ exprimée dans le repère terrestre. Néanmoins ces informations sont bruitées car obtenues par dérivation.

2.2.2.3 Baromètre

Le baromètre est utilisé pour calculer l'altitude du véhicule z . L'instrument mesure une différence de pression atmosphérique par rapport à un niveau de référence, généralement celui de la mer. Un modèle atmosphérique est alors utilisé pour déduire l'altitude à partir de la différence de pression.

2.2.2.4 Magnétomètre

Le magnétomètre ou compas permet de calculer l'angle de lacet ψ . Cet instrument donne une direction de référence, généralement le nord magnétique, sur le plan horizontal. Dans le cas de l'Altimum 4, le compas est électronique. L'angle de lacet est mesuré par différence entre la direction de l'engin et la direction de référence.

2.2.2.5 Inclinomètre

Un inclinomètre ou clinomètre est un capteur permettant de mesurer des angles par rapport au plan horizontal. L'inclinomètre est fixé sur un système et permet de déterminer l'angle d'inclinaison par rapport à la ligne horizontale. L'avion jaune possède deux inclinomètres, un mesurant l'angle de tangage θ et un mesurant l'angle de roulis ϕ . L'inclinomètre fournit une relation linéaire entre l'angle à mesurer et le signal de sortie.

2.2.2.6 Sondes d'incidence et de dérapage

Les sondes d'incidence et de dérapage permettent de relever respectivement la mesure d'angle d'incidence local α et d'angle de dérapage local β . Ces angles correspondent à la valeur de l'angle existant entre le vent relatif et l'axe de référence des

sondes. La sonde d'incidence est placée dans le plan $\{x, y\}$ du véhicule et la sonde de dérapage dans le plan $\{x, z\}$ du véhicule.

2.2.2.7 Autres capteurs

D'autres instruments, absents sur l'Altimum 4, peuvent être présents sur d'autres drones :

- les tubes Pitot qui mesurent les pressions statiques et totales ;
- les capteurs de température qui mesurent les températures statiques et totale ;
- les capteurs à ultrasons qui mesurent la distance du véhicule par rapport au sol ;
- les radioaltimètres qui mesurent la hauteur d'un aéronef par rapport au sol ou à la surface de l'eau ;
- les caméras.

Ces capteurs peuvent fournir des mesures d'autres variables d'intérêt. La vitesse air peut être reconstruite grâce aux capteurs de pression et de température, la vitesse calibrée peut être calculée à partir des capteurs de pression. De plus, le traitement d'images fournies par les caméras peut donner des indications sur la vitesse grâce au flot optique et sur la position grâce à la reconnaissance d'amers.

2.2.2.8 Des capteurs complémentaires

Comme le montre le Tableau 2.2, le GPS et l'IMU sont complémentaires en ce qui concerne la mesure de la position. Le GPS permet de recalibrer périodiquement la localisation obtenue par la mesure de l'IMU. En sens contraire, les mesures GPS ne sont obtenues qu'à relativement basse fréquence, rendant nécessaire la fusion avec l'IMU qui permet d'estimer ce qui se passe entre deux points GPS. Ainsi, pour améliorer significativement la précision, une fusion de données est nécessaire entre les mesures du GPS et de l'IMU. Le GPS fournit des positions et vitesses régulières ; l'IMU, qui mesure l'accélération et la rotation, lui est complémentaire. C'est pourquoi l'Altimum 4, comme la majorité des drones, est équipé d'un IMU en complément du GPS.

	GPS	IMU
Précision à long terme	Bonne	Mauvaise
Précision à court terme	Mauvaise	Bonne

Tableau 2.2 – Avantages et désavantages du GPS et de l'IMU pour la mesure de la position

Par ailleurs, l'association du GPS et de l'IMU à d'autres capteurs permet d'avoir des informations plus précises sur le système. Par exemple, l'utilisation d'un baromètre permet d'avoir une redondance au niveau de l'altitude du drone et donc une meilleure précision. Pour utiliser au mieux l'ensemble des mesures, une solution consiste à effectuer une fusion de données avec les différents capteurs.

2.2.3 Modèle de simulation

Un modèle dynamique non linéaire a été développé sous Simulink. Ce modèle comprend différents sous-modèles [Lesprier 2015b] :

- la cinématique du système ;
- la dynamique de propulsion du moteur et de l'hélice ;
- la dynamique des actionneurs des surfaces mobiles ;
- les coefficients aérodynamiques identifiés ;
- les forces et moments aérodynamiques ;
- les effets environnementaux avec les conditions atmosphériques et le vent.

Sur la Figure 2.10, la connexion entre les différents sous-modèles est montrée. Les variables rouges correspondent aux entrées : les actionneurs et le vecteur vent. Les variables bleues correspondent aux sorties mesurées par les capteurs disponibles. Les variables noires ne sont pas mesurées mais sont indispensables pour calculer la dynamique du système ; V_a correspond à la vitesse air, q_d à la pression dynamique, F à la somme des forces et M à la somme des moments. Pour plus d'informations sur le modèle, voir l'Annexe A.

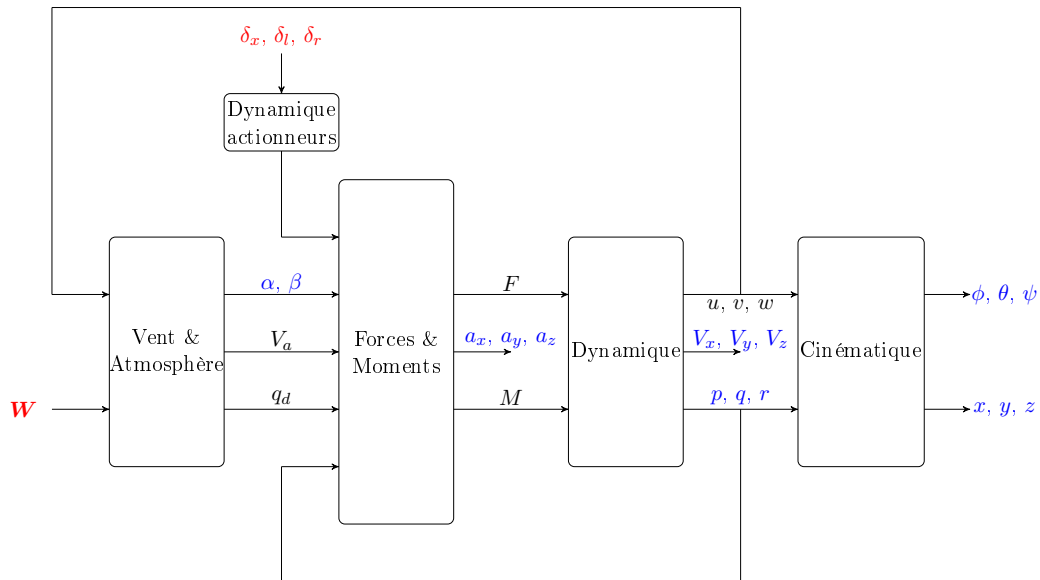


FIGURE 2.10 – Structure du modèle Simulink

Un modèle d'état linéaire peut être obtenu par linéarisation de ce modèle Simulink autour d'un point de vol.

2.2.3.1 Éléments essentiels du modèle complet

On présente ici les éléments principaux concernant le modèle complet ; sa présentation détaillée est reportée en annexe A.

L'état, le vecteur d'entrée et le vecteur de sortie sont les suivants :

$$\mathbf{x} = [u \ v \ w \ p \ q \ r \ x \ y \ z \ \phi \ \theta \ \psi \ E \ \delta_e \ \delta_a]$$

$$\mathbf{u} = [\delta_x \quad \delta_l^1 \quad \delta_l^2 \quad \delta_r^1 \quad \delta_r^2 \quad w_x \quad w_y \quad w_z]$$

$$\mathbf{z} = [p \quad q \quad r \quad a_{sx} \quad a_{sy} \quad a_{sz} \quad \psi \quad \phi \quad \theta \quad \alpha \quad \beta \quad z_B \quad x \quad y \quad z \quad V_x \quad V_y \quad V_z]$$

Où E correspond au moteur, δ_e à la profondeur, δ_a à l'aileron, $\{w_x, w_y, w_z\}$ correspondent aux composantes du vent et $\mathbf{a}_s = -\frac{\vec{F}_{ext}}{m_a}$ correspond au signe près à la force spécifique.

Les sorties ont été ordonnées en fonction de la fréquence d'acquisition des capteurs comme montré sur le Tableau 2.3.

Composant	IMU	Magnétomètre	Inclinomètre/Sonde	Baromètre	GPS
Fréquence	100Hz	20Hz	10Hz	9Hz	4Hz

Tableau 2.3 – Fréquence des différents capteurs

Remarque 2.3

Les entrées et les sorties du modèle de la littérature ne correspondent pas parfaitement à l'Altium 4 réel. En effet, certaines composantes comme V_a , u , v ou w ne sont pas mesurées par les capteurs disponibles, par conséquent ces variables ne sont pas considérées pour la sortie \mathbf{z} . De plus, les redondances des élevons absentes du modèle de départ sont prises en compte.

2.2.3.2 Modèle longitudinal

Comme indiqué plus haut, le développement de la méthode est réalisé avec un modèle de complexité et représentativité croissantes. Dès lors, un modèle uniquement longitudinal est utilisé pour certaines simulations. Il est en effet classique de considérer que pour de petits mouvements autour d'un équilibre, les dynamiques longitudinales et latérales sont découplées. Si on suppose que la dynamique longitudinale peut être découplée de la dynamique latérale, les hypothèses suivantes sont faites pour modéliser la dynamique longitudinale :

- la vitesse latérale est nulle, $v = 0$
- les vitesses angulaires de roulis et de lacet sont nulles, $p = 0$ et $r = 0$
- la position latérale est nulle, $y = 0$
- les angles de cap et de gîte sont nuls, $\psi = 0$ et $\phi = 0$

Les variables définissant l'état latéral (v , p , r , y , ψ et ϕ) ne sont alors pas considérées. L'état, le vecteur d'entrée et le vecteur de sorties deviennent donc les suivants :

$$\mathbf{x} = [u \quad w \quad q \quad x \quad z \quad \theta \quad E \quad \delta_e]$$

$$\mathbf{u} = [\delta_x \quad \delta_l \quad \delta_r \quad w_x \quad w_z]$$

$$\mathbf{z} = [q \quad a_{sx} \quad a_{sz} \quad \theta \quad \alpha \quad z_B \quad x \quad z \quad V_x \quad V_z]$$

Remarque 2.4

Pour le modèle longitudinal, il est supposé que les deux élevons gauches sont contrôlés par le même actionneur. La même hypothèse est faite pour les deux élevons droits.

2.3 Scénarios de panne pour le drone Altimum 4

2.3.1 La phase d'atterrissage

Les phases de décollage et d'atterrissage sont les deux phases les plus critiques de par la proximité du drone vis à vis du sol. Ainsi, de façon à fixer un scénario, la méthode est évaluée en considérant le cas où le drone est en phase d'atterrissage et qu'une panne survient. La Figure 2.11 permet d'illustrer le type de problème rencontré lorsqu'une panne survient si la loi de commande n'a pas été conçue pour traiter ce problème. La loi de commande est généralement conçue pour le cas où le système est nominal. Or si une panne survient, cette loi n'est pas capable de la gérer et la trajectoire ne peut plus être suivie, comme montré sur la Figure 2.11 avec la trajectoire bleue.

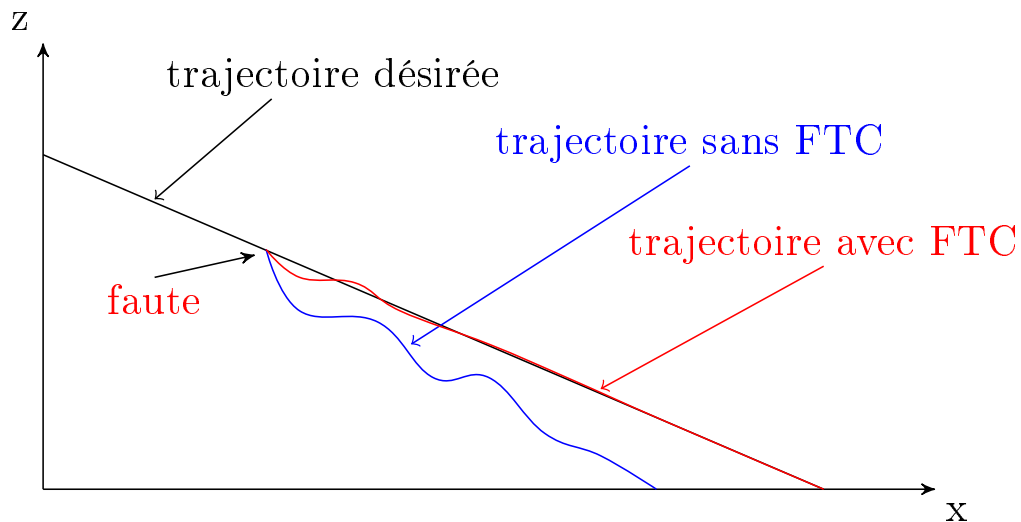


FIGURE 2.11 – Exemple de trajectoires avec et sans FTC pour un scénario d'atterrissage avec l'apparition soudaine d'une faute

Le problème considéré consiste donc à définir une loi de commande FTC permettant d'assurer le suivi d'une trajectoire de référence malgré la survenue de pannes, comme montré sur la Figure 2.11 avec la trajectoire rouge.

Concernant la reconfiguration en cas de panne actionneur, on distingue deux types de problèmes à résoudre ; la reconfiguration en cas de panne actionneurs redondants, comme par exemple pour une panne élevo, et la reconfiguration en cas de panne actionneurs non-redondants, comme par exemple pour une panne moteur. Une panne capteur peut être traitée en utilisant des capteurs redondants s'ils sont disponibles ou en reconstruisant la valeur à partir des mesures des autres capteurs et des connaissances sur le modèle du drone. La panne capteur n'implique pas nécessairement une reconfiguration.

2.3.2 Les types de fautes envisagés

Les fautes du système de commande ne sont pas étudiées. En effet, sans redondance de carte électronique, il est difficile de compenser la faute grâce à l'utilisation d'une méthode FTC qui par conséquent ne permet pas de contribuer à un niveau de sécurité acceptable.

Les fautes des composants ne sont pas non plus étudiées. Elles sont généralement très graves, par conséquent il est difficile de trouver un contrôleur permettant d'avoir des performances acceptables. De plus, il est difficile mathématiquement de modéliser ce type de fautes. Par exemple, si une partie de l'aile du drone est arrachée en vol il est difficile de prévoir la dynamique de vol du drone résultante. Par conséquent, une solution souvent envisagée pour ce type de pannes, est de réaliser un "crash" en douceur par exemple avec l'aide d'un parachute. A cause de la difficulté à gérer ces pannes avec une méthode FTC, les fautes des composants ne sont pas traitées.

Les scénarios se focalisent donc sur les fautes actionneurs et capteurs. D'après la section 2.1.3.1, les actionneurs peuvent tomber en panne de différentes façons. Les pannes oscillatoires étant rares [Goupil 2010], elles ne sont pas étudiées ici. De plus, lorsque l'actionneur est flottant, il est difficile d'avoir un modèle mathématique, ce type de panne n'est donc pas non plus traité. On se focalise sur les fautes de type blocage, biais ou perte d'efficacité pour les actionneurs. Concernant les fautes capteurs, celles-ci sont sujettes à différents types de fautes en fonction de leurs mécanismes. Parmi les types de fautes évoqués à la section 2.1.3.2, il est rare pour un capteur d'être sujet à l'ensemble de ces fautes. Ainsi pour les capteurs, les pannes étudiées sont les suivantes : les biais, les dérives, les blocages et les pertes d'efficacité. Pour des raisons d'observabilité, il est supposé que le GPS ne peut pas tomber en panne (cf. Section 4.2.6.2).

Par ailleurs, on traite le cas où une seule panne arrive à la fois, l'hypothèse étant basée sur le fait que pour des pannes indépendantes le produit des probabilités de pannes associé à la conjonction de celles-ci est bien plus faible que chaque probabilité de panne prise individuellement. De plus, on souhaite traiter les cas où l'on peut finir la mission, ce qui paraît plus difficile avec des cas de pannes multiples. Au vue de la redondance matérielle existante sur le drone utilisé, si plusieurs pannes arrivent en même temps, la solution la plus probable sera le crash. Finalement, les pannes permanentes et intermittentes vont être traitées.

2.4 Conclusion

La méthode que l'on souhaite développer lors de cette thèse doit posséder les caractéristiques suivantes. D'un point de vue technique, on souhaite traiter un nombre varié de pannes uniques ; des pannes actionneurs et capteurs de différents types. Pour les actionneurs, les blocages, les biais et les pertes d'efficacité sont étudiés. Pour les capteurs, les blocages, les biais, les pertes d'efficacité et les dérives sont considérés. De plus, on souhaite aussi bien gérer les pannes permanentes que les pannes intermittentes. D'un point de vue opérationnel, on souhaite développer une méthode

peu coûteuse pour pouvoir l'embarquer en temps réel et qui puisse être aisément certifiable.

Etat de l'art

Sommaire

3.1 La commande tolérante aux fautes	33
3.1.1 Les méthodes passives	33
3.1.2 Les méthodes actives	35
3.2 Modélisation des systèmes et de leurs fautes	38
3.2.1 Systèmes à état continu	38
3.2.2 Systèmes à état discret	40
3.2.3 Systèmes hybrides	41
3.3 Détection et diagnostic de faute	43
3.3.1 Classification des méthodes	44
3.3.2 Génération de résidus	44
3.3.3 Des variantes du filtre de Kalman	48
3.3.4 Estimation d'état sous contraintes	54
3.3.5 Signaux facilitant l'estimation des paramètres	55
3.3.6 Tests statistiques	55
3.3.7 Méthodes discrètes et hybrides	57
3.4 Les méthodes de reconfiguration pour la FTC indirecte . .	60
3.4.1 Les méthodes de restructuration	61
3.4.2 Les méthodes de commutation	65
3.5 Conclusion	66

Résumé

LA commande tolérante aux fautes est un thème qui a suscité de nombreuses études au cours des dernières décennies. Ce chapitre vise à présenter un état de l'art concernant ce domaine. Dans un premier temps, des notions de base seront abordées concernant les différentes approches possibles. Par la suite, l'état de l'art vise en particulier à présenter les différentes méthodes existantes concernant la commande tolérante aux fautes active et indirecte.

3.1 La commande tolérante aux fautes

Les systèmes complexes dépendent de systèmes de commande sophistiqués pour répondre aux exigences de performance et de sécurité. Une commande en boucle fermée classique peut aboutir à des performances non satisfaisantes et même à une instabilité du système en cas de dysfonctionnement des actionneurs, des capteurs ou des autres composantes du système. De nombreux systèmes de commandes sont sujets aux défaillances de composants. Ainsi, en raison de la demande croissante de fiabilité et de sécurité dans les systèmes complexes, des approches FTC voient le jour pour surmonter ce problème. En effet, une méthode FTC vise à maintenir la stabilité du système, ainsi que d'obtenir des performances acceptables en présence de fautes. Au cours de ces dernières années, de nombreuses recherches ont été faites sur ce sujet [Zhang 2008] et notamment pour des drones [Sadeghzadeh 2011].

3.1.1 Les méthodes passives

Généralement, les méthodes FTC sont classées en deux types : on distingue les méthodes passives des méthodes actives [Jiang 2012]. Dans le cas des méthodes actives, le système réagit face aux fautes en reconfigurant sa structure ou les paramètres de sa loi de commande. Dans le cas de la tolérance passive, le système est construit de manière à obtenir la meilleure robustesse face aux fautes, sans détection ou identification de ces fautes ni même adaptation à ces fautes. Il n'y a donc pas de modification de la structure ou des paramètres du contrôleur en réaction à la faute. Dans ce cas, des défaillances possibles sont étudiées en même temps que le cas nominal. L'objectif est de synthétiser un seul contrôleur fixe qui fonctionne le mieux possible malgré l'occurrence de fautes. Ce contrôleur doit permettre de maintenir la stabilité et un certain niveau de performance malgré la présence de fautes.

Les méthodes passives sont généralement basées sur la théorie de la commande robuste. Parmi celle-ci, on trouve différentes approches ; la commande par mode glissant, en anglais *Sliding Mode Control* (SMC) [Merheb 2013] ; la méthode H_∞ [Yang 2001] ; la méthode Linéaire Quadratique (LQ) [Veillette 1995] [Yang 2000].

La technique SMC fait partie des commandes dites à structure variable. Elle s'applique à la fois aux systèmes linéaires et aux systèmes non linéaires. Cette technique a l'avantage d'être robuste aux perturbations externes et aux incertitudes ou variations des paramètres. L'idée de base de la commande par mode glissant est constituée de deux étapes ; amener le système sur un hyperplan de commutation stable, la surface de glissement, et converger sur la surface de glissement vers le point d'équilibre désiré. La technique SMC a la capacité de maintenir une performance désirée en cas de faute sans module FDD, ce qui en fait une méthode très utilisée pour les méthodes FTC passives [Merheb 2013]. Merheb propose deux commandes tolérantes aux fautes basées sur la théorie SMC. La première commande peut tolérer une perte d'efficacité du moteur jusqu'à 50%, alors que la seconde peut tolérer une perte d'efficacité du moteur jusqu'à 55%. Les deux commandes permettent d'améliorer le suivi de trajectoire mais une petite erreur au niveau des angles augmente

avec l'intensité de la panne.

La technique H_∞ fait partie des commandes d'optimisation. Elle prend en compte une définition mathématique des contraintes en ce qui concerne le comportement attendu en boucle fermée. C'est une méthode très utilisée pour la commande robuste, par conséquent elle est très utilisée pour les méthodes passives. Il existe une procédure pour concevoir un contrôleur fiable en cas de pannes actionneurs et capteurs qui peuvent être modélisées par un facteur d'échelle et une perturbation [Yang 2001]. Ce contrôleur garantit une stabilité asymptotique et des performances H_∞ en mode nominal et en mode dégradé. Cependant, en cas de pannes totales d'un capteur, la performance H_∞ ne peut plus être garantie.

La commande LQ fait partie des commandes par retour d'état. Elle garantit une certaine robustesse en boucle fermée. Une procédure pour concevoir un contrôleur tolérant aux pannes d'actionneurs se base sur la méthode LQ [Veillette 1995]. Cette procédure permet bien d'augmenter la fiabilité du contrôleur mais peut détériorer la performance en mode nominal. Un contrôleur basé sur la méthode LQ garantit une stabilité asymptotique malgré la présence de panne actionneurs [Yang 2000]. Cependant, seules les pertes d'efficacité sont considérées.

Les approches actives et passives utilisent les redondances du système mais la différence vient de la manière de les utiliser. La différence entre les deux méthodes est exposée sur la Figure 3.1.

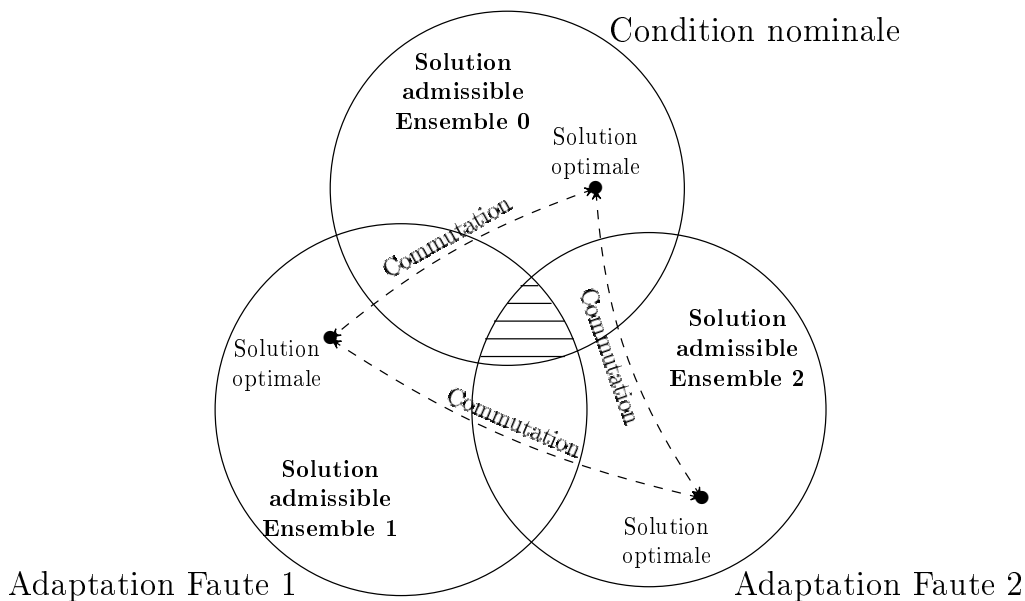


FIGURE 3.1 – Espace des solutions admissibles pour les méthodes passive et active [Jiang 2012]

Pour chaque cas de faute, il est supposé qu'il y a un espace de solution admissible dans lequel il existe un ou des contrôleurs ; c'est à dire un ensemble de contrôleurs présentant des performances acceptables étant donné la faute. Dans le cas de l'approche active, il y a un contrôleur dans chaque espace. Ce contrôleur peut être

construit à partir d'une solution optimale, comme montré sur la Figure 3.1, mais aussi à partir d'une solution sous-optimale. L'existence de ces solutions dépend des contraintes et des techniques d'optimisation utilisées. Un nouveau contrôleur est mis en place par le processus de reconfiguration pour compenser les effets dus à la faute. Dans le cas de l'approche passive, l'objectif est de trouver un contrôleur présent au niveau de l'intersection des différentes espaces. Sur la Figure 3.1, cette zone correspond à la zone rayée. Cependant, si une faute a un ensemble de solutions admissibles ne présentant pas d'intersection avec cette zone rayée, le contrôleur ne permettra plus d'obtenir des performances acceptables. Par ailleurs, dans certains cas, il n'existe pas de zone commune à l'ensemble des espaces, et il est alors impossible de construire un contrôleur unique répondant à l'ensemble des fautes.

Il est en général plus difficile pour une méthode passive d'obtenir de bonnes performances pour chaque type de fautes qu'avec une méthode active. En effet, avec les méthodes passives, il n'y a pas d'adaptation face à la faute mais un unique contrôleur qui fonctionne le mieux possible globalement, ce qui est plus compliqué que d'autoriser une adaptation à la situation. De plus, pour les méthodes passives la solution pour chaque type de faute est généralement sous-optimale alors qu'avec les méthodes actives, la solution optimale peut être obtenue. Par exemple, sur la Figure 3.1 la partie hachurée ne contient pas de solutions optimales, la solution choisie est donc sous-optimale pour chaque type de faute. La stabilité est la principale considération dans une approche passive. Puisqu'une méthode passive n'inclut pas de mécanisme de reconfiguration, il n'y a pas de transitoires dus aux commutations de contrôleur qui peuvent déstabiliser le système, mais uniquement des transitoires dus aux fautes. Par ailleurs, le contrôleur fixe d'une approche passive est conçu pour un certain type de fautes, il n'est donc pas possible de prédire le comportement du système pour un type de fautes non considéré lors de la conception du contrôleur. Le système peut donc seulement tolérer un nombre limité de fautes qui sont supposées connues lors de la conception du contrôleur. Le contrôleur est robuste pour les fautes supposées. Par conséquent, pour traiter de nombreuses pannes, il est un peu risqué de se baser sur une méthode passive [Patton 1997].

“ce qu'il faut retenir”

Étant donné les limitations relatives aux méthodes passives, le travail de cette thèse s'inscrit dans le cadre des méthodes actives. Par conséquent, l'état de l'art relatif aux méthodes passives n'est pas développé de manière détaillée ici et la suite de cette section vise à présenter les méthodes actives.

3.1.2 Les méthodes actives

Différentes classifications existent concernant les méthodes FTC actives. Généralement les méthodes sont classifiées en deux catégories : les méthodes qui comprennent une banque de contrôleurs non-adaptatifs et qui commutent entre eux en fonction de la faute détectée, et les méthodes qui possèdent un seul contrôleur adap-

tatif capable de compenser différents types de fautes. Il est également possible de distinguer les méthodes FTC basées sur une détection explicite des fautes, appelées approches indirectes, des méthodes qui ne passent pas par cette détection, appelées méthodes directes.

3.1.2.1 Les approches adaptatives

Ces approches utilisent un contrôleur dont les gains changent pour assurer des performances acceptables en cas de fautes. Dans le cas de méthodes adaptatives directes les modifications des gains du contrôleur sont calculées à partir de la commande et des mesures sans pour autant passer par une reconstitution des paramètres de fautes [Kim 2003] [Ye 2006] [Xiao-Zheng 2009].

La technique de commande adaptative avec modèle de référence est adaptée aux approches adaptatives directes. Le but est d'avoir la sortie du système proche de la sortie du modèle de référence. Le modèle de référence permet donc de calculer l'erreur entre la sortie réelle du système et la sortie du modèle de référence. Cette erreur est ensuite utilisée par le système de commande pour adapter le gain du contrôleur. Cette technique a été appliquée aux fautes actionneurs [Kim 2003]. Le contrôleur permet de tolérer différents types de pannes actionneurs, celles-ci étant modélisées à l'aide d'une série polynomiale. Les résultats numériques démontrent une adaptabilité rapide malgré la panne actionneur. De plus, la stabilité est prouvée grâce à la théorie de Lyapunov. D'autres techniques utilisent des règles d'adaptation directes. Une méthode développe un contrôleur adaptatif pour lequel les paramètres adaptatifs sont estimés en ligne [Xiao-Zheng 2009]. Le modèle de fautes actionneurs permet de couvrir les blocages, ainsi que les pertes d'efficacité. Le contrôleur développé permet de garantir la stabilité du système. Une autre méthode considère un contrôleur adaptatif qui se base sur l'estimation en ligne d'une faute éventuelle et sur l'ajout d'un nouveau contrôleur pour réduire les effets des fautes [Ye 2006]. Cette méthode permet de continuer à suivre la trajectoire malgré la présence de pertes d'efficacité sur les actionneurs. L'efficacité de la méthode est démontrée sur des simulations.

Contrairement à l'adaptation directe, l'adaptation indirecte inclut un processus de détection et d'isolation de la faute. La détection permet de surveiller l'apparition de fautes, l'isolation permet de localiser la faute et l'identification permet d'estimer l'amplitude de la faute. Dans la littérature, on retrouve ce processus sous le nom de détection et d'isolation de la faute, en anglais *Fault Detection and Isolation* (FDI). Lorsque que la faute est en plus identifiée, le processus se nomme détection et diagnostic de faute, en anglais *Fault Detection and Diagnosis* (FDD). Les gains du contrôleur sont alors calculés en fonction des paramètres de fautes. Sur la Figure 3.2, on peut voir à gauche 3.2a un exemple d'adaptation directe et à droite 3.2b un exemple d'adaptation indirecte.

Le bloc FDD permet également de surveiller l'état du système. Il existe aussi deux types de FDD, les méthodes FDD passives et les méthodes FDD actives [Ducard 2009]. Un FDD passif ne prend aucune action et attend qu'une faute apparaisse, alors qu'un FDD actif teste le système en effectuant des manœuvres tests ou en in-

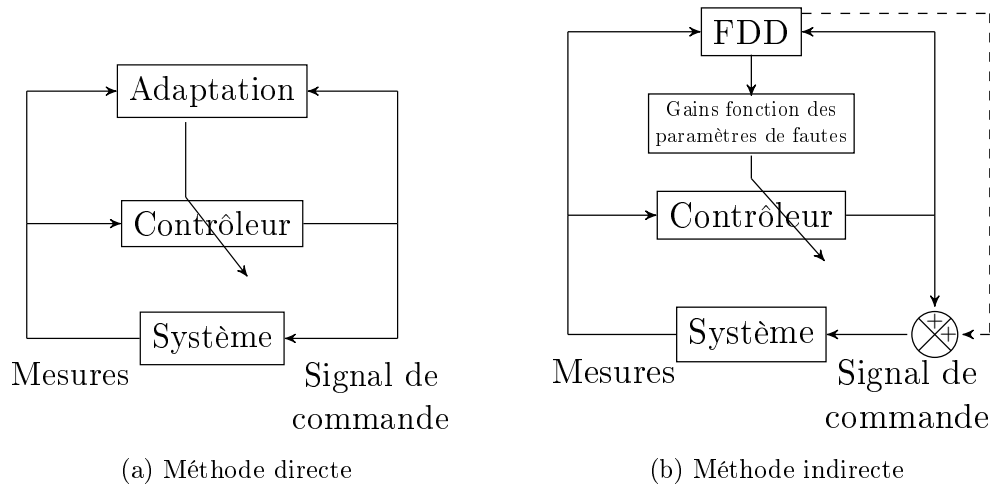


FIGURE 3.2 – Exemples de méthodes d'adaptation : *FDD* : *Détection et diagnostic de faute*, pointillés *FDD* actif

jectant des entrées tests aux commandes d'actionneurs. Une fois la faute détectée, isolée et identifiée, une action appropriée doit être prise pour la sécurité du drone.

3.1.2.2 Les méthodes de commutation

Sur la Figure 3.3, on peut voir des exemples d'approches basées sur la commutation de contrôleurs.

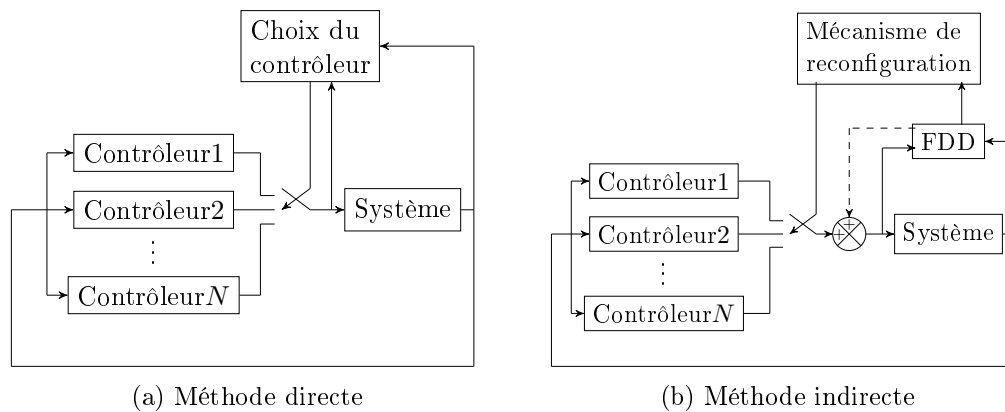


FIGURE 3.3 – Exemples de méthodes de commutation *FDD* : *Détection et diagnostic de faute*, pointillés *FDD* actif

En ce qui concerne les méthodes directes, on a par exemple la falsification de contrôleur qui consiste à calculer pour chaque contrôleur le signal d'erreur qui aurait conduit à la commande observée et à sélectionner le contrôleur correspondant au signal le plus faible [Ingimundarson 2008].

En ce qui concerne les méthodes indirectes, le mécanisme de reconfiguration, la méthode *FDD* et les contrôleurs doivent être compatibles pour pouvoir mener à

bien les tâches de commande. Basée sur cette architecture, les méthodes de reconfiguration doivent prendre en compte les critères suivants pour obtenir de bonnes performances [Zhang 2008] :

- développer une méthode FDD efficace ;
- développer des contrôleurs qui permettent d'obtenir la stabilité du système et des performances acceptables en boucle fermée ;
- développer un système de reconfiguration qui minimise les effets dus aux transitions entre contrôleurs.

Une faute peut rendre le système instable, et par conséquent le temps disponible pour réagir face à la faute est limité. Ce temps correspond au temps critique de réaction. Si le temps de réaction est plus long que le temps critique de réaction, le système atteint un état d'erreur irréparable. Le temps total de réaction d'une méthode active FTC dépend :

- du temps nécessaire au FDD pour obtenir les informations sur la faute et ;
- du temps nécessaire au mécanisme de reconfiguration pour déployer le nouveau système de commande.

Les méthodes de reconfiguration qui possèdent un bloc FDI ou FDD peuvent être référencées sous différents noms dans la littérature, notamment on retrouve souvent les méthodes de détection et d'identification des fautes et de reconfiguration *Fault Detection Identification and Reconfiguration* (FDIR).

“ce qu'il faut retenir”

La présence d'un bloc FDD permet, au-delà de la pure tolérance aux fautes, de présenter des explications sur le comportement du système à un opérateur. Ceci est un avantage certain des méthodes indirectes sur les méthodes directes. La suite de cette analyse de l'état de l'art se focalise donc sur les méthodes indirectes, les modèles qu'elles utilisent, le développement de leur FDD et l'adaptation ou le choix en ligne de leur contrôleur.

3.2 Modélisation des systèmes et de leurs fautes

Le choix de reconstituer les paramètres de fautes conduit naturellement à s'intéresser à la modélisation de celles-ci et à l'intégration de tels modèles avec le modèle du système en absence de fautes.

3.2.1 Systèmes à état continu

La communauté Automatique utilise des modèles continus pour représenter le comportement des systèmes. Elle se base principalement sur la représentation de systèmes continus, qui fait intervenir des variables pouvant prendre toute valeur réelle. L'évolution d'états continus est dirigée par le temps. Suivant les auteurs, le problème est abordé pour une modélisation à temps continu ou pour une modélisation à temps discret.

Les fautes dans un système sont généralement classifiées en trois types : les fautes actionneurs, les fautes capteurs et les fautes composants. Partant d'une modélisation à temps et états continus linéaire et stationnaire (cf. Annexe B), une modélisation permettant de représenter ces différents types de faute peut être proposée selon les équations [Hwang 2010] :

$$\dot{\mathbf{x}}(t) = (A + A_c)\mathbf{x}(t) + (B + B_c)\mathbf{u}(t) + B\mathbf{f}_a(t) + \mathbf{w}(t) \quad (3.1)$$

$$\mathbf{z}(t) = (C + C_c)\mathbf{x}(t) + (D + D_c)\mathbf{u}(t) + \mathbf{f}_s(t) + \mathbf{v}(t) \quad (3.2)$$

Où \mathbf{f}_a représente les fautes actionneurs, \mathbf{f}_s représente les fautes capteurs et A_c , B_c , C_c et D_c représentent les fautes composants.

Les fautes capteurs et actionneurs sont modélisées par des fautes additives (cf. Figure 3.4 (a)) alors que les fautes composants sont modélisées par des fautes multiplicatives (cf. Figure 3.4 (b)), c'est à dire qu'elles sont modélisées comme des changements au niveau des matrices du système [Mahmoud 2003]. Les fautes capteurs et actionneurs interviennent au niveau des sorties et des entrées respectivement mais elles ne modifient pas la dynamique propre du système. A l'inverse, les fautes composants modifient la dynamique du système. Par exemple, si un trou apparaît sur une aile d'un drone à aile fixe, les coefficients aérodynamiques vont être changés et le drone n'aura plus la même dynamique.

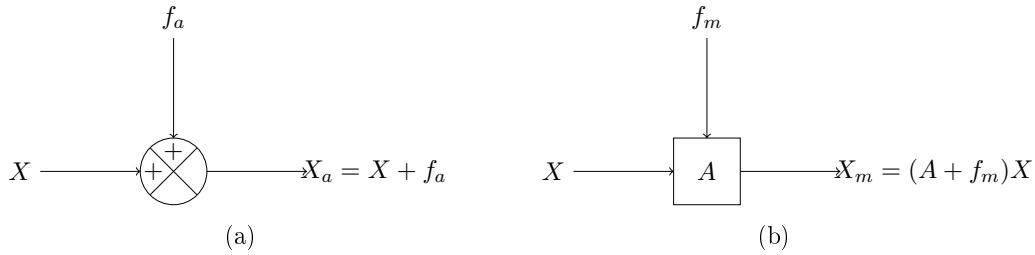


FIGURE 3.4 – (a) Faute additive, (b) Faute multiplicative

Il existe différents types de modélisation pour les pannes actionneurs et capteurs, c'est à dire pour les vecteurs $\mathbf{f}_a(t)$ et $\mathbf{f}_c(t)$.

En temps continu, une faute actionneur peut être modélisée en utilisant une série polynomiale et un processus de bruit aléatoire de la façon suivante [Kim 2003] :

$$u^f(t) = b_0 + b_1 u(t) + b_2 u^2(t) + \dots + n(t) \quad (3.3)$$

Où les termes b_i correspondent aux coefficients de fautes et $n(t)$ correspond au processus de bruit. Le terme b_0 agit comme une perturbation et le terme b_1 détermine l'efficacité de l'actionneur. Les termes d'ordre plus élevés b_j , $j \geq 2$ représentent les effets non linéaires. Cette modélisation est simple et permet de représenter différents types de fautes. Si il n'y a pas de fautes, la situation nominale est obtenue avec $b_1 = 1$ et les autres termes nuls. Si l'actionneur est bloqué, le terme b_0 représente la valeur à laquelle il est bloqué et les autres termes sont nuls. Le cas où l'actionneur est

flottant peut être modélisé en mettant l'ensemble des termes à zéro à l'exception de $n(t)$. Encore, en choisissant $0 < b_1 < 1$ et les autres termes nuls, la modélisation représente une perte d'efficacité.

En temps discret, il est courant de modéliser les pannes actionneurs comme des pertes d'efficacité de $100\gamma\%$ [Wu 1998] [Wu 2000] [Zhang 2002] :

$$u_k^f = (1 - \gamma_k)u_k \quad (3.4)$$

Ou comme des pertes d'efficacité et des blocages [Caliskan 2016] :

$$u_k^f = \delta_k + (1 - \gamma_k)u_k \quad (3.5)$$

En temps continu comme en temps discret, les fautes capteurs sont généralement modélisées avec des biais [Hwang 2010] :

$$z^f(t) = z(t) + \delta(t) \quad (3.6)$$

3.2.2 Systèmes à état discret

A plus haut niveau, le comportement d'un ensemble de variables à dynamiques continues peut être vu comme un système à état discret. La communauté Intelligence Artificielle utilise des modèles discrets ou logiques pour décrire le système. Elle se base principalement sur l'utilisation de variables pouvant prendre un nombre fini de valeurs. Parmi les systèmes discrets, il existe les systèmes à événements discrets, en anglais *Discrete Event Systems* (DES). Les DES ne sont pas régis par le temps, comme pour les systèmes continus, mais par des événements au fil du temps. La transition d'état discret est déclenchée par l'événement. Les événements peuvent être contrôlables ou spontanés, observables ou non observables. De plus, un DES est composé d'un certain nombre de modes de fonctionnement, les états discrets. Plusieurs formalismes existent pour représenter les DES (cf. Annexe C). Les plus connus sont :

- les automates ou machines à état finis, et
- les chaînes de Markov.

Sampath fournit une méthodologie pour construire les DES adaptés au diagnostic de panne [Sampath 1996]. Les fautes dans un modèle discret sont généralement modélisées de deux façons ; par un événement ou par un mode. Les événements non-observables peuvent correspondre à des fautes. La Figure 3.5 montre un exemple de modélisation de fautes sur un DES. Les événements a , b et c correspondent à des événements observables alors que f correspond à un événement non observable, ici une faute. De plus, l'état 4 correspond à un mode de faute étant donné que celui-ci est atteint une fois la faute apparue.

Un DES fournit donc une séquence d'événements permettant de passer d'un mode, nominal ou dégradé, à un autre. Les observations des événements viennent d'un espace discret. Néanmoins, les problèmes complexes ont souvent un espace d'observation plus riche, posant un problème pour les DES classiques. Par exemple,

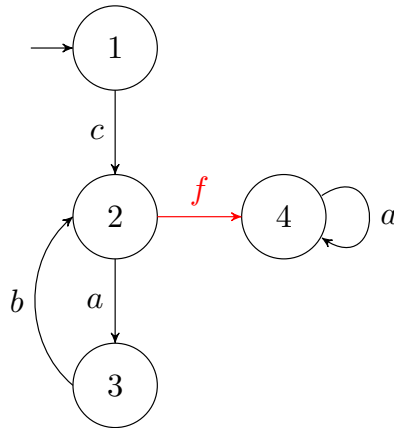


FIGURE 3.5 – Exemple de modélisation de fautes sur un DES

les capteurs fournissent des mesures dans un espace d'observation continu. Ce problème peut être approché en imposant une discrétisation de l'espace d'observation, qui peut être sous-optimal pour la prise de décision. Cependant, il est possible de partitionner l'espace d'observation de façon à séparer uniquement les observations ne donnant pas la même politique [Hoey 2005].

3.2.3 Systèmes hybrides

La modélisation discrète fait abstraction de la dynamique continue du système et des bruits des capteurs. La modélisation hybride permet de pallier ce problème. Un système hybride comprend un processus continu interagissant avec un processus discret. Typiquement, un modèle hybride prend en compte, d'une part, un certain nombre de modes de fonctionnement, c'est-à-dire un mode nominal et des modes dégradés, et d'autre part, à l'intérieur de chaque mode un ensemble de paramètres variables. L'état continu du système évolue selon la dynamique continue associée au mode courant du système. Les changements de modes sont décrits par un système à événements discrets. Un changement de modes est marqué par l'occurrence d'un événement interne ou externe au système, par exemple, l'apparition soudaine d'une faute. Les fautes peuvent donc être modélisées, d'une part, par des paramètres de fautes pour la partie continue, et d'autre part, par des événements et des modes pour la partie discrète.

La modélisation hybride peut être réalisée :

- soit par combinaisons explicites des dynamiques discrètes et continues, c'est-à-dire une continuité par morceaux,
- soit par des dynamiques intrinsèquement hybrides, c'est-à-dire que l'évolution est basée sur des informations continues et discrètes.

3.2.3.1 Systèmes hybrides à états discrets entiers

Un système hybride peut être défini de la façon suivante [Branicky 1998] :

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), q(t), \mathbf{u}(t)) \quad (3.7)$$

$$q(t) = \nu(\mathbf{x}(t), q(t^-), \mathbf{u}(t)) \quad (3.8)$$

où $\mathbf{x}(t) \in \mathbb{R}^n, \mathbf{u}(t) \in \mathbb{R}^l, q \in Q \simeq \{1, \dots, N\}$, $f(., q, .) : \mathbb{R}^n \times \mathbb{R}^l \rightarrow \mathbb{R}^n$ est la dynamique continue du système et est supposée globalement Lipschitz continue et $\nu : \mathbb{R}^n \times Q \times \mathbb{R}^l \rightarrow Q$ est la dynamique discrète de commutation où t^- représente un instant infinitésimalement proche de t et précédent t .

A partir de ce modèle, la stabilité des systèmes hybrides a été étudiée ; au sens de Lyapunov à l'aide de fonctions de Lyapunov multiples et au sens de Lagrange grâce à la théorie des systèmes de fonctions itérées [Branicky 1998]. Différents théorèmes permettent donc de garantir la stabilité des systèmes hybrides. Cependant, l'existence de théorèmes permettant, dans certaines situations, d'obtenir une loi de commande avec de bonnes propriétés de stabilité, de performances et de robustesse ne doit pas masquer la difficulté existante pour en obtenir dans le cas général. Par ailleurs, un point particulièrement délicat vient du basculement entre les différents modes de fonctionnement. En effet, il faut pouvoir gérer la transition pour ne pas rendre le système instable.

Différents formalismes existent pour les systèmes hybrides à états discrets entiers. Les automates hybrides sont souvent utilisés pour modéliser les fautes [Henninger 2000] [Chanthery 2013].

3.2.3.2 Systèmes mixtes à dynamique continue et logique

Pour des systèmes avec des logiques discrètes et des dynamiques continues il est nécessaire d'établir comment construire des propositions logiques à partir des événements possibles concernant la dynamique continue. L'idée est de mettre les propositions logiques sous la forme d'inégalités linéaires comprenant des variables continues, dans le but de les utiliser comme des contraintes dans des problèmes d'optimisation mathématique. Ainsi, à partir de la description standard d'un système, les propositions logiques sont transformées en équations présentant une certaine dynamique, similaire à ce qui est réalisé pour la partie continue, mais avec des contraintes additionnelles issues des variables logiques.

Une approche très utilisée consiste à utiliser une modélisation de système mixte à dynamique continue et logique, en anglais *Mixed Logic Dynamical* (MLD). Cette méthode s'adapte parfaitement au problème car elle intègre des variables logiques, des dynamiques continues et des contraintes de fonctionnement.

La forme générale d'un système MLD est la suivante [Mignone 2002] :

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B_1\mathbf{u}(k) + B_2\boldsymbol{\delta}(k) + B_3\mathbf{z}(k) \quad (3.9)$$

$$\mathbf{y}(k) = C\mathbf{x}(k) + D_1\mathbf{u}(k) + D_2\boldsymbol{\delta}(k) + D_3\mathbf{z}(k) \quad (3.10)$$

$$E_2\boldsymbol{\delta}(k) + E_3\mathbf{z}(k) \leq E_1\mathbf{u}(k) + E_4\mathbf{x}(k) + E_5 \quad (3.11)$$

où $\mathbf{x} \in \mathbb{R}^{n_c} \times \{0, 1\}^{n_l}$, correspond aux parties continues et binaires de l'état, $\mathbf{u} \in \mathbb{R}^{l_c} \times \{0, 1\}^{l_l}$ correspond aux parties continues et binaires de la commande, $\mathbf{y} \in \mathbb{R}^{m_c} \times \{0, 1\}^{m_l}$ correspond aux parties continues et binaires de la sortie, $\boldsymbol{\delta} \in \{0, 1\}^{r_l}$ correspond à la partie binaire de la variable auxiliaire et $\mathbf{z} \in \mathbb{R}^{r_c}$ correspond à la partie continue de la variable auxiliaire.

3.3 Détection et diagnostic de faute

Concernant le diagnostic, on distingue généralement deux approches du diagnostic : l'approche FDI/FDD et l'approche DX, de l'anglais *Diagnosis* [Cordier 2004]. L'approche FDI/FDD utilise les modèles continus alors que l'approche DX utilise les modèles discrets ou logiques. Cette section traite en grande partie le problème de détection et de diagnostic pour des modèles continus, et donc l'approche FDI/FDD. L'approche DX et l'approche hybride font l'objet d'une sous-section 3.3.7.

Le principe de la détection et du diagnostic de faute est montré sur la Figure 3.6. Il repose sur une comparaison entre un modèle simulé et le système réel.

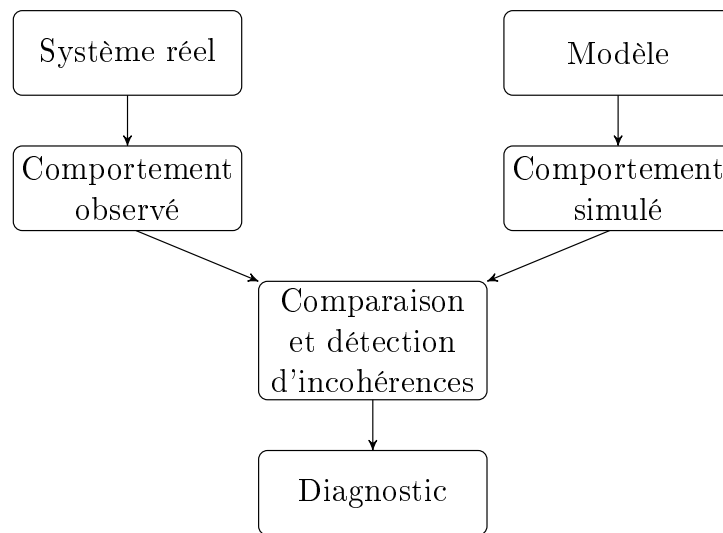


FIGURE 3.6 – Principe du FDD

Le bloc FDI ou FDD est très important pour obtenir de bonnes performances. Un bloc FDD a trois objectifs :

- La détection de la faute, qui indique que quelque chose ne va pas dans le système et qui fournit des informations sur l'occurrence de la faute et l'instant

- d'apparition de la faute ;
- l'isolation de la faute, qui détermine quel composant, capteur ou actionneur est en faute ;
- l'identification de la faute, qui détermine l'intensité et le type de la faute.

La détection de la faute nécessite normalement un modèle représentant le comportement nominal du système, alors que l'isolation et l'identification de la faute nécessite aussi un modèle représentant le comportement du système en cas de pannes.

Un schéma FDD doit être précis et rapide. Cependant, le temps pris par le processus FDD dépend de la nature de la faute. Lorsqu'une faute apparaît, le temps de reconfiguration est limité avant d'arriver à un point de non-retour. Les caractéristiques désirables pour un système FDD sont les suivantes :

- minimiser le temps de détection des situations anormales ;
- ne pas déclencher de fausses alarmes ;
- fournir les informations nécessaires concernant la faute.

Pour remplir l'ensemble de ses objectifs, une méthode FDD peut être passive ou active. Une méthode FDD est dite active lorsqu'elle déclenche une excitation au niveau du système pour pouvoir détecter, isoler et identifier la faute. Une solution consiste à envoyer des signaux excitants au niveau de la commande. Cette méthode est souvent utilisée pour l'identification de systèmes et notamment en aéronautique (cf. Section 3.3.5).

3.3.1 Classification des méthodes

Dans la littérature il existe différentes classifications. Généralement et comme indiqué sur la Figure 3.7, les méthodes FDD sont classifiées en deux types ; les méthodes basées modèles et les méthodes basées données.

Ici, on s'intéresse aux méthodes quantitatives basées sur les modèles. Ces méthodes sont particulièrement adaptées dans le cas où un modèle physique ou un modèle de comportement est disponible. Or, comme discuté au Chapitre 2, un modèle du drone est disponible. Concernant les autres méthodes, le lecteur peut se référer à des articles de synthèse pour plus de détails [Gao 2015a] [Gao 2015b].

À partir d'un modèle continu du système, les méthodes basées modèle comprennent deux étapes : la génération de résidus et l'évaluation de ces résidus à l'aide d'un outil de décision.

3.3.2 Génération de résidus

La génération de résidus permet d'obtenir des informations sur le système et donc de détecter et localiser les fautes. Ces résidus représentent les incohérences entre le comportement observé du système réel et le comportement simulé du modèle mathématique du système. Un résidu est donc un signal temporel qui dépend du vecteur d'entrée $\mathbf{u}(t)$ et des mesures $\mathbf{z}(t)$ [Hwang 2010].

$$\mathbf{r}(t) = g(\mathbf{u}(t), \mathbf{z}(t)) \quad (3.12)$$

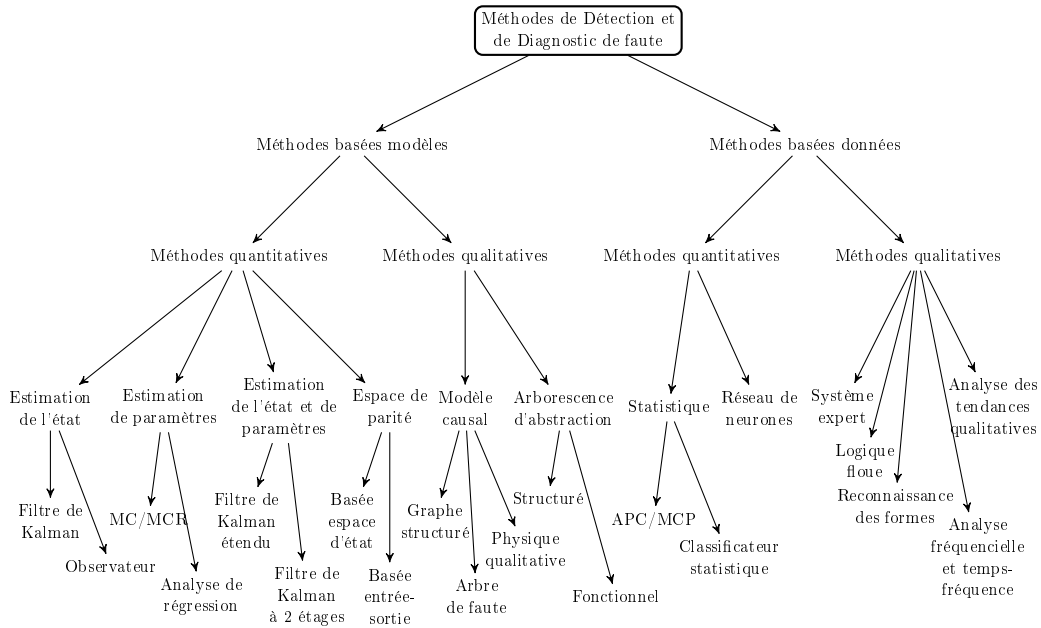


FIGURE 3.7 – Classification des méthodes FDD [Zhang 2008] : Légende : MC/MCR : Moindres Carrés (Rekursifs), ACP : Analyse en composantes principales, MCP : Moindres Carrés Partiels

Un résidu est associé à une différence résultant de la comparaison de mesures de capteurs $z(t)$ à des valeurs calculées analytiquement de la variable considérée dans le système modélisé, par exemple les estimations des mesures $\hat{z}(t)$:

$$\mathbf{r}(t) = \mathbf{z}(t) - \hat{\mathbf{z}}(t) \quad (3.13)$$

L'analyse des résidus permet de détecter la présence ou non de fautes. Dans le cas d'un système idéal sans faute, sans incertitude et sans bruit, ces derniers devraient être nuls et évoluer de manière significative lors de l'apparition d'une faute. En réalité, les résidus sont constitués d'un grand nombre de phénomènes physiques tels que du bruit, des perturbations ou encore, dans certains cas, de fautes. Pour détecter une faute, un résidu doit posséder les caractéristiques suivantes :

- avoir une moyenne nulle en condition nominale, c'est à dire sans faute ;
- être peu sensible aux bruits, aux perturbations ou aux incertitudes du modèle dans le but de maximiser la sensibilité aux fautes.

Pour isoler une faute, les résidus doivent être capables de différencier l'effet de fautes par des composants différents. Dans la littérature, il y a deux approches pour générer des résidus qui facilitent l'isolation des fautes. L'approche du résidu directionnel permet de générer un vecteur de résidus qui se dirige dans une direction spécifique propre à chaque type de fautes. L'autre approche est celle du résidu structuré pour laquelle chaque vecteur de résidus est conçu de façon à être sensible à une faute et peu sensible au reste. Cette méthode est aussi connue sous le nom de

signature de fautes.

À partir du modèle continu du système, deux types de techniques sont employées pour réaliser le diagnostic du système : les techniques à base de redondances analytiques et les techniques à base d'estimateurs (cf. Section 3.3.2.2).

3.3.2.1 La redondance analytique

Les techniques basées sur les relations de redondance analytique sont souvent appliquées dans le cadre du diagnostic de systèmes continus [Cordier 2004]. Une relation de redondance analytique est une équation dans laquelle toutes les variables inconnues ont été remplacées par des expressions ne faisant intervenir que des variables observées. Ces méthodes se basent sur la génération de résidus du modèle pour déterminer les valeurs pertinentes à mesurer. La redondance analytique peut prendre deux formes :

- La redondance directe porte sur les mesures instantanées des capteurs et se base sur des relations algébriques statiques liant les différents signaux.
- La redondance temporelle porte sur les mesures antécédentes des capteurs et des entrées et se base sur l'utilisation de relations dynamiques.

L'approche la plus classique est celle dite de l'espace de parité [Patton 1991]. Le principe de cette approche est de réécrire les équations du modèle analytique du système de manière à obtenir des relations de redondances analytiques [Maquin 1997]. Ces relations ont pour propriété de ne lier que des grandeurs disponibles en ligne. Les méthodes par espace de parité consistent à projeter les estimations des résidus dans un espace binaire afin de reconnaître ce qu'on appelle des signatures des fautes. Les signatures de fautes sont des vecteurs d'indicateurs booléens représentatifs de chaque mode de fonctionnement du système. Les relations de parité peuvent utiliser à la fois des relations de redondance directe et des relations de redondance temporelle.

3.3.2.2 Les méthodes d'estimation

Les techniques à base d'estimateurs consistent à estimer les valeurs des variables d'état à partir du modèle et d'un ensemble d'observations sur le système. Dans ces techniques, on utilise la représentation d'état du système (cf. Annexe B). On utilise la fonction d'évolution du modèle continu pour calculer une estimation de l'état $\hat{\mathbf{x}}$ qui est ensuite corrigée en fonction de la différence entre la sortie estimée $\hat{\mathbf{z}}$, calculée avec la fonction de sortie, et le vecteur des sorties mesurées $\hat{\mathbf{z}}$.

Pour la FTC, il est intéressant d'avoir une estimation de l'état et des paramètres de fautes. Techniquement, cela revient à ajouter les paramètres à estimer dans le vecteur des variables d'état. L'estimation de l'état peut être utilisée pour la conception du contrôleur. De plus, en cas de fautes capteurs, l'estimation de l'état est nécessaire pour pouvoir reconstruire la mesure en faute et pour poursuivre le vol en sécurité. L'estimation des paramètres est aussi importante pour pouvoir compenser les fautes actionneurs. En effet, l'estimation nous permet d'évaluer l'intensité de la

faute et donc de réagir en fonction. Par conséquent, on s'intéresse plus particulièrement aux méthodes d'estimation de l'état et de paramètres de fautes. On retrouve différentes méthodes d'estimation de l'état et des paramètres sur la Figure : 3.8 :

- les variantes des observateurs ;
- les filtres particulières ;
- les variantes du filtre de Kalman ;
- les méthodes d'optimisation.

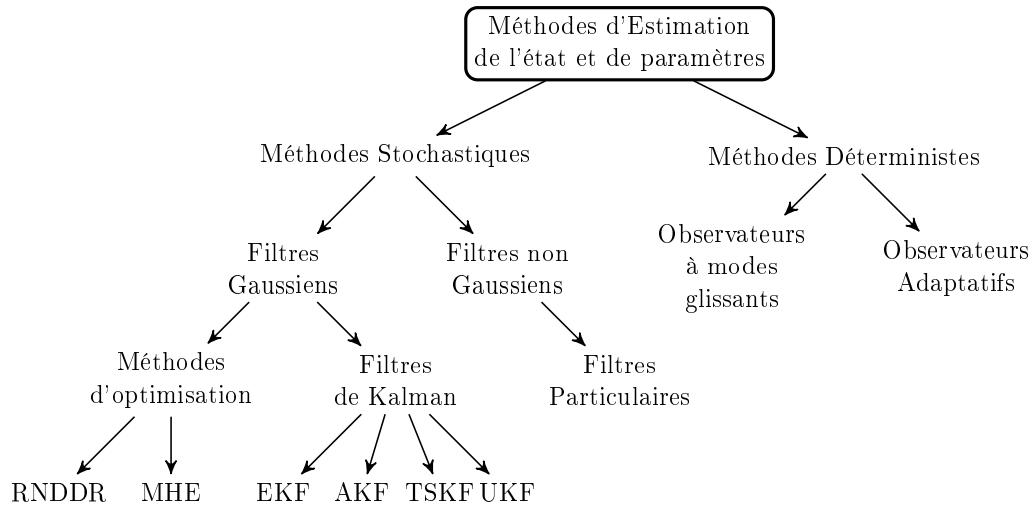


FIGURE 3.8 – Classification des méthodes d'estimation de l'état et de paramètres : *Légende : RNDDR Recursive Nonlinear Dynamic Data Reconciliation, MHE : Moving Horizon Estimation, EKF : Extended Kalman Filter, AKF : Augmented Kalman Filter, TSKF : Two-Stage Kalman Filter, UKF : Unscented Kalman Filter*

Dans les variantes des observateurs, on retrouve les observateurs à modes glissants [Tan 2002] et les observateurs adaptatifs [Wang 1996]. Le principe des approches basées sur les observateurs est d'estimer les variables du système grâce à un observateur, tel que l'observateur de Luenberger. Cette approche fonctionne pour les cas déterministes, contrairement aux autres méthodes (filtre de Kalman, filtres particuliers et méthodes d'optimisation) qui fonctionnent pour les cas stochastique. Les méthodes déterministes traitent le problème d'estimation en considérant uniquement la moyenne des signaux. Les méthodes stochastiques s'intéressent à la description de la densité de probabilité des variables aléatoires. Il est intéressant d'avoir la densité de probabilité, notamment pour avoir des informations sur la précision de l'estimation de l'état et des paramètres de fautes. Pour faire le lien entre les états continus et discrets, ces informations peuvent être utilisées, les méthodes stochastiques sont donc préférées aux méthodes déterministes.

Le filtre particulière a été introduit par Gordon en 1993 [Gordon 1993]. Cette approche s'appuie sur une technique de Monte-Carlo séquentielle. Dans les filtres particuliers, la distribution de croyances se compose d'un ensemble fini de particules, représentant chacune une valeur possible de l'état du système. Durant une étape de

prédiction bayésienne, chaque particule est mise à jour avec la fonction d'évolution du modèle à laquelle est ajouté un bruit aléatoire. Une étape de correction associe alors un poids à chaque particule en fonction de l'erreur entre sa sortie estimée et les observations sur le système, puis utilise ce poids pour ré-échantillonner l'ensemble des particules. Le ré-échantillonnage consiste à sélectionner les particules ayant les poids les plus élevés pour former la nouvelle distribution de particules estimant l'état courant du système. Aucune hypothèse n'est faite sur la densité *a posteriori*. Elle est donc applicable à des modèles non linéaires et des bruits quelconques, contrairement aux filtres de Kalman qui fonctionnent avec des bruits Gaussiens. Néanmoins, cette méthode nécessite une puissance de calcul importante.

Parmi les algorithmes stochastiques appliqués aux systèmes impliquant des variables aléatoires à densités de probabilités gaussiennes, plusieurs découlent du filtre de Kalman [Kalman 1960] [Kalman 1961]. La première variante correspond au filtre de Kalman étendu, en anglais *Extended Kalman Filter* (EKF), qui s'applique aux systèmes non-linéaires. Ces filtres utilisent une étape de prédiction et une étape de correction. La phase de prédiction utilise l'état estimé de l'instant précédent pour produire une estimation de l'état courant. Dans l'étape de mise à jour, les mesures de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise. Pour le lecteur peu familier avec ce sujet, l'Annexe D présente les bases du filtre de Kalman.

Concernant les filtres gaussiens, on retrouve des méthodes d'optimisation. L'algorithme d'estimation à horizon glissant, en anglais *Moving Horizon Estimation* est décrit par Robertson en 1996 [Robertson 1996]. Cet algorithme a été comparé aux filtres de Kalman et aux filtres particulaires [Rawlings 2006]. C'est une méthode efficace pour des problèmes d'estimations en ligne de systèmes linéaires contraints. Cependant, les applications en temps réel sur des systèmes non-linéaires nécessitent une puissance calculatoire importante ce qui limite son utilisation. L'algorithme de rapprochement de données dynamique non-linéaire récursif, en anglais *Recursive Nonlinear Dynamic Data Reconciliation* combine l'efficacité d'un filtre EKF avec un problème d'optimisation [Vachhani 2005]. L'état estimé prédit et la matrice de covariance de l'erreur qui lui est associée sont donnés par les mêmes équations que celles de l'EKF tandis que la mise à jour des états par les mesures s'obtient par le résultat d'un problème d'optimisation.

3.3.3 Des variantes du filtre de Kalman

3.3.3.1 Le filtre de Kalman augmenté

Dans le cadre de l'estimation conjointe d'états et de paramètres, il est possible d'avoir recours à des filtres de Kalman augmentés, en anglais *Augmented Kalman Filter* (AKF) [Keller 1997]. Il est commun d'augmenter le vecteur d'état du problème d'origine par les paramètres. Le filtre estime donc les paramètres additionnels et le vecteur d'état.

Ce filtre a été développé de façon à pouvoir traiter le cas de système dynamique

avec biais. On considère le système d'équations linéaires à temps discret suivant :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + F\mathbf{b}_k + \mathbf{w}_k^x \quad (3.14)$$

$$\mathbf{b}_{k+1} = \mathbf{b}_k + \mathbf{w}_k^b \quad (3.15)$$

$$\mathbf{z}_k = C\mathbf{x}_k + G\mathbf{b}_k + \mathbf{v}_k \quad (3.16)$$

Où $\mathbf{b}_k \in \mathbb{R}^{n_b}$ est le vecteur de biais de magnitude inconnue. Les matrices F et G permettent de prendre en compte le vecteur biais dans la dynamique du système. \mathbf{w}_k^x , \mathbf{w}_k^b et \mathbf{v}_k sont gaussiens, centrés et décorrélés :

$$\mathbb{E} \left\{ \begin{bmatrix} \mathbf{w}_k^x \\ \mathbf{w}_k^b \\ \mathbf{v}_k \end{bmatrix} \begin{bmatrix} \mathbf{w}_j^x & \mathbf{w}_j^b & \mathbf{v}_j \end{bmatrix} \right\} = \begin{bmatrix} Q_k^x & 0 & 0 \\ 0 & Q_k^b & 0 \\ 0 & 0 & R_k \end{bmatrix} \delta_{kj} \quad (3.17)$$

On pose :

$$\bar{A} = \begin{bmatrix} A & F \\ 0 & \mathbb{I}_l \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \text{ et } \bar{C} = [C \ G].$$

Le système augmenté est décrit par les équations suivantes :

$$\mathbf{X}_{k+1} = \bar{A}\mathbf{X}_k + \bar{B}\mathbf{u}_k + \bar{\mathbf{w}}_k \quad (3.18)$$

$$\mathbf{z}_k = \bar{C}\mathbf{X}_k + \mathbf{v}_k \quad (3.19)$$

Où $\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{b}_k \end{bmatrix}$, $\bar{\mathbf{w}}_k = \begin{bmatrix} \mathbf{w}_k^x \\ \mathbf{w}_k^b \end{bmatrix}$ et $\bar{Q}_k = \begin{bmatrix} Q_k^x & 0 \\ 0 & Q_k^b \end{bmatrix} \delta_{kj}$

Il est possible d'appliquer un filtre de Kalman classique à ce système augmenté :

$$\hat{\mathbf{X}}_{k+1|k} = \bar{A}\hat{\mathbf{X}}_{k|k} + \bar{B}\mathbf{u}_k \quad (3.20)$$

$$P_{k+1|k} = \bar{A}P_{k|k}\bar{A}^T + \bar{Q}_k \quad (3.21)$$

$$\boldsymbol{\mu}_{k+1} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k} = \mathbf{z}_{k+1} - \bar{C}\hat{\mathbf{X}}_{k+1|k} \quad (3.22)$$

$$K_{k+1} = P_{k+1|k}\bar{C}^T (\bar{C}P_{k+1|k}\bar{C}^T + R_{k+1})^{-1} \quad (3.23)$$

$$\hat{\mathbf{X}}_{k+1|k+1} = \hat{\mathbf{X}}_{k+1|k} + K_{k+1}\boldsymbol{\mu}_{k+1} \quad (3.24)$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1}\bar{C}P_{k+1|k} \quad (3.25)$$

Remarque 3.1

La matrice de covariance $P_{k|k}$ peut se décomposer en matrice par bloc de la façon suivante :

$$P_{k|k} = \begin{bmatrix} P_{k|k}^x & P_{k|k}^{xb} \\ P_{k|k}^{bx} & P_{k|k}^b \end{bmatrix} \quad (3.26)$$

Où $P_{k|k}^x$ est la matrice de covariance de l'état, $P_{k|k}^b$ est la matrice de covariance des paramètres de fautes, $P_{k|k}^{xb}$ et $P_{k|k}^{bx}$ sont les matrice de covariance du couplage entre l'état et les paramètres de fautes.

Cette méthode est efficace quand le nombre de paramètres est relativement faible

par rapport au nombre de variables d'état du problème d'origine ; dans ce cas, la taille du vecteur d'état n'est pas significativement augmentée. Quand le nombre de paramètres est comparable au nombre de variables d'états d'origine, le temps de calcul du filtre peut devenir excessif.

3.3.3.2 Le filtre de Kalman à deux étages

Le filtre de Kalman à deux étages, en anglais *Two Stage Kalman Filter* (TSKF), a été développé pour éviter le temps de calcul excessif du AKF. Friedland a développé une nouvelle méthode séparant l'estimation du biais inconnu de l'estimation de la dynamique du système [Friedland 1969]. L'idée de ce filtre est de découpler le filtre en deux filtres parallèles, un basé sur l'hypothèse que le biais n'existe pas, un autre qui fournit une estimation du vecteur biais. La sortie du premier filtre est alors corrigée avec la sortie du second filtre. Pour ce filtre le biais est constant mais inconnu. Une extension de cette technique considère le cas d'un biais dont l'évolution est modélisée par un bruit blanc qui est décorrélé du bruit d'état [Ignagni 1990]. Cependant, ce filtre est sous-optimal. Alouani [Alouani 1993] a considéré le cas d'un biais modélisé par un bruit blanc Gaussien corrélé avec le bruit d'état. L'optimalité de ce filtre est obtenue quand une contrainte algébrique est satisfaite sur la corrélation entre le bruit du biais et le bruit d'état. Hsieh [Hsieh 1995] a développé une autre variante de ce filtre contenant un filtre sans biais modifié conduit par une entrée externe pour compenser le bruit biais-état. Cependant l'optimalité de la solution n'est pas prouvée. En 1997, une solution optimale est proposée par Keller et Darouach [Keller 1997] sous la condition spécifique que le bruit d'état et le bruit du biais sont décorrélés.

On considère la même représentation d'état que pour le filtre AKF 3.14, 3.15, 3.16 et on suppose aussi que \mathbf{w}_k^x , \mathbf{w}_k^b et \mathbf{v}_k sont gaussiens, centrés et décorrélés 3.17.

Le filtre optimal de Kalman à deux étages peut être décrit par les équations suivantes :

$$\hat{\mathbf{x}}_{k+1|k} = \tilde{\mathbf{x}}_{k+1|k} + \beta_{k+1|k} \mathbf{b}_{k+1|k} \quad (3.27)$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \tilde{\mathbf{x}}_{k+1|k+1} + \beta_{k+1|k+1} \mathbf{b}_{k+1|k+1} \quad (3.28)$$

$$\mathbf{P}_{k+1|k}^x = \tilde{\mathbf{P}}_{k+1|k}^x + \beta_{k+1|k} \mathbf{P}_{k+1|k}^b \beta_{k+1|k}^T \quad (3.29)$$

$$\mathbf{P}_{k+1|k+1}^x = \tilde{\mathbf{P}}_{k+1|k+1}^x + \beta_{k+1|k+1} \mathbf{P}_{k+1|k+1}^b \beta_{k+1|k+1}^T \quad (3.30)$$

avec un filtre d'état modifié sans biais :

$$\tilde{\mathbf{x}}_{k+1|k+1} = \tilde{\mathbf{x}}_{k+1|k} + \tilde{K}_{k+1}^x \tilde{\boldsymbol{\mu}}_{k+1} \quad (3.31)$$

$$\tilde{P}_{k+1|k+1}^x = (\mathbb{I} - \tilde{K}_{k+1}^x C) \tilde{P}_{k+1|k}^x \quad (3.32)$$

$$\tilde{K}_{k+1}^x = \tilde{P}_{k+1|k}^x C^T \tilde{G}_{k+1}^{-1} \quad (3.33)$$

$$\tilde{\boldsymbol{\mu}}_{k+1} = \mathbf{z}_{k+1} - C \tilde{\mathbf{x}}_{k+1|k} \quad (3.34)$$

$$\tilde{G}_{k+1} = C \tilde{P}_{k+1|k}^x C^T + R_{k+1} \quad (3.35)$$

$$\tilde{\mathbf{x}}_{k+1|k} = A \tilde{\mathbf{x}}_{k|k} + B \mathbf{u}_k + r_k \mathbf{b}_{k|k} - \beta_{k+1|k} \mathbf{b}_{k|k} \quad (3.36)$$

$$\tilde{P}_{k+1|k}^x = A \tilde{P}_{k|k}^x A^T + Q_k^x + r_k P_{k|k}^b r_k^T - \beta_{k+1|k} P_{k+1|k}^b \beta_{k+1|k}^T \quad (3.37)$$

et un filtre optimal du biais :

$$\mathbf{b}_{k+1|k} = \mathbf{b}_{k|k} \quad (3.38)$$

$$P_{k+1|k}^b = P_{k|k}^b + Q_k^b \quad (3.39)$$

$$\mathbf{b}_{k+1|k+1} = \mathbf{b}_{k|k} + K_{k+1}^b (\tilde{\boldsymbol{\mu}}_{k+1} - H_{k+1|k} \mathbf{b}_{k|k}) \quad (3.40)$$

$$K_{k+1}^b = P_{k+1|k}^b H_{k+1|k}^T (H_{k+1|k} P_{k+1|k}^b H_{k+1|k}^T + \tilde{G}_{k+1})^{-1} \quad (3.41)$$

$$P_{k+1|k+1}^b = (\mathbb{I} - K_{k+1}^b H_{k+1|k}) P_{k+1|k}^b \quad (3.42)$$

et finalement des équations de couplage :

$$H_{k+1|k} = G + C \beta_{k+1|k} \quad (3.43)$$

$$\beta_{k+1|k+1} = \beta_{k+1|k} - \tilde{K}_{k+1}^x H_{k+1|k} \quad (3.44)$$

$$r_k = A \beta_{k|k} + F \quad (3.45)$$

$$\beta_{k+1|k} = r_k P_{k|k}^b (P_{k+1|k}^b)^{-1} \quad (3.46)$$

Le terme $\beta_{k+1|k}$ permet de prendre en compte le couplage entre l'état et les paramètres de fautes. En effet, $\beta_{k+1|k}$ est définie dans la preuve [Keller 1997] par l'équation suivante :

$$\beta_{k+1|k} = P_{k+1|k}^{xb} P_{k+1|k}^b^{-1} \quad (3.47)$$

Remarque 3.2

Pour pouvoir traiter les non-linéarités, ce filtre a été étendu sous le nom de filtre de Kalman à deux étages étendu, en anglais *Two Stage Extended Kalman Filter (TSEKF)* [Kim 2009]. Comme dans le cas de l'EKF, le filtrage est sous optimal.

3.3.3.3 Le filtre TSKF adaptatif

Les précédents filtres modélisent les biais par des marches aléatoires. Par conséquent, les filtres ne sont pas conçus pour gérer des biais sujets à des changements abruptes ce qui est souvent le cas des fautes actionneurs ou capteurs. Pour y remédier, on peut avoir recours à des méthodes adaptatives comme dans le cas du

filtre adaptatif TSKF avec facteur d'oubli [Wu 1998]. Une mesure complémentaire est prise en compte pour modifier le TSKF de façon à ce que l'estimé devienne plus sensible aux variations brutales. L'idée est de permettre à l'algorithme récursif d'oublier l'information passée de façon à ce que le filtre soit plus apte à reconnaître un changement dans le système. Pour cela, l'algorithme introduit un facteur d'oubli λ au niveau de la prédiction de la covariance du biais :

$$P_{k+1|k}^b = \frac{P_{k|k}^b}{\lambda} + Q_k^b, \quad 0 < \lambda \leq 1 \quad (3.48)$$

Dans ce cas, l'ancienne information est oubliée uniformément dans le temps et dans l'espace puisque la division par λ revient à augmenter P^b . La prédiction de la covariance ne doit pas être trop grande ou trop petite. La technique utilisée est de sélectionner des facteurs d'oubli qui forcent la covariance à rester entre certaines bornes :

$$\sigma_{min}\mathbb{I} \leq P_{k+1|k}^b \leq \sigma_{max}\mathbb{I}, \quad 0 < \sigma_{min} < \sigma_{max} < \infty \quad (3.49)$$

Soit le développement dyadique de $P_{k|k}^b$ donné par :

$$P_{k|k}^b = \sum_{i=1}^l \alpha_{k|k}^i e^i (e^i)^T \quad (3.50)$$

Où $\alpha_{k|k}^1, \dots, \alpha_{k|k}^l$ sont les valeurs propres de $P_{k|k}^b$ et e_k^1, \dots, e_k^l sont les vecteurs propres correspondants avec $\|e_k^1\| = \dots = \|e_k^l\| = 1$.

La prédiction de la covariance $P_{k+1|k}^b$ peut être remplacée par l'équation suivante dans le filtre TSKF classique :

$$P_{k+1|k}^b = \sum_{i=1}^l \frac{\alpha_{k|k}^i}{\lambda_k^i} e^i (e^i)^T + Q_k^b, \quad 0 < \lambda_k^i \leq 1 \quad (3.51)$$

Le facteur d'oubli λ_k^i est choisi de la façon suivante :

$$\lambda_k^i = \begin{cases} 1, & \alpha_{k|k}^i > \alpha_{max} \\ \alpha_{k|k}^i (\alpha_{min} + \frac{\alpha_{max} - \alpha_{min}}{\alpha_{max}} \alpha_{k|k}^i)^{-1}, & \alpha_{k|k}^i \leq \alpha_{max} \end{cases} \quad (3.52)$$

En insérant des facteurs d'oubli, l'algorithme perd son optimalité mais il est maintenant plus sensible aux changements brusques. Cette méthode est efficace et a été appliquée dans différents cas ; pour des pertes d'efficacité actionneurs [Wu 1998] [Wu 2000] [Zhang 2002], pour des pertes d'efficacité et des blocages actionneurs [Caliskan 2016] et pour des fautes capteurs [Zhang 1999b] [Kim 2009] [Wu 2006].

3.3.3.4 Les autres variantes

D'autres approches ont été développées de façon à être sensible aux pannes. L'approche multi-modèle nommée l'estimation adaptative multi-modèle, en anglais *Multiple Model Adaptive Estimation*, (MMAE) s'est imposée comme une solution efficace pour l'estimation. L'algorithme MMAE a été présenté en 1976 par Athans [Athans 1976]. Cet algorithme utilise plusieurs filtres de type filtre de Kalman en parallèle. Chaque filtre fait une hypothèse concernant le modèle du système pour estimer l'état et les paramètres inconnus. Les sorties de chaque filtre sont ensuite combinées selon les vraisemblances de leurs hypothèses respectives - déterminées à partir des résidus des filtres - pour fournir une seule estimation de l'état et des paramètres inconnus. Cette approche utilise des sous-modèles homogènes, c'est-à-dire qu'ils partagent la même structure et le même espace d'état. La Figure 3.9 illustre le fonctionnement de l'algorithme MMAE. Les éléments r_i , P_i , \hat{x}_i et μ_i correspondent respectivement aux résidus, aux matrices de covariance de l'erreur d'estimation d'état, aux estimations de l'état et aux poids associés aux états x_i .

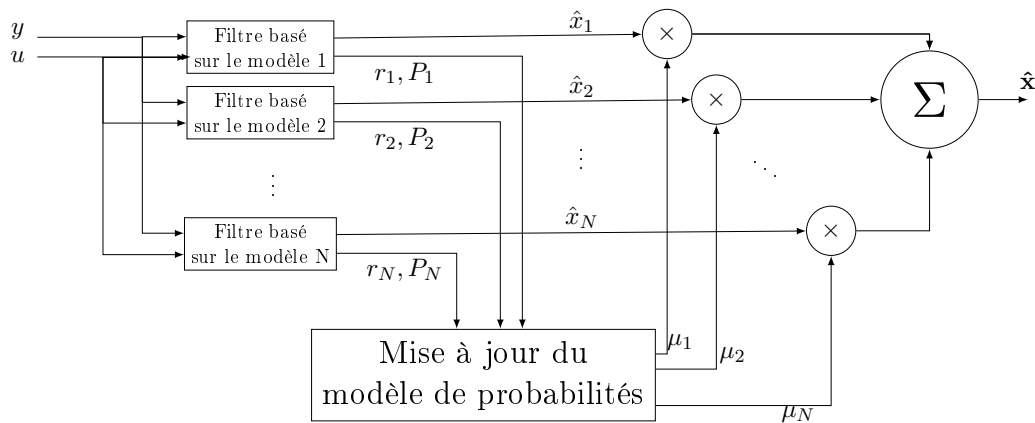


FIGURE 3.9 – Schéma de principe de l'algorithme MMAE

De nombreux travaux utilisent cette méthode pour la détection de fautes sur les avions [Menke 1995] [Maybeck 1999] [Ducard 2008]. Néanmoins, les approches multi-modèle nécessitent une puissance calculatoire importante dû à l'implémentation de plusieurs filtres en parallèle, avec parfois la nécessité d'un grand nombre de modèles pour décrire l'ensemble des configurations possibles. Ainsi, cette approche peut être utilisée uniquement pour certaines applications et elle est conçue pour des pannes capteurs et actionneurs connues.

Une autre approche consiste à adapter les matrices de bruit Q et R en les multipliant par une variable dépendant du temps. Généralement, une adaptation de la matrice Q permet de détecter des fautes actionneurs alors que l'adaptation de la matrice R permet de détecter des fautes capteurs. En effet, une faute actionneur impacte l'équation d'état alors qu'une panne capteur impacte l'équation de mesure. Il est possible d'adapter la matrice de bruit en utilisant un seul facteur multiplicatif [Hu 2003]. Hu *et al* ont développé deux algorithmes adaptatifs permettant

à un filtre de Kalman d'être plus robuste aux changements soudains du mouvement du véhicule et aux erreurs de mesures. Néanmoins pour les systèmes complexes il est préférable d'utiliser plusieurs facteurs multiplicatifs, l'effet d'une faute n'étant pas le même sur chaque état [Geng 2008]. Il faut donc examiner attentivement l'effet de chaque faute sur chaque état pour construire les facteurs multiplicatifs de façon à pondérer l'adaptation pour chaque état. Ces deux approches considèrent uniquement l'adaptation de la matrice de bruit Q et l'adaptation de R n'est pas étudiée [Hu 2003] [Geng 2008]. A l'inverse, il est possible d'étudier l'adaptation de la matrice R pour robustifier le filtre de Kalman face aux erreurs de mesures [Hajiyev 2007]. L'algorithme développé fournit des résultats précis quand les mesures sont saines et quand celles-ci sont en panne. Il est aussi possible d'adapter les matrices Q et R pour détecter les fautes capteurs et actionneurs [Hajiyev 2013]. L'adaptation de la matrice R dépend d'une matrice de facteurs et rend le filtre plus robuste face aux fautes capteurs. De façon similaire à l'adaptation de la matrice R , l'adaptation de la matrice Q rend le filtre plus robuste face aux fautes actionneurs.

Finalement, une approche utilise un filtre de Kalman à trois étapes [Gillijns 2007]. Contrairement au filtre TSKF, l'état n'est pas augmenté avec le biais et aucune hypothèse n'est faite sur la dynamique du biais, celle-ci étant en général inconnue. L'état étant supposé non biaisé, si une erreur apparaît au niveau de l'innovation, le biais en est la cause. L'estimation du biais se fait donc à partir de l'innovation. Cette approche a été adaptée pour des systèmes non linéaires et appliquée à l'estimation de fautes capteurs [Lu 2015].

3.3.4 Estimation d'état sous contraintes

Soit un système non linéaire à temps discret (cf. Annexe B) soumis à la contrainte suivante :

$$D_k \mathbf{x}_k \geq \mathbf{d}_k \quad (3.53)$$

Une manière de résoudre le problème de contraintes d'inégalités est d'utiliser la technique d'activation de la limite contrainte [Sircoulomb 2008]. Cela consiste à tester à chaque instant k les inégalités. Pour la i^{me} inégalité, deux scénarios sont envisageables :

- l'inégalité est vérifiée donc la contrainte n'est pas prise en compte ;
- l'inégalité n'est pas vérifiée, on applique une contrainte d'égalité à la borne :

$$D_{i,k} \mathbf{x}_k = d_{i,k} ;$$

L'approche par projection permet d'estimer de manière optimale sous contraintes sur l'état [Sircoulomb 2008]. Le principe consiste à résoudre le problème d'optimisation sous contrainte suivant :

$$\min_{\mathbf{x}_k^c} ((\mathbf{x}_k^c - \mathbf{x}_k^{nc})^T P_k^{nc-1} (\mathbf{x}_k^c - \mathbf{x}_k^{nc})) \quad s.t. \quad D_k \mathbf{x}_k^c = \mathbf{d}_k \quad (3.54)$$

La solution de 3.54 est donnée par les équations suivantes :

$$\mathbf{x}_k^c = \mathbf{x}_k^{nc} + L_k(\mathbf{d}_k - D_k \mathbf{x}_k^{nc}) \quad (3.55)$$

$$P_k^c = (\mathbb{I} - L_k D_k) P_k^{nc} (\mathbb{I} - L_k D_k)^T \quad (3.56)$$

$$L_k = W_k D_k^T (D_k W_k D_k^T)^{-1} \quad (3.57)$$

Si $W_k = \mathbb{I}$, le filtre contraint est sous optimal au sens du minimum de variance. Si $W_k = P_k^{nc}$ alors le filtre contraint est à variance minimale.

3.3.5 Signaux facilitant l'estimation des paramètres

Dans l'aéronautique, il est classique d'envoyer un signal doublet pour exciter le système [Raol 2004]. C'est un signal correspondant à une série de paliers alternatifs (cf. Figure 3.10). Ce signal excite une large bande de fréquence et est utilisé pour exciter le mode longitudinal. Un autre signal classique pour exciter le système est le signal 3211 [Raol 2004]. C'est un signal correspondant à une série de paliers alternatifs dont les durées satisfont le ratio 3,2,1,1 (cf. Figure 3.10). Il est préféré aux doublets car il excite un spectre dynamique plus large dans une courte période tout en étant facile à générer.

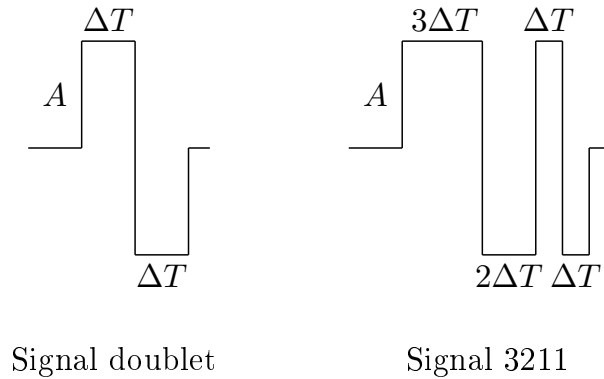


FIGURE 3.10 – Signaux excitants

3.3.6 Tests statistiques

Une fois les résidus générés, il faut déterminer si une faute est apparue et, si c'est le cas, la localiser. Pour cela, des tests statistiques sont généralement effectués sur les résidus. Le problème peut être formulé comme un processus permettant de tester un changement au niveau d'un paramètre η du système. Une décision est prise à chaque pas de temps sur les deux hypothèses suivantes [Hwang 2010] :

$$H_0 : \eta = \eta_0 \quad (3.58)$$

$$H_1 : \eta = \eta_1 \neq \eta_0 \quad (3.59)$$

Où l'hypothèse nulle H_0 considère le paramètre η comme nominal, et l'hypothèse alternative H_1 considère le paramètre η comme en faute.

En réalité, quatre situations peuvent être considérées :

- l'hypothèse H_0 est vraie et H_0 a été choisie ;
- l'hypothèse H_1 est vraie et H_1 a été choisie ;
- l'hypothèse H_0 est vraie mais H_1 a été choisie, ce qui correspond à une fausse alarme ;
- l'hypothèse H_1 est vraie mais H_0 a été choisie, ce qui correspond à une panne non détectée.

De nombreuses approches statistiques existent pour la détection de fautes [Basseville 1993] [Gustafsson 2000]. La façon la plus simple est de déterminer un seuil sur les résidus à partir duquel il est décidé qu'une faute est apparue. De cette manière, seules les fautes générant une amplitude supérieure à un certain niveau seront détectées. Les seuils doivent être dimensionnés de façon à avoir une faible probabilité de fausse alarme et une bonne probabilité de détection. Des méthodes plus sophistiquées existent :

- la méthode du seuil adaptatif ;
- le test de Student (cf. Annexe E) ;
- le test du χ^2 (cf. Annexe E) ;
- le test séquentiel du rapport de probabilité, *Sequential Probability Ratio Test* (SPRT) ;
- l'algorithme de somme cumulative, *CUmulative SUM* (CUSUM) ;
- le rapport de vraisemblance généralisée, *Generalized Likelihood Ratio* (GLR).

3.3.6.1 Le test SPRT et l'algorithme CUSUM

Le SPRT est un test d'hypothèse séquentiel développé par Wald [Wald 1945].

Soit une séquence d'observations z_0, z_1, \dots, z_k . Le test SPRT calcule la somme cumulative du logarithme du rapport de vraisemblance suivant :

$$S_k = S_{k-1} + \log \frac{p(z_0, \dots, z_k | H_1)}{p(z_0, \dots, z_k | H_0)} \quad (3.60)$$

Où $p(z_0, \dots, z_k | H)$ est la densité de probabilité conditionnelle de l'observation, qui dépend du paramètre η .

La règle d'interruption est la suivante :

- si $S_k \geq b$, H_1 est accepté ;
- si $S_k \leq a$, H_0 est accepté ;
- sinon le test continu.

Les paramètres a et b dépendent des probabilités de fausse alarme et de pannes non détectées.

Page est le premier à avoir introduit l'algorithme CUSUM [Page 1954]. L'algorithme CUSUM est équivalent à un test SPRT répété. Le test est recommencé tant

que la décision est prise en faveur de H_0 . La règle d'interruption devient la suivante :

$$S_k = \max(0, S_{k-1} + \log \frac{p(z_0, \dots, z_k | H_1)}{p(z_0, \dots, z_k | H_0)}) \quad (3.61)$$

La décision est prise en faveur de H_1 quand $S_k \geq b$.

Plusieurs variantes de cet algorithme ont été développées en fonction du nombres d'hypothèses et des contraintes d'opération en ligne ou hors ligne [Basseville 1993] [Gustafsson 2000].

Contrairement à l'approche du seuil, le test SPRT et l'algorithme CUSUM permettent de détecter des changements brutaux, tel que des biais, mais aussi des petits changements, tel que des dérives. En effet, il est difficile de trouver un seuil unique permettant de détecter des petits changements mais aussi d'éviter les fausses alarmes.

3.3.6.2 L'algorithme GLR

L'algorithme GLR a été introduit par Willsky et Jones dans les années 70 [Willsky 1976]. Il permet de tester les hypothèses 3.58 et 3.59 quand le paramètre η_1 est inconnu. Dans ce cas, il y a deux inconnues indépendantes ; le temps d'apparition et la valeur du paramètre η_1 . L'approche standard est d'utiliser le maximum de vraisemblance généralisé de ces deux paramètres, et donc de calculer une double maximisation :

$$S_k = \max_{0 \leq j \leq k} \max_{\eta_1} S_j^k(\eta_1) \quad (3.62)$$

avec $S_j^k(\eta_1) = \sum_{i=j}^k \log \frac{p(z_0, \dots, z_i | H_1)}{p(z_0, \dots, z_i | H_0)}$ est le rapport de vraisemblance pour les observations du temps j au temps k .

La règle d'interruption est la suivante :

- si $S_k \geq b$, H_1 est accepté ;
- si $S_k \leq b$, H_0 est accepté ;

3.3.7 Méthodes discrètes et hybrides

La plupart des techniques FDD fonctionnent avec des systèmes continus, néanmoins certains travaux portent sur les techniques FDD pour des systèmes qui contiennent des états discrets uniquement ou qui contiennent des états discrets avec des états continus, c'est-à-dire des systèmes hybrides.

3.3.7.1 Les méthodes discrètes

Le diagnostic de faute à partir des DES est un domaine de recherche actif. Différentes méthodes ont été développées au cours de ces dernières années [Zaytoon 2013]. Sampath fournit un concept de base en proposant une approche DES au problème de diagnostic de panne [Sampath 1995]. L'approche se base sur le modèle d'un automate classique où les fautes sont prises en compte au niveau des modes et au niveau des événements non observables. Le problème de diagnostic consiste

à déterminer quelle faute, s'il y en a, explique la séquence d'événements observés en se basant sur le modèle du système. Il faut donc construire un observateur qui considère les différentes séquences possibles à partir des événements observables. Par exemple, sur la Figure 3.11, (b) représente l'observateur du système (a). a , b et c représentent des événements observables et f et u_o représentent des événements non observables. L'observateur est initialisé à l'état $\{1\}$ puisque 1 correspond à l'état initial de l'automate. L'événement c active l'état 2 qui n'est pas différentiable de l'état 4 à cause de l'événement faute inobservable f . L'observateur se déplace donc à l'état estimé $\{2, 4\}$. Ici, le seul événement observable est a qui active l'état 3, si l'état précédent était 2; l'état 5, si l'état précédent était 4, qui n'est pas différentiable de l'état 6 à cause de l'événement inobservable u_o . L'événement a mène donc à l'état $\{3, 5, 6\}$ où le prochain événement observable est a ou b . On construit ainsi de suite l'observateur.

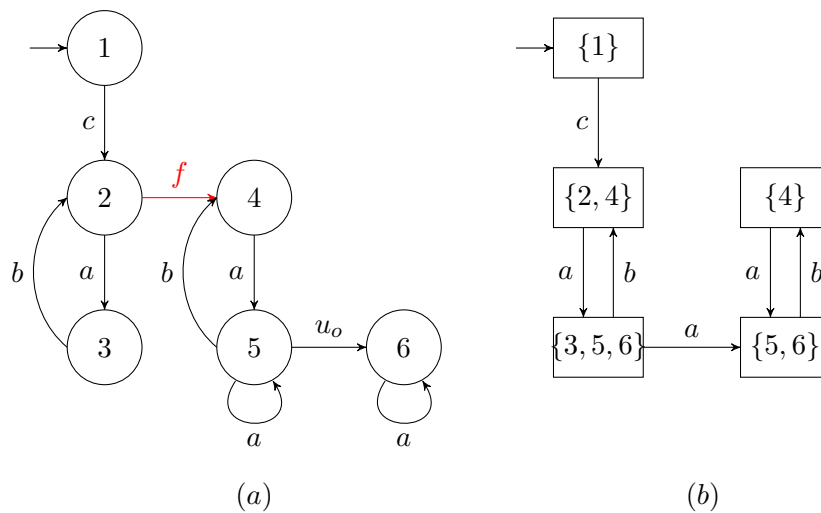


FIGURE 3.11 – (a) Automate à 6 états, (b) Son Observateur [Zaytoon 2013]

Cette méthodologie est appropriée pour les fautes capteurs et actionneurs. Néanmoins, la construction d'observateur peut entraîner à un grand nombre d'états en fonction de la complexité du système. De plus, établir la séquence permettant de diagnostiquer la faute n'est pas toujours une chose aisée. A partir de l'observateur de la Figure 3.11, il est facile de détecter l'occurrence de l'événement f si la séquence est c , a puis de nouveau a . Cependant si la séquence est c , puis a , puis b , l'observateur ne peut pas conclure que l'événement f est apparu car cette séquence peut correspondre à la transition entre 2 et 3, qui n'implique pas l'événement f ou bien à la transition entre 4 et 5 qui implique l'événement f en amont. Le diagnostic de faute n'est donc pas toujours évident.

Par ailleurs, les systèmes discrets permettent généralement de traiter les fautes drastiques, telles que les blocages d'actionneurs ou les pertes totales de capteurs. Ils ne contiennent pas assez d'informations pour traiter les fautes partielles, telles que les dérives ou les pertes d'efficacité. Ces informations sont en général détaillées

dans des variables continues. Pour pouvoir traiter les deux problèmes, les systèmes hybrides ont vu le jour.

3.3.7.2 Les méthodes hybrides

Avec l'utilisation de systèmes hybrides pour l'estimation et la détection de panne, les modes de fautes du système sont modélisés comme les états discrets du système hybride.

Les approches multi-modèles ont fait leur apparition dans de nombreux domaines et se sont imposées comme des solutions efficaces pour la détection. On retrouve notamment l'interaction multi-modèle, en anglais *Interacting Multiple Model*, (IMM) [Zhang 1998] [Zhang 1999a] [Zhang 2001]. Le système est représenté comme un système hybride stochastique ; le processus discret représente les changements paramétriques et/ou structurels possibles dus aux fautes, et le processus continu représente l'état du système. L'approche IMM utilise donc des sous-modèles hétérogènes, c'est-à-dire ayant des structures et des espaces d'états différents. Cette approche considère différents modes de fonctionnements, un fonctionnement nominal et un ou plusieurs fonctionnements dégradés. Ces modes, c'est-à-dire le processus discret, sont modélisés par une chaîne de Markov, par conséquent la probabilité de transition entre chaque mode est explicitement connue. Le changement abrupt dû à la faute est explicitement modélisé par la commutation d'un mode à un autre. La Figure 3.12 illustre le principe de l'algorithme IMM. Les éléments r_i , P_i , \hat{x}_i et μ_i correspondent respectivement aux résidus, aux matrices de covariance de l'erreur d'estimation d'état, aux estimations de l'état et aux poids associés aux états x_i . Les notations P_i^0 et x_i^0 correspondent aux matrices de covariance et aux estimations de l'état après l'étape d'interaction, qui consiste à corriger les états obtenus au pas de temps k à partir des pondérations μ_i .

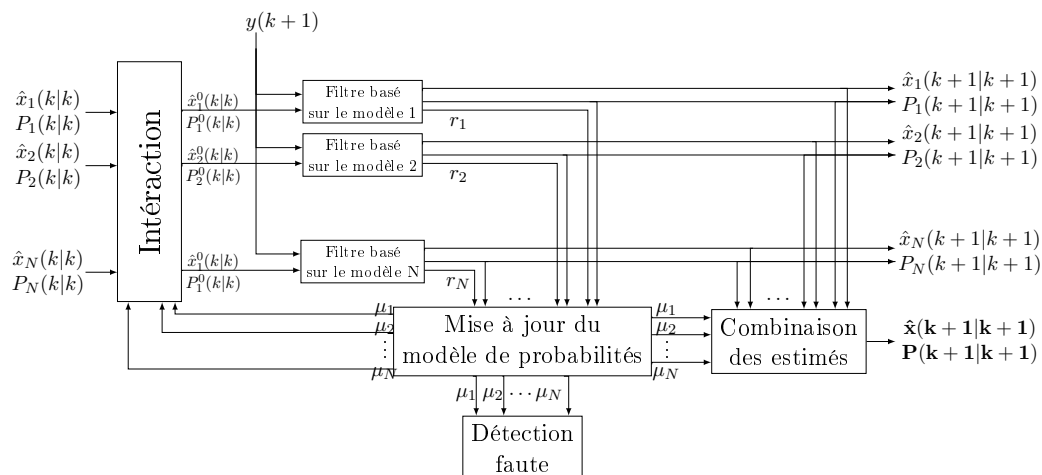


FIGURE 3.12 – Schéma de principe de l'algorithme IMM

Les méthodes multi-modèles ont de bonnes performances, parfois supérieures aux

méthodes plus classiques. Elles se sont imposées comme des solutions efficaces pour la détection de panne simple. Elles permettent une modélisation du système avec des paramètres de pannes spécifiques. Chaque sous-modèle correspond à un cas de panne particulier et l'étude des résidus à chaque pas de temps permet de déterminer le sous modèle le plus probable et donc le cas de panne le plus probable au regard des mesures et des entrées du système. Par ailleurs, l'approche IMM a été étendue à des systèmes non-linéaires en utilisant des filtres particulières au sein de son architecture [Boers 2003] [Blom 2007]. Une autre approche consiste à utiliser des systèmes hybrides à sauts markoviens. Cette technique est une variante de l'algorithme IMM. Elle peut être appliquée à des systèmes linéaires [Blom 1988], mais aussi à des systèmes non-linéaires [Doucet 2001].

Une autre approche consiste à modéliser le système hybride sous forme d'automates hybrides. Les fautes progressives sont modélisées à l'aide du processus continu et les fautes abruptes avec le processus discret [Zhao 2005]. Le modèle hybride est utilisé pour générer hors ligne une table des symptômes de fautes qui est ensuite transformée en arbre de décision qui est utilisé par le diagnostic en ligne.

Une autre approche consiste à coupler une méthode de diagnostic standard pour les systèmes continus, la méthode d'espace de parité, et une méthode de diagnostic standard pour les systèmes discrets [Bayoukh 2008] [Chanthery 2013]. La méthodologie du diagnostic hybride peut être décomposée entre trois parties :

- le diagnostic de la partie continue du système,
- la discrétisation de la partie continue en événements discrets pour enrichir la partie discrète du système, et
- le diagnostic de la partie discrète du système.

Finalement, une approche considère le problème FDI en utilisant un automate hybride, qui combine un automate avec des modèles continus [Cocquempot 2004]. Deux types de modèles sont utilisés ; le modèle nominal qui permet de détecter une faute et les modèles dégradés qui permettent d'identifier la faute grâce à la méthode de l'espace de parité. Un exemple prenant en compte la faute de deux capteurs permet de mettre en avant l'efficacité de la méthode.

3.4 Les méthodes de reconfiguration pour la FTC indirecte

Le but de la reconfiguration du système de commande est de compenser l'effet d'une panne de l'un des composants. Il est important de déterminer l'action de commande la plus appropriée suivant le type de panne dans le but d'assurer la sécurité du système. Cette action peut être effectuée en reconfigurant le contrôleur en ligne. Le bloc de reconfiguration doit être compatible avec le bloc FDD. Cependant, la plupart des études traitent les deux blocs de façon séparée [Patton 1997]. Lunze et Richter classifient les méthodes de reconfiguration selon la Figure 3.13 [Lunze 2008].

Comme évoqué à la Section 2.1.4, la redondance matérielle n'est pas souhaitée pour des drones.

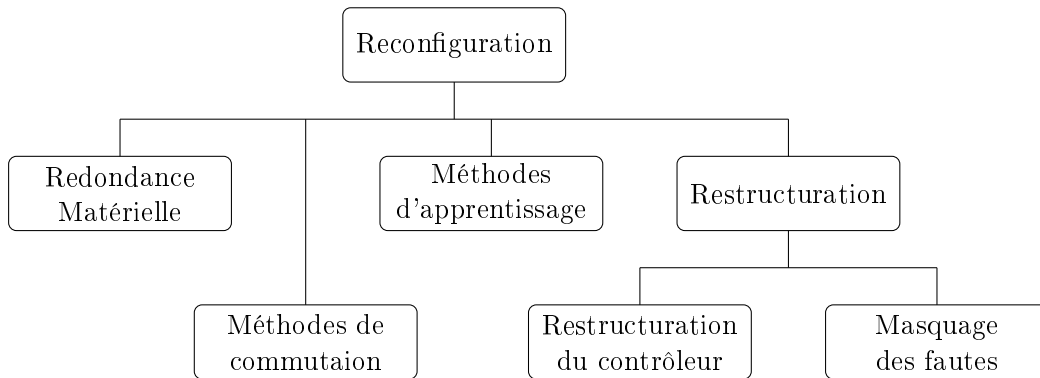


FIGURE 3.13 – Classification des méthodes de reconfiguration

Les méthodes d'apprentissage appartiennent à la famille des approches utilisant l'Intelligence Artificielle. On retrouve les techniques suivantes ; les réseaux de neurones, les systèmes experts ou encore la logique floue.

3.4.1 Les méthodes de restructuration

Le problème de restructuration commence à partir de la solution du contrôleur conçue pour le mode nominal. Il faut donc dans un premier temps concevoir un contrôleur permettant d'obtenir une bonne stabilité et de bonnes performances en situation nominale. Le but de la restructuration est de pouvoir garder les mêmes performances qu'en mode nominal. Pour cela, il existe deux approches. La première consiste à trouver un contrôleur qui en boucle fermée permet d'obtenir le même comportement qu'avec la boucle fermée nominale. Dans ce cas, la loi de commande est adaptée en ligne. A l'opposé, la seconde approche essaye de "cacher" la faute au contrôleur nominal, ainsi les signaux d'entrée et de sortie du composant en panne sont reconfigurés.

La suite de cette section consiste à présenter différentes techniques de restructuration. La première sous-section présente la technique de masquage des fautes. Les autres sous-sections présentent des techniques de restructuration du contrôleur ; la technique d'allocation de la commande, en anglais *Control Allocation* (CA), plusieurs techniques par retour d'état ainsi que deux techniques basées sur la commande prédictive. Cependant, de nombreuses autres techniques de restructuration du contrôleur existent pour contrôler un système [Zhang 2008]. On retrouve la méthode du séquençement de gain qui propose différentes lois de commandes pour différents points de fonctionnement donnés, pour réaliser une loi de commande globale. On trouve aussi la commande classique PI/PID (Proportionnel, Intégral, Dérivé) où un régulateur PI peut être utilisé pour compenser des fautes actionneurs de type blocage ou de type perte d'efficacité [Caliskan 2016].

3.4.1.1 Les méthodes de masquage des fautes

Les méthodes de masquage de faute permettent de garder le contrôleur nominal, de donc de conserver la structure du contrôleur. L'idée est de "cacher" la faute au contrôleur nominal. Pour cela, un module de reconfiguration est inséré entre le système en faute et le contrôleur nominal. Le module de reconfiguration est équivalent à un bloc qui crée des actionneurs et des capteurs virtuels [Rotondo 2015] [Rotondo 2014].

Une méthode consiste à masquer les fautes actionneurs et capteurs en adaptant les outils virtuels du contrôleur au lieu d'adapter ce contrôleur [Rotondo 2014]. Il est également possible d'utiliser des mesures virtuelles, c'est-à-dire des mesures estimées pour cacher la faute. L'estimateur est alors construit de façon à ce que l'estimation des sorties soient insensibles aux effets des fautes [Wu 2006].

3.4.1.2 L'allocation de commande

Le principe de la technique CA est le suivant : une commande virtuelle est générée par le contrôleur. Cette commande virtuelle est passée à l'allocation de commande qui connaît les bornes et l'efficacité de chaque actionneur sur le signal de commande. La technique CA fournit des lois explicites qui permettent d'avoir des opérations rapides avec une faible puissance calculatoire. De plus, elle permet de compenser les fautes actionneurs sans avoir besoin de modifier la structure des lois de commandes. Cependant, ceci entraîne aussi le désavantage de ne pas prendre en compte la dynamique et les limitations des actionneurs après une faute. Par conséquent, le contrôleur essaye d'atteindre les performances du système nominal même si l'actionneur n'est pas capable de l'effectuer.

La technique CA est utilisée pour reconfigurer un système de commande en cas de faute d'une surface de commande de type blocage ou flottant [Ducard 2008] [Ducard 2009]. Le contrôleur génère une commande virtuelle $\mathbf{v} = [C_L \ C_M \ C_N]^T$ définissant les couples de roulis, de tangage et de lacet désirés. Cette commande virtuelle est passée à l'allocation de commande qui prend en compte les bornes de chaque actionneur ainsi que leur efficacité à produire un couple. Un problème d'optimisation contraint est alors résolu en ligne pour générer les signaux de commandes envoyés aux actionneurs. L'allocation de commande permet de distribuer les moments désirés parmi l'ensemble des actionneurs. Il doit donc avoir l'information de santé des surfaces de commande. La méthode utilisée consiste à remplacer les limites de la surface de commande en panne par sa valeur de blocage estimée. Ainsi, la technique CA peut générer des signaux de commande adaptés.

Une autre approche propose de compenser des pannes actionneurs simultanées grâce à une architecture combinant la technique CA et la technique SMC [Wang 2017]. La technique CA permet de redistribuer le signal de commande selon les actionneurs fonctionnels, alors que la technique SMC permet d'ajuster le gain du contrôleur pour stabiliser le système.

3.4.1.3 La commande par retour d'état

La commande par retour d'état permet de modifier le comportement en boucle fermée d'un système dynamique donné par une représentation d'état. La commande s'écrit sous la forme suivante :

$$\mathbf{u} = K\mathbf{x} \quad (3.63)$$

Ainsi, un système linéaire nominal en boucle fermée peut s'écrit sous la forme suivante :

$$\dot{\mathbf{x}} = (A + BK)\mathbf{x} \quad \mathbf{z} = \mathbf{x} \quad (3.64)$$

Pour la commande tolérante aux fautes, le but des commandes par retour d'état est de générer un gain K_f qui stabilise la dynamique du système en faute :

$$\dot{\mathbf{x}} = (A_f + B_f K_f)\mathbf{x} \quad \mathbf{z} = \mathbf{x} \quad (3.65)$$

La technique LQ et la technique de placement de la structure propre, en anglais *Eigenstructure Assignment* (EA) sont les plus utilisées pour concevoir des contrôleurs pour les systèmes multi-entrées, multi-sorties.

La commande LQ permet de calculer la matrice de gains d'une commande par retour d'état. Cette technique minimise un critère de performance, quadratique en l'état \mathbf{x} et la commande \mathbf{u} , qui prend en compte l'énergie de \mathbf{x} et celle de \mathbf{u} . Le but est de ramener aussi rapidement que possible l'état à sa valeur d'équilibre tout en respectant les contraintes du système. La commande LQ possède de bonnes propriétés de robustesse. Lorsqu'une faute est identifiée, un nouveau contrôleur est conçu en recalculant le gain K_f à partir du modèle en faute, c'est à dire avec les matrices A_f et B_f :

$$K_f = R^{-1} B_f^T P_f \quad (3.66)$$

$$0 = Q + A_f P_f + P_f A_f - P_f B_f^T R^{-1} B_f P_f \quad (3.67)$$

Pour plus d'informations sur le calcul du gain, le lecteur peut se référer à l'annexe F.

L'idée de la technique EA est de placer la structure du système, les pôles, de manière appropriée. Si les spécifications du système sont données sous forme de structure propre, la technique EA peut maintenir la stabilité et atteindre exactement la performance désirée. Puisque la stabilité et le comportement dynamique de la boucle fermée du système sont régis par la structure propre du système, en cas de faute, le but est d'avoir la structure propre du système en faute aussi proche que possible de celle du système nominal. Ainsi, chaque degré de liberté du gain K_f est utilisé de façon à minimiser la distance entre les vecteurs propres obtenus et ceux désirés :

$$\lambda(A_f + B_f K_f) = \lambda(A + BK) \quad (3.68)$$

Où $\lambda(\cdot)$ représente les valeurs propres d'une matrice.

Ces deux techniques peuvent aussi bien être utilisées pour les approches passives

[Veillette 1995] [Yang 2000], que pour les approches actives [Wu 2000] [Zhang 2001] [Zhang 2002]. Dans le cas de perte d'efficacité, la faute intervient sur la matrice B . En utilisant les informations fournies par le module FDD, la matrice B_f est estimée et le contrôleur est conçu automatiquement en se basant sur la technique LQ ou EA.

La technique Pseudo Inverse essaye d'atteindre la performance du système nominal en calculant une approximation inverse d'une matrice. La commande Pseudo Inverse calcule K_f de façon à minimiser la norme matricielle de la différence entre $(A + BK)$ et $(A_f + B_f K_f)$. La solution du gain est obtenue à partir de la pseudo inverse de B_f :

$$K_f = B_f^+(A + BK - A_f) \quad (3.69)$$

La stabilité de la boucle fermée n'est pas garantie. Pour faire face à ce manque de garantie, Gao and Antsaklis ont développé la méthode Pseudo Inverse modifiée [Gao 1991]. La méthode Pseudo Inverse est combinée avec la méthode CA pour reconfigurer un drone octorotor en cas de multiples pannes au niveau des moteurs ou des hélices [Saied 2017]. Deux approches pour compenser les fautes actionneurs ont été proposées [Theilliol 2008] ; la première est basée sur la méthode Pseudo Inverse pour garantir performance et stabilité du système et la seconde considère l'ajout d'un terme à la loi de commande nominale. De plus, dans le but de réduire l'énergie utilisée pour compenser la faute, la référence est modifiée pendant la transition.

Remarque 3.3

L'approche par retour d'état suppose que l'état \mathbf{x} est connu. Or l'état est rarement entièrement connu. Il est alors nécessaire d'utiliser un estimateur d'état de manière à le reconstruire.

3.4.1.4 La commande prédictive

La commande prédictive, en anglais *Model Predictive Control* (MPC), repose sur le modèle dynamique afin d'anticiper le comportement futur du système. Cette technique est intuitive et facile à comprendre. Elle a l'avantage de prendre en compte les contraintes sur les variables du système comme par exemple les taux de saturation des actionneurs. L'idée de base est la suivante ; à chaque instant, le but est d'élaborer une séquence de commande futures qui minimise une fonction de coût sur un horizon H grâce à la prédiction des variables du système et d'appliquer le premier élément de cette séquence au système. La commande MPC est bien adaptée au problème de suivi de trajectoires, pour lesquels la trajectoire à suivre est parfaitement connue à l'avance. La prise en compte des fautes est en général faite en ajoutant des contraintes d'inégalités sur des positions et des taux limites. De cette façon, les redondances peuvent être exploitées et le système peut être stabilisé.

La commande MPC a pour avantage d'être applicable aux systèmes linéaires, non-linéaires mais aussi aux systèmes mixtes à dynamique continue et logique [Mignone 2002]. Bien que, sa complexité de calcul augmente très vite avec le nombre de décisions possibles, elle permet des stratégies de reconfiguration utilisant la re-

dondance des composants pour compenser la défaillance des systèmes. Plusieurs algorithmes permettant de résoudre le problème pour des systèmes hybrides linéaires associent des composants à des variables du modèle. Le fait de pouvoir optimiser par rapport à une variable correspond alors au fait d'utiliser le composant correspondant [Tsuda 2001].

La commande prédictive stochastique, en anglais *Stochastic Model Predictive Control* (SMPC), considère des modèles dynamiques qui correspondent à des réalisations possibles d'une variable aléatoire. Une probabilité est associée à chaque réalisation possible. Ainsi le contrôleur va prendre en compte la probabilité d'avoir plusieurs réalisations futures possibles [Bernardini 2009]. Cette méthode peut être adaptée aux fautes, en considérant les actionneurs ou les capteurs comme des variables aléatoires pouvant avoir plusieurs réalisations possibles : une réalisation nominale et des réalisations dégradées.

3.4.2 Les méthodes de commutation

Les méthodes de commutation traitent du choix d'un contrôleur à partir d'une liste prédéfinie de contrôleurs. Les contrôleurs sont développés hors ligne pour un nombre fini de cas de pannes. Le choix du contrôleur se fait en ligne de façon à choisir le meilleur contrôleur suivant la situation courante. Il est donc nécessaire d'utiliser un outil de décision en ligne, généralement une technique d'optimisation.

Les méthodes multi-modèles comprennent un ensemble de contrôleurs. Il est donc nécessaire d'avoir un mécanisme qui permet ; soit de choisir le signal de commande le plus adapté à la faute ; soit de faire une combinaison entre les différents signaux de commande. Les méthodes de commutation préférées en FTC sont ; la commutation et l'adaptation de modèle multiple, en anglais *MMST : Multiple Model Switching and Tuning* (MMST) [Boskovic 2001], la commande adaptative multi-modèles, en anglais *Multiple Model Adaptive Control* (MMAC) [Maybeck 1991] et celle associée à l'IMM [Zhang 2001].

Avec la méthode MMST, la dynamique de chaque scénario de faute est décrite à l'aide de son propre modèle. Chaque modèle a un contrôleur associé. Le système est reconfiguré en choisissant la paire modèle/contrôleur la plus appropriée à chaque pas de temps. Pour cela, un critère J_i , qui correspond à l'erreur entre la sortie du modèle i et la mesure du vecteur \mathbf{z} , est calculé à chaque instant, pour chaque modèle. Le modèle i qui fournit le critère J_i le plus faible est le plus représentatif via à via du système réel et donc le contrôleur i est activé. La technique MMST a l'avantage d'être simple. Cependant, elle est limitée à des scénarios de fautes prédéfinies ; elle n'est pas capable de gérer les pannes non envisagées et elle n'est pas capable de gérer les pannes multiples. De plus, le nombre de paires modèle/contrôleur peut devenir excessif.

La méthode MMST peut être implémentée en construisant un contrôleur pour chaque modèle. Ceux-ci correspondent à un mode nominal et des modes de pannes [Boskovic 2001]. Chaque contrôleur est ensuite conçu avec la technique Pseudo Inverse. Ainsi, la loi de commande est explicitement donnée. L'aspect adaptation est

réalisé par l'ajout d'une matrice de CA au système de commande [Boskovic 2001].

Lorsque le module FDD est réalisé avec la méthode IMM, il est courant d'avoir un ensemble de contrôleurs associés aux différents modèles pour accomplir la tâche de reconfiguration. Pour N modes et N contrôleurs associés à chaque mode, le signal de commande est calculé en faisant une combinaison des signaux de commande à partir des probabilités de modes [Zhang 2001]. Ainsi, la commande a la forme suivante :

$$\mathbf{u}_k = \sum_{i=1}^N \mu_k^i (L_i \mathbf{r}_k - K_i \widehat{\mathbf{x}}_{i_k|k}) \quad (3.70)$$

Où $\widehat{\mathbf{x}}_{i_k|k}$ correspond à l'estimation de l'état du i^{me} modèle ; K_i et L_i sont les gains de contre-réaction et de consigne du i^{me} modèle ; μ_k^i correspond à la probabilité de mode du i^{me} modèle.

Dans ce contexte, les gains K_i sont construits avec les techniques LQ, pour le contrôleur nominal, et EA, pour les contrôleurs pour les modes de fautes. Les gains L_i sont construits en utilisant le théorème de la valeur finale pour améliorer la performance en régime permanent.

L'implémentation de la méthode MMAC comprend également une combinaison des signaux de commande à partir des probabilités de modes [Maybeck 1991].

3.5 Conclusion

Cet état de l'art permet tout d'abord de situer l'approche relative à la démarche suivie ici, c'est-à-dire de situer les méthodes de FTC actives et indirecte dans le contexte général de la FTC. Il indique également que la FTC présente un caractère transverse par rapport aux méthodes continues, discrètes et hybrides existantes.

La partie de l'état de l'art dédiée à la FDD indique que l'estimation conjointe de l'état et des paramètres liés aux fautes est une approche attractive pour celle-ci. De part le nombre de pannes que l'on souhaite traiter lors de cette thèse, les méthodes multi-modèles ne semblent pas adaptées. A l'inverse, les variantes du filtre de Kalman semble bien adapté pour ce problème, néanmoins des verrous le concernant demandent à être levés ; en particulier les limites d'observabilité et les sauts brusques de paramètres de fautes.

La partie de l'état de l'art dédiée à la reconfiguration indique des pistes intéressantes concernant la CA, le MPC et le SMPC. Ces pistes peuvent être combinées ou comparées. En particulier l'état de l'art ne permet pas de trancher entre une FTC adaptative basée sur la restructuration et une FTC basée sur la commutation. Il est fort probable que la combinaison des deux approches soit intéressante pour tirer profit des avantages de chacune, c'est-à-dire avoir un nombre limité de contrôleur tout en gardant des contrôleurs adaptatifs pratiques, c'est-à-dire simple à concevoir et le plus générique possible.

Définition d'un cadre et de modèles génériques pour la commande tolérante aux fautes

Sommaire

4.1	Structure générale de la méthode proposée combinant les méthodes indirectes de commutation et d'adaptation	69
4.2	Modélisation à états continus	70
4.2.1	Représentation d'état du modèle dynamique sans faute	71
4.2.2	Modélisation des fautes actionneurs	72
4.2.3	Modélisation des fautes capteurs	74
4.2.4	Observabilité théorique du système augmenté avec les paramètres de fautes	75
4.2.5	Représentation d'état du modèle dynamique avec fautes	78
4.2.6	Application au drone Altimum 4	78
4.3	Modélisation à états discrets	81
4.3.1	États, Actions et Récompenses	81
4.3.2	Probabilités de transition	82
4.3.3	Observations et probabilités d'observation	83
4.3.4	Application au drone Altimum 4	85
4.4	Conclusion	86

Résumé

CE chapitre vise à décrire l'architecture de la méthode active de commande tolérante proposée, ainsi qu'à définir le cadre de modélisation associé. L'architecture de la méthode combine les approches de type commutatif et adaptatif. De plus, l'objectif de la méthode est de pouvoir prendre avantage des approches utilisant les techniques d'Automatique classique de celles exploitant les méthodes d'Intelligence Artificielle. Ainsi ce chapitre décrit les cadres de modélisation continus et discrets utilisés pour cette méthode.

4.1 Structure générale de la méthode combinant les méthodes indirectes de commutation et d'adaptation

Dans cette section, le principe général de la méthode FTC est décrit. Cette méthode est développée pour traiter le problème d'atterrissage d'un drone en panne, cependant ce chapitre est, autant que possible, général de façon à la rendre applicable à d'autres systèmes. La méthode proposée est présentée sur la Figure 4.1.

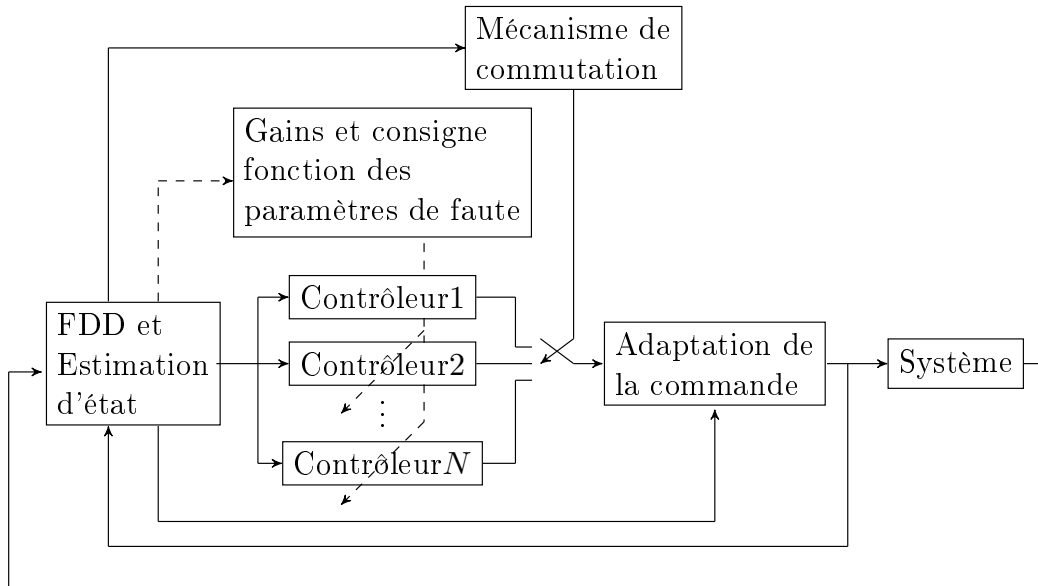


FIGURE 4.1 – Méthode proposée combinant la commutation et l'adaptation

La méthode possède donc quatre caractéristiques :

- elle est indirecte, c'est-à-dire qu'elle prend en compte une méthode FDD ;
- la méthode FDD développée est active, c'est-à-dire qu'elle excite le système pour pouvoir accomplir sa tâche, d'où un bloc d'adaptation de la commande ;
- elle combine les approches adaptatives et commutatives, c'est-à-dire qu'elle commute entre plusieurs contrôleurs adaptatifs ;
- tous les contrôleurs se basent sur la même estimation d'état et peuvent présenter des consignes dépendantes des paramètres de fautes ;
- le mécanisme de reconfiguration a une période d'échantillonnage T_r alors que la méthode FDD et les contrôleurs sont échantillonnés avec une période T_s égale à la période d'échantillonnage du système.

La méthode proposée est indirecte dans le but de concevoir plus facilement les contrôleurs. En effet, lorsque l'on connaît la panne du système, c'est à dire son temps d'apparition, sa localisation et son amplitude, le contrôleur ainsi conçu devrait être plus performant. Par ailleurs, la méthode FDD est active dans le but de résoudre d'éventuels problèmes d'observabilités.

La méthode combine les approches adaptatives et commutatives dans une architecture unique de façon à réduire les désavantages de ces deux méthodes. En

effet, lorsqu'il y a de nombreux types de pannes à gérer, il est difficile de développer un unique contrôleur adaptatif permettant d'être performant face à l'ensemble des fautes. Cependant, avoir un contrôleur pour chaque type de panne peut aussi avoir certains inconvénients. En effet, la conception de l'ensemble des contrôleurs peut nécessiter beaucoup de temps, et l'implémentation de ces contrôleurs est plus fastidieuse. L'objectif de coupler les méthodes adaptatives et commutatives (cf. Section 3.1.2) est donc d'avoir un faible nombre de contrôleurs adaptatifs, faciles à concevoir.

L'échantillonnage T_r du mécanisme de reconfiguration est différent de T_s dans le but de minimiser la puissance de calcul nécessaire. En effet, ce mécanisme peut être coûteux et donc il est intéressant d'avoir une période T_r plus grande que T_s . Le pas de temps associé à l'échantillonnage T_s est noté k alors que le pas de temps associé à l'échantillonnage T_r est noté κ .

4.2 Modélisation à états continus

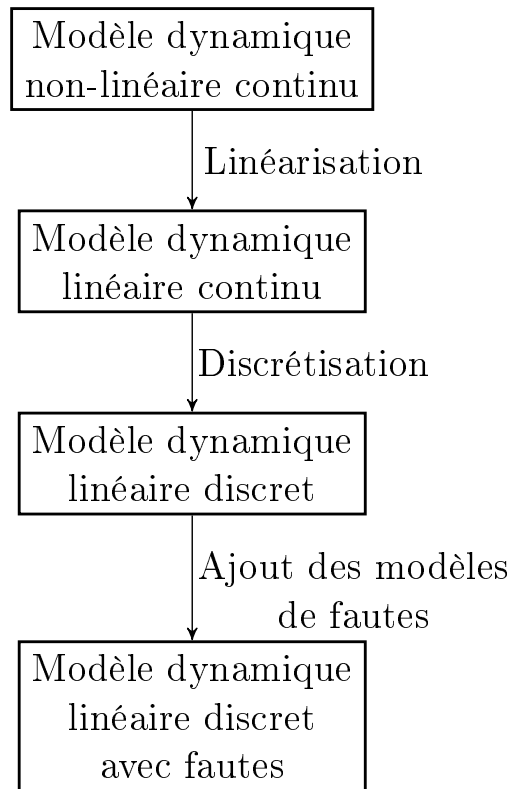


FIGURE 4.2 – Étapes d'obtention du modèle à état continu

Le point de départ considère une représentation d'état non linéaire à temps continu. A partir de ce modèle non-linéaire, trois étapes, représentées sur la Figure 4.2, sont nécessaires pour obtenir un modèle plus adapté à la synthèse d'un filtre et à la conception de contrôleurs :

1. Dans un premier temps, ce modèle peut être linéarisé autour d'un point de fonctionnement en faisant l'hypothèse de petits mouvements autour de ce point. Cette linéarisation est réalisée non seulement pour un point de fonctionnement mais aussi pour des paramètres donnés. Ainsi, après linéarisation, le modèle peut être représenté par un système d'état linéaire stationnaire à temps continu (cf. Annexe B, équations B.3 B.4).
2. Dans un second temps, pour des facilités d'implémentation, le modèle est discrétisé dans le temps, le modèle peut donc être représenté par un système d'état linéaire stationnaire à temps discret (cf. Annexe B, équations B.11 B.12). Il faut faire attention à respecter le théorème de Shannon au moment de choisir la période d'échantillonnage T_s .
3. Finalement, à ce modèle sont ajoutés les modèles de fautes. Ces ajouts doivent prendre en compte l'identifiabilité des paramètres de fautes. Le modèle obtenu permet alors la synthèse d'un filtre et le calcul de contrôleurs.

4.2.1 Représentation d'état du modèle dynamique sans faute

Le modèle linéarisé à la période d'échantillonnage T_s est décrit par les équations suivantes :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k \quad (4.1)$$

$$\mathbf{y}_k = C\mathbf{x}_k \quad (4.2)$$

$$\mathbf{z}_k = \mathbf{y}_k + \mathbf{v}_k \quad (4.3)$$

Où $\mathbf{x}_k \in \mathbb{R}^n$ correspond à l'état ; $\mathbf{u}_k \in \mathbb{R}^l$ correspond au signal de commande ; $\mathbf{w}_k \in \mathbb{R}^n$ correspond aux erreurs de modélisation ; $\mathbf{v}_k \in \mathbb{R}^m$ correspond aux bruits de mesures ; $\mathbf{y}_k \in \mathbb{R}^m$ correspond aux sorties ; $\mathbf{z}_k \in \mathbb{R}^m$ correspond aux mesures.

4.2.1.1 Contrôlabilité et redondance d'actionneurs

Le système est un système discret à paramètres invariants dans le temps, c'est-à-dire que les matrices A , B et C ne varient pas au court du temps. Il est nécessaire de préalablement vérifier la contrôlabilité du système en utilisant le critère de Kalman (cf. Annexe B). De plus, en cas de panne il faut s'assurer que la panne ne rend pas le système non contrôlable. Si la panne apparaît sur un actionneur redondant, le système va rester contrôlable, mais si la panne apparaît sur un actionneur non-redondant, il est possible de rendre le système non contrôlable. Il est donc également nécessaire d'identifier les redondances entre couples ou ensembles d'actionneurs en s'appuyant sur les définitions 4.1 et 4.2.

Définition 4.1 (*Redondance entre deux actionneurs*)

Soit le système défini par les équations 4.1 et 4.2, et sa matrice de fonction de transfert $G(z) \in \mathbb{R}^{m \times l}$. Deux actionneurs, c'est-à-dire deux entrées u_k^i et u_k^j , sont dits redondants si la $i^{\text{ème}}$ et la $j^{\text{ème}}$ colonne de la fonction de transfert de $G(z)$

sont linéairement dépendantes.

Définition 4.2 (*Redondance entre plusieurs actionneurs*)

Soit le système défini par les équations 4.1 et 4.2, et sa matrice de fonction de transfert $G(z) \in \mathbb{R}^{m \times l}$. Il y a des redondances entre des actionneurs si la fonction de transfert de $G(z)$ n'est pas de rang plein.

Les deux définitions se basent sur la matrice de transfert du système. Si la matrice B possède deux colonnes linéairement dépendantes, alors la fonction de transfert possède deux colonnes linéairement dépendantes (Définition 4.1). La réciproque n'est pas forcément vraie. Le même raisonnement est possible pour la définition 4.2.

4.2.1.2 Observabilité

Il est également nécessaire de vérifier l'observabilité du système avec tous ses capteurs et pour toutes les possibilités de ne pas utiliser un des capteurs. L'observabilité du système étudié est essentielle. Elle permet de s'assurer que chacun des états définissant le système est bien défini en fonction des entrées et des sorties disponibles. Dans un système à paramètres invariants, l'observabilité est évaluée facilement à partir du critère de Kalman (cf. Annexe B).

4.2.2 Modélisation des fautes actionneurs

Le modèle choisi pour les fautes actionneurs possède trois caractéristiques :

- il se situe en amont d'une éventuelle dynamique de l'actionneur,
- il est de type affine comme celui de l'équation 3.5 et,
- il s'applique aux variables du modèle non-linéaire du système.

Notons U_k le signal de commande pour le modèle non linéaire. La modélisation pour les fautes d'actionneur proposée se présente selon l'équation suivante :

$$U_k^{f^i} = b_{0k}^i + (1 - b_{1k}^i)U_k^i \quad (4.4)$$

Où $U_k^{f^i}$ correspond au mouvement physique effectué par le i^{me} actionneur pour le modèle non linéarisé, b_{0k}^i et b_{1k}^i correspondent aux paramètres de fautes associés au i^{me} actionneur.

Avec ce cadre, différents types de fautes sont modélisés :

- Si il n'y a pas de faute, $b_{0k}^i = 0$ et $b_{1k}^i = 0$;
- si l'actionneur est bloqué à une valeur δ_0 , $b_{0k}^i = \delta_0$ et $b_{1k}^i = 1$;
- Si un biais d'une valeur δ_0 apparaît au niveau de l'actionneur, $b_{0k}^i = \delta_0$ et $b_{1k}^i = 0$;
- Si l'actionneur est en perte d'efficacité, $b_{0k}^i = 0$ et $b_{1k}^i \in]0, 1[$.

Cependant, cette équation convient pour un modèle non linéarisé. Les relations entre les variables du modèle linéarisé et celles du modèle non linéarisé sont les

suivantes :

$$U_k^i = u_k^i + u^{*i} \quad (4.5)$$

$$U_k^{f^i} = u_k^{f^i} + u^{*i} \quad (4.6)$$

Où u^{*i} est la valeur d'équilibre de la i^{me} commande. Comme le modèle est linéarisé pour un état d'équilibre nominal, c'est à dire sans faute, ce terme apparaît de la même manière dans les équations 4.5 et 4.6.

En remplaçant les équations 4.5 et 4.6 dans l'équation 4.4, on obtient la relation suivante :

$$u_k^{f^i} = b_{0k}^i + (1 - b_{1k}^i)u_k^i - b_{1k}^i u^{*i} \quad (4.7)$$

La sortie \mathbf{u}_k^f correspond au mouvement physique effectué par l'actionneur à \mathbf{u}^* près, qui est limitée selon l'amplitude de l'actionneur. Donc, \mathbf{u}_k^f , et \mathbf{u}_k sont bornées $[\underline{\mathbf{u}}, \bar{\mathbf{u}}]$ avec $\underline{\mathbf{u}} = \underline{\mathbf{U}} - \mathbf{u}^*$ et $\bar{\mathbf{u}} = \bar{\mathbf{U}} - \mathbf{u}^*$.

La Figure 4.3 décrit la structure générale du modèle linéarisé avec fautes actionneurs. F_1 est la fonction permettant de modéliser les fautes d'actionneurs, $F_1(u_k^i) = u_k^{f^i}$. Pour un modèle linéarisé, la modélisation de la faute se présente donc selon l'équation 4.7.

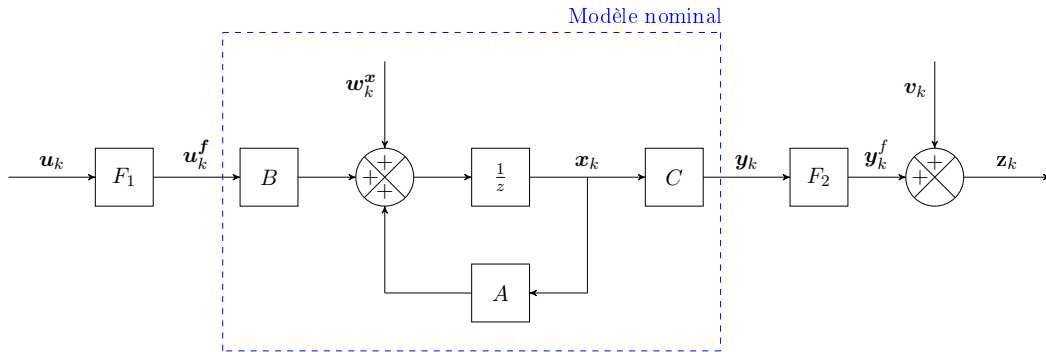


FIGURE 4.3 – Modélisation du système avec fautes actionneurs et capteurs

Remarque 4.1

Le terme $b_{1k}^i u^{*i}$ de l'équation 4.7 n'apparaît pas lorsque le modèle initial est linéaire.

Les fautes actionneurs envisagées sont des biais, des blocages et des pertes d'efficacité. D'après la modélisation choisie, pour ces trois types de fautes, les paramètres de fautes b_0 et b_1 sont constants, exceptés à l'apparition de la faute où ils subissent un saut de valeur. La connaissance de ces paramètres de fautes est indispensable pour pouvoir compenser la faute. En effet, ces paramètres fournissent une information sur le type de faute ; il n'est pas possible de savoir si l'actionneur est bloqué ou en perte d'efficacité et il est alors plus facile de calculer des gains de retour d'état estimé permettant de compenser la faute.

4.2.3 Modélisation des fautes capteurs

Le modèle choisi pour les fautes capteurs possède trois caractéristiques :

- il se situe en amont du bruit de mesure,
- c'est un biais, équivalent discret de celui de l'équation 3.6 et,
- il s'applique aux variables du modèle non-linéaire du drone.

Notons \mathbf{Y}_k le vecteur de sortie pour le modèle non linéarisé. La modélisation pour les fautes de capteurs proposée dans cette thèse se présente selon les équations suivantes :

$$Y_k^{f^i} = c_{0k}^i + Y_k^i \quad (4.8)$$

$$Z_k^{f^i} = Y_k^{f^i} + v_k^i \quad (4.9)$$

Où $Y_k^{f^i}$ correspond au modèle de sortie sans bruit en cas de faute, $Z_k^{f^i}$ correspond aux mesures effectués par le i^{me} capteur en faute pour le modèle non linéarisé, c_{0k}^i correspond au paramètre de faute associés au i^{me} capteur.

Avec ce cadre, contrairement aux fautes actionneurs, il est supposé que le paramètre de faute c_{0k}^i peut varier au cours du temps. Différents types de fautes sont alors modélisés : les biais, les dérives, ainsi que les blocages.

- Si il n'y a pas de faute, $c_{0k}^i = 0$;
- Si un biais d'une valeur δ_0 apparaît au niveau du capteur, $c_{0k}^i = \delta_0$;

Cependant, cette équation convient pour un modèle non linéarisé.

Les relations entre les variables du modèle linéarisé et celles du modèle non linéarisé sont les suivantes :

$$Y_k^i = y_k^i + z^{*i} \quad (4.10)$$

$$Y_k^{f^i} = y_k^{f^i} + z^{*i} \quad (4.11)$$

Où z^{*i} est la valeur d'équilibre de la i^{me} mesure physique. Comme le modèle est linéarisé pour un état d'équilibre nominal, c'est à dire sans faute, ce terme apparaît de la même manière dans les équations 4.10 et 4.11.

En remplaçant les équations 4.10 et 4.11 dans l'équation 4.13, on obtient la relation suivante :

$$z_k^{f^i} = c_{0k}^i + y_k^i + v_k^i \quad (4.12)$$

La sortie z_k^f correspond à la mesure effectués par le capteur.

La Figure 4.3 décrit la structure générale du modèle linéarisé avec fautes capteurs. F_2 est la fonction permettant de modéliser les fautes capteurs, $F_2(y_k^i) = y_k^{f^i}$. Pour un modèle linéarisé, la modélisation de la faute se présente selon l'équation 4.12.

Les fautes capteurs sont modélisées de façon différente à celles des actionneurs. Contrairement aux paramètres b_0 et b_1 , le paramètre de faute c_0 peut varier au cours du temps. Les fautes capteurs ne sont pas modélisées comme les fautes actionneurs pour plusieurs raisons. En supposant une modélisation similaire à celle des fautes

actionneurs, l'équation de sortie s'écrirait de la façon suivante :

$$z_k^{f^i} = c_{0k}^i + (1 - c_{1k}^i)z_k^i - c_{1k}^i z_k^{*i} \quad (4.13)$$

L'équation 4.13 possède un terme non linéaire $-c_{1k}^i z_k^i$, l'utilisation d'un estimateur non linéaire est donc nécessaire avec cette modélisation. De plus, cette modélisation rend le système peu observable. Pour un système non-linéaire, il faut regarder localement l'observabilité (cf. Annexe B). Autour du vecteur d'état $\mathbf{x}_0 = 0$ et du vecteur d'entrée $\mathbf{u} = -K\mathbf{x}_0 = 0$, le système n'est pas observable.

Par ailleurs, contrairement aux fautes actionneurs, la connaissance du type de fautes n'est pas nécessaire pour les fautes capteurs. En effet, une faute capteur n'a pas besoin d'être compensée par un autre capteur. On va chercher à reconstruire la valeur de l'état grâce à la dynamique du modèle. Finalement il est néfaste d'augmenter la dimension du vecteur des variables à estimer avec des paramètres de fautes c_1 , car ceci augmente le coût calculatoire.

4.2.4 Observabilité théorique du système augmenté avec les paramètres de fautes

L'ajout de paramètres de fautes peut détériorer l'observabilité du système. Par conséquent l'observabilité du système augmenté avec les paramètres de fautes doit elle aussi être évaluée. Le système d'état prenant en compte les fautes actionneurs et capteurs est décrit par les équations suivantes :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + B\mathbf{b}_{0k} + D_k\mathbf{b}_{1k} + \mathbf{w}_k^x \quad (4.14)$$

$$\mathbf{z}_k = C\mathbf{x}_k + \mathbf{c}_{0k} + \mathbf{v}_k \quad (4.15)$$

avec $D_k = D(\mathbf{u}_k) = B \cdot \text{diag}\{-\mathbf{u}_k - \mathbf{u}^*\}$.

Les valeurs des paramètres de fautes \mathbf{b}_0 , \mathbf{b}_1 et \mathbf{c}_0 sont inconnues, par ailleurs, pour prendre en compte la possibilité de variations, l'évolution de ces paramètres est modélisée à l'aide de marches aléatoires :

$$\mathbf{b}_{0k+1} = \mathbf{b}_{0k} + \mathbf{w}_k^{b_0} \quad (4.16)$$

$$\mathbf{b}_{1k+1} = \mathbf{b}_{1k} + \mathbf{w}_k^{b_1} \quad (4.17)$$

$$\mathbf{c}_{0k+1} = \mathbf{c}_{0k} + \mathbf{w}_k^{c_0} \quad (4.18)$$

Notons $\mathbf{b} = [\mathbf{b}_0 \ \mathbf{b}_1 \ \mathbf{c}_0]^T$. Pour étudier l'observabilité du système, on écrit le système précédent sous la forme d'un système augmenté :

$$\begin{aligned} \mathbf{x}_{k+1} &= A_k^{aug} \mathbf{x}_k^{aug} + B^{aug} \mathbf{u}_k + \mathbf{w}_k \\ \mathbf{z}_k &= C^{aug} \mathbf{x}_k^{aug} + \mathbf{v}_k \end{aligned} \quad (4.19)$$

$$\text{avec } A_k^{aug} = \begin{bmatrix} A & B & D_k & \mathbb{O} \\ \mathbb{O} & \mathbb{I} & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \mathbb{I} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{I} \end{bmatrix}, B^{aug} = \begin{bmatrix} B \\ \mathbb{O} \\ \mathbb{O} \\ \mathbb{O} \end{bmatrix}, C^{aug} = [C \quad \mathbb{O} \quad \mathbb{O} \quad \mathbb{I}], \mathbf{x}_k^{aug} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{b}_{0k} \\ \mathbf{b}_{1k} \\ \mathbf{c}_{0k} \end{bmatrix} \text{ et } \mathbf{w}_k = \begin{bmatrix} \mathbf{w}_k^x \\ \mathbf{w}_k^{b_0} \\ \mathbf{w}_k^{b_1} \\ \mathbf{w}_k^{c_0} \end{bmatrix}.$$

Dans ce cas, le système est un système discret à paramètres variants dans le temps. En effet, dans ce cas la matrice A_k^{aug} dépend du temps.

Alors que l'analyse de l'observabilité est plutôt simple pour les systèmes discret à temps invariant, l'analyse pour des systèmes à temps variant est complexe et prend en compte l'évaluation du grammien d'observabilité (cf. Annexe B). Le calcul du grammien d'observabilité est fait numériquement plutôt que de façon analytique. Par conséquent, il est plus difficile d'étudier l'observabilité du système et d'en déduire des règles générales.

Ici, la matrice C^{aug} est invariante dans le temps, donc le grammien d'observabilité s'exprime de la façon suivante :

$$W_o(k_0, k_0 + s) = \sum_{i=0}^s \Phi^T(k_0 + i, k_0) C^{augT} C^{aug} \Phi(k_0 + i, k_0) \quad (4.20)$$

où Φ est la matrice de transition ($i > 0$) :

$$\Phi(k_0 + i, k_0) = A_{k_0+i-1}^{aug} A_{k_0+i-2}^{aug} \cdots A_{k_0}^{aug} \quad (4.21)$$

$$\Phi(k_0, k_0) = \mathbb{I} \quad (4.22)$$

D'après l'écriture par bloc de A_k^{aug} , la matrice de transition peut s'exprimer de la façon suivante :

$$\Phi(k_0 + i, k_0) = \begin{bmatrix} A^s & \sum_{j=1}^i A^{i-j} B & \sum_{j=1}^i A^{i-j} D_{k_0+j-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{I} & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \mathbb{I} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{I} \end{bmatrix} \quad (4.23)$$

Le grammien d'observabilité est le suivant :

$$W_o(k_0, k_0 + s) = \begin{bmatrix} \Lambda_{1,1} & \Lambda_{1,2} & \Lambda_{1,3} & \Lambda_{1,4} \\ \Lambda_{1,2}^T & \Lambda_{2,2} & \Lambda_{2,3} & \Lambda_{2,4} \\ \Lambda_{1,3}^T & \Lambda_{2,3}^T & \Lambda_{3,3} & \Lambda_{3,4} \\ \Lambda_{1,4}^T & \Lambda_{2,4}^T & \Lambda_{3,4}^T & \Lambda_{4,4} \end{bmatrix} \quad (4.24)$$

avec

$$\Lambda_{1,1} = \sum_{i=0}^s A^{iT} C^T C A^i \quad (4.25)$$

$$\Lambda_{1,2} = \sum_{i=1}^s A^{iT} C^T C \left(\sum_{j=1}^i A^{i-j} \right) B \quad (4.26)$$

$$\Lambda_{1,3} = \sum_{i=1}^s A^{iT} C^T C \left(\sum_{j=1}^i A^{i-j} D_{k_0+j-1} \right) \quad (4.27)$$

$$\Lambda_{1,4} = \sum_{i=0}^s A^{iT} C^T \quad (4.28)$$

$$\Lambda_{2,2} = B^T \sum_{i=1}^s \left(\sum_{j=1}^i A^{i-j} \right)^T C^T C \left(\sum_{j=1}^i A^{i-j} \right) B \quad (4.29)$$

$$\Lambda_{2,3} = B^T \sum_{i=1}^s \left(\sum_{j=1}^i A^{i-j} \right)^T C^T C \left(\sum_{j=1}^i A^{i-j} D_{k_0+j-1} \right) \quad (4.30)$$

$$\Lambda_{2,4} = B^T \sum_{i=1}^s \left(\sum_{j=1}^i A^{i-j} \right)^T C^T \quad (4.31)$$

$$\Lambda_{3,3} = \sum_{i=1}^s \left(\sum_{j=1}^i A^{i-j} D_{k_0+j-1} \right)^T C^T C \left(\sum_{j=1}^i A^{i-j} D_{k_0+j-1} \right) \quad (4.32)$$

$$\Lambda_{3,4} = \sum_{i=1}^s \left(\sum_{j=1}^i A^{i-j} D_{k_0+j-1} \right)^T C^T \quad (4.33)$$

$$\Lambda_{4,4} = \sum_{i=0}^s \mathbb{I} \quad (4.34)$$

Le système est observable si le grammien d'observabilité est inversible, c'est-à-dire si le déterminant est non nul. Le déterminant du grammien d'observabilité est non nul si aucune ligne (ou colonne) n'est une combinaison linéaire des autres lignes (ou colonnes).

La matrice $[\Lambda_{1,1} \ \Lambda_{1,2} \ \Lambda_{1,3} \ \Lambda_{1,4}]$ ne possède pas de lignes proportionnelles. En effet, la paire (A, C) est observable et donc $\Lambda_{1,1}$ ne possède pas de lignes proportionnelles.

Mathématiquement, on peut constater au niveau du grammien d'observabilité que les paramètres \mathbf{b}_0 associés à des actionneurs redondants ne peuvent être identifiables. En effet, si la matrice B possède au moins deux colonnes qui forment une combinaison linéaire entre elles, la matrice $[\Lambda_{1,2}^T \ \Lambda_{2,2} \ \Lambda_{2,3} \ \Lambda_{2,4}]$ possède au moins deux lignes qui forment une combinaison linéaire entre elles et donc le déterminant du grammien d'observabilité est nul. Il faut donc choisir un ou des b_0 communs pour les actionneurs redondants. Pour cela, il est nécessaire de calculer le rang de

la matrice formée par les colonnes de B associées aux actionneurs redondants. Ce rang fournit le nombre de b_0 à considérer dans le modèle.

Mathématiquement, on peut constater au niveau du grammien d'observabilité que les paramètres \mathbf{b}_1 associés à des actionneurs redondants sont plus difficilement différentiables que pour un actionneur non redondant. En effet, si la matrice D_{k_0+j-1} possède au moins deux colonnes proportionnelles entre elles, la matrice $[\Lambda_{1,3}^T \ \Lambda_{2,3}^T \ \Lambda_{3,3} \ \Lambda_{3,4}]$ possède au moins deux lignes proportionnelles et donc le déterminant du grammien d'observabilité est nul. Sachant que D_{k_0+j-1} vaut $-B \cdot \text{diag}\{\mathbf{u}_{k_0+j-1} + \mathbf{u}^*\}$, il faut envoyer une commande différente pour les actionneurs redondants pour rendre identifiable les paramètres b_1 associés aux actionneurs redondants. Ainsi, sans aucune excitation au niveau des commandes, il n'est pas possible de différencier les deux actionneurs.

Pour l'identifiabilité des paramètres c_0 , on regarde les colonnes du grammien d'observabilité et notamment les colonnes de $\Lambda_{.,1}$ et de $\Lambda_{.,4}$. Si le produit de matrices $CA^j, \forall j$ donne des colonnes identiques à la matrice identité \mathbb{I} , alors $\Lambda_{.,1}$ et $\Lambda_{.,4}$ auront des colonnes proportionnelles et le grammien ne sera donc pas inversible. La matrice C peut se décomposer en deux matrices : $C = [C_o \ C_{io}]^T$. Où C_o correspond aux lignes de la matrice C associées aux mesures pouvant être en panne sans entraver l'observabilité du système, et C_{io} correspond aux lignes de la matrice C associées aux mesures dont la panne fait perdre l'observabilité du système.

4.2.5 Représentation d'état du modèle dynamique avec fautes

Le système d'état prenant en compte les fautes actionneurs et capteurs est décrit par les équations suivantes :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + E\mathbf{b}_0 + D_k\mathbf{b}_1 + \mathbf{w}_k^x \quad (4.35)$$

$$\mathbf{z}_k = \begin{bmatrix} C_o \\ C_{io} \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbb{I} \\ \mathbb{O} \end{bmatrix} \mathbf{c}_0 + \mathbf{v}_k \quad (4.36)$$

où $D_k = B \cdot \text{diag}\{-\mathbf{u}_k - \mathbf{u}^*\}$ et E est la matrice formée à partir de combinaisons linéaires des colonnes de B associées aux actionneurs redondants.

Avec $\mathbf{b} = [\mathbf{b}_0 \ \mathbf{b}_1 \ \mathbf{c}_0]^T$, le système peut s'écrire sous la forme suivante :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + [E \ D_k \ \mathbb{O}] \mathbf{b}_k + \mathbf{w}_k^x \quad (4.37)$$

$$\mathbf{b}_{k+1} = \mathbf{b}_k + \mathbf{w}_k^b \quad (4.38)$$

$$\mathbf{z}_k = \begin{bmatrix} C_o \\ C_{io} \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbb{O} & \mathbb{O} & \mathbb{I} \\ \mathbb{O} & \mathbb{O} & \mathbb{O} \end{bmatrix} \mathbf{b}_k + \mathbf{v}_k \quad (4.39)$$

4.2.6 Application au drone Altimum 4

4.2.6.1 Analyse du modèle dynamique sans faute

Le modèle Simulink présenté en Section 2.2.3 a été conçu à partir d'équations non linéaires correspondantes à une représentation d'état non-linéaire à temps continu

(cf. Annexe B, équations B.1 B.2). Pour appliquer la méthode à un scénario d'atterrissage du drone Altimum 4, le modèle est linéarisé pour le point de vol suivant : une vitesse air de $20m/s$ et une pente de -3° . Les matrices numériques A , B et C obtenues pour le modèle complet et le modèle longitudinal sont données en Annexe A.

Remarque 4.2

Sur l'Altimum 4, la dynamique la plus rapide correspond à celle des élevons. La fréquence de coupure des élevons est de $12Hz$. Pour respecter le théorème de Shannon, l'échantillonnage doit être d'au moins $24Hz$. La fréquence de pilotage est de $50Hz$, la période d'échantillonnage T_s choisie est donc de $1/50s$.

Contrôlabilité et redondances des actionneurs :

Que ce soit avec le modèle longitudinal ou avec le modèle complet de l'Altimum 4, le système est contrôlable d'après le critère de Kalman (cf. Annexe B). Sur le modèle longitudinal de l'Altimum 4 (cf. Annexe A), les deux élevons δ_l et δ_r sont redondants, les colonnes de B associées aux élevons sont identiques. Le moteur est un actionneur non-redondant. Sur le modèle complet de l'Altimum 4 (cf. Annexe A), les quatre élevons δ_l^1 , δ_l^2 , δ_r^1 et δ_r^2 sont redondants, les colonnes de B associées aux élevons forment deux combinaisons linéaires. Le moteur est un actionneur non-redondant.

Observabilité :

Que ce soit avec le modèle longitudinal ou avec le modèle complet de l'Altimum 4, le système est observable d'après le critère de Kalman (cf. Annexe B). Avec le modèle longitudinal, si l'on perd la mesure du GPS de la position x , le système devient inobservable. Si l'on perd la mesure x , le rang de la matrice d'observabilité devient $n - 1$. Avec le modèle complet, on observe le même phénomène sur les mesures du GPS des positions x et y . Que ce soit avec le modèle longitudinal ou avec le modèle complet, ce phénomène n'est pas présent sur la mesure de position z . En effet, la position z est redondante, il y a la mesure du GPS et la mesure du baromètre. Ainsi si l'on perd l'une des deux mesures, le système est toujours observable. Cependant, si l'on perd les deux mesures, le système est inobservable. Pour les autres mesures, il est possible de les perdre sans pour autant rendre le système inobservable, le rang de la matrice du critère de Kalman reste de n . Ainsi par la suite, on suppose le GPS fonctionnel pour des raisons d'observabilité. Par conséquent, pour le modèle longitudinal, on a $C_o = C_{1:6}$. et $C_{io} = C_{7:10}$, pour le modèle complet, on a $C_o = C_{1:12}$. et $C_{io} = C_{13:18}$.

4.2.6.2 Identifiabilité des paramètres c_0 de la position

L'inobservabilité du système sans les mesures des positions x , et y pour le modèle complet, est simple à démontrer. En utilisant le modèle complet, et en supposant que seule la mesure de la position du GPS, x , y et z peut tomber en panne, les matrices augmentées sont les suivantes :

$$A^{aug} = \begin{bmatrix} A & \mathbb{O}_{n,3} \\ \mathbb{O}_{3,n} & \mathbb{I}_3 \end{bmatrix} \text{ et } C^{aug} = \begin{bmatrix} C_o & \mathbb{O}_{m-3,3} \\ C_{io} & \mathbb{I}_{3,3} \end{bmatrix}$$

En utilisant les valeurs numériques des matrices A et C de l'Annexe A et en utilisant le critère de Kalman, le rang de la matrice d'observabilité est de 16 ce qui n'est pas égal à $n = 18$ (15 états et 3 paramètres c_0). Le système n'est donc pas observable. Ainsi pour des raisons d'observabilité, le GPS est supposé fiable ; le modèle doit indiquer que le GPS ne peut pas tomber en panne, c'est-à-dire qu'il n'y a pas de paramètres de fautes c_0 associés aux mesures du GPS, et aucun scénario ne va simuler des pannes GPS lors de la validation de l'algorithme. Il est possible de supposer le GPS en panne si la mesure de la position est redondante, par exemple si deux GPS sont présents sur le drone. Il existe aussi des méthodes, tel que le GLR présenté en Section 3.3.6.2, permettant de détecter les pannes GPS. L'utilisation de ces possibilités est discuté lors du Chapitre 9.

Concernant la mesure de l'altitude z_B du baromètre, on peut supposer que celle-ci peut tomber en panne. En effet, la mesure est redondante avec celle du GPS que l'on suppose fonctionnel, il est donc possible d'associer un paramètre de faute c_0 à la sortie z_B tout en restant observable.

4.2.6.3 Identifiabilité des paramètres b_0 des actionneurs redondants

La non identifiabilité des biais est montrée sur le modèle longitudinal. En utilisant le modèle longitudinal et en supposant que seuls les actionneurs peuvent tomber en panne avec des biais, les matrices augmentées sont les suivantes :

$$A^{aug} = \begin{bmatrix} A & B \\ \mathbb{O}_{3,n} & \mathbb{I}_3 \end{bmatrix} \text{ et } C^{aug} = [C \quad \mathbb{O}_{m,3}]$$

En utilisant les valeurs numériques des matrices A , B et C de l'Annexe A, et en utilisant le critère de Kalman, le rang de la matrice d'observabilité est de 10 ce qui n'est pas égal à $n = 11$ (8 états et 3 paramètres b_0). Le système n'est donc pas observable. Ce résultat était attendu mathématiquement et physiquement. Physiquement, l'effet des élevons est strictement équivalent sur le drone. Mathématiquement, la matrice B possède deux colonnes identiques pour les élevons, il n'est donc pas possible de différencier les paramètres b_0 associés aux deux élevons.

Pour le modèle longitudinal, le rang de la matrice $[B_{.2} \quad B_{.3}]$ est de 1. On prend donc un seul b_0 pour modéliser les biais des deux élevons, la matrice E est donc la suivante : $E = [B_{.1} \quad B_{.2}]$.

Pour le modèle complet, les colonnes de B associées aux quatre élevons forment une combinaison linéaire entre elles. Le rang de la matrice formée par ces quatre colonnes est de 2. Le rang étant de 2, la dimension de l'espace des effets des actionneurs est de 2, donc on peut associer à chacune un paramètre b_0 , qui sera ainsi observable. Deux paramètres sont choisis, b_0^e qui correspond au biais sur l'élévateur virtuel et un b_0^a qui correspond au biais sur l'aileron virtuel. Ces paramètres sont

associés aux colonnes suivantes :

$$B_e = -B_2 - B_3 - B_4 - B_5 \quad (4.40)$$

$$B_a = B_2 + B_3 - B_4 - B_5 \quad (4.41)$$

La matrice E est donc la suivante : $E = [B_1 \ B_e \ B_a]$.

4.3 Modélisation à états discrets

Le choix du contrôleur dans le mécanisme de commutation présenté sur la Figure 4.1 correspond à une action discrète décidée à chaque instant κ . Ce choix peut dépendre du contrôleur actuellement utilisé, c'est-à-dire d'un état discret. Il peut également dépendre des valeurs et des incertitudes courantes des paramètres de pannes. En faisant l'hypothèse que des ensembles de valeurs de paramètres de pannes peuvent être représentés de manière abstraite par un mode, on peut définir un problème de décision discret basée sur des états discrets. D'un instant à l'instant suivant, le mode peut changer de manière non déterministe et on ne l'observe pas directement. Le problème de reconfiguration nécessite une prise de décision en vol. C'est pourquoi on choisit d'utiliser le cadre des processus de décision Markovien partiellement observable, en anglais *Partially Observable Markov Decision Process* (POMDP) comme modélisation à états discrets.

Un POMDP (cf. Annexe C) prend en compte des états, des actions, une fonction de coût, une transition de probabilité entre les états, des observations ainsi qu'une probabilité d'observation. Chaque ensemble est défini ci-dessous de façon à être adapté au problème de commutation de contrôleur.

4.3.1 États, Actions et Récompenses

Différents modes m sont considérés ; le mode nominal et les modes dégradés. Par exemple, un mode dégradé correspond à un actionneur ou à un capteur en panne. Chaque mode m a un contrôleur associé c qui est construit de façon à obtenir la meilleure performance dans ce mode. Il est possible d'avoir un contrôleur associé à plusieurs modes, ainsi le nombre de contrôleurs est inférieur ou égal au nombre de modes. Un état s est défini par un mode m et un contrôleur c : $s = \{m, c\}$. Il y a donc $n_{\text{contrôleur}} \times n_{\text{mode}}$ états.

Une action a consiste à sélectionner un contrôleur c pour le pas suivant. La dynamique du système est présentée sur la Figure 4.4.

La transition entre contrôleurs est déterministe alors que la transition entre les modes est probabiliste.

Un POMDP prend en compte une fonction de coût qui permet d'avoir une récompense à chaque pas de temps. Comme montré sur la Figure 4.5, cette récompense dépend de l'état courant, de l'état précédent, ainsi que de l'action. Contrairement au cadre POMDP classique, la fonction de coût n'est pas stationnaire.

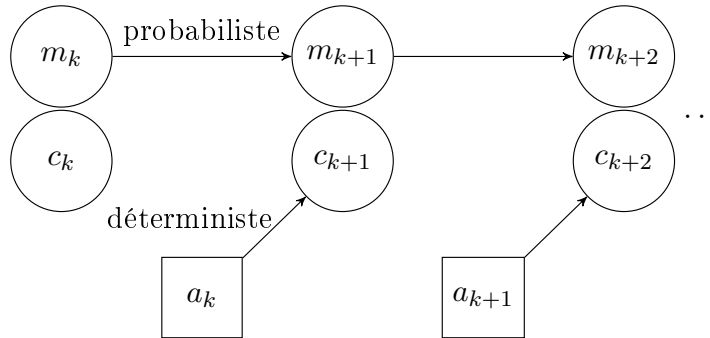


FIGURE 4.4 – Dynamique discrète du système

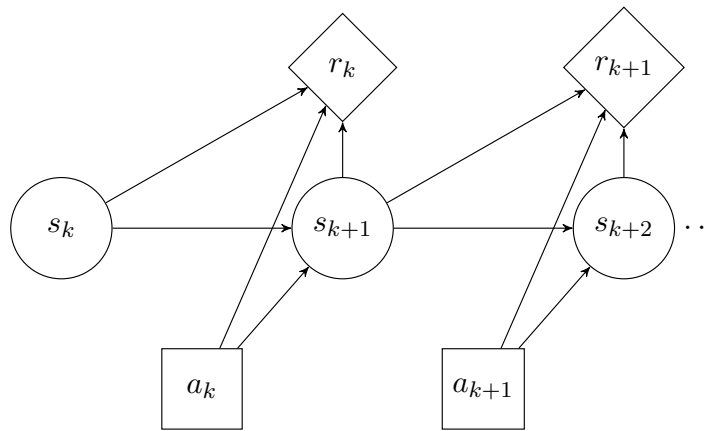


FIGURE 4.5 – Récompense du système

Remarque 4.3

La transition entre contrôleurs est déterministe puisqu'on considère que le contrôleur choisi est effectivement enclenché car on ne considère pas la possibilité d'une faute ici. Néanmoins, avec le formalisme POMDP, il est possible de considérer cette possibilité de faute.

4.3.2 Probabilités de transition

La forme générale des probabilités de transition entre les états $p(s'|s, a)$ dépend de l'action. Contrairement à un POMDP classique, le problème considéré ici est plus simple, la probabilité de transition ne dépendant pas des actions $p(s'|s)$. En effet, le fait de sélectionner un contrôleur plutôt qu'un autre ne change pas la probabilité d'un composant de tomber en panne. De plus, comme montré sur la Figure 4.4, la dynamique des contrôleurs est déterministe et les transitions entre les modes ne dépendent pas des contrôleurs. On s'intéresse donc uniquement aux probabilités entre les modes $p(m'|m)$, qui sont stationnaires.

La matrice de transition T_m entre les modes est la suivante :

$$T_m = \begin{bmatrix} p(m_0|m_0) & \cdots & p(m_0|m_{n_{mode}-1}) \\ \vdots & \ddots & \vdots \\ p(m_{n_{mode}-1}|m_0) & \cdots & p(m_{n_{mode}-1}|m_{n_{mode}-1}) \end{bmatrix} \quad (4.42)$$

La Figure 4.6 montre un exemple pour un cas où il y a trois modes.

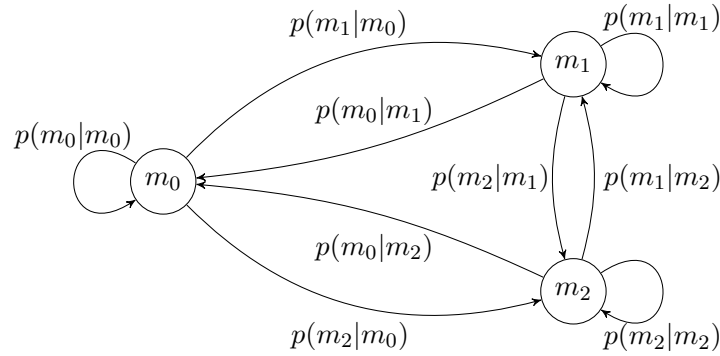


FIGURE 4.6 – Exemple de graphe de transition entre modes

Pour définir un modèle réaliste, la matrice de transition T_m doit :

- soit considérer la documentation des actionneurs et des capteurs qui peuvent contenir des informations sur les taux de défaillances ;
- soit se baser sur des études concernant la défaillance des actionneurs et capteurs en général.

4.3.3 Observations et probabilités d'observation

Le mode courant n'est pas connu, néanmoins il est possible d'avoir certaines observations liées au mode courant. Les observations d'un POMDP classique viennent d'un ensemble discret, cependant ici, l'observation ω est basée sur l'information du processus continu issue de l'estimation des paramètres de fautes $\omega = \{\hat{\mathbf{b}}\}$.

La probabilité d'observation est égale à la probabilité d'avoir l'observation ω sachant l'état s , $p(\omega|s)$. Ici encore, le problème peut être simplifié en considérant uniquement la probabilité d'avoir l'observation ω sachant le mode m , $p(\omega|m)$, puisqu'elle ne dépend pas du choix du contrôleur. Comme ω correspond aux estimations des paramètres de fautes, pour calculer la probabilité d'observation, il faut faire des hypothèses sur la distribution des paramètres de fautes b^i .

Pour chaque mode, il faut définir une loi de probabilité pour l'observation ayant f_m pour densité de probabilité. La densité de probabilité doit être paire vis à vis de l'erreur entre \mathbf{b} et $\hat{\mathbf{b}}$ de façon à ce que les paramètres de fautes puisse être échangés avec leurs estimations. On peut alors faire l'hypothèse que $p(\mathbf{b}|\omega) = p(\omega|\mathbf{b})$.

Les probabilités d'observation sont calculées en intégrant sur le domaine d'ob-

servation la quantité suivante :

$$p(\omega|m) = \int f_m(\mathbf{b})p(\omega|\mathbf{b})d\mathbf{b} \quad (4.43)$$

où $f_m(\mathbf{b})$ dépend du mode.

Dans le but de concevoir un schéma d'estimation capable de déterminer le mode courant le plus probable, différentes hypothèses sont faites sur la distribution des paramètres de fautes, pour chaque mode.

4.3.3.1 Probabilité d'observation en supposant les paramètres de fautes dépendant entre eux

Dans le cas où les paramètres de fautes sont dépendants, il faut utiliser une loi de probabilité multi-variée. On fait l'hypothèse que ω suit une loi multi-normale paramétrée par le vecteur \mathbf{b} et par la matrice semi-définie positive Σ (cf. Annexe E) :

$$p(\omega|\mathbf{b}) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp^{-\frac{1}{2}((\omega-\mathbf{b})^T \Sigma^{-1}(\omega-\mathbf{b}))} \quad (4.44)$$

Il faut maintenant faire des hypothèses sur la fonction f_m pour chaque mode. Les hypothèses prises sont présentées dans le Tableau 4.1. Dans le cas du mode nominal, les paramètres de fautes doivent être rigoureusement égaux à zéro, donc ici proches de zéro. Dans le cas où le $i^{\text{ème}}$ capteur est en panne, le paramètre de faute associé c_0^i peut prendre n'importe quelle valeur alors que les autres paramètres de fautes doivent être proches de zéro. Dans le cas de mode où le $i^{\text{ème}}$ actionneur est en panne, les paramètres de fautes associés $\mathbf{b}_0^i \in \mathbb{R}^{|J(i)|}$, où \mathbf{b}_0^i est le vecteur des b_0^j associés au $i^{\text{ème}}$ actionneur et $J(i)$ correspond aux indices des éléments de \mathbf{b}_0 qui sont associés au $i^{\text{ème}}$ actionneur, et b_0^i peuvent prendre n'importe quelle valeur alors que les autres paramètres de fautes doivent être proches de zéro. Si le $i^{\text{ème}}$ actionneur est un actionneur non redondant, alors un seul paramètre de faute b_0 lui est associé et donc $|J(i)| = 1$. Si le $i^{\text{ème}}$ actionneur est un actionneur redondant, alors plusieurs paramètres de fautes b_0 peuvent lui être associé et donc $|J(i)| \geq 1$.

Mode	$f_m(\mathbf{b})$
Nominal	$\prod_j^{n_{b_0}} \delta(b_0^j) \prod_j^{n_{b_1}} \delta(b_1^j) \prod_j^{n_{c_0}} \delta(c_0^j)$
$i^{\text{ème}}$ capteur en panne	$\prod_j^{n_{b_0}} \delta(b_0^j) \prod_j^{n_{b_1}} \delta(b_1^j) \prod_{j \neq i}^{n_{c_0}} \delta(c_0^j)$
$i^{\text{ème}}$ actionneur en panne	$\prod_{j \in J(i)}^{n_{b_0}} \delta(b_0^j) \prod_{j \notin J(i)}^{n_{b_1}} \delta(b_1^j) \prod_j^{n_{c_0}} \delta(c_0^j)$

Tableau 4.1 – Hypothèse sur la fonction f_m pour chaque type de mode

4.3.3.2 Probabilité d'observation en supposant les paramètres de fautes indépendant entre eux

Pour simplifier les calculs de probabilité en ligne, on peut supposer que les paramètres de fautes sont indépendants entre eux. On peut alors modifier l'expression de la probabilité d'observation :

$$p(\omega|m) = \prod_j^{n_b} \int f_m(b^j) p(\widehat{b}^j|b^j) db^j \quad (4.45)$$

Dans ce cas, il est nécessaire d'avoir n_b lois normales. Chaque paramètre de faute \widehat{b}^i suit une loi normale de moyenne b^i et d'écart-type σ_b (cf. Annexe E) :

$$p(\widehat{b}^i|b^i) = \frac{1}{\sigma_b \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{\widehat{b}^i - b^i}{\sigma_b}\right)^2} \quad (4.46)$$

Les hypothèses faites pour la fonction f_m sont les mêmes que précédemment (cf. Tableau 4.1).

Dans le cas du mode nominal, la probabilité d'observation est la suivante :

$$p(\omega|m) = \prod_j^{n_b} \frac{1}{\sqrt{2\pi}\sigma_{b_j}} e^{-\frac{1}{2} \left(\frac{\widehat{b}^j}{\sigma_{b_j}}\right)^2} \quad (4.47)$$

Dans le cas où le i^{me} actionneur est en panne, la probabilité d'observation est la suivante :

$$p(\omega|m) = \prod_{j \in J(i)}^{n_{b_0}} \frac{1}{\sqrt{2\pi}\sigma_{b_0^j}} e^{-\frac{1}{2} \left(\frac{\widehat{b}_0^j}{\sigma_{b_0^j}}\right)^2} \prod_{j \in i}^{n_{b_1}} \frac{1}{\sqrt{2\pi}\sigma_{b_1^j}} e^{-\frac{1}{2} \left(\frac{\widehat{b}_1^j}{\sigma_{b_1^j}}\right)^2} \prod_j^{n_{c_0}} \frac{1}{\sqrt{2\pi}\sigma_{c_0^j}} e^{-\frac{1}{2} \left(\frac{\widehat{c}_0^j}{\sigma_{c_0^j}}\right)^2} \quad (4.48)$$

Dans le cas où le i^{me} capteur est en panne, la probabilité d'observation est la suivante :

$$p(\omega|m) = \prod_j^{n_{b_0}} \frac{1}{\sqrt{2\pi}\sigma_{b_0^j}} e^{-\frac{1}{2} \left(\frac{\widehat{b}_0^j}{\sigma_{b_0^j}}\right)^2} \prod_j^{n_{b_1}} \frac{1}{\sqrt{2\pi}\sigma_{b_1^j}} e^{-\frac{1}{2} \left(\frac{\widehat{b}_1^j}{\sigma_{b_1^j}}\right)^2} \prod_{j \in i}^{n_{c_0}} \frac{1}{\sqrt{2\pi}\sigma_{c_0^j}} e^{-\frac{1}{2} \left(\frac{\widehat{c}_0^j}{\sigma_{c_0^j}}\right)^2} \quad (4.49)$$

Remarque 4.4

Pour calculer la probabilité d'observation, la covariance des paramètres de fautes est nécessaire. Il faut donc utiliser une méthode stochastique pour l'estimation de l'état et des paramètres de fautes.

4.3.4 Application au drone Altimum 4

Pour le modèle complet, respectivement longitudinal, de l'Altimum 4, on définit 15, respectivement 9, modes ; un mode nominal, un mode lorsque le moteur est en

panne, un mode lorsque l'un des élevons est en pannes et 12, respectivement 6, modes dégradés capteurs, un pour chaque capteur en panne.

Que ce soit avec le modèle longitudinal ou avec le modèle complet, on définit 3 contrôleurs ; un contrôleur nominal pour le mode nominal et les modes dégradés capteurs, un contrôleur capable de gérer les pannes moteur et un contrôleur capable de gérer les pannes élevons.

Concernant la matrice de transition T_m , les valeurs suivantes sont prises pour la calculer :

- le moteur a un taux de défaillance de 10^{-7} ;
- les élevons ont un taux de défaillance de 10^{-8} ;
- les capteurs ont un taux de défaillance de 10^{-6} ;
- les actionneurs et les capteurs ont un taux de “réparation” de 10^{-9} .

Le taux de “réparation” permet de prendre en compte les pannes intermittentes dans la modélisation.

Remarque 4.5

On choisit d'avoir un mode pour chaque capteur en panne. Ce choix permet notamment de présenter des explications à un opérateur sur le comportement du système. Il est néanmoins possible d'avoir un mode global prenant en compte le mode nominal et les modes dégradés capteurs étant donné que le contrôleur associé à ces modes est le même : le contrôleur nominal.

4.4 Conclusion

L'architecture générale de la méthode de FTC proposée présente trois fonctions principales :

- une FDD fournissant une estimation de l'état et des paramètres de fautes,
- une adaptation de gains de retour d'état estimé dépendante des paramètres de fautes, et
- un mécanisme de commutation.

L'estimation et l'adaptation de gains doivent se baser sur un modèle linéaire à temps discret qui prend en compte les fautes alors que le mécanisme de commutation peut se baser soit sur un modèle POMDP qui considère le résultat de l'estimation non seulement en terme de moyenne mais aussi en terme de variance, soit directement sur le modèle linéaire, soit sur les deux. Les chapitres suivants sont dédiés à l'utilisation de ces modèles pour assurer les trois fonctions.

Cette méthode générique a fait l'objet d'une publication [Boche 2016].

Développement d'une méthode active de détection et de diagnostic des fautes

Sommaire

5.1 Développement d'un filtre de Kalman adaptatif avec réouverture de la covariance	89
5.1.1 La méthode d'estimation retenue : le filtre de Kalman adaptatif	89
5.1.2 Utilisation de signatures de pannes pour isoler la faute	90
5.1.3 Adaptation de la covariance du filtre	93
5.1.4 Application de contraintes	95
5.1.5 Réglage du filtre de Kalman	96
5.2 Développement d'une méthode active	98
5.2.1 Excitation des commandes redondantes pour l'observabilité .	98
5.2.2 Excitation des commandes pour l'Altimum 4	99
5.3 Utilisation des données de l'estimation pour la détection et le diagnostic des fautes	100
5.4 Conclusion	101

Résumé

CE chapitre vise à présenter la méthode de détection et diagnostic de fautes. L'observabilité du système étant vérifiée, la première partie de ce chapitre vise à présenter l'estimateur développé. Cet estimateur est un filtre de Kalman adapté de façon à pouvoir estimer aussi les paramètres de fautes subissant des discontinuités, c'est à dire de façon à rendre le filtre sensible aux sauts de panne. La seconde partie décrit la méthode permettant le déclenchement de l'excitation rendant le système plus observable de façon temporaire. Finalement, la troisième partie présente la technique permettant de détecter et d'identifier la faute.

5.1 Développement d'un filtre de Kalman adaptatif avec réouverture de la covariance

5.1.1 La méthode d'estimation retenue : le filtre de Kalman adaptatif

Parmi les méthodes d'estimation présentées à la Section 3.3.2.2, le filtre de Kalman a été retenu au vu de ses avantages. Son coût de calcul est modéré, comparé à d'autres méthodes envisageables, et peut être encore réduit grâce au traitement séquentiel des mesures. Ce filtre et ses variantes semblent combiner facilité d'implémentation et qualité de l'estimation. Ils sont donc choisis pour traiter le problème de FDD relatif à cette thèse.

Les types de pannes que l'on souhaite traiter sont équivalentes à des sauts de valeurs au niveau des paramètres de fautes ce qui n'est pas considéré dans les hypothèses sous-jacentes au filtre de Kalman. En effet, les filtres de Kalman, tels que les filtres TSKF, AKF, EKF ou ETSKF, ne sont pas adaptés aux changements soudains au niveau des paramètres de fautes puisqu'ils ne couvrent pas un tel comportement. Il faut donc adapter le filtre de façon à traiter les discontinuités, c'est à dire les sauts au niveau des paramètres de fautes. Comme présenté à la Section 3.3.3, il existe des variantes permettant de résoudre le problème de pannes. Les approches multi-modèles ne sont pas retenues à cause de leur coût calculatoire trop important. La méthode retenue est donc celle du filtre de Kalman adaptatif. L'estimation de grandeurs telles que des états est ainsi rendue possible tout en assurant une surveillance des principaux capteurs et actionneurs.

Parmi les filtres adaptatifs permettant de gérer les pannes, il existe le filtre adaptatif TSKF de Wu décrit au paragraphe 3.3.3.3. Ce filtre a montré a donné de bons résultats pour l'estimation de paramètres d'efficacité \mathbf{b}_1 . Néanmoins, ici on souhaite estimer des biais \mathbf{b}_0 et des paramètres d'efficacité \mathbf{b}_1 . Or, avec ces paramètres combinés, le signal de commande \mathbf{u}_k doit varier et être excité continuellement pour permettre à l'estimateur de distinguer \mathbf{b}_0 et \mathbf{b}_1 , même pour des paramètres associés à un actionneur non-redondant. Cette méthode n'est pas adaptée à notre application pour deux raisons ; il n'est pas souhaitable de devoir exciter une commande non redondante et il n'est pas possible d'avoir confiance en l'estimation des paramètres de fautes. En effet, avec les facteurs d'oublis, l'ancienne information est oubliée uniformément dans le temps et dans l'espace, ce qui revient à avoir une grande covariance, et donc peu de confiance en l'estimation des paramètres de fautes, tout au long de la simulation. Donc l'utilisation de facteurs d'oublis au niveau de la prédiction de la covariance des paramètres de fautes n'est pas acceptable pour le calcul des probabilités de modes (cf. Section 4.3.3).

L'adaptation proposée ici se base sur le principe suivant. Quand un capteur ou un actionneur tombe en panne, l'estimateur va délivrer une estimation erronée qui va tôt ou tard augmenter la valeur de l'innovation, c'est-à-dire l'erreur entre la mesure du capteur et la sortie estimée. Lorsque cette erreur est importante, la prédiction de la matrice de covariance $P_{k|+1|k}^b$ est mise à jour de façon à oublier

les informations antérieures et permettre une estimation des paramètres de fautes. Cette mise à jour revient à dire au filtre de ne pas faire confiance à l'estimation courante des paramètres de fautes et donc de lui laisser une chance de se corriger.

Remarque 5.1

L'adaptation a été testée sur les filtres linéaires AKF et TSKF. Néanmoins, la méthode est facilement transposable aux filtres non-linéaires EKF et ETSKF.

Les sections suivantes permettent de présenter le principe du filtre développé. La première section présente le principe permettant d'isoler la provenance de la faute. La seconde section présente l'adaptation effectuée au niveau de la covariance des paramètres de fautes. La troisième section présente le principe permettant de contraindre les valeurs des paramètres de fautes. Et finalement la dernière section propose une méthode pour régler le filtre de Kalman développé.

5.1.2 Utilisation de signatures de pannes pour isoler la faute

5.1.2.1 Principe général

De façon à ne pas oublier l'ensemble des informations sur la prédiction de la matrice de covariance des paramètres de fautes $P_{k+1|k}^b$, le principe des signatures de fautes est utilisé de façon à isoler les fautes possibles sur le système. Par ailleurs, cette isolation des fautes permet aussi d'envoyer le signal excitant 3211 lorsque la panne possible est celle d'un actionneur redondant.

Les signatures de fautes se basent sur des résidus. Un résidu est construit pour chaque mesure disponible sur le système à partir de la valeur courante de l'innovation et de la matrice de covariance fournie par le filtre :

$$r_{k+1}^i = \mu_{k+1}^i (C_i P_{k+1|k} C_i^T + R_{ii}) \mu_{k+1}^i \quad (5.1)$$

Lorsqu'une faute apparaît, le résidu peut augmenter rapidement ou lentement. Si la faute est abrupte, le résidu va augmenter rapidement. Cependant, le résidu peut aussi augmenter doucement par exemple lorsqu'un capteur dérive ou à cause de la dynamique du système. Dans le but de détecter les deux types de fautes, les seuils des résidus sont basés sur l'algorithme CUSUM présenté en Section 3.3.6.1.

Les résidus sont plus ou moins impactés en fonction du capteur ou de l'actionneur en panne. Il est donc nécessaire de construire une matrice liant les pannes des composants et les résidus. Dans cette matrice, la valeur 1 pour une signature signifie que l'une ou plusieurs des expressions suivantes sont vraies :

$$\sum_{j=k+1-n_{CS}}^{k+1} r_j^i > t(i, n_{CS}), \quad n_{CS} = \{0, \dots, 5\} \quad (5.2)$$

Où $t(i, n_{CS})$ est le seuil de la i^{th} mesure, associé à la somme cumulative de n_{CS} résidu dans le temps.

Pour les capteurs, les signatures de panne sont faciles à obtenir ; une panne capteur impacte uniquement la mesure de ce capteur, c'est donc le résidu associé au capteur en panne qui capte l'erreur. Pour les actionneurs, il faut s'intéresser à la dynamique du système pour pouvoir définir les signatures de panne. Les signatures définies peuvent se baser sur des résultats obtenus en simulation et sur les équations différentielles permettant de modéliser le système. Néanmoins, les signatures des actionneurs doivent être adaptées pour chaque système. Par ailleurs, les actionneurs redondants ont la même signature de panne. En effet, ceux-ci étant redondants, ils ont le même effet sur le système.

5.1.2.2 Signatures de pannes pour l'Altium 4

Les signatures de fautes pour le modèle complet sont données sur le Tableau 5.1.

Composant	Signature											
	r_p	r_q	r_r	$r_{a_{sx}}$	$r_{a_{sy}}$	$r_{a_{sz}}$	r_{z_B}	r_ϕ	r_θ	r_ψ	r_α	r_β
Moteur	0	0	0	1	0	0	0	0	0	0	0	0
Elevons	0	1	0	0	0	1	0	0	0	0	0	0
Elevons	1	1	0	0	0	0	0	0	0	0	0	0
IMU p	1	0	0	0	0	0	0	0	0	0	0	0
IMU q	0	1	0	0	0	0	0	0	0	0	0	0
IMU r	0	0	1	0	0	0	0	0	0	0	0	0
IMU a_{sx}	0	0	0	1	0	0	0	0	0	0	0	0
IMU a_{sy}	0	0	0	0	1	0	0	0	0	0	0	0
IMU a_{sz}	0	0	0	0	0	1	0	0	0	0	0	0
Baromètre z_B	0	0	0	0	0	0	1	0	0	0	0	0
Inclinomètre ϕ	0	0	0	0	0	0	0	1	0	0	0	0
Inclinomètre θ	0	0	0	0	0	0	0	0	1	0	0	0
Inclinomètre ψ	0	0	0	0	0	0	0	0	0	1	0	0
Capteur α	0	0	0	0	0	0	0	0	0	0	1	0
Capteur β	0	0	0	0	0	0	0	0	0	0	0	1

Tableau 5.1 – Signatures de fautes pour le modèle complet

Sur le Tableau 5.1, il y a deux lignes pour les élevons. Ces deux lignes correspondent en fait à un “ou”, soit on a la signature prenant en compte r_q et $r_{a_{sz}}$, soit on a la signature prenant en compte r_p et r_q . De plus, comme attendu, tous les élevons ont la même signature de pannes.

Pour les actionneurs, il faut s'intéresser à la dynamique du drone pour pouvoir définir les signatures de panne. Les signatures définies ici se basent sur des résultats obtenus en simulation et sur les équations différentielles permettant de modéliser le drone. L'Annexe A fournit les équations différentielles ainsi que les matrices A , B et C du modèle linéarisé utilisées lors des simulations. En se basant sur ces matrices, il est possible d'expliquer les signatures de pannes des actionneurs. L'actionneur

moteur a une dynamique lente, par conséquent on s'intéresse uniquement à son effet direct, c'est-à-dire à l'effet qu'on peut observer sur la mesure via la matrice C . D'après la matrice C , l'état δ_x a un effet sur la mesure de l'accélération spécifique horizontale a_{sx} . C'est donc le résidu associé au capteur a_{sx} de l'IMU qui capte en premier une panne moteur. Les actionneurs élevons ont des dynamiques rapides, par conséquent on s'intéresse à leur effet direct et à leur effet indirect, c'est-à-dire à l'effet qu'on peut observer sur l'état via la matrice A . D'après la matrice C , l'effet prédominant provient de l'état δ_e sur la mesure de l'accélération spécifique verticale a_{sz} . D'après la matrice A , l'état δ_e a un effet prédominant sur l'état q et l'état δ_a a un effet prédominant sur l'état p . Par conséquent, une panne élevon impacte les mesures p , q et a_{sz} . C'est donc les résidus associés aux capteurs a_{sz} , q et p de l'IMU qui captent en premier une panne élevon.

Que ce soit pour le modèle longitudinal ou le modèle complet, chaque composant a sa propre signature de faute, excepté pour le moteur et le capteur a_{sx} de l'IMU. En effet, pour ces deux fautes l'accélération spécifique a_{sx} est impactée en premier lieu. Pour pallier l'ambiguïté soulevée par ces deux composants, la mesure du capteur a_{sx} est ignorée -sa mesure n'est plus considérée dans le filtre- durant au moins $2s$ si cette signature de faute est obtenue. Cette action permet de distinguer une panne moteur, d'une panne sur le capteur a_{sx} de l'IMU grâce au résidu r_{V_x} : si r_{V_x} dépasse son seuil, le moteur est supposé en panne ; si le résidu reste en dessous de son seuil, le capteur mesurant a_{sx} est supposé en panne. En effet, une panne moteur a un effet indirect sur la vitesse V_x . D'après la matrice A , une panne moteur impacte l'état u .

Il est possible que plusieurs résidus dépassent leurs seuils et donc que plusieurs fautes soient supposées. Comme l'hypothèse de panne unique est faite, les résidus qui dépassent leurs seuils sont comparés en fonction de leur intensité pour choisir le composant supposé en panne. Pour cela, le résidu ayant la plus grande magnitude vis à vis du rapport entre le résidu et son seuil est choisi.

Les seuils sont déterminés de façon expérimentale en utilisant un filtre de Kalman classique, c'est-à-dire sans paramètre de faute, sans apparition de panne. 500 simulations sont réalisées avec des séquences de bruits de mesures différentes à chaque fois et pour lesquelles les résidus sont calculés. Les seuils sont alors calculés de la façon suivante :

- le seuil $t(i, 0)$ est égal à la valeur maximale du résidu i obtenue parmi les 500 simulations et
- le seuil $t(i, n_{CS})$ est égal à la valeur maximale de la somme consécutive de $n_{CS} - 1$ résidu i obtenue parmi les 500 simulations.

Ainsi, ces seuils sont supposés ne pas être franchis lorsqu'il n'y a pas de panne.

Remarque 5.2

Lorsque la signature de panne est celle d'un élevon, le signal 3211 est envoyé au niveau des commandes redondantes.

5.1.3 Adaptation de la covariance du filtre

5.1.3.1 Adaptation du filtre AKF

La prédiction de la matrice de covariance est de la forme suivante :

$$P_{k+1|k} = \begin{bmatrix} V_{x^1} & \cdots & C_{x^1, b_0^1} & \cdots & C_{x^1, b_1^1} & \cdots & C_{x^1, c_0^1} & \cdots & C_{x^1, c_0^{nc_0}} \\ \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ C_{b_0^1, x^1} & \vdots & V_{b_0^1} & \cdots & C_{b_0^1, b_1^1} & \cdots & C_{b_0^1, c_0^1} & \cdots & C_{b_0^1, c_0^{nc_0}} \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ C_{b_1^1, x^1} & \vdots & C_{b_1^1, b_0^1} & \vdots & V_{b_1^1} & \cdots & C_{b_1^1, c_0^1} & \cdots & C_{b_1^1, c_0^{nc_0}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ C_{c_0^1, x^1} & \vdots & C_{c_0^1, b_0^1} & \vdots & C_{c_0^1, b_1^1} & \vdots & V_{c_0^1} & \cdots & C_{c_0^1, c_0^{nc_0}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ C_{c_0^{nc_0}, x^1} & \cdots & C_{c_0^{nc_0}, b_0^1} & \cdots & C_{c_0^{nc_0}, b_1^1} & \cdots & C_{c_0^{nc_0}, c_0^1} & \cdots & V_{c_0^{nc_0}} \end{bmatrix} \quad (5.3)$$

Où V_x correspond à la variance de la variable x et $C_{x,y}$ correspond à la covariance des variables x et y .

Lorsqu'un composant est supposé en panne, les variances et les covariances des paramètres de fautes associés au composant sont mise à jour. Les covariances sont mises à zéros alors que la valeur des variances sont augmentées.

Par exemple, si un actionneur est associé aux paramètres de fautes b_0^1 et b_1^1 , la mise à jour est faite de la façon suivante :

$$P_{k+1|k}^{reset} = \begin{bmatrix} V_{x^1} & \cdots & 0 & \cdots & 0 & \cdots & C_{x^1, c_0^1} & \cdots & C_{x^1, c_0^{nc_0}} \\ \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \vdots & \nu_{b_0^1} & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \vdots & 0 & \vdots & \nu_{b_1^1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ C_{c_0^1, x^1} & \vdots & 0 & \vdots & 0 & \vdots & V_{c_0^1} & \cdots & C_{c_0^1, c_0^{nc_0}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ C_{c_0^{nc_0}, x^1} & \cdots & 0 & \cdots & 0 & \cdots & C_{c_0^{nc_0}, c_0^1} & \cdots & V_{c_0^{nc_0}} \end{bmatrix} \quad (5.4)$$

Où $\nu_{b_0^1}$ et $\nu_{b_1^1}$ valent :

$$\nu_{b_0^1} = \left(\frac{u_{max} - u_{min}}{5} \right)^2 \quad (5.5)$$

$$\nu_{b_1^1} = \left(\frac{1}{5} \right)^2 \quad (5.6)$$

Ces valeurs ont été choisies de façon à couvrir l'ensemble des cas de panne possible. D'après la règle des 5 sigmas, l'estimation de b_1^1 doit respecter la règle suivante :

$$b_1^1 - 5\sigma \leq \hat{b}_1^1 \leq b_1^1 + 5\sigma \quad (5.7)$$

b_1^1 peut prendre sa valeur entre 0 et 1, ainsi le saut maximal qui peut apparaître au niveau de ce paramètre de faute est de 1. Supposons qu'à l'instant où la panne survient $\hat{b}_1^1 = 0$ et $b_1^1 = 1$. Alors, pour respecter la règle des 5 sigmas, il faut que :

$$1 - 5\sigma = 0 \quad (5.8)$$

D'où la valeur de $\nu_{b_1^1}$. Le raison pour le choix de $\nu_{b_0^1}$ est équivalent.

La valeur de $\nu_{c_0^i}$ est choisie de façon arbitraire. En effet, les capteurs peuvent en théorie prendre n'importe quelle valeur réelle, il est donc difficile de couvrir l'ensemble des cas de panne possible. On prend donc $\nu_{c_0^i} = (\frac{1}{5})^2$.

5.1.3.2 Adaptation du filtre TSKF

Avec le filtre TSKF, la prédiction de la matrice de covariance des paramètres de fautes $P_{k+1|k}^b$ est séparée de la matrice de covariance de l'état. La matrice de covariance $P_{k+1|k}^b$ est de la forme suivante :

$$P_{k+1|k}^b = \begin{bmatrix} V_{b_0^1} & \cdots & C_{b_0^1, b_1^1} & \cdots & C_{b_0^1, c_0^1} & \cdots & C_{b_0^1, c_0^{nc_0}} \\ \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ C_{b_1^1, b_0^1} & \vdots & V_{b_1^1} & \cdots & C_{b_1^1, c_0^1} & \cdots & C_{b_1^1, c_0^{nc_0}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ C_{c_0^1, b_0^1} & \vdots & C_{c_0^1, b_1^1} & \vdots & V_{c_0^1} & \cdots & C_{c_0^1, c_0^{nc_0}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ C_{c_0^{nc_0}, b_0^1} & \cdots & C_{c_0^{nc_0}, b_1^1} & \cdots & C_{c_0^{nc_0}, c_0^1} & \cdots & V_{c_0^{nc_0}} \end{bmatrix} \quad (5.9)$$

De façon similaire au filtre AKF, lorsqu'un composant est supposé en panne, les variances et les covariances des paramètres de fautes associés au composant sont mise à jour. Les covariances sont mises à zéros alors que les variances sont augmentées.

Par exemple, si un actionneur est associé aux paramètres de fautes b_0^1 et b_1^1 , la

mise à jour est faite de la façon suivante :

$$P_{k+1|k}^b \stackrel{reset}{=} \begin{bmatrix} \nu_{b_0^1} & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \vdots & \nu_{b_1^1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & \vdots & 0 & \vdots & V_{c_0^1} & \cdots & C_{c_0^1, c_0^{n_{c_0}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ 0 & \cdots & 0 & \cdots & C_{c_0^{n_{c_0}}, c_0^1} & \cdots & V_{c_0^{n_{c_0}}} \end{bmatrix} \quad (5.10)$$

Contrairement au filtre AKF, d'autres paramètres doivent être mis à jour lorsque la matrice de covariance $P_{k+1|k}^b$ est mise à jour. En effet d'après les équations du filtre TSKF (cf. Section 3.3.3.2), il faut faire la mise à jour des paramètres suivants :

- $\beta_{k+1|k}$ qui dépend de la matrice de covariance $P_{k+1|k}^b$;
- $\tilde{P}_{k+1|k}^x$ qui dépend de la matrice de covariance $P_{k+1|k}^b$ et de la prédiction de la matrice $\beta_{k+1|k}$;
- $\tilde{\mathbf{x}}_{k+1|k}$ qui dépend de la prédiction de la matrice $\beta_{k+1|k}$;
- $\hat{\mathbf{x}}_{k+1|k}$ qui dépend de la prédiction de l'état modifié sans biais $\tilde{\mathbf{x}}_{k+1|k}$ et ;
- $\tilde{\boldsymbol{\mu}}_{k+1}$ qui dépend aussi de la prédiction de l'état modifié sans biais $\tilde{\mathbf{x}}_{k+1|k}$.

Remarque 5.3

Avec le modèle complet de l'Altimum 4, le nombre de paramètres de fautes (20 paramètres de fautes) est relativement élevé par rapport au nombre de variables d'état du problème d'origine (15 états). Le filtre peut donc être coûteux en temps de calcul. L'utilisation du filtre TSKF semble plus adaptée au problème.

5.1.4 Application de contraintes

Le paramètre de faute $\mathbf{b}_1 \in [0, 1]^{n_{b_1}}$ est borné. Or avec les filtres de Kalman, aucune contrainte ne peut être appliquée sur le paramètre \mathbf{b}_1 . Pour de meilleures performances au niveau du filtre, il est possible de contraindre ce paramètre, prenant ainsi en compte une connaissance exogène sur le système.

L'approche par projection permet d'estimer de manière optimale sous contraintes sur l'état (cf. Section 3.3.4). Les variables b_1^i du vecteur \mathbf{b}_1 sont testées à chaque pas de temps. Trois scénarios sont envisageables :

- $0 \leq b_1^i \leq 1$, l'inégalité est vérifiée donc la contrainte n'est pas prise en compte ;
- $b_1^i < 0$, l'inégalité n'est pas vérifiée, on applique une contrainte d'égalité à la borne : $b_1^i = 0$;
- $b_1^i > 1$, l'inégalité n'est pas vérifiée, on applique une contrainte d'égalité à la borne : $b_1^i = 1$.

Si une ou plusieurs variables b_1^i ne respectent pas l'inégalité, il est possible de construire une matrice D_k et un vecteur \mathbf{d}_k de façon à appliquer la contrainte d'éga-

lité :

$$D_k \begin{bmatrix} \mathbf{x}_{k|k} \\ \mathbf{b}_{0k|k} \\ \mathbf{b}_{1k|k} \\ \mathbf{c}_{0k|k} \end{bmatrix} = \mathbf{d}_k \quad (5.11)$$

La solution de la contrainte d'égalité est donnée par les équations suivantes (cf. Section 3.3.4) :

$$\mathbf{x}_{k|k}^c = \mathbf{x}_{k|k}^{nc} + L_k(\mathbf{d}_k - D_k \mathbf{x}_{k|k}^{nc}) \quad (5.12)$$

$$L_k = P_{k|k}^{nc} D_k^T (D_k P_{k|k}^{nc} D_k^T)^{-1} \quad (5.13)$$

L'équation de contrainte associée à la covariance (cf. Équation 3.56) n'est pas prise en compte. Cette équation corrige la matrice de covariance en diminuant sa valeur. L'effet est donc inverse à celui que l'on souhaite appliquer lorsque l'on suppose l'apparition d'une faute. Ainsi, la prise en compte de cette équation peut annuler l'effet permettant de rendre le filtre sensible aux fautes.

L'application des contraintes au filtre TSKF peut se faire grâce aux trois équations suivantes :

$$\mathbf{b}_{k|k}^c = \mathbf{b}_{k|k}^{nc} + L_k^b(\mathbf{d}_k^b - D_k^b \mathbf{b}_{k|k}^{nc}) \quad (5.14)$$

$$L_k^b = P_{k|k}^{b\ nc} D_k^{bT} (D_k^b P_{k|k}^{b\ nc} D_k^{bT})^{-1} \quad (5.15)$$

$$\mathbf{x}_{k|k}^c = \tilde{\mathbf{x}}_{k|k} + \beta_{k|k} \mathbf{b}_{k|k}^c \quad (5.16)$$

Où

$$D_k^b \begin{bmatrix} \mathbf{b}_{0k|k} \\ \mathbf{b}_{1k|k} \\ \mathbf{c}_{0k|k} \end{bmatrix} = \mathbf{d}_k^b \quad (5.17)$$

5.1.5 Réglage du filtre de Kalman

Le travail de réglage du filtre de Kalman est un passage délicat pour assurer le bon fonctionnement du filtre. Il faut donc choisir les valeurs des matrices de covariance du bruit de mesure R et du bruit de processus Q^x et Q^b , ainsi que les valeurs des états à l'instant initial $x_{0|0}$ et $b_{0|0}$ et de la confiance qu'on leur associe au travers de la matrice de covariance des erreurs d'estimation $P_{0|0}$ et $P_{0|0}^b$.

La matrice R traduit la confiance que l'on associe aux mesures. Elle dépend donc des spécifications associées aux capteurs utilisés. Les valeurs des écarts types σ étant connues, R est définie de la façon suivante :

$$R = \text{diag}[\sigma_p^2, \sigma_q^2, \sigma_r^2, \sigma_{a_{sx}}^2, \sigma_{a_{sy}}^2, \sigma_{a_{sz}}^2, \sigma_\psi^2, \sigma_\phi^2, \sigma_\theta^2, \sigma_\alpha^2, \sigma_\beta^2, \sigma_{z_B}^2, \sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_{V_x}^2, \sigma_{V_y}^2, \sigma_{V_z}^2] \quad (5.18)$$

avec $\sigma_p = 0.3675 \text{deg/s}$, $\sigma_q = 0.216 \text{deg/s}$, $\sigma_r = 0.2313 \text{deg/s}$, $\sigma_{a_{sx}} = 0.0035$, $\sigma_{a_{sy}} = 0.0035$, $\sigma_{a_{sz}} = 0.1$, $\sigma_\psi = 0.5 \text{deg}$, $\sigma_\phi = 0.1 \text{deg}$, $\sigma_\theta = 0.1 \text{deg}$, $\sigma_\alpha = 0.2 \text{deg}$, $\sigma_\beta = 0.2 \text{deg}$, $\sigma_{z_B} = 0.1 \text{m}$, $\sigma_x = 0.1 \text{m}$, $\sigma_y = 0.1 \text{m}$, $\sigma_z = 0.1 \text{m}$, $\sigma_{V_x} = 0.0274 \text{m/s}$,

$$\sigma_{V_y} = 0.0391m/s, \sigma_{V_z} = 0.0711m/s$$

La matrice R étant connue, le bon fonctionnement du filtre va dépendre du réglage de la matrice Q et de l'état initial. Si la matrice Q n'est pas bien réglée, l'erreur entre l'état réel et l'état estimé sera incohérente avec la matrice de covariance estimée. Il est donc important de vérifier cette propriété.

La matrice de bruit du processus Q^x est plus délicate à obtenir. Elle traduit la confiance que l'on associe au modèle de prédiction. Ainsi, deux cas s'offrent à nous :

- le cas où les mesures sont générées à partir du modèle linéarisé et
- le cas où les mesures sont générées à partir du modèle non linéaire.

Lorsque que les mesures sont générées à partir du modèle linéarisé, il n'y a théoriquement aucune erreur liée à la modélisation, la matrice Q devrait donc être nulle. Néanmoins, l'utilisation d'une matrice Q nulle dans un filtre de Kalman donne très peu de poids aux mesures, ce qui n'est pas acceptable pour la gestion des pannes. Une matrice Q arbitraire est alors choisie :

$$Q^x = ts \times 0.001^2 \times \mathbb{I}_n \quad (5.19)$$

La matrice du processus des fautes Q_b est elle aussi choisie de façon arbitraire :

$$Q^b = ts \times 0.0001^2 \times \mathbb{I}_{n_b} \quad (5.20)$$

Lorsque que les mesures sont générées à partir du modèle non linéaire, les erreurs viennent de la linéarisation. Pour quantifier ses erreurs, les deux modèles linéaire et non linéaire sont simulés en parallèle, comme sur la Figure 5.1, et l'erreur entre les deux états permet alors d'avoir un écart-type.

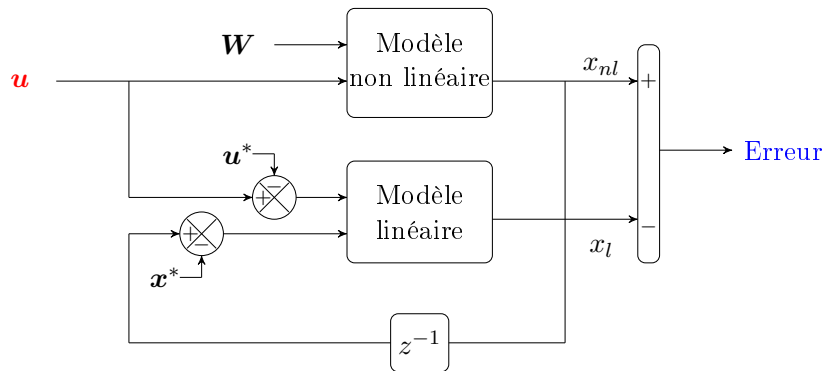


FIGURE 5.1 – Évaluation de l'erreur de linéarisation

Le modèle non linéaire peut prendre en compte l'effet du vent W . Néanmoins, on suppose ici que le vent est nul. L'erreur dépend de la commande u . Pour être robuste, il faut choisir une commande qui va s'éloigner de la commande d'équilibre.

Il est possible de vérifier la cohérence des valeurs obtenues pour Q^x . D'après les équations du filtre de Kalman l'erreur d'estimation $x - \hat{x}$ est une gaussienne centrée en zéro, de variance P . Une façon de régler la matrice Q^x est de minimiser

la quantité suivante :

$$J = \left\| P - \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T \right\|^2 \quad (5.21)$$

Dans le cas où on utilise un filtre de Kalman étendu, la matrice Q^x peut être calculée grâce à l'équation suivante :

$$Q = F_k^u Cov(\mathbf{u}) F_k^{uT} \quad (5.22)$$

Où $F_k^u = \frac{\partial f}{\partial \mathbf{u}} |_{\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k}$

Il faut maintenant choisir les valeurs des états à l'instant initial $x_{0|0}$ et $b_{0|0}$. L'état initial $x_{0|0}$ est choisi nul, le modèle étant linéarisé autour d'un point d'équilibre. Le vecteur initial des paramètres de fautes $b_{0|0}$ est aussi choisi nul. En effet, on suppose que l'on commence la simulation sans faute. Il faut maintenant choisir la confiance qu'on leur associe au travers des matrices de covariance $P_{0|0}$ et $P_{0|0}^b$. Il est classique de prendre :

$$P_{0|0} = 10P_{N|N} \quad (5.23)$$

Où $P_{N|N}$ correspond à la valeur de convergence de la covariance en cas de simulation sans panne.

5.2 Développement d'une méthode active

5.2.1 Excitation des commandes redondantes pour l'observabilité

Pour exciter les commandes redondantes et rendre identifiable les paramètres de fautes b_1 , un signal excitant est ajouté au signal de commande. Le signal utilisé est un signal 3211 (cf. Section 3.3.5). Cependant, il n'est pas souhaitable d'avoir cette excitation en permanence. Celle-ci est donc envoyée à des moments astucieux, c'est-à-dire lorsque la signature des actionneurs redondants est obtenue, de façon à rendre le système plus observable de façon ponctuelle.

La structure du système de commande est comme montrée sur la Figure 5.2. Quand une excitation sur le système est requise, le signal 3211 est ajouté aux signaux de contre réaction et de consigne.

\mathbf{u}_k^c est la commande obtenue à partir ; de la contre-réaction qui réagit en fonction de l'erreur présente sur l'état et ; d'une consigne qui permet de compenser les fautes. A_k correspond à l'amplitude A du signal excitant 3211 \mathbf{s}_k quand il est présent et vaut zéro le reste du temps. Finalement \mathbf{u}_k correspond au signal envoyé aux actionneurs.

La commande \mathbf{u}_k^c est saturée de façon à laisser passer le signal 3211 quel que soit la commande. Le signal de commande est le suivant :

$$u_k^i = f(u_k^{ci}, s_k^i, A_k) = \min\{\bar{u}^i - A_k, \max\{\underline{u}^i + A_k, u_k^{ci}\}\} + s_k^i \quad (5.24)$$

La plupart du temps, le système ne requiert aucune excitation. Dans ce cas, le

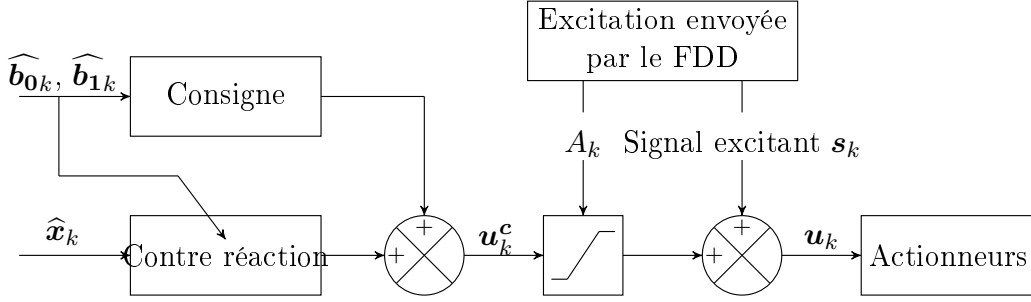


FIGURE 5.2 – Structure du système de commande

signal s_k^i est nul, l'amplitude A_k est nulle et le signal de commande est le suivant :

$$u_k^i = \min\{\bar{u}^i, \max\{\underline{u}^i, u_k^{ci}\}\} \quad (5.25)$$

Dans le cas où le signal excitant s_k est présent, il n'est pas souhaitable de rendre le système instable. Le signal excitant doit donc permettre de rendre les paramètres de fautes identifiables sans avoir d'effet néfaste sur le système, ce qui n'est pas un objectif trivial.

Remarque 5.4

Il n'est pas nécessaire de rajouter un deuxième saturateur avant les actionneurs. En effet, avec la saturation construite, le signal u_k est déjà contraint entre les bornes de l'actionneur $[\underline{u}, \bar{u}]$.

5.2.2 Excitation des commandes pour l'Altimum 4

On propose de construire le signal de façon à avoir un effet nul sur les dynamiques longitudinales et latérales, c'est-à-dire avoir $\delta_e = 0$ et $\delta_a = 0$.

Dans le cas du modèle longitudinal, il faut satisfaire l'équation suivante :

$$\frac{-(1 - b_1^2)s_k^2 - (1 - b_1^3)s_k^3}{2} = 0 \quad (5.26)$$

Le signal excitant choisi est le suivant :

$$\mathbf{s}_k = \begin{bmatrix} 0 \\ (1 - b_1^3)S_{3211} \\ -(1 - b_1^2)S_{3211} \end{bmatrix} \quad (5.27)$$

Dans le cas du modèle complet, il faut satisfaire les équations suivantes :

$$\frac{-0.5(1 - b_1^2)s_k^2 - 0.5(1 - b_1^3)s_k^3 - 0.5(1 - b_1^4)s_k^4 - 0.5(1 - b_1^5)s_k^5}{2} = 0 \quad (5.28)$$

$$\frac{0.4(1 - b_1^2)s_k^2 + 0.6(1 - b_1^3)s_k^3 - 0.4(1 - b_1^4)s_k^4 - 0.6(1 - b_1^5)s_k^5}{2} = 0 \quad (5.29)$$

Le signal excitant choisi est le suivant :

$$\mathbf{s}_k = \begin{bmatrix} 0 \\ 0.6(1 - b_1^3)(1 - b_1^4)(1 - b_1^5)S_{3211} \\ -0.4(1 - b_1^2)(1 - b_1^4)(1 - b_1^5)S_{3211} \\ -0.6(1 - b_1^2)(1 - b_1^3)(1 - b_1^5)S_{3211} \\ 0.4(1 - b_1^2)(1 - b_1^3)(1 - b_1^4)S_{3211} \end{bmatrix} \quad (5.30)$$

Remarque 5.5

Le nombre d'équations permettant de définir l'excitation des commandes est égal au rang de la matrice formée par les colonnes de la matrice B associées aux actionneurs redondants.

5.3 Utilisation des données de l'estimation pour la détection et le diagnostic des fautes

Lorsque la détection de faute est basée sur des résidus, il est courant d'avoir des fausses alarmes lorsque l'un des résidus dépasse son seuil sans qu'il y ait une erreur -ce qui par nature peut arriver, avec une probabilité faible mais non nulle. Pour robustifier la détection contre les fausses alarmes, la méthode FDD développée se base sur la modélisation discrète et donc sur la probabilité de mode.

Avec la modélisation discrète, à chaque période de temps k , le système est dans un état $s \in S$ (par exemple l'état nominal ou un état de panne de l'actionneur i). Une action est prise (par exemple utilisation du contrôleur nominal), ce qui conduit le système à un nouvel état $s' \in S$. Cet état s' n'est pas connu mais des observations, les estimations des paramètres de fautes, permettent d'estimer cet état. On peut construire un vecteur, nommé état de croyance $\boldsymbol{\rho}_k$, composé des probabilités d'être dans un certain état. Ce vecteur est mis à jour grâce à la règle de Bayes pour chaque état $\boldsymbol{\rho}_k(s)$ en fonction de l'action choisie et de l'observation.

$$\boldsymbol{\rho}_{k+1}^{a,\omega}(s) = \frac{p(\omega|s)p(s|\boldsymbol{\rho}_k, a)}{p(\omega|\boldsymbol{\rho}_k, a)} \quad (5.31)$$

$$\text{Avec } \begin{cases} p(s|\boldsymbol{\rho}_k, a) = \sum_{s'} p(s|s', a)\boldsymbol{\rho}_k(s') \\ p(\omega|\boldsymbol{\rho}_k, a) = \sum_{s'} p(\omega|s')p(s'|\boldsymbol{\rho}_k, a) \end{cases}$$

Comme expliqué à la Section 4.3 on est dans un cas particulier où les actions n'ont pas d'influence sur les probabilités des états, et il est donc possible de s'intéresser uniquement à l'état de croyance des modes :

$$\boldsymbol{\rho}_{k+1}^\omega(m) = \frac{p(\omega|m)p(m|\boldsymbol{\rho}_k)}{p(\omega|\boldsymbol{\rho}_k)} \quad (5.32)$$

$$\text{Avec } \begin{cases} p(m|\boldsymbol{\rho}_k) = \sum_{m'} p(m|m')\boldsymbol{\rho}_k(m') \\ p(\omega|\boldsymbol{\rho}_k) = \sum_{m'} p(\omega|m')p(m'|\boldsymbol{\rho}_k) \end{cases}$$

Ainsi la détection de la panne prend en compte ; l'état précédent du système, c'est-à-dire l'état de croyance à l'instant k et la probabilité d'observation. Or, comme vu à au Chapitre 4, la probabilité d'observation dépend des valeurs d'estimation des paramètres de fautes, mais aussi de la confiance qu'on leurs associe.

L'état de croyance a plusieurs objectifs :

- détecter et isoler la panne ;
- robustifier la détection face aux fausses alarmes ;
- être utilisé pour le choix du contrôleur.

5.4 Conclusion

La Figure 5.3 montre le processus de détection et d'identification de la faute.

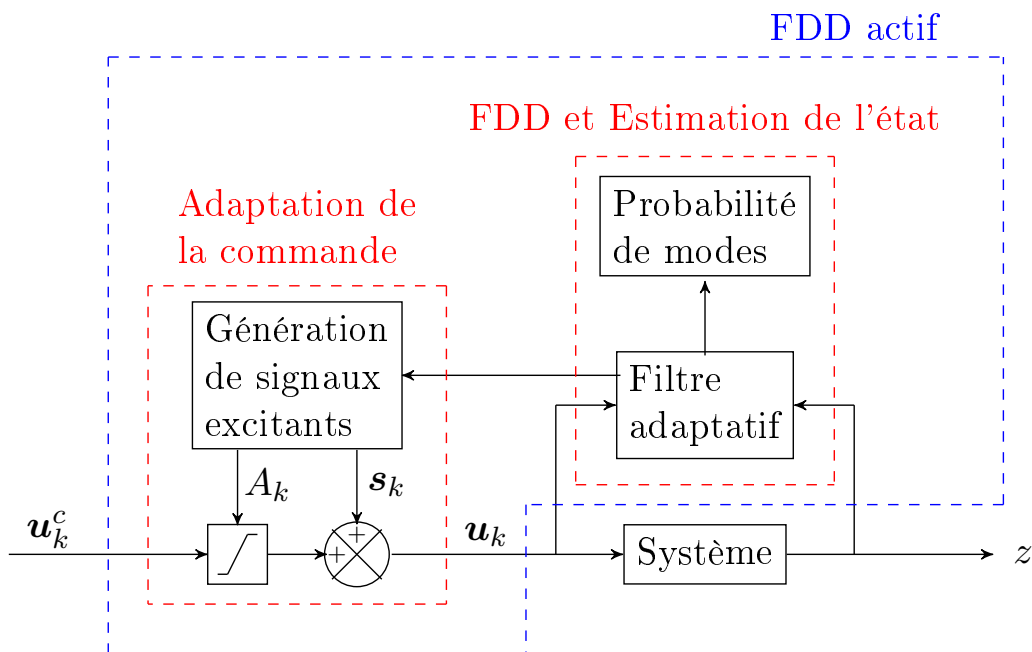


FIGURE 5.3 – Schéma du processus de détection et d'identification de la faute

Pour la méthode FDD développée, on utilise des filtres de Kalman adaptatif, le filtre AKF et TSKF. Néanmoins, il est envisageable d'utiliser d'autres types de filtres, tant que celui-ci calcule une distribution de probabilités pour les paramètres de fautes. En effet, la méthode de détection développée utilise la covariance de l'estimation pour calculer une probabilité de mode, et pour ensuite détecter la faute. Ainsi, l'utilisation d'une méthode d'estimation déterministe, telle qu'un observateur, n'est donc pas envisageable dans notre étude. Pour une meilleure précision, il est possible d'utiliser un filtre permettant de traiter les systèmes non linéaires. L'EKF et l'ETSKF, version non linéaire du filtre de Kalman et du TSKF, sont facilement adaptables à la méthode FDD développée.

Par ailleurs, la méthode FDD doit être facilement transférable à un autre système

complexe, c'est-à-dire qu'elle doit être facile à implémenter et simple à ajuster. Une adaptation complète pour chaque application n'est pas acceptable d'un point de vue industriel. Pour la méthode développée, il faut pour chaque système :

- définir les paramètres de fautes en fonction de l'observabilité du système ;
- régler le filtre de Kalman, c'est-à-dire la matrices R en fonction des capteurs disponibles, la matrice Q de façon à avoir un bon fonctionnement pour le filtre, ainsi que l'état initial $x_{0|0}$ et $P_{0|0}$;
- définir les signatures de fautes pour le système ;
- régler les seuils permettant de définir ces signatures de fautes.

Les adaptations à effectuer pour un autre système complexe sont par conséquent assez peu nombreuses pour la méthode développée. Il est donc envisageable de l'appliquer à un autre système.

Validation de la méthode active de détection et de diagnostic des fautes

Sommaire

6.1	Démonstration expérimentale de la pertinence du modèle de fautes	105
6.1.1	Identifiabilité des paramètres b_0 des actionneurs redondants	105
6.1.2	Identifiabilité des paramètres b_1 des actionneurs redondants	105
6.2	Critères d'évaluation	106
6.2.1	Critère d'évaluation quantitative pour l'estimation	106
6.2.2	Critère d'évaluation quantitative pour la détection et l'identification	109
6.3	Validation sur mesures générées à partir du modèle linéaire	109
6.3.1	Cas où l'on considère uniquement les pannes actionneurs	110
6.3.2	Cas où l'on considère uniquement les pannes capteurs	125
6.3.3	Cas où l'on considère les pannes actionneurs et capteurs	134
6.4	Validation sur mesures générées à partir du modèle non linéaire	137
6.4.1	Cas où l'on considère uniquement les pannes actionneurs	137
6.4.2	Cas où l'on considère uniquement les pannes capteurs	143
6.5	Conclusion	143

Résumé

CE chapitre vise à valider notre méthode de détection et diagnostique de fautes. Dans un premier temps, la méthode est validée en générant les mesures à partir du modèle linéaire. Ces simulations permettent de valider la méthode lorsque le modèle disponible du système est très représentatif de la réalité, c'est-à-dire lorsqu'il y a peu d'incertitude. Dans un second temps, la méthode est validée en générant les mesures à partir du modèle non linéaire. Le but est d'évaluer la méthode face aux incertitudes de modèle.

Avant d'implémenter le code sur le système réel, des simulations sont effectuées en utilisant le logiciel Matlab/Simulink pour tester la méthode. Afin de générer des mesures, le modèle de simulation développé sur Simulink est utilisé. Par la suite, les courbes peuvent provenir de deux types de simulations : soit en ayant des mesures générées par le modèle non linéaire, soit en ayant des mesures générées par le modèle linéarisé autour d'un point de vol. Néanmoins, il est précisé dans chaque cas la nature de la simulation.

6.1 Démonstration expérimentale de la pertinence du modèle de fautes

Deux principaux résultats sont montrés ici :

- la non identifiabilité des biais b_0 associés à des actionneurs redondants, c'est-à-dire l'impossibilité de différencier les b_0 ;
- l'identifiabilité partielle des facteurs d'échelle b_1 associés à des actionneurs redondants, c'est-à-dire la nécessité d'exciter le système pour rendre les b_1 différentiable.

Les résultats montrés dans cette section ont été obtenus en utilisant le filtre TSKF et en générant les mesures à partir du modèle linéarisé.

6.1.1 Identifiabilité des paramètres b_0 des actionneurs redondants

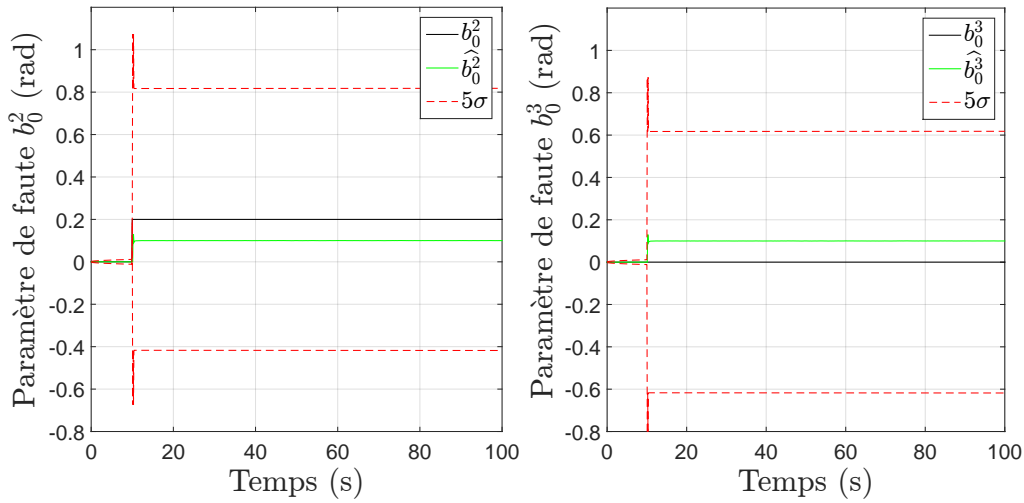
Il est possible d'observer la non identifiabilité des b_0 des actionneurs redondants lors d'une simulation. La simulation utilisée est la suivante : avec le modèle longitudinal, l'élève gauche tombe en panne au bout de 10s avec un biais de 0.2. Les paramètres de faute b_0 associés aux élèves sont alors comme montrés sur la Figure 6.1.

La valeur du biais de b_0^2 est partagée, au niveau de l'estimation, sur les deux paramètres b_0^2 et b_0^3 , ce qui est bien conforme avec le résultat obtenu avec le critère de Kalman. De plus, la covariance reste élevée, ce qui traduit aussi un manque d'observabilité. Ainsi, il n'est pas utile d'avoir plusieurs paramètres b_0 .

6.1.2 Identifiabilité des paramètres b_1 des actionneurs redondants

Les paramètres b_1 sont partiellement identifiables, il faut les exciter pour les rendre identifiable. A partir d'une simulation, différentes méthodes peuvent montrer cette indentifiabilité partielle. La simulation utilisée est la suivante : avec le modèle longitudinal, l'élève gauche tombe en panne au bout de 10s avec une perte d'efficacité de 80%. La Figure 6.2 montre l'estimation des paramètres b_1 sans envoyer de signal excitant et la Figure 6.3 montre l'estimation des paramètres b_1 en envoyant le signal excitant.

Sans le signal excitant, la valeur du facteur d'échelle de b_1^2 est partagée, au niveau de l'estimation, sur les deux paramètres b_1^2 et b_1^3 , ce qui traduit bien l'inobservabilité du système. De plus, la covariance reste élevée, ce qui traduit aussi un manque

FIGURE 6.1 – Paramètres de fautes b_0 associés aux élevons

d'observabilité. A l'opposé, avec le signal excitant, l'estimation des paramètres b_1^2 et b_1^3 est la bonne. De plus, la covariance est significative réduite après la réouverture, ce qui traduit une bonne observabilité.

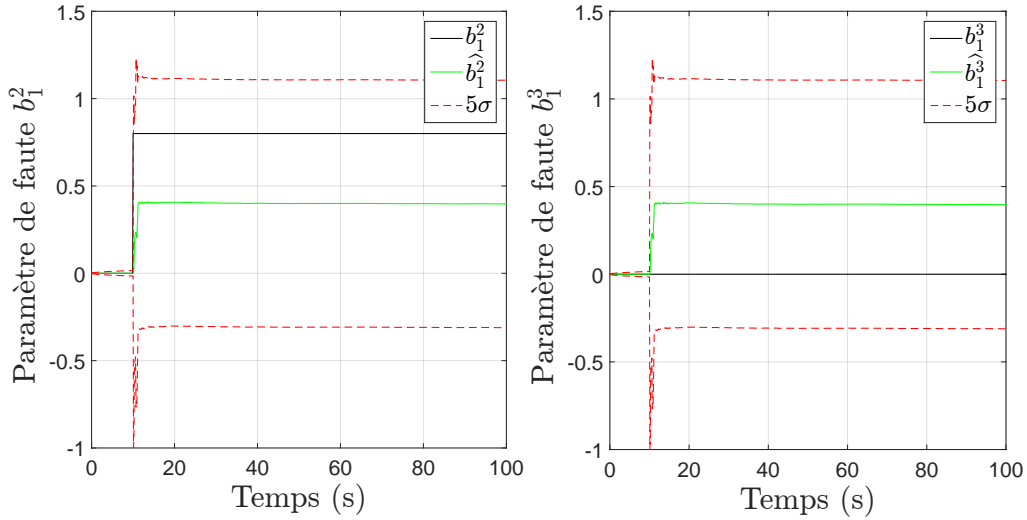
L'observabilité peut aussi être traduite via le calcul du grammien. Le grammien $W_o(k_0, k_0 + s)$, avec $s = 3$ et k_0 étant le temps à partir duquel le signal 3211 est envoyé, vaut 4.10^{10} . A contrario, sans le signal 3211, au même instant, le grammien vaut 6.10^{-15} .

6.2 Critères d'évaluation

6.2.1 Critère d'évaluation quantitative pour l'estimation

Dans un premier temps, il faut vérifier que le filtre de Kalman est bien réglé. Pour cela, la matrice de bruit Q doit être bien adaptée au problème. En statistique, la règle des trois sigmas indique que pour une loi normale, la quasi-totalité (99,73%) des valeurs se situent dans un intervalle centré autour de la moyenne et dont les bornes se situent à 3 écarts-types de part et d'autre. Ainsi, dans le cas du filtre de Kalman, il faut que l'erreur entre l'état réel et l'état estimé se situent dans les bornes $[-3\sqrt{P}, 3\sqrt{P}]$. Si le filtre respecte cette condition, on dit qu'il est intègre. Dans un second temps, il faut vérifier qu'une mauvaise initialisation de $x_{0|0}$ et $P_{0|0}$ n'a pas d'impact important sur le reste de la simulation et que le filtre de Kalman finit bien par converger.

Une fois, l'estimateur bien réglé, il est important d'évaluer sa précision quant à l'estimation de l'état et des paramètres de fautes. Une façon courante de caractériser la précision d'un estimateur consiste à calculer l'erreur quadratique moyenne, en

FIGURE 6.2 – Paramètres de fautes b_1 associés aux éleveurs sans signal excitant

anglais *Root Mean Squared Error* (RMSE). La formule est la suivante :

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (6.1)$$

Où N correspond au nombre de valeurs recueillies au cours d'une simulation et x correspond à un état ou à un paramètre de faute.

Il est nécessaire d'évaluer dans un premier temps la précision du filtre dans le cas nominal, c'est à dire sans panne capteur ou actionneur. Dans un second temps, il faut évaluer la précision du filtre en cas de panne. Dans ce cas, deux cas se distinguent ; les pannes actionneurs et les pannes capteurs.

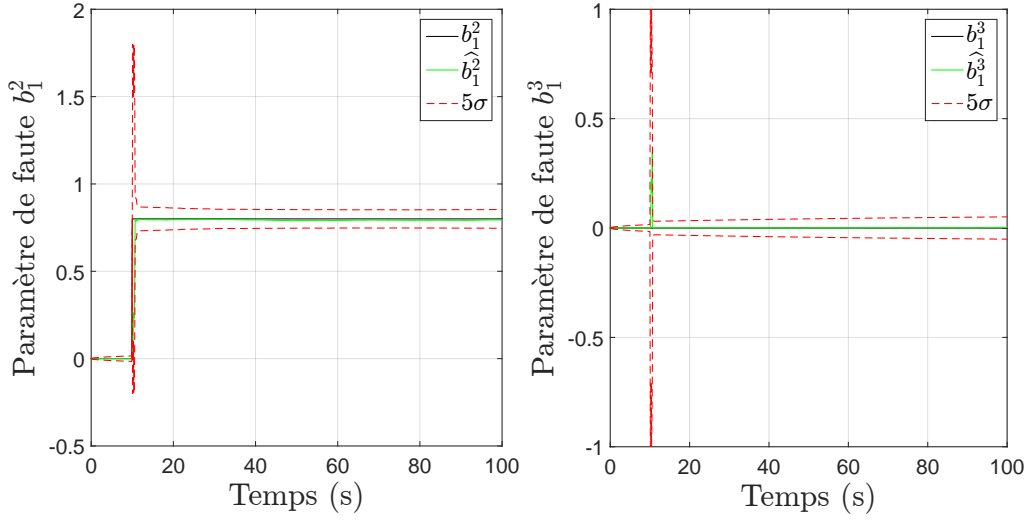
6.2.1.1 Critère d'évaluation en cas de pannes actionneurs

Les pannes actionneurs sont simulées grâce à l'équation mathématique suivante :

$$u^f = B_0 + (1 - B_1)u - B_1u^* \quad (6.2)$$

Les paramètres de fautes estimés \hat{b}_0 et \hat{b}_1 correspondent aux paramètres B_0 et B_1 respectivement. Il est donc possible de calculer les erreurs d'estimation de ces paramètres. Par ailleurs, il est important que le filtre soit précis au niveau de ces paramètres de faute, de façon à pouvoir les utiliser pour la reconfiguration du contrôleur.

Il faut aussi vérifier l'intégrité du filtre. Lorsqu'une panne survient, le filtre doit estimer précisément les paramètres de fautes b_0 et b_1 de façon à retrouver un état "nominal", c'est-à-dire non biaisé. Ainsi, en cas de panne, l'erreur entre l'état réel et

FIGURE 6.3 – Paramètres de fautes b_1 associés aux éleveurs avec le signal excitant

l'état estimé doit rester dans les bornes $[-3\sqrt{P}, 3\sqrt{P}]$, où P est la variance en cas nominal.

6.2.1.2 Critère d'évaluation en cas de pannes capteurs

Les pannes capteurs sont simulées grâce à deux équations mathématiques. En cas de blocage, l'équation est la suivante :

$$z^f = C_0 - z^* \quad (6.3)$$

Pour les autres types de pannes, l'équation est la suivante :

$$z^f = C_0 + (1 - C_1)(Cx) - C_1 z^* + v \quad (6.4)$$

L'estimation de la mesure dépend des paramètres suivant :

$$\hat{z} = \hat{c}_0 + C\hat{x} \quad (6.5)$$

Comme on peut le constater en comparant les deux équations précédentes, le paramètre de faute estimé \hat{c}_0 ne correspond pas aux paramètres C_0 et C_1 . Ainsi, pour les capteurs, il n'est pas possible de calculer le RMSE des paramètres estimés. Néanmoins, l'erreur d'estimation de la mesure peut être calculée. Pour pouvoir évaluer la précision du filtre en cas de pannes capteurs, il est aussi nécessaire de comparer l'état réel du système avec son estimation et de vérifier l'intégrité du filtre.

6.2.2 Critère d'évaluation quantitative pour la détection et l'identification

La méthode FDD doit être performante et robuste. L'évaluation de la méthode est basée sur un ensemble de définitions et d'équations présentées ci-dessous.

Les paramètres de temps ont les définitions suivantes :

- t_{faute} est l'instant où la faute apparaît.
- $t_{détection}$ est l'instant où la faute est détectée par la méthode FDD.
- T_D est la durée nécessaire à la détection de la faute, c'est-à-dire le temps entre l'instant d'apparition et l'instant de détection, $T_D = t_{détection} - t_{faute}$.

Avec ces définitions, l'évaluation quantitative est définie à partir des quatre critères suivants :

1. La performance du temps de détection définie par le temps nécessaire à la détection de la faute T_D .
2. Le taux de fausse alarme (FA), définie par le ratio entre le nombre de cas de fausse alarme n_{FA} et le nombre total de simulations Monte Carlo n_{MC} pour lesquelles il n'y a pas de faute :

$$FA\% = \frac{n_{FA}}{n_{MC}} 100 \quad (6.6)$$

3. Le taux de non détection, en anglais *Missed Detection rate* (MD), définie par le ratio entre le nombre de non détection n_{MD} et le nombre total de simulations Monte Carlo n_{MC} :

$$MD\% = \frac{n_{MD}}{n_{MC}} 100 \quad (6.7)$$

4. Le taux de mauvaise identification (MI), définie par le ratio entre le nombre de mauvaise identification n_{MI} et le nombre total de simulations Monte Carlo n_{MC} :

$$MI\% = \frac{n_{MI}}{n_{MC}} 100 \quad (6.8)$$

5. Le temps d'exécution (en μs), c'est-à-dire le coût de fonctionnement d'une méthode FDD dans un environnement temps-réel.

Remarque 6.1

Concernant la détection, on suppose qu'une panne est détectée et identifiée lorsque que la probabilité du mode associée est supérieure à 0.5.

6.3 Validation sur mesures générées à partir du modèle linéaire

Le but de cette section est d'évaluer la méthode lorsque le modèle du système disponible est très proche de la réalité. En générant les mesures à partir du modèle

linéaire, on se place dans un cas où le modèle est parfaitement connu. L'incertitude provient uniquement du bruit des capteurs. Le modèle utilisé a été linéarisé pour une trajectoire de descente, c'est-à-dire pour une vitesse air de $20m/s$ et une pente de -3° . Les matrices A , B et C utilisées sont définies dans l'Annexe A

6.3.1 Cas où l'on considère uniquement les pannes actionneurs

Dans le cas où l'on considère uniquement les pannes actionneurs, le tableau des signatures de fautes est considérablement simplifié comme montré sur le Tableau 6.1.

Composant	Signature											
	r_p	r_q	r_r	$r_{a_{sx}}$	$r_{a_{sy}}$	$r_{a_{sz}}$	r_{z_B}	r_ϕ	r_θ	r_ψ	r_α	r_β
Moteur	0	0	0	1	0	0	0	0	0	0	0	0
Elevons	0	1	0	0	0	1	0	0	0	0	0	0
Elevons	1	1	0	0	0	0	0	0	0	0	0	0

Tableau 6.1 – Signatures de fautes actionneurs pour les actionneurs

Le moteur a donc une signature unique, la panne du capteur a_{sx} n'étant plus considérée. De plus, les fautes considérées pour les actionneurs sont des pannes soudaines, qui vont par conséquent être observées rapidement par les résidus. Ainsi lorsque l'on considère uniquement les pannes actionneurs, l'algorithme CUSUM n'est pas utilisé et un seuil unique est utilisé. Ainsi dans le Tableau 6.1, la valeur 1 pour une signature signifie que l'expression suivante est vraie :

$$r_{k+1}^i > t(i, 0) \quad (6.9)$$

6.3.1.1 Comportement en mode nominal

Pour valider le filtre en cas de pannes, différents scénarios de pannes sont testés. Pour chaque type de panne, par exemple une perte d'efficacité actionneur, 500 (ou 100) simulations sont réalisées avec une séquence de bruits sur les capteurs différente pour chaque simulation. Pour avoir une base de comparaison, 500 simulations sont réalisées en mode nominal avec 500 séquences de bruits sur les capteurs et cela pour chaque filtre : le TSKF, l'AKF et l'AKF séquentiel.

Ces simulations permettent d'évaluer le taux de FA . Quel que soit le filtre utilisé, sur les 500 simulations, il n'y a pas de fausse alarme, ainsi le taux est nul, $FA\% = 0$. De plus, le filtre est intègre pour l'ensemble des simulations. Le filtre donne donc de bons résultats dans le mode nominal.

6.3.1.2 Intérêt des contraintes sur les paramètres b_1 pour l'estimation

Le but de cette section est de montrer l'intérêt de contraindre les paramètres de fautes b_1 entre 0 et 1. Les résultats montrés lors de cette section proviennent de

simulations réalisées avec le filtre adaptatif TSKF prenant en compte les contraintes sur b_1^1 et le filtre adaptatif TSKF sans prise en compte les contraintes. Pour plus de visibilité sur les courbes, la comparaison entre le filtre contraint et le filtre non contraint est réalisée pour 100 simulations pour chaque type de panne.

Remarque 6.2

Les résultats provenant des simulations utilisant le filtre adaptatif AKF contraint et non contraint sont semblables à ceux montrés lors de cette section.

Le scénario suivant est utilisé : lors d'un atterrissage une panne permanente survient sur un actionneur à 10s ; la simulation se termine au bout de 100s. La panne peut être un biais, une perte d'efficacité ou un blocage. Les figures suivantes montrent l'erreur RMSE calculé après l'apparition de la faute, en régime établi, c'est-à-dire une seconde après l'apparition de la faute, de 11s à 100s.

Lors des simulations, il y aura différents cas possibles :

- les cas de non détections MD ;
- les cas de détections pour lesquels il n'y a pas eu de réouverture de la covariance D_{ssRC} et
- les cas de détections pour lesquels il y a eu une réouverture de la covariance D_{acRC} .

Simulations de pannes élevoon 1 :

Les Figures 6.4 et 6.5 montrent les erreurs d'estimation des paramètres de fautes associés aux élevoons, c'est-à-dire b_0^2 , b_0^3 , b_1^2 , b_1^3 , b_1^4 et b_1^5 , en cas de perte d'efficacité de l'élevoon 1.

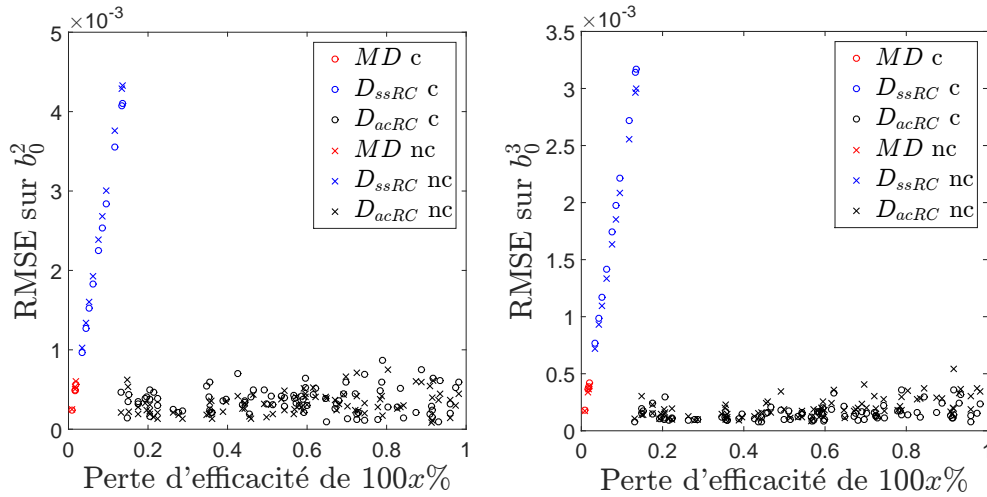


FIGURE 6.4 – Erreur d'estimation en cas de perte d'efficacité élevoon 1 sur 100 simulations

Les erreurs d'estimation sur b_0^2 et b_0^3 sont très faibles ; inférieures à 0.005 pour b_0^2 et inférieures à 0.0035 pour b_0^3 . On remarque néanmoins des erreurs plus importantes pour de petites pertes d'efficacité (inférieures à 13.5%). De plus, pour ces paramètres

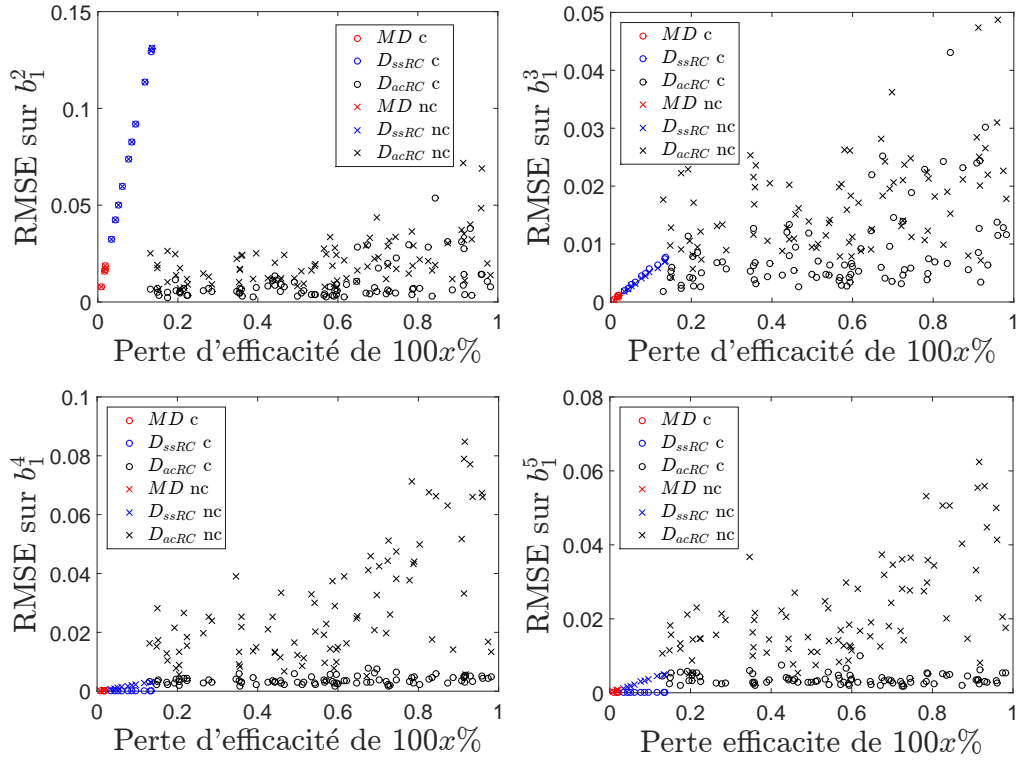


FIGURE 6.5 – Erreur d'estimation en cas de perte d'efficacité élewon 1 sur 100 simulations

la différence entre le filtre contraint et non contraint est très faible. Les erreurs d'estimations sur les paramètres b_1^2 , b_1^3 , b_1^4 et b_1^5 sont légèrement plus élevées. Pour les paramètres b_1^4 et b_1^5 , les erreurs sont plus importantes avec le filtre ne prenant pas en compte les contraintes, entre 0 et 0.08 pour b_1^4 et entre 0 et 0.07 pour b_1^5 . Alors que pour le filtre contraint, les erreurs valent entre 0 et 0.008 pour b_1^4 et entre 0 et 0.01 pour b_1^5 . Pour le paramètre b_1^3 , l'erreur est aussi légèrement supérieure avec le filtre non contraint mais reste inférieure à 0.04. Finalement, pour le paramètre b_1^2 associé à l'élewon 1 en panne, l'erreur est plus importante pour des petites pertes d'efficacité (inférieures à 13.5%). Entre 0% et 13.5%, l'erreur augmente de façon linéaire avec la perte d'efficacité. Ces simulations correspondent à des détections sans réouverture de la covariance. Pour les autres simulations, l'erreur sur b_1^2 est faible (inférieures à 0.08 pour le filtre non contraint et inférieures à 0.06 pour le filtre contraint).

Concernant la détection, il y a plusieurs cas. Pour 14 simulations pour lesquelles on a des faibles pertes d'efficacité, il y a, que ce soit avec le filtre contraint ou avec le filtre non contraint :

- quatre *MD* (pour des pertes d'efficacité entre 0.8% et 1.9%) pour lesquelles il n'y a pas eu de réouverture de la covariance
- dix détections sans réouverture de la covariance (pour des pertes d'efficacité entre 3.3% et 13.5%).

Dans le premier cas, les MD sont trop faibles pour être détectées. Dans le second cas, les valeurs des paramètres de fautes dévient légèrement de zéro en gardant une covariance faible, d'où la détection malgré de mauvaises valeurs d'estimation. Malgré tout, ces 14 simulations correspondent à des très faibles pertes d'efficacité qui auront par conséquent un très faible impact sur le comportement du drone.

Les 86 autres simulations correspondent à des détections après réouverture de la covariance. La Figure 6.6 montre le temps de détection en fonction de la perte d'efficacité. Les temps de détection sont semblables avec les deux filtres, le contraint et le non contraint.

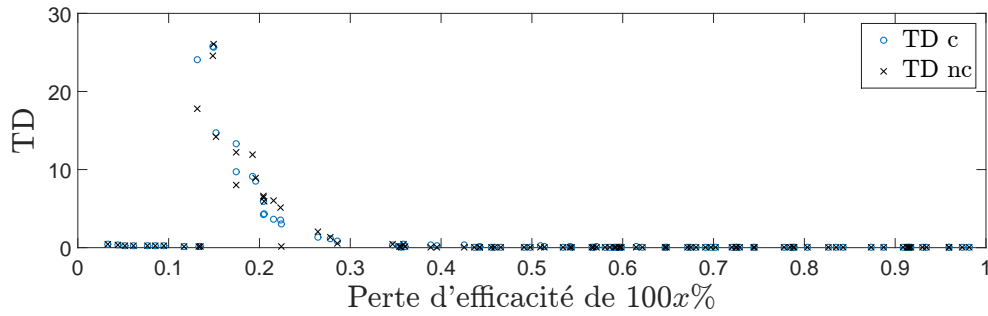


FIGURE 6.6 – Temps de détection en cas de perte d'efficacité élecon 1 sur 96 simulations

Le temps de détection est inférieur à 0.5s entre 0% et 13% de perte d'efficacité. Ces temps de détection correspondent au cas où la covariance n'est pas réouverte. Malgré la non réouverture, l'estimation des paramètres de fautes dévient légèrement de zéro, en gardant une covariance assez faible et par conséquent en étant plutôt confiant concernant la valeur. La détection d'une panne dépend de la valeur de l'estimation du paramètre de fautes mais aussi de la confiance que l'on associe à cette valeur. Ainsi, dans le cas où la covariance n'est pas réouverte à cause d'une faible perte d'efficacité, la panne est détectée, malgré une faible valeur de l'estimation du paramètre de fautes, mais grâce à une valeur assez faible de la covariance. Après 13% de perte d'efficacité, la covariance est réouverte. Ainsi entre 13% et 25% de perte d'efficacité, la valeur de l'estimation du paramètre de fautes est assez faible et la covariance est élevée à cause de la réouverture. Ainsi la détection est assez tardive, le temps que la covariance diminue suffisamment pour détecter la panne. Après 25% de perte d'efficacité, la valeur de l'estimation du paramètre de fautes est suffisamment grande pour pouvoir détecter rapidement ($TD \leq 2s$) la panne malgré une covariance assez grande. Après 30% de perte d'efficacité, le temps de détection est inférieur à 0.5s.

Les Figures 6.7 et 6.8 montrent les erreurs d'estimation des paramètres de fautes associés aux élecons en cas de blocage de l'élecon 1.

Les erreurs d'estimation sur b_0^2 et b_0^3 sont faibles. Excepté cinq simulations, les erreurs sont inférieures à 0.005 pour b_0^2 et inférieure à 0.004 pour b_0^3 . De plus, la différence entre le filtre contraint et non contraint est très faible. Les erreurs d'es-

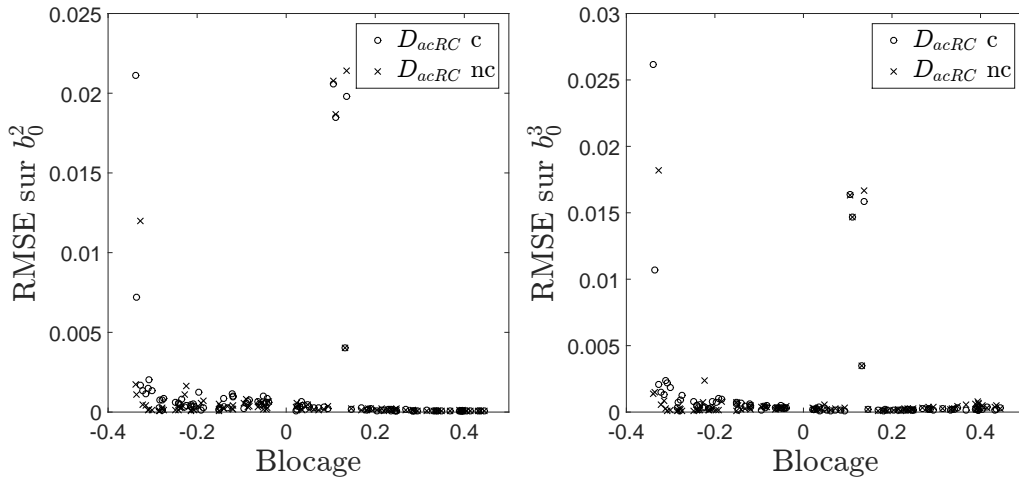


FIGURE 6.7 – Erreur d'estimation en cas de blocage éleçon 1 sur 100 simulations

timations sur les paramètres b_1^2 , b_1^3 , b_1^4 et b_1^5 sont légèrement plus élevées. Pour les paramètres b_1^4 et b_1^5 , les erreurs sont plus importantes avec le filtre ne prenant pas en compte les contraintes, entre 0 et 0.12 pour b_1^4 et entre 0 et 0.09 pour b_1^5 . Alors que pour le filtre contraint, les erreurs valent entre 0 et 0.012 pour b_1^4 et entre 0 et 0.017 pour b_1^5 . Pour le paramètre b_1^3 , l'erreur est aussi légèrement supérieure avec le filtre non contraint mais reste inférieure à 0.07 pour 99 simulations. Pour le filtre contraint, l'erreur est inférieure à 0.05 pour 98 simulations. Finalement, pour le paramètre b_1^2 associé à l'éleçon 1 en panne, l'erreur est plus importante pour des blocages proches de la valeur d'équilibre. Excepté les simulations pour lesquelles le blocage est proche de la valeur d'équilibre, l'erreur sur b_1^2 est faible (inférieures à 0.11 pour le filtre non contraint et inférieures à 0.06 pour le filtre contraint).

Pour les simulations ayant des erreurs plus importantes, quatre correspondent à des valeurs de blocage proche de la valeur d'équilibre (0.1326, 0.1361, 0.1058 et 0.1107 pour une valeur d'équilibre de 0.125). Les deux autres simulations, qui ont une valeur de blocage de -0.3387 et -0.3363 , ont une erreur plus importante à cause d'une période de transition plus longue, c'est-à-dire que le filtre met plus de temps à converger vers les bonnes valeurs.

La covariance est réouverte pour l'ensemble des simulations et il n'y a pas de MD que ce soit avec ou sans contrainte. Cependant, avec le filtre contraint, il y a 2 détections tardives ($TD = [33.52 \ 28.32]$), et avec le filtre non contraint, il y a une détection tardive ($TD = 28.42$). Ces détections tardives interviennent pour des valeurs de blocage proches de celle de l'équilibre. La Figure 6.9 montre le temps de détection des autres simulations en fonction du blocage.

Le temps de détection est très rapide pour toutes les simulations. Il reste néanmoins plus long lorsque l'on se rapproche d'un blocage autour de la valeur de trim.

Les Figures 6.10 et 6.11 montrent les erreurs d'estimation des paramètres de fautes associés aux éleçons en cas de biais sur l'éleçon 1.

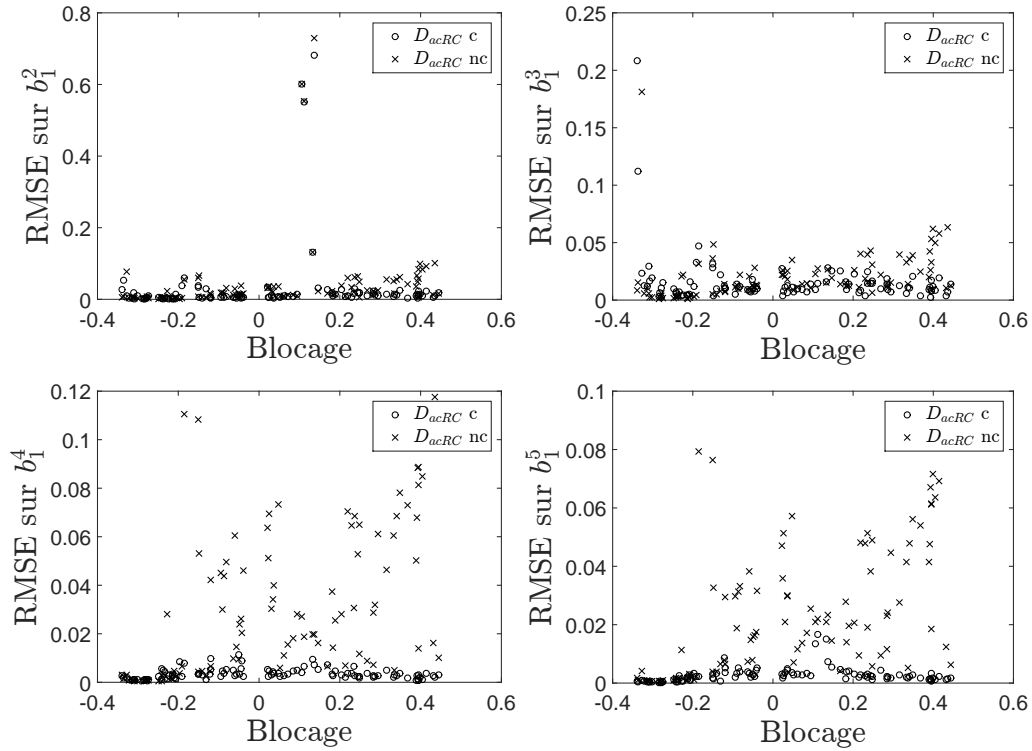


FIGURE 6.8 – Erreur d’estimation en cas de blocage élevé 1 sur 100 simulations

Les erreurs d’estimation sur b_0^2 et b_0^3 sont faibles, inférieures à 0.0015 pour b_0^2 et inférieur à 0.0011 pour b_0^3 . De plus, la différence entre le filtre contraint et non contraint est très faible. Les erreurs d’estimations sur les paramètres b_1^2 , b_1^3 , b_1^4 et b_1^5 sont légèrement plus élevées. Les erreurs sont plus importantes avec le filtre ne prenant pas en compte les contraintes, entre 0 et 0.1 pour b_1^2 , entre 0 et 0.06 pour b_1^3 , entre 0 et 0.15 pour b_1^4 et entre 0 et 0.11 pour b_1^5 . Alors que pour le filtre contraint, les erreurs valent entre 0 et 0.014 pour b_1^2 , entre 0 et 0.012 pour b_1^3 , entre 0 et 0.014 pour b_1^4 et entre 0 et 0.008 pour b_1^5 .

Concernant la détection, il n’y a pas de MD que ce soit avec le filtre contraint ou avec le filtre non contraint. Cependant, la covariance n’est pas réouverte pour 4 simulations qui correspondent à des biais très faibles et qui auront, par conséquent, peu d’effet sur le comportement du drone. La covariance est réouverte pour les 96 simulations restantes. La Figure 6.12 montre le temps de détection en fonction du biais.

Le temps de détection est très rapide pour l’ensemble des simulations, excepté pour une simulation avec le filtre contraint pour un biais très faible ($TD = 74.44s$) qui n’apparaît pas sur la Figure 6.12. Le temps de détection est plus long lorsque l’on se rapproche d’un biais proche de zéro. De plus, pour les biais entre ± 0.2 , le temps de détection avec le filtre non contraint est plus rapide qu’avec le filtre contraint. Pour des biais supérieurs à ± 0.2 , le temps de détection est inférieur à 0.06s.

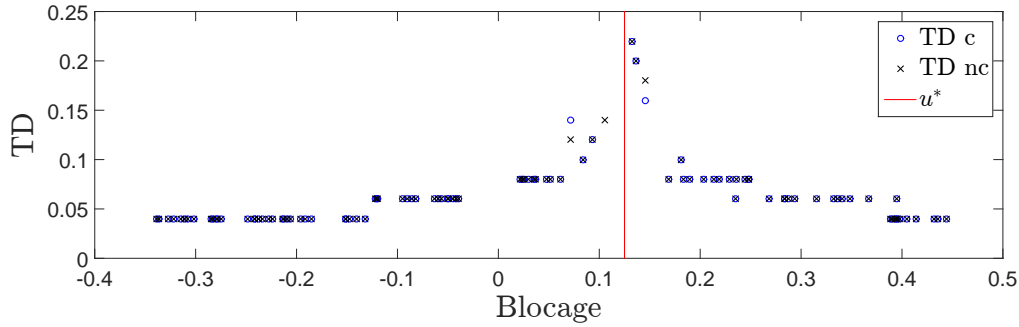


FIGURE 6.9 – Temps de détection en cas de blocage éleçon 1 sur 99 simulations avec le filtre non contraint et sur 98 simulations avec le filtre contraint

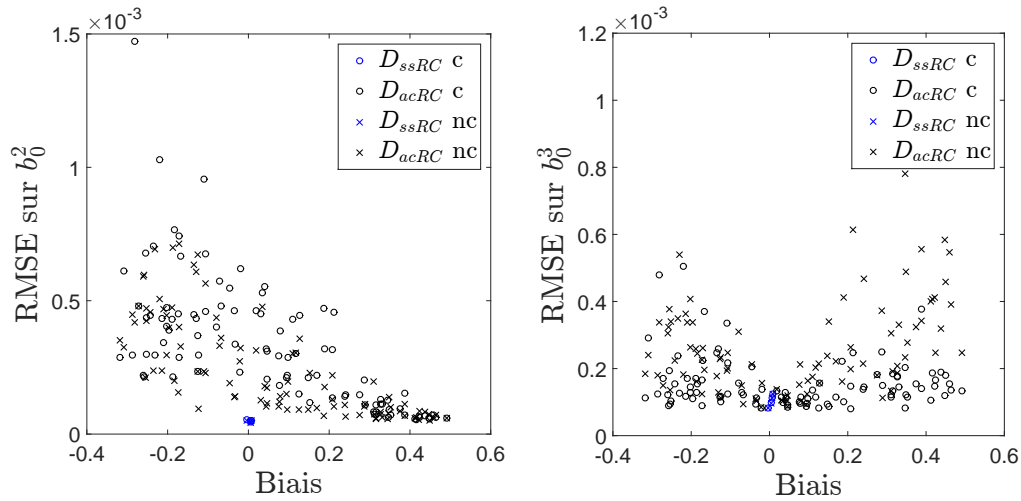


FIGURE 6.10 – Erreur d'estimation en cas de biais éleçon 1 sur 100 simulations

Contrairement aux pannes moteur (cf. Annexe H), des pannes éleçons peuvent entraîner une réouverture inappropriée de la covariance des paramètres de fautes associés au moteur. La Figure 6.13 montre les erreurs d'estimation des paramètres de fautes associés au moteur en cas de blocage de l'éleçon 1.

Avec le filtre contraint, il y a 29 réouvertures inappropriées. Les erreurs d'estimation des paramètres de fautes associés au moteur sont alors plus importantes mais elles restent assez faibles, inférieures à 0.004 pour b_0^1 et inférieures à 0.012 pour b_1^1 . Avec le filtre non contraint, il y a 41 réouvertures inappropriées et les erreurs d'estimation deviennent beaucoup plus importantes, inférieures à 0.05 pour b_0^1 et inférieures à 0.18 pour b_1^1 .

Les figures montrant les erreurs d'estimation des paramètres de fautes associés au moteur en cas de biais et en cas de perte d'efficacité sur l'éleçon 1 sont montrées en Annexe H. Quel que soit la panne éleçon, si il y a une réouverture inappropriée sur les paramètres de fautes du moteur, le filtre contraint permet d'avoir une meilleure précision.

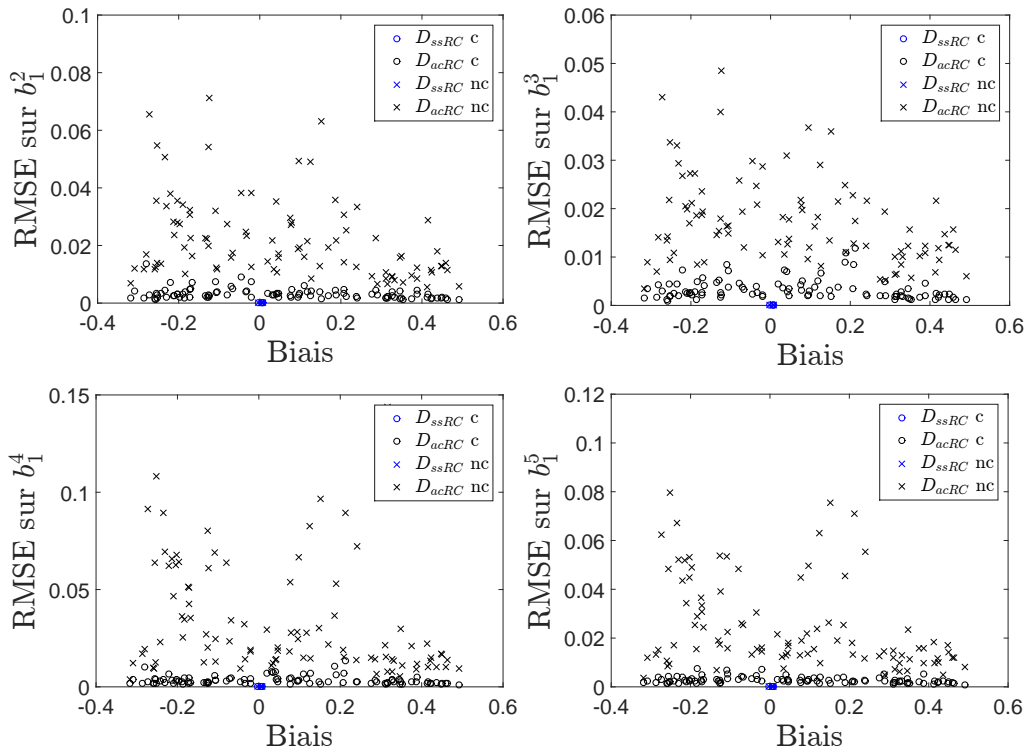


FIGURE 6.11 – Erreur d’estimation en cas de biais élevo 1 sur 100 simulations

Remarque 6.3

Les courbes pour les pannes des autres élevo sont semblables à celle montrées pour l’élevo 1.

Simulations de pannes moteur :

Les courbes et les explications de la comparaison entre le filtre TSKF non contraint et contraint en cas de panne moteur sont présentées en Annexe H. De façon générale, les deux filtres donnent des résultats à peu près similaires. Le filtre TSKF contraint donne néanmoins de meilleurs résultats pour des faibles pannes, c’est-à-dire pour des faibles pertes d’efficacité, pour des blocages proches de la valeur d’équilibre et pour des faibles biais. Cependant, le temps de détection est légèrement plus rapide avec le filtre non contraint. Les simulations fournissent donc des résultats similaires à ceux obtenus en cas de pannes de l’élevo 1.

Conclusion

Concernant la détection, il est important de noter que pour l’ensemble des simulations, il n’y a pas eu une seule *MI*. En effet, en cas de pannes moteur, une panne élevo n’a jamais été identifié et inversement. Ainsi, le taux de *MI* est nul, $MI_{\%} = 0$. Le nombre de *MD* est faible, il y a 20 *MD* pour le filtre contraint et 11 *MD* pour le filtre non contraint. De plus, les *MD* correspondent à des fautes très faibles et qui auront par conséquent un très faible impact sur le comportement du

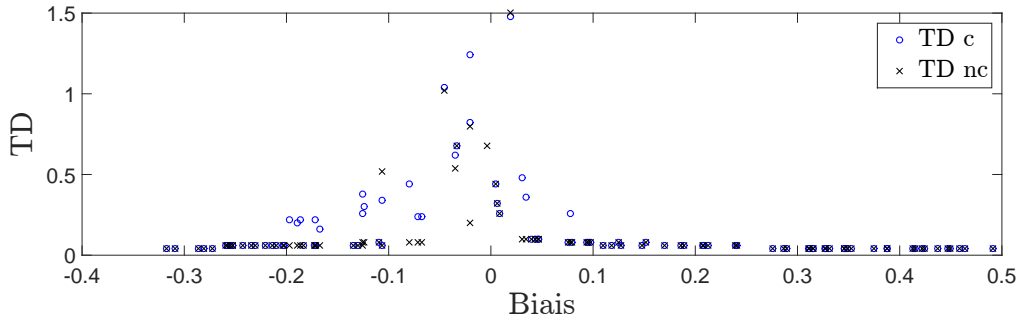


FIGURE 6.12 – Temps de détection en cas de biais élevé 1 sur 100 simulations

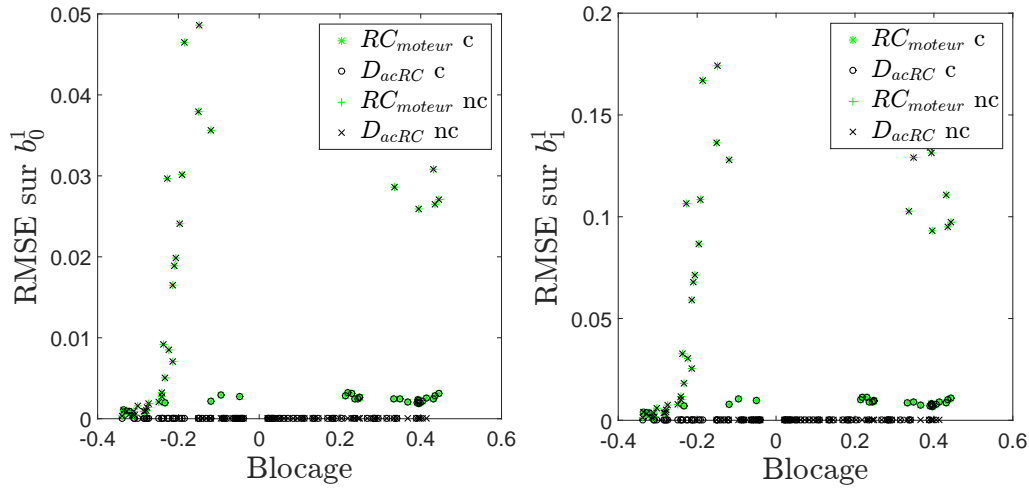


FIGURE 6.13 – Erreur d'estimation en cas de blocage élevé 1 sur 100 simulations

drone. Parmi les pannes assez faibles, il y a quelques détections tardives, c'est-à-dire supérieures à $5s$. Il y a 12 détections tardives pour le filtre contraint et 22 détections tardives pour le filtre non contraint. Ces détections tardives ne sont pas satisfaisantes. Ainsi, pour les résultats non satisfaisants, c'est-à-dire pour les MD et les détections tardives ($> 5s$), le taux est semblable pour les deux filtres. Néanmoins, ces résultats étant obtenus pour des faibles pannes, ils n'auront pas un effet néfaste pour le drone. Les autres simulations correspondent à des détections rapides. Le filtre non contraint est légèrement plus rapide au niveau de la détection mais les résultats avec le filtre contraint sont proches et tout à fait satisfaisants.

Concernant l'estimation, le filtre est plus précis avec les contraintes sur \mathbf{b}_1 , que ce soit pour les pannes moteur ou les pannes élevons. De plus, il est important d'avoir une bonne précision au niveau des paramètres de fautes actionneurs dans le but de pouvoir les utiliser pour compenser les fautes au niveau du système de commande. Ainsi, par la suite, le filtre utilisé prendra en compte les contraintes sur \mathbf{b}_1 .

“ce qu’il faut retenir”

L’application de contraintes sur le paramètre de faute \mathbf{b}_1 permet d’avoir une meilleure précision de l’estimation au niveau des paramètres de fautes et notamment en cas de pannes sur un actionneur redondant. En effet, la contrainte sur \mathbf{b}_1 permet de limiter le nombre de combinaisons linéaires possibles.

6.3.1.3 Comparaison entre les différents filtres contraints

Avec la méthode développée, il est possible d’utiliser les filtres AKF et TSKF classiques, mais aussi le filtre AKF séquentiel. Les simulations réalisées à la Section précédente 6.3.1.2 avec le filtre TSKF contraint ont aussi été réalisées pour le filtre AKF contraint et pour le filtre AKF contraint séquentiel. Les résultats obtenus au niveau de la précision du filtre sont semblables pour les trois filtres.

Concernant les taux de détection, 7500 simulations ont été lancées pour les calculer. Ces taux ont été calculés en réalisant pour chaque actionneur :

- 500 simulations en cas de perte d’efficacité comprise entre 0% et 100% ;
- 500 simulation en cas de blocage compris entre \underline{u} et \bar{u} ;
- 500 simulation en cas de biais compris entre $u^* - \underline{u}$ et $u^* - \bar{u}$.

Le Tableau 6.2 montre les taux de détections pour les trois filtres.

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0.5, 1]s$ (%)	$TD \leq 0.5s$ (%)
Filtre TSKF	1.67%	2.55%	3.13%	3.89%	88.76%
Filtre AKF	1.73%	2.81%	4.4%	5%	86.06%
Filtre AKF séquentiel	1.53%	1.23%	4.41%	5.29%	87.54%

Tableau 6.2 – Récapitulatif des détections

Les taux de détection sont semblables. Par ailleurs, pour chaque filtre, le taux de mauvaise identification est nul.

Cependant, l’intégrité du filtre n’est pas respectée pour certains cas. En cas de pannes moteur, l’intégrité est toujours respectée. Pour les filtres AKF et TSKF, les cas suivants ne respectent pas l’intégrité du filtre : les blocages proches de la valeur d’équilibre pour les éleveurs 1 et 3. L’intégrité n’est pas respecté sur les états w , p , q et δ_e . La Figure 6.14 montre les erreurs RMSE sur ces états en fonction de la perte d’efficacité sur l’éleveur 3.

Les simulations pour lesquelles le seuil 3σ est dépassé ne sont pas intègres. Ces simulations correspondent à des cas de blocage très proche de la valeur d’équilibre.

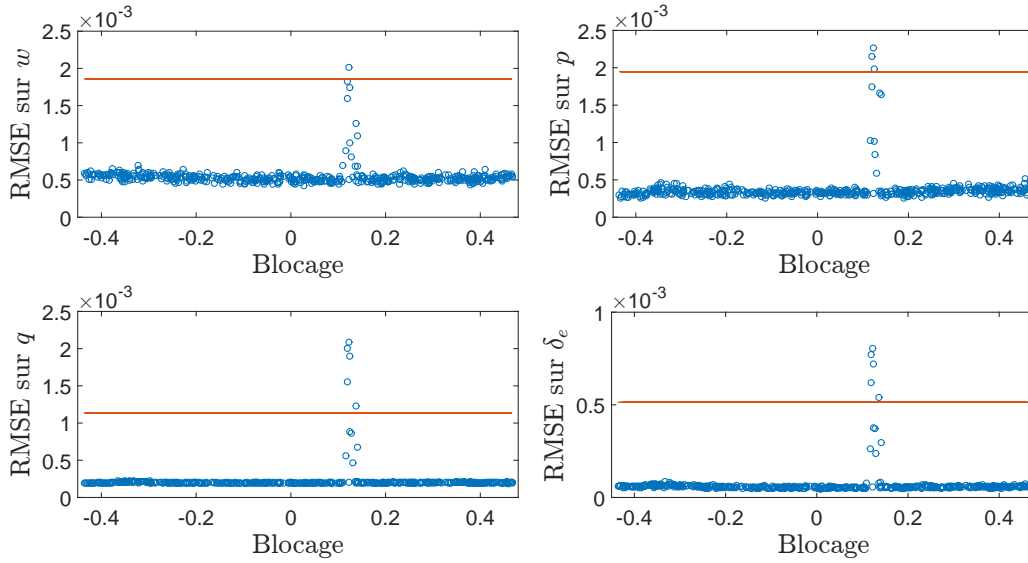


FIGURE 6.14 – Erreurs RMSE sur l'état en cas de blocage sur l'élevon3

Pour le filtre AKF séquentiel, les cas suivants ne respectent pas l'intégrité du filtre : les faibles pertes d'efficacité, entre 10% et 20%, pour l'élevon 2 et 4. L'intégrité n'est pas respecté sur les états u , w , p , q , δ_e et δ_a . La Figure 6.15 montre les erreurs RMSE sur ces états en fonction de la perte d'efficacité sur l'élevon 4.

Les simulations pour lesquelles le seuil 3σ est dépassé ne sont pas intègres. Ces simulations correspondent à des cas où la covariance n'est pas réouverte, pour des pertes d'efficacité comprises entre 10% et 20%. Comme la covariance n'est pas réouverte, les paramètres de faute ainsi que l'état vont chercher à compenser l'erreur dû à la panne, d'où l'apparition d'un faible biais au niveau de l'état.

“ce qu'il faut retenir”

Quel que soit le filtre utilisé, le taux de FA et le taux de MI sont nuls. De plus, le taux de MD est faible, entre 1% et 2% et correspond à des pannes de faibles amplitudes et la plupart des détections sont très rapides.

6.3.1.4 Influence du choix de Q

Dans cette section, les simulations ont été réalisées avec le filtre AKF contraint. Le but est de voir l'influence des matrices Q^x et Q^b sur la précision du filtre ainsi que sur le taux de détection.

Influence de la matrice Q^x :

Deux types de simulations ont été lancés :

- 100 simulations avec $Q^x = 2 \times 10^{-8} \mathbb{I}_n$ et $Q^b = 2 \times 10^{-10} \mathbb{I}_{n_b}$ et
- 100 simulations avec $Q^x = 2 \times 10^{-6} \mathbb{I}_n$ et $Q^b = 2 \times 10^{-10} \mathbb{I}_{n_b}$.

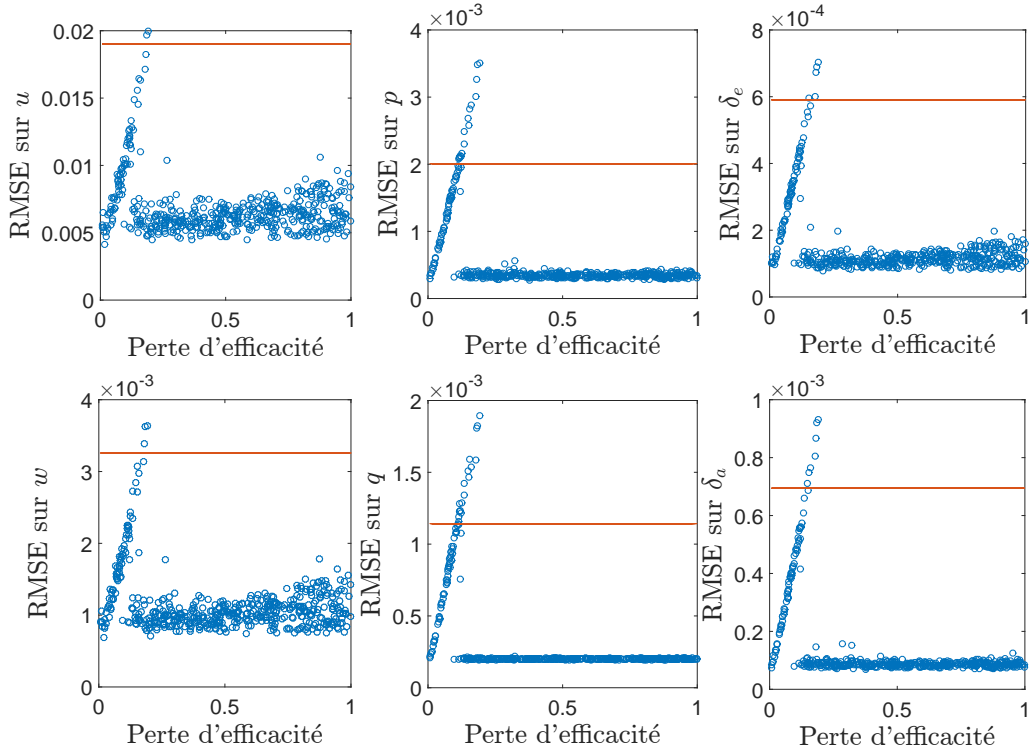


FIGURE 6.15 – Erreurs RMSE sur l'état en cas de perte d'efficacité sur l'élevon4

Le Tableau 6.3 montre les taux de détection en cas de pannes du moteur et en cas de pannes de l'élevon 1, en fonction de la matrice de bruit Q^x .

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0.5, 1]s$ (%)	$TD \leq 0.5s$ (%)
Moteur $Q^x = 2 \times 10^{-8}$	5.67%	0%	6.33%	8%	80%
Moteur $Q^x = 2 \times 10^{-6}$	9.33%	7%	10.33%	12.33%	61%
Élevon1 $Q^x = 2 \times 10^{-8}$	1%	3.33%	3%	2%	90.67%
Élevon1 $Q^x = 2 \times 10^{-6}$	2.33%	1.33%	5%	5%	86.33%

Tableau 6.3 – Taux de détections pour différentes matrices Q^x

Lorsque l'on a moins confiance dans le modèle, le temps de détection est plus lent et il y a plus de MD notamment en cas de pannes moteur. Malgré tout, les MD et les détections tardives correspondent à des pannes faibles, tels que de petites pertes d'efficacité ou de petits biais. De plus, le taux de mauvaise identification reste

nul lorsque l'on a moins confiance dans le modèle.

Concernant la précision de l'estimation, les erreurs RMSE sur les paramètres de fautes associés au moteur b_0^1 et b_1^1 sont plus élevées en cas de perte d'efficacité moteur et en cas de blocage moteur comme montrées sur la Figure 6.16. Pour les autres types de pannes, les erreurs sont semblables.

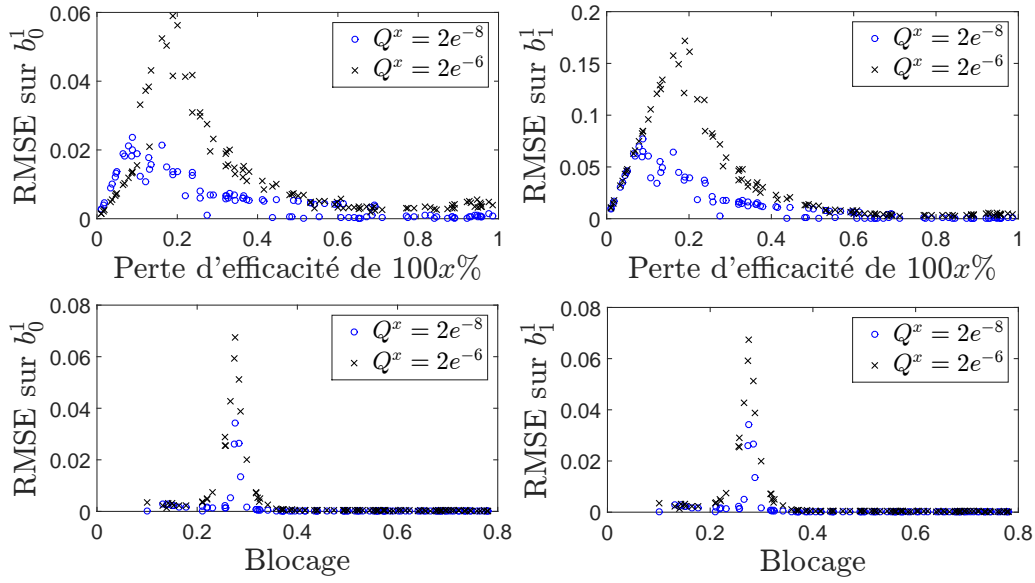


FIGURE 6.16 – Comparaison de l'erreur avec $Q^x = 2e^{-8}$ et $Q^x = 2e^{-6}$

Il est aussi important de noter qu'il n'y a pas de réouvertures inappropriées de la covariance lorsque $Q^x = 2 \times 10^{-6}$ alors qu'il y en a lorsque $Q^x = 2 \times 10^{-8}$ (cf. Section 6.3.1.2).

Concernant l'intégrité, le filtre avec $Q^x = 2 \times 10^{-6}$ est intègre pour l'ensemble des simulations.

Influence de la matrice Q^b :

Deux types de simulations ont été lancés :

- 100 simulations avec $Q^x = 2 \times 10^{-8}\mathbb{I}_n$ et $Q^b = 2 \times 10^{-10}\mathbb{I}_{n_b}$ et
- 100 simulations avec $Q^x = 2 \times 10^{-8}\mathbb{I}_n$ et $Q^b = 2 \times 10^{-8}\mathbb{I}_{n_b}$.

Le Tableau 6.4 montre les taux de détection en cas de pannes du moteur et en cas de pannes de l'élevon 1, en fonction de la matrice de bruit Q^b .

Lorsque l'on a moins confiance dans les paramètres de faute, le temps de détection est légèrement plus lent et il y a plus de MD . Malgré tout, les MD et les détections tardives correspondent à des pannes faibles, tels que de petites pertes d'efficacité ou de petits biais. De plus, le taux de mauvaise identification reste nul lorsque l'on a moins confiance dans le modèle.

Concernant la précision de l'estimation, les erreurs RMSE sur l'ensemble des paramètres de fautes sont légèrement plus élevées. En effet, celle-ci sont plus bruitées.

La valeur de Q^b ne change pas l'intégrité. L'intégrité n'est pas respectée pour des blocages proches de la valeur d'équilibre.

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0.5, 1]s$ (%)	$TD \leq 0.5s$ (%)
Moteur $Q^b = 2 \times 10^{-10}$	5.67%	0%	6.33%	8%	80%
Moteur $Q^b = 2 \times 10^{-8}$	8%	0.33%	3.33%	7%	81.33%
Élevon1 $Q^b = 2 \times 10^{-10}$	1%	3.33%	3%	2%	90.67%
Élevon1 $Q^b = 2 \times 10^{-8}$	5.33%	2.67%	5%	3%	84%

Tableau 6.4 – Taux de détections pour différentes matrices Q^b

“ce qu’il faut retenir”

Une matrice de bruit Q^x plus élevée, c’est-à-dire avec plus d’incertitudes, entraîne une détérioration de la précision de l’estimation et une augmentation du temps de détection.

6.3.1.5 Cas des pannes intermittentes

Cas d’une panne intermittente de 10s :

Sur 100 simulations de pannes de types pertes d’efficacité du moteur, il y a une *FA*, c’est à dire que la panne est détectée de façon permanente. En effet, pour cette simulation, les estimations des paramètres de fautes b_0^1 et b_1^1 sont mauvaises au cours de la simulation, b_0^1 converge vers 0.2162 et b_1^1 converge vers 0.776. Or, ces fausses estimations permettent d’obtenir une commande équivalente à la commande d’équilibre, par conséquent le drone n’est pas impacté par cette mauvaise estimation. Parmi les 99 autres simulations, il y a 12 *MD* et 87 détections temporaires. Le temps de détection du retour au mode nominal est plus long pour des pertes d’efficacité supérieures à 60%, comme montrée sur la Figure 6.17. En effet, lorsque le moteur est en perte d’efficacité d’une valeur supérieure à 60%, la commande du moteur est en saturation pour pouvoir compenser la panne. Lorsque l’actionneur redevient fonctionnel, la commande reste en saturation le temps que le drone rattrape la trajectoire, ainsi il est difficile de détecter un retour au mode nominal à ce moment là. Par conséquent, plus la perte d’efficacité est importante, plus le drone s’éloigne de la trajectoire, plus la commande reste en saturation le temps que le drone rattrape la trajectoire, plus le temps de détection du retour au mode nominal est long.

Sur les 100 simulations de pannes de types blocage du moteur, il y a 100 détections temporaires. De façon similaire aux pertes d’efficacité moteur, la saturation de la commande rend le temps de détection du retour au mode nominal plus long. Plus la commande se bloque à une valeur loin de la valeur d’équilibre, plus le temps de détection du retour au mode nominal est long, comme montrée sur la Figure 6.17.

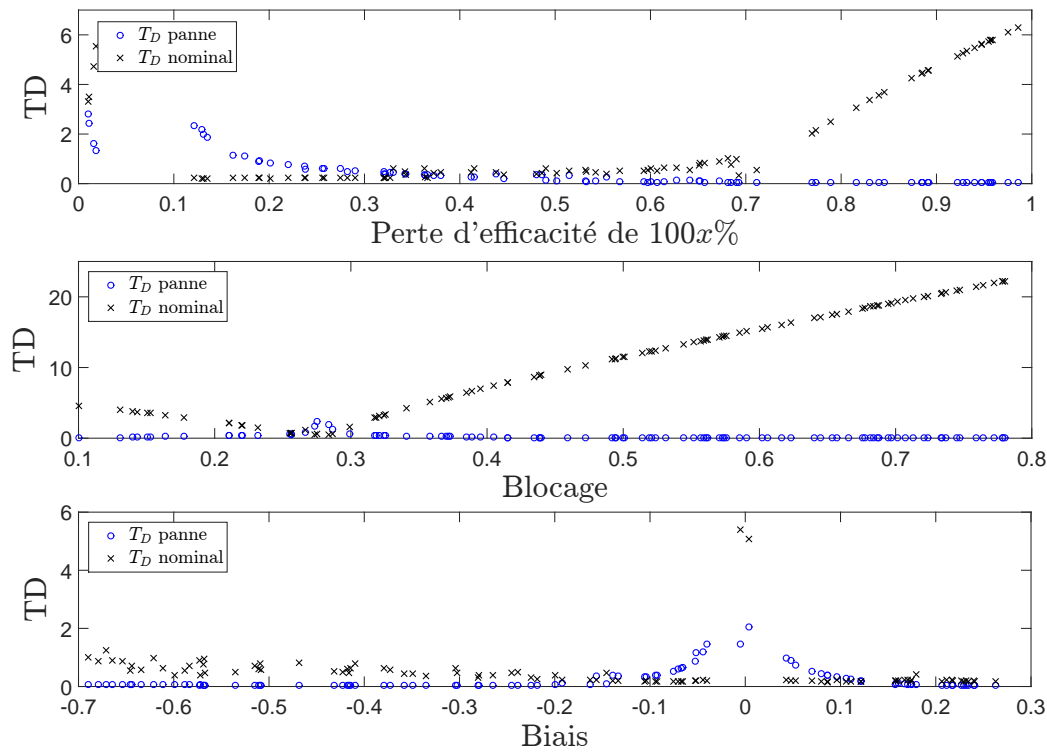


FIGURE 6.17 – Temps de détection en cas de panne intermittente de 10s

Sur les 100 simulations de pannes de types biais sur moteur, il y a 4 *MD* et 96 détections temporaires. Le temps de détection du retour au mode nominal est légèrement plus long que le temps de détection de la panne mais reste inférieur à 1s pour 95% des cas.

Pour les pannes élévons, le temps de détection du retour au mode nominal est semblable au temps de détection de la panne. La Figure 6.18 montre le temps de détection en cas de panne intermittente sur l'élevon 1.

Cas général d'une panne intermittente de durée variable :

Les résultats sont similaires pour des pannes intermittentes de temps différents. Néanmoins, plus la panne intermittente est courte, moins il y aura de détection. En effet, les pannes intermittentes de X s sont détectées uniquement si la panne est détectée en moins de X s.

“ce qu'il faut retenir”

Un retour à la normal après une panne est géré de la même façon qu'une panne. Ainsi, une panne intermittente peut être traitée par la méthode FDD.

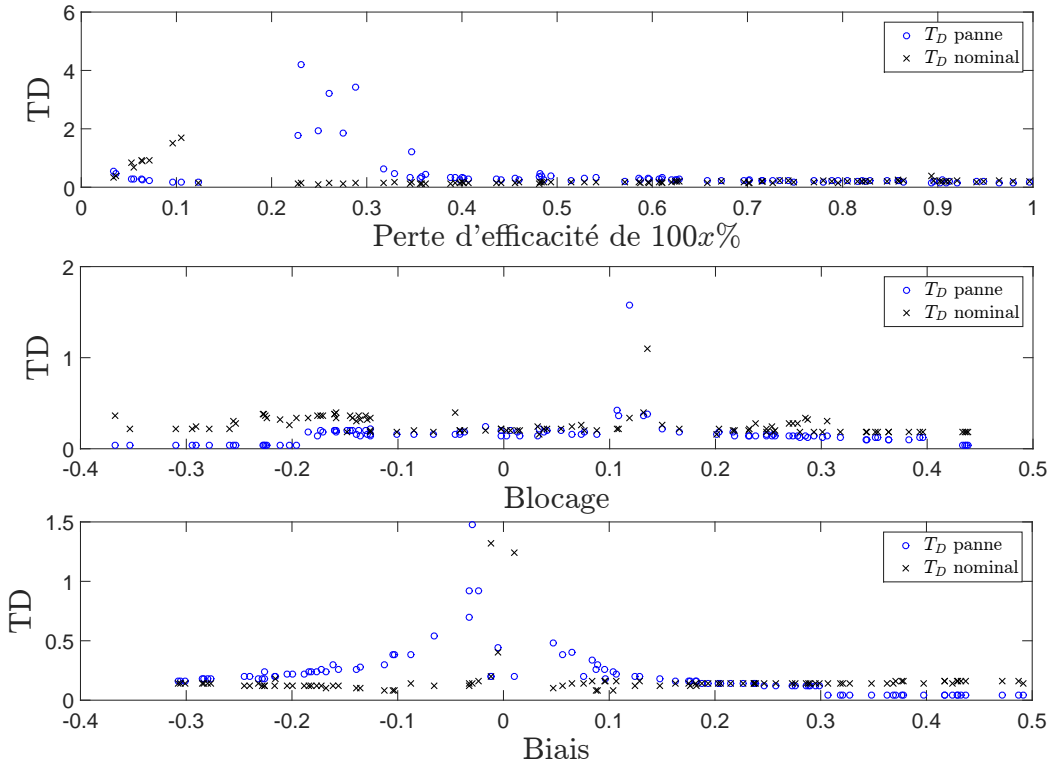


FIGURE 6.18 – Temps de détection en cas de panne intermittente de 10s

6.3.2 Cas où l'on considère uniquement les pannes capteurs

Dans le cas où l'on considère uniquement les pannes capteurs, le tableau des signatures de fautes est simplifié comme montré sur le Tableau 6.5.

Le capteur a_{sx} a donc une signature unique, la panne moteur n'étant plus considérée. De plus, contrairement aux actionneurs, les dérives sont des pannes courantes sur les capteurs. Ainsi, l'algorithme CUSUM est pris en compte lors des simulations suivantes.

6.3.2.1 Comportement en mode nominal

500 simulations sont réalisées en mode nominal avec 500 séquences de bruits sur les capteurs et cela pour chaque filtre : l'AKF et l'AKF séquentiel.

Ces simulations permettent d'évaluer le taux de FA . Quel que soit le filtre utilisé, sur les 500 simulations, il n'y a pas de fausse alarme, ainsi le taux est nul, $FA\% = 0$. Néanmoins il y a 36 réouvertures inappropriées de la covariance des paramètres de fautes associés aux capteurs pour le filtre AKF et 32 réouvertures pour le filtre séquentiel. Le Tableau 6.6 montre le nombre de réouverture inappropriée associée à chaque paramètre de faute c_0 . Il faut noter qu'il n'y a pas de réouverture de la covariance pour les paramètres de faute associés aux capteurs r et z_B .

Composant	Signature											
	r_p	r_q	r_r	$r_{a_{sx}}$	$r_{a_{sy}}$	$r_{a_{sz}}$	r_{z_B}	r_ϕ	r_θ	r_ψ	r_α	r_β
IMU p	1	0	0	0	0	0	0	0	0	0	0	0
IMU q	0	1	0	0	0	0	0	0	0	0	0	0
IMU r	0	0	1	0	0	0	0	0	0	0	0	0
IMU a_{sx}	0	0	0	1	0	0	0	0	0	0	0	0
IMU a_{sy}	0	0	0	0	1	0	0	0	0	0	0	0
IMU a_{sz}	0	0	0	0	0	1	0	0	0	0	0	0
Baromètre z_B	0	0	0	0	0	0	1	0	0	0	0	0
Inclinomètre ϕ	0	0	0	0	0	0	0	1	0	0	0	0
Inclinomètre θ	0	0	0	0	0	0	0	0	1	0	0	0
Inclinomètre ψ	0	0	0	0	0	0	0	0	0	1	0	0
Capteur α	0	0	0	0	0	0	0	0	0	0	1	0
Capteur β	0	0	0	0	0	0	0	0	0	0	0	1

Tableau 6.5 – Signatures de fautes pour les capteurs

Composant	p	q	r	a_{sx}	a_{sy}	a_{sz}	ψ	ϕ	θ	α	β	z_B
AKF	3	7	0	1	3	4	4	2	2	6	4	0
AKF_{sq}	3	6	0	1	3	4	3	1	3	5	3	0

Tableau 6.6 – Nombres de réouverture inappropriées de la covariance des paramètres de fautes c_0

De plus, le filtre est intègre pour l'ensemble des simulations. Le filtre donne donc de bons résultats dans le mode nominal.

6.3.2.2 Analyse des résultats en utilisant le filtre AKF

Le scénario suivant est utilisé : lors d'un atterrissage, une panne permanente survient sur un capteur à 10s ; la simulation se termine au bout de 100s. La panne peut être un biais, une perte d'efficacité, un blocage ou une dérive. Le filtre utilisé lors de cette section est le filtre AKF avec $Q^x = 2 \times 10^{-8} \mathbb{I}_n$ et $Q^b = 2 \times 10^{-10} \mathbb{I}_{n_b}$. Les figures suivantes montrent l'erreur RMSE calculé après l'apparition de la faute, en régime établi, c'est-à-dire une seconde après l'apparition de la faute, de 11s à 100s.

Simulations de perte d'efficacité :

Bien que l'on souhaite traiter les cas de pertes d'efficacité pour l'ensemble des capteurs, il est difficile pour certains capteurs de détecter la panne. En effet, le modèle du drone a été linéarisé pour une certaine valeur d'équilibre, pour laquelle certaines valeurs des mesures à l'équilibre sont nulles. Dans le cas où la mesure à l'équilibre est nulle, l'équation mathématique permettant de simuler les pertes

d'efficacité capteurs (cf. Section 6.2.1.2) devient :

$$z^f = Cx - c_1 Cx + v \quad (6.10)$$

L'estimation \hat{c}_0 est donc équivalente à un bruit centré en zéro ($-c_1 Cx + v$). Avec la méthode de détection développée, il est difficile de détecter une panne sur un bruit centré en zéro. En effet, une panne est détectée lorsque la valeur du paramètre de faute s'éloigne de zéro. Avec un bruit centré en zéro, la valeur de \hat{c}_0 restera proche de zéro, en oscillant autour de zéro. Ainsi, en fonction de la valeur de \hat{c}_0 , la panne va être détectée de façon intermittente, comme montré sur la Figure 6.19.

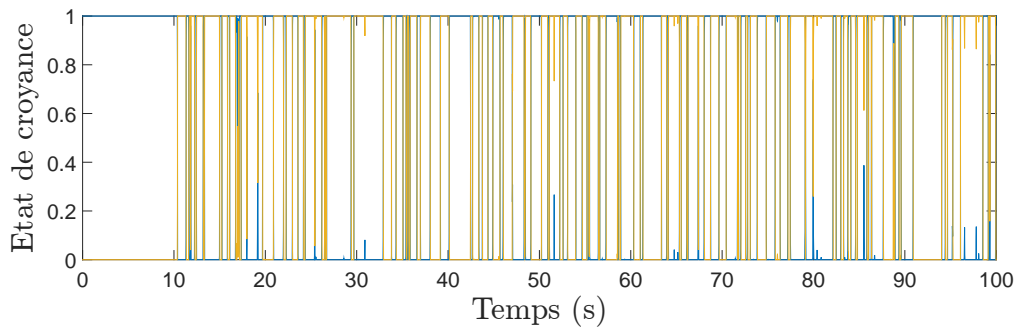


FIGURE 6.19 – État de croyance en cas de perte d'efficacité de 79% au niveau du capteur q

Ainsi, il est difficile de détecter une perte d'efficacité sur une mesure qui est très proche de zéro. C'est le cas pour les mesures suivantes : les mesures des vitesses angulaires p , q et r , la mesure de l'accélération spécifique latérale a_{sy} , et les mesures des angles ψ , ϕ et β . Malgré la difficulté de détection, l'intégrité du filtre est respectée sur les mesures. L'intégrité est aussi respectée sur l'état excepté en cas de panne sur ϕ et sur r . En cas de perte d'efficacité sur ϕ , l'intégrité de l'état ϕ n'est pas toujours respectée (pour 4 simulations). En cas de perte d'efficacité sur r , l'intégrité de l'état v et r n'est pas toujours respectée pour des pertes d'efficacité supérieures à 60%. De plus, sur 100 simulations de perte d'efficacité de ϕ , il y a une MI temporaire de a_{sy} . Malgré tout, l'intégrité sur les mesures étant respectée, l'estimation est suffisamment précise pour avoir un bon comportement au niveau de la trajectoire du drone.

Pour les capteurs ayant une valeur d'équilibre différente de zéro, il est plus facile de détecter les pertes d'efficacité. Les mesures concernés sont : les mesures de l'accélération spécifique horizontale et verticale a_{sx} et a_{sz} , les mesures des angles θ et α et la mesure de la position verticale z_B . Pour ces simulations, l'intégrité du filtre est toujours respectée sur les mesures. L'intégrité de l'état n'est pas toujours respectée. En cas de perte d'efficacité sur a_{sx} , l'intégrité des états u , z et E n'est pas toujours respectée. En cas de perte d'efficacité sur z_B , l'intégrité de l'état z n'est pas respectée pour des faibles pertes d'efficacité. La Figure 6.20 montre les erreurs d'estimation sur les états u , z et E en cas de perte d'efficacité du capteur a_{sx} . Bien que l'intégrité ne soit pas respectée, les erreurs au niveau de ces états sont faibles.

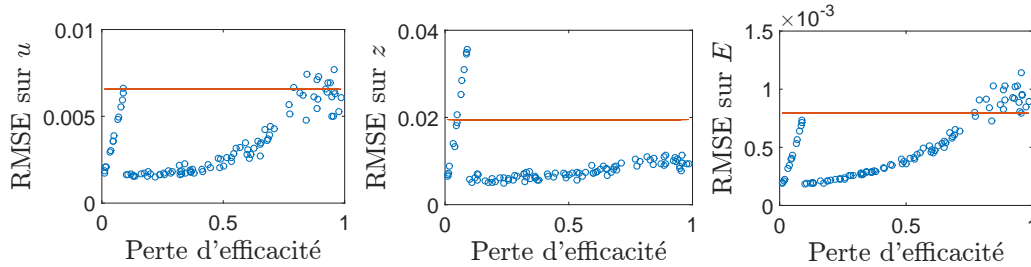


FIGURE 6.20 – Erreur d'estimation en cas de perte d'efficacité sur a_{sx} sur 100 simulations

Concernant la détection, le Tableau 6.7 récapitule les taux de détections pour les différents capteurs.

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0, 1]s$ (%)	$TD = 0s$ (%)
$p, q, r, a_{sy},$ ψ, ϕ, β	/	/	/	/	/
a_{sx}	0%	8%	7%	17%	71%
a_{sz}	0%	0%	0%	0%	100%
θ	4%	0%	16%	29%	51%
α	0%	3%	8%	23%	66%
z_B	0%	0%	0%	0%	100%

Tableau 6.7 – Taux de détection en cas de perte d'efficacité

Parmi les 100 simulations sur a_{sx} , 15 détections se font sans réouverture de la covariance. Celles-ci correspondent aux temps de détection supérieurs à 1s. De plus, parmi ces 15 simulations, 3 prennent en compte une MI temporaire. Dans un premier temps, la panne détectée provient du capteur θ , puis la panne détectée devient le capteur a_{sx} . Pour les simulations modélisant des pertes d'efficacité sur a_{sz} et celles modélisant des pertes d'efficacité sur z_B , le temps de détection est toujours immédiat $T_D = 0$. Pour les simulations modélisant des pertes d'efficacité sur θ , il y a uniquement 4 MD qui correspondent à des pertes d'efficacité très faibles (inférieures à 2%). Les 16 simulations ayant un temps de détection supérieur à 1s correspondent à des détections sans réouverture de la covariance. De façon similaire, les 11 simulations modélisant des pertes d'efficacité sur α ayant un temps de détection supérieur à 1s correspondent à des détections sans réouverture de la covariance. De façon générale, le temps de détection diminue lorsque la perte d'efficacité augmente.

Simulations de biais :

En cas de biais sur n'importe quel capteur, l'intégrité sur les mesures est toujours respectée. De plus, l'intégrité sur les états est elle aussi toujours respectée, exceptée pour deux cas de biais sur ϕ . En cas de faibles biais sur ϕ (0.33° et -0.35°), l'intégrité

n'est pas respectée sur l'état ϕ , l'erreur est légèrement supérieure au 3σ . De plus, pour ces deux simulations, il y a des *MI* temporaires. Dans un premier temps, la panne détectée provient du capteur r , puis la panne détectée devient le capteur ϕ .

Concernant la détection, le Tableau 6.8 récapitule les taux de détections pour différents capteurs en cas de biais.

	$MD_{\%}$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0, 1]s$ (%)	$TD = 0s$ (%)
p	0%	0%	2%	4%	94%
q	0%	0%	1%	3%	96%
r	0%	0%	2%	2%	96%
a_{sx}, a_{sy}, a_{sz}	0%	0%	0%	0%	100%
ψ	0%	0%	3%	9%	88%
ϕ	0%	2%	0%	1%	97%
θ	0%	0%	0%	2%	98%
α, β	0%	0%	0%	3%	97%
z_B	0%	1%	0%	6%	93%

Tableau 6.8 – Récapitulatif des détections en cas de biais

Quel que soit le capteur, il n'y a pas de *MD*. Pour le capteur z_B , la simulation ayant un temps de détection supérieur à $5s$ correspond à un biais très faible, de $-0.0476m$ pour laquelle il n'y a pas eu de réouverture de la covariance. Il y a aussi deux détections tardives en cas de faibles biais sur ϕ , de 0.33° et de -0.35° , qui correspondent là aussi à des simulations pour lesquelles il n'y a pas eu de réouverture de la covariance. Les autres simulations donnent des temps de détections rapides. De façon générale, le temps de détection est légèrement plus lent pour des biais proche de zéro. Cependant, la plupart des détections sont immédiates.

Simulations en cas de blocage :

En cas de blocage sur n'importe quel capteur, l'intégrité sur les mesures est toujours respectée. L'intégrité sur l'état n'est pas respectée pour certaines simulations en cas de blocage sur r , sur a_{sx} , sur ϕ et sur θ . Pour les autres simulations, l'intégrité de l'état est respectée. En cas de blocage sur r , les états v et r ne sont jamais intègre. En cas de blocage sur a_{sx} , les états u et E ne sont jamais intègre. En cas de blocage sur θ , l'état θ n'est jamais intègre. Finalement, en cas de blocage sur ϕ , l'état ϕ n'est jamais intègre. De plus, en cas de blocage à 1.15° du capteur ϕ , il y a une *MI* temporaire de r . La Figure 6.21 montre la valeur des erreurs dans le cas où les états ne sont pas intègres.

Concernant la détection, le Tableau 6.9 récapitule les taux de détections pour les différents capteurs en cas de blocage.

Il n'y a de *MD* quel que soit le blocage. Mais il y a quelques détections tardives. Comme remarqué sur le Tableau 6.9, en cas de blocage sur le capteur p , il y a 13 détections tardives. Ces détections tardives correspondent à des blocages compris entre $-0.1rad/s$ et $0.1rad/s$. En cas de blocage, le paramètre de faute \hat{c}_0 est équi-

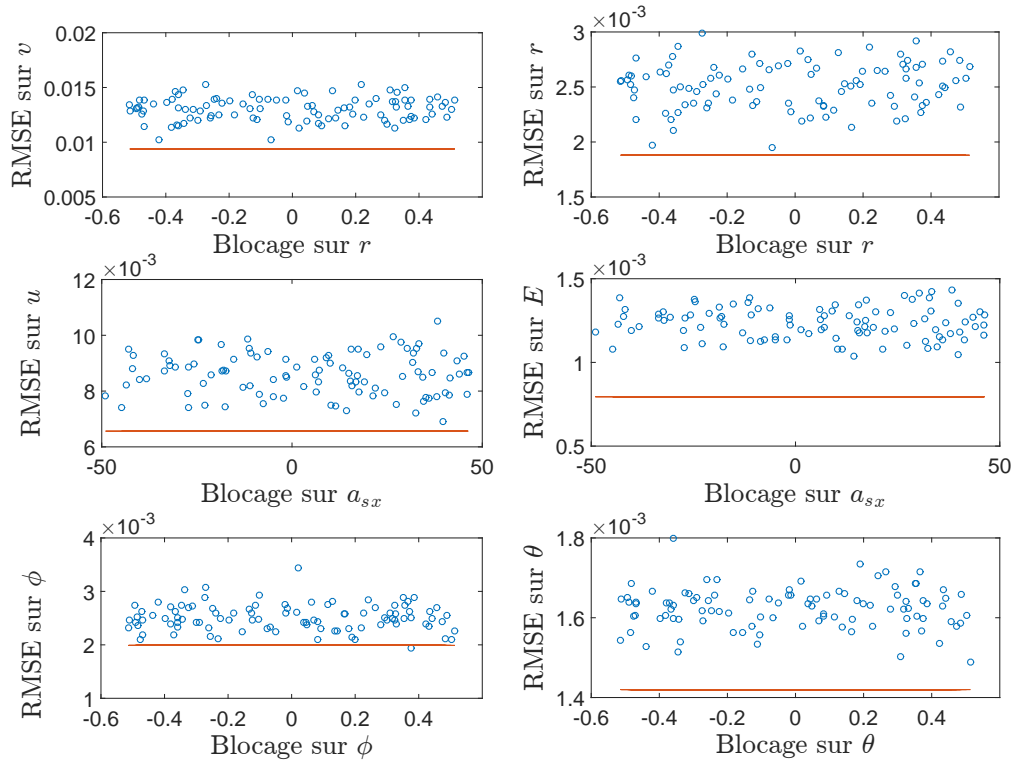


FIGURE 6.21 – Erreur d'estimation en cas de blocage sur différents capteurs sur 100 simulations

valent à un bruit blanc autour de la valeur de blocage. Ainsi, si le blocage est proche de zéro, certaines valeurs de \hat{c}_0 vont elle aussi être proche de zéro, dans quel cas, le mode nominal sera détecté par intermittence. Les détections tardives du capteur p correspondent en réalité à des détections de la panne par intermittence comme montré sur la Figure 6.22.

Il serait logique d'avoir autant de détections tardives sur les capteurs q et r . Cependant les bruits de ces capteurs sont plus faibles que celui du capteur p . Ainsi, le même phénomène est observé mais pour des valeurs de blocage plus faible : entre $-0.01rad/s$ et $0.01rad/s$. Il y a alors uniquement 6 détections tardives pour q et 2 détections tardives pour r . De façon similaire, les détections tardives sur les autres capteurs, tels que ϕ , a_{sz} et θ , correspondent en réalité à des détections intermittentes, dû au même phénomène que sur p .

De façon générale, le temps de détection est légèrement plus lent pour des blocages proches de l'équilibre. Néanmoins, la plupart des détections sont immédiates.

Remarque 6.4

Le blocage du baromètre a une particularité. Par exemple, lors de l'atterrissage, à 95m d'altitude le baromètre se bloque à une valeur de 70m. La panne est détectée très rapidement et le drone continue son atterrissage. Lorsque le drone

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0, 1]s$ (%)	$TD = 0s$ (%)
p	0%	13%	0%	0%	87%
q	0%	6%	0%	0%	94%
r	0%	2%	0%	3%	95%
a_{sx}, a_{sy}	0%	0%	0%	0%	100%
a_{sz}	0%	2%	0%	0%	98%
ψ	0%	0%	5%	4%	91%
ϕ	0%	5%	0%	0%	95%
θ	0%	2%	0%	0%	98%
α	0%	0%	2%	3%	95%
β	0%	0%	0%	4%	96%
z_B	0%	0%	0%	4%	96%

Tableau 6.9 – Récapitulatif des détections en cas de blocage

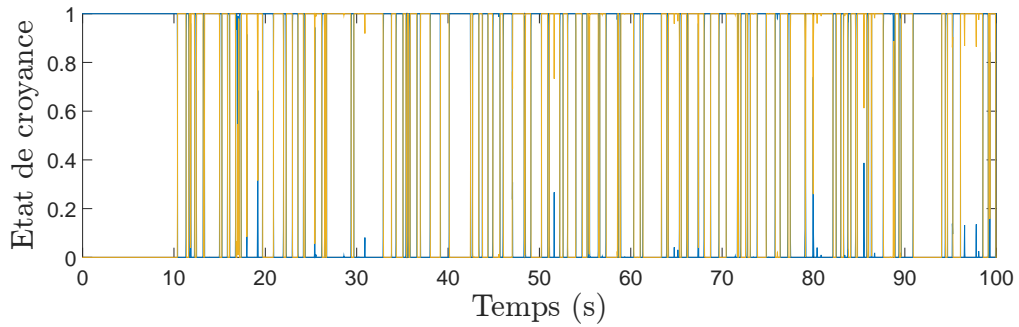


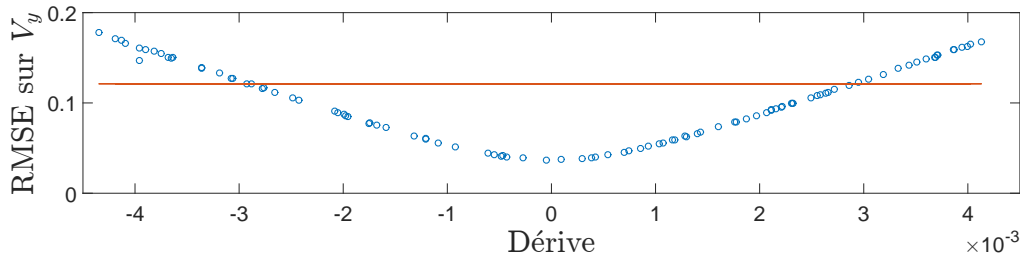
FIGURE 6.22 – État de croyance en cas de blocage à 0.0794 au niveau du capteur p

atteint l'altitude de 70m, il est possible que le drone détecte un retour à la normal mais de façon temporaire. En effet, la panne est de nouveau détectée lorsque le drone continue sa descente.

Simulations en cas de dérive :

En cas de dérive, l'intégrité sur les mesures est toujours respectée, exceptée pour la mesure de V_y en cas de fortes dérives sur ϕ , comme montrée sur la Figure 6.23.

L'intégrité sur l'état est respectée sur les capteurs $p, q, a_{sy}, \psi, \alpha$ et β . Pour les autres capteurs, l'intégrité n'est pas toujours respectée sur l'ensemble des états. En cas de faibles dérives sur a_{sx} , les états u, w, x, z et E ne sont pas intègre pour de faibles dérives. En cas de faibles dérives sur a_{sz} , les états u, w, x, z et q ne sont pas intègre là aussi pour de faibles dérives. En cas de dérive sur ϕ , les états v, p, r, y et ϕ sont de plus en plus biaisés lorsque la pente de la dérive augmente. En cas de dérive sur θ , les états u, q, x, z et θ ne sont pas intègre pour des pentes de dérive importante. En cas de dérive sur r , l'intégrité des états r et ϕ n'est pas toujours respectée. En cas de dérive sur z_B , l'intégrité de l'état z n'est pas respectée pour de

FIGURE 6.23 – Erreur sur la mesure V_y en cas de dérive sur le capteur ϕ

faibles biais. Ainsi, en cas de dérives, il y a quelques problèmes d'intégrité. Malgré tout, l'intégrité sur les mesures étant respectée, la trajectoire du drone reste proche de la trajectoire de référence.

Par ailleurs, en cas de dérives il y a de nombreuses *MI* qui restent malgré tout temporaire ($< 1s$ pour la plupart). Il y a 9 *MI* sur r (capteur ϕ détecté), 1 sur a_{sx} (capteur θ détecté), 1 sur a_{sz} (capteur θ détecté), 13 sur θ (capteur a_{sz} détecté) et 92 sur ϕ (capteur r détecté).

De façon générale, une faible dérive, c'est-à-dire proche du taux nul, est plus longue à être détectée. Le Tableau 6.10 montre les taux de détections pour les différents capteurs en cas de dérive.

	$MD_{\%}$	$TD > 5s$ (%)	$TD \in [1, 5]s$ (%)	$TD \in]0, 1]s$ (%)	$TD = 0s$ (%)
p	0%	23%	77%	0%	0%
q	0%	9%	91%	0%	0%
r	0%	15%	85%	0%	0%
a_{sx}	0%	0%	6%	94%	0%
a_{sy}	0%	0%	3%	97%	0%
a_{sz}	0%	0%	5%	95%	0%
ψ	0%	38%	62%	0%	0%
ϕ	0%	58%	42%	0%	0%
θ	0%	3%	97%	0%	0%
α	0%	10%	90%	0%	0%
β	0%	11%	89%	0%	0%
z_B	0%	4%	15%	81%	0%

Tableau 6.10 – Récapitulatif des détections en cas de dérive

“ce qu'il faut retenir”

Il n'y a pas de *FA* mais il est possible d'avoir des *MI* temporaires. Il y a très peu de *MD* et la plupart des détections sont immédiates, excepté pour les cas de dérives. Les dérives sont plus lentes à être détectées et entraîne quelques problèmes d'intégrité, c'est-à-dire que l'état est légèrement biaisé.

6.3.2.3 Influence du choix de Q^b

Les matrices Q^x et Q^b choisies sont arbitraires, ainsi cette section a pour but de montrer l'influence de ces matrices. Comme vu lors de la Section précédente, il y a quelques problèmes d'intégrités notamment en cas de dérives sur les capteurs. Il peut alors être intéressant d'augmenter la matrice Q^b pour laisser aux paramètres de fautes, plus de "liberté" de mouvement. Ainsi les dérives sont plus faciles à capter et l'état devrait moins dériver et donc être moins biaisé.

Les simulations de la section précédente ont été rejouées avec la même matrice $Q^x = 2 \times 10^{-8} \mathbb{I}_n$ et une matrice $Q^b = 2 \times 10^{-8} \mathbb{I}_{n_b}$ plus élevée.

De façon générale, le temps de détection n'est pas vraiment impacté par le choix de la matrice Q^b . Le Tableau 6.11 montre les différents taux de détection en fonction de la matrice Q^b .

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0, 1]s$ (%)	$TD = 0s$ (%)
$Q^b = 2 \times 10^{-8} \mathbb{I}_{n_b}$	0.48%	3.48%	15%	10.5%	70.5%
$Q^b = 2 \times 10^{-10} \mathbb{I}_{n_b}$	0.08%	4.48%	14.8%	10.1%	70.6%

Tableau 6.11 – Taux de détections en cas de pannes capteurs sur 4800 simulations

Les taux de détection ayant un temps de détection inférieur à 5s sont semblables. Il y a légèrement plus de MD avec la matrice Q^b plus élevée mais le taux de détection tardives est inférieur. Par ailleurs, alors qu'il y avait des MI temporaires avec $Q^b = 2 \times 10^{-10} \mathbb{I}_{n_b}$ (123 dont 92 en cas de dérive sur ϕ), il n'y a plus de MI avec une matrice Q^b plus élevée.

Concernant l'estimation, avec une matrice Q^b plus élevée, il n'y a plus de problème d'intégrité sur la mesure V_y en cas de dérive sur ϕ et il y a moins de problèmes d'intégrité sur l'état. Contrairement à la section précédente,

- l'intégrité de l'état en cas de perte d'efficacité sur r , a_{sx} , ϕ et θ est respectée ;
- l'intégrité de l'état en cas de biais sur ϕ est respectée ;
- l'intégrité de l'état en cas de blocage sur ϕ et de θ est respectée ;
- l'intégrité de l'état en cas de dérive sur r et θ est respectée.

Ainsi certains problèmes d'intégrité sont réglés, mais certains persistent :

- en cas de faibles perte d'efficacité sur z_B , des problèmes d'intégrité sont toujours présents sur l'état z ;
- en cas de blocage sur r des problèmes d'intégrité sont toujours présents sur les états v et r et en cas de blocage sur a_{sx} des problèmes d'intégrité sont toujours présents sur l'état E (mais l'état u est maintenant intègre) ;
- en cas de dérive sur a_{sx} , sur a_{sz} , sur ϕ et sur z_B des problèmes d'intégrité persistent.

“ce qu'il faut retenir”

Une valeur pour les termes de Q^b équivalente à celle des termes de Q^x permet d'améliorer l'intégrité du filtre et le temps de détection n'est pas pour autant dégradé.

6.3.3 Cas où l'on considère les pannes actionneurs et capteurs

Le but de cette section est de montrer l'impact sur la précision de l'estimation, ainsi que sur le temps de détection, suite à la prise en compte des paramètres de fautes actionneurs et capteurs. Dans cette section, le filtre utilisé est le filtre AKF séquentiel.

6.3.3.1 Comportement en mode nominal

500 simulations sont réalisées en mode nominal avec 500 séquences de bruits sur les capteurs. Ces simulations permettent d'évaluer le taux de FA . Sur les 500 simulations, il n'y a pas de fausses alarmes. Ainsi le taux est nul, $FA_{\%} = 0\%$. Néanmoins il y a 31 réouvertures inappropriées de la covariance associés aux capteurs ; 3 pour p , 6 pour q , 3 pour a_{sy} , 3 pour a_{sz} , 5 pour ψ , 3 pour ϕ , 5 pour α et 3 pour β . Il n'y a donc pas de réouverture de la covariance pour les capteurs r , a_{sx} , θ et z_B . Par ailleurs, le filtre est intègre pour l'ensemble des simulations. Le filtre donne donc de bons résultats dans le mode nominal.

6.3.3.2 Comportement en cas de pannes

Dans cette section la matrice Q^x utilisée est égale à $Q^x = 2 \times 10^{-8} \mathbb{I}_n$. La matrice Q^b vaut $Q^b = 2 \times 10^{-10} \mathbb{I}_{n_b}$.

Simulations de pannes moteur :

L'allure des courbe est similaire à celles présentées lors de la Section 6.3.1.2. La précision de l'estimateur en cas de perte d'efficacité ou de biais est très bonnes et semblables au cas où l'on considère uniquement les pannes actionneurs. En cas de blocage d'une valeur supérieure à 0.5, l'erreur d'estimation sur le paramètre b_1^1 est plus importante comme on peut le voir sur la Figure 6.24.

L'erreur est importante pour des blocages élevés pour deux raisons. Contrairement au cas où l'on considèrerait uniquement les actionneurs, la covariance des paramètres de faute s'ouvre plus tardivement. En effet, il faut attendre de voir l'effet de la panne sur la mesure du capteur V_x avant de pouvoir ouvrir la covariance. Pendant ce temps, le système de commande tente de compenser l'effet de la panne en mettant la commande en saturation. Ainsi lorsque la covariance des paramètres de faute associé au moteur est réouverte, la commande est déjà en saturation, il est donc difficile d'évaluer précisément la valeur des paramètres de fautes.

Contrairement au cas où l'on considèrerait uniquement les actionneurs, il est importante de noter qu'il y a 11 MI temporaires des élevons ; 1 pour les pertes d'effi-

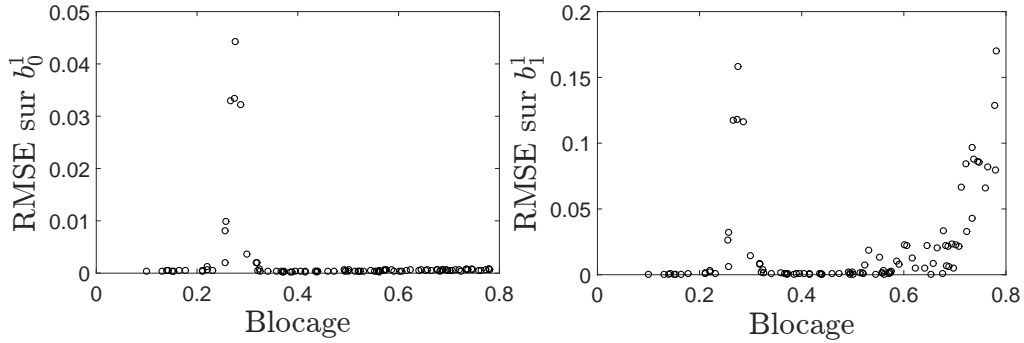


FIGURE 6.24 – Erreur sur les paramètres de fautes associés au moteur en cas de blocage sur 99 simulations

capacité, 1 en cas de blocage et 9 en cas de biais. Le taux est donc de $MI\% = 3.67\%$. Pour le cas de perte d'efficacité, la MI n'est pas due à une réouverture inapproprié de la covariance. Les élevons sont détectés en panne entre 12.36s et 13.3s. Comme la covariance des paramètres de fautes associées au moteur est ouverte plus tardivement, les paramètres de fautes des élevons vont essayer de compenser la faute en dérivant légèrement de zéro. Les autres MI , c'est-à-dire celles en cas de blocage ou de biais, sont dues à des réouvertures inappropriées de la covariance des paramètres de fautes associées aux élevons. Néanmoins ces MI durent moins de 1s (détection temporaire entre 10s et 11s), par conséquent elles auront peu d'impact. En effet, la tâche de décision est activée avec une période de 1s. Or à 11s, le moteur est bien détecté en panne.

Le Tableau 6.12 montre les taux de détection en cas de pannes du moteur avec le filtre prenant en compte uniquement les paramètres de fautes actionneurs (F_{act}) et celui prenant en compte les paramètres de fautes actionneurs et capteurs ($F_{act/cap}$).

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0.5, 1]s$ (%)	$TD \leq 0.5s$ (%)
F_{act}	4.67%	1.33%	6.33%	7.67%	80%
$F_{act/cap}$	6%	7%	18%	1%	68%

Tableau 6.12 – Taux de détections en cas de pannes moteur sur 300 simulations

De façon générale, le temps de détection est plus long avec le filtre $F_{act/cap}$ qu'avec le filtre F_{act} . En effet, avec le filtre $F_{act/cap}$ il faut attendre de voir l'impact sur la vitesse V_x . Le taux de détection pour lesquels $T_D < 0.5s$ correspondent à des pannes importantes qui auront pas conséquent un impact quasiment immédiat sur la vitesse V_x . Pour les pannes moins importantes, le temps de détection est plus lent avec le filtre $F_{act/cap}$, le temps de voir l'impact sur la vitesse V_x .

Simulations de pannes élevo1 :

L'allure des courbe est similaire à celles présentées lors de la Section 6.3.1.2. La

précision de l'estimateur en cas de perte d'efficacité, de blocage ou de biais est très bonne et semblables au cas où l'on considère uniquement les pannes actionneurs.

Contrairement au cas où l'on considèrerait uniquement les actionneurs, il est importante de noter qu'il y a 180 *MI* temporaires des capteurs ; 96 pour les pertes d'efficacité, 42 en cas de blocage et 42 en cas de biais. Le taux est très élevé, $MI_{\%} = 60\%$. Parmi les 180 *MI*, 27 correspondent à des *MI* temporaires du capteur q et 153 correspondent à des *MI* temporaires des capteurs a_{sx} . Ces *MI* ont le désavantage de retarder le temps de détection des élevons comme on peut le voir sur le Tableau 6.13.

	$MD_{\%}$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0.5, 1]s$ (%)	$TD \leq 0.5s$ (%)
F_{act}	1%	2.33%	3%	2.67%	91%
$F_{act/cap}$	1%	4.67%	6.33%	1%	87%

Tableau 6.13 – Taux de détections en cas de pannes élévon1 sur 300 simulations

Simulations de pannes capteur :

Pour les pannes capteurs, les simulations réalisées avec le filtre prenant en compte les paramètres de fautes actionneurs et capteurs ($F_{act/cap}$) sont plutôt semblables aux simulations réalisées avec le filtre prenant en compte uniquement les paramètres de fautes capteurs (F_{cap}) excepté pour le capteur a_{sx} .

Pour le capteur a_{sx} , il est évident que le filtre ne peut pas donner le même résultat. En effet, lorsqu'une panne est supposée, celui-ci est mis en quarantaine pendant $2s$ de façon à pouvoir distinguer les pannes moteur des pannes a_{sx} . Le temps de détection est forcément plus long avec le filtre $F_{act/cap}$. Le Tableau 6.14 montre la différence obtenue au niveau des temps de détection.

	$MD_{\%}$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0, 1]s$ (%)	$TD = 0s$ (%)
F_{cap}	0%	2%	3.25%	27%	67.75%
$F_{act/cap}$	0.25%	4%	95.75%	0%	0%

Tableau 6.14 – Taux de détections en cas de pannes du capteur a_{sx} sur 400 simulations

La *MD* correspond en réalité à une *MI* permanente du moteur, c'est-à-dire que le moteur est détecté en panne. Cette simulation correspond à une dérive très faible, par conséquent le seuil de panne n'est pas dépassé et les paramètres de fautes associés au moteur dévient légèrement de zéro pour compenser la faute. De plus, il y a 7 *MI* temporaires du moteur, 3 en cas de dérive et 4 en cas de pertes d'efficacité. Ces *MI* peuvent avoir un impact sur la tâche de décision du contrôleur. Ces impacts sont étudié au Chapitre 8.

Remarque 6.5

Il est possible d'avoir des MI temporaires des éleveurs ou du moteur pour d'autres cas de pannes capteur. En cas de blocage sur ϕ , il y a une MI des éleveurs ; cette MI est inférieure à 1s. En cas de dérive sur p , il y a 78 MI des éleveurs. En cas de dérive sur a_{sz} , il y a 6 MI temporaires sur les éleveurs. L'impact de ces MI est étudié au Chapitre 8.

“ce qu'il faut retenir”

La prise en compte des pannes capteurs et actionneurs entraîne une augmentation des MI temporaires qui restent pour la plupart inférieure à 1s et qui auront par conséquent peu d'impacts sur le comportement du drone. La précision de l'estimation est très peu dégradée. Cependant, comme attendu, les temps de détection en cas de pannes moteur et en cas de pannes du capteur a_{sx} sont dégradés. Ces deux types de pannes ayant la même signature de faute, il faut attendre d'observer ou non une erreur sur la vitesse V_x pour pouvoir discriminer la panne.

6.4 Validation sur mesures générées à partir du modèle non linéaire

En utilisant le modèle non linéaire pour générer les mesures, on se rapproche de la réalité mais on augmente l'incertitude au niveau du filtre de Kalman. Les incertitudes peuvent provenir des bruits des capteurs mais aussi des approximations dues aux linéarisations. Le but de cette section est d'évaluer la méthode face aux incertitudes de modèle, ce mot recouvrant toutes les différences et sources d'erreur existant entre le comportement du système et son modèle : par exemple les non linéarités ou les imprécisions de paramètres. Par ailleurs, pour être plus fidèle à la réalité, les filtres utilisés lors de cette section sont séquentiels de façon à mettre à jour l'estimation uniquement lorsque les mesures sont reçues.

6.4.1 Cas où l'on considère uniquement les pannes actionneurs

6.4.1.1 Utilisation du filtre AKF séquentiel

Pour cette section deux types de simulations ont été réalisées :

- Celles pour lesquelles les mesures sont générées avec un modèle simplifié au niveau des éleveurs, c'est-à-dire que les éleveurs sont modélisés uniquement avec un premier ordre.
- Celles pour lesquelles les mesures sont générées avec un modèle plus réaliste au niveau des éleveurs, c'est-à-dire que les éleveurs sont modélisés par la combinaison d'un retard, d'une saturation en vitesse et d'un premier ordre.

Ces deux types de simulations ont deux matrices Q^x différentes. En effet, la matrice Q^x est calculé selon la méthode décrite dans la Section 5.1.5. La matrice

Q^x obtenue avec le modèle simplifié a des termes de faible valeur pour l'ensemble des états. Celle obtenue avec le modèle complexe possède deux termes assez élevés : 0.0062 pour l'état p et 0.0013 pour l'état δ_a .

En cas de pannes moteur, les résultats sont semblables avec ces deux modélisations. En effet, la modélisation n'impacte pas l'actionneur moteur. Par conséquent, les résultats des cas de pannes moteur montrés ci-après sont tirés du modèle prenant en compte le modèle le plus réaliste. En cas de pannes élevons, il est intéressant de comparer les deux résultats étant donné que le modèle linéaire, celui utilisé par l'AKF, modélise la dynamique des élevons avec un unique premier ordre.

Simulations de pannes moteur :

Pour des fautes importantes, c'est-à-dire des fautes pour lesquelles le contrôleur n'arrive pas à ramener à l'état à son état d'équilibre, la méthode FDD ne parvient plus à estimer correctement les paramètres de faute et l'état n'est plus intègre. C'est le cas pour les fautes suivantes :

- les blocages et
- les pertes d'efficacité supérieures à 70%.

En effet, lorsque l'on s'éloigne trop du point d'équilibre, le modèle linéaire utilisé dans l'AKF n'est plus suffisamment représentatif du modèle non linéaire. Par conséquent différents seuils vont être dépassés, les paramètres de fautes du moteur et des élevons vont être modifiés sans jamais converger vers une valeur satisfaisante. Pour les autres types de pannes, l'estimation réussie à converger vers une valeur. La Figure 6.25 montre les erreurs sur les paramètres de fautes associés au moteur en cas de perte d'efficacité et en cas de biais sur le moteur.

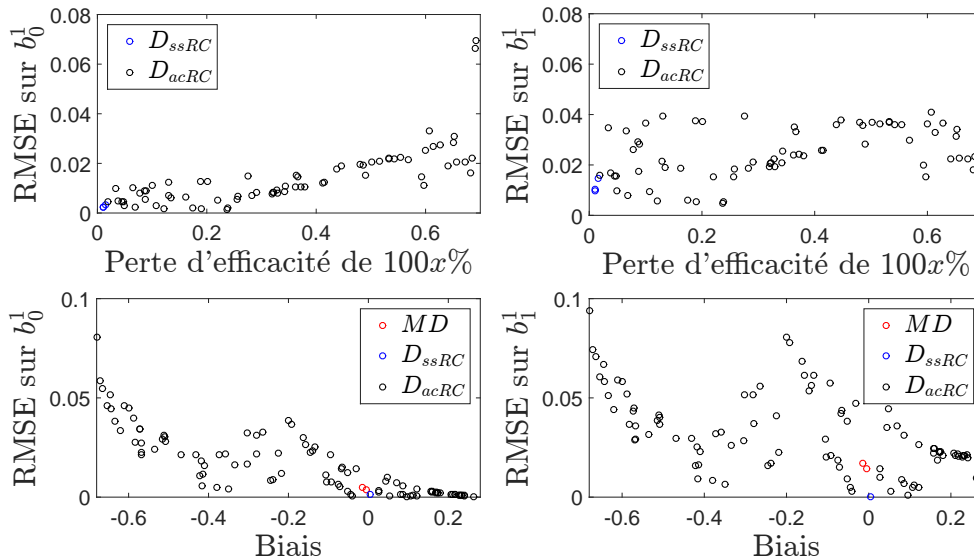


FIGURE 6.25 – Erreurs sur les paramètres de fautes associés au moteur en cas de perte d'efficacité et de biais sur le moteur

Ces erreurs sont plus importantes que lorsque les mesures sont générées par

6.4. Validation sur mesures générées à partir du modèle non linéaire 39

le modèle linéaire. Néanmoins, le temps de détection est semblable au cas où les mesures sont générées par le modèle linéaire. Le temps de détection est plus long pour des faibles pertes d'efficacité, ainsi que pour des faibles biais, ce qui correspond bien aux résultats obtenus lors de la Section précédente. Il y a 7 détections tardives, pour des pertes d'efficacité entre 1.8% et 6.5%. Les autres temps de détection sont montrés sur la Figure 6.26.

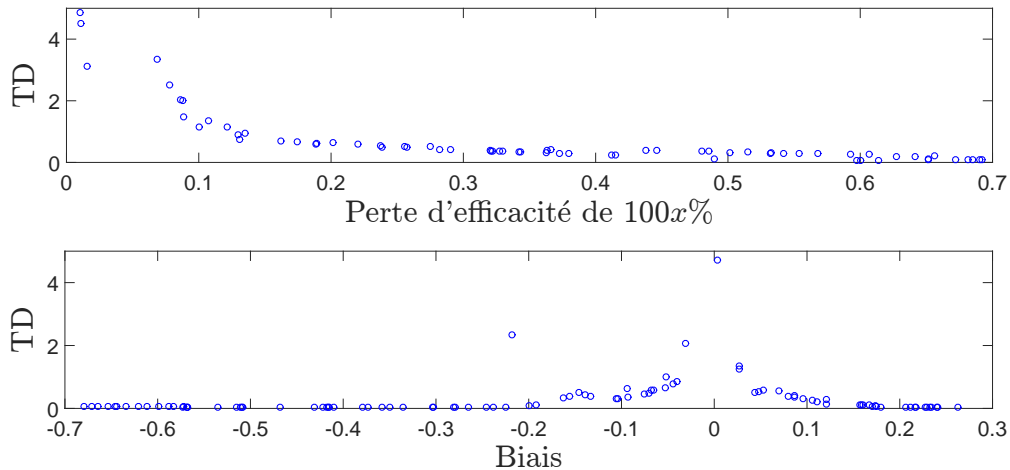


FIGURE 6.26 – Temps de détection en cas de perte d'efficacité et de biais sur le moteur

Simulations de pannes élevo 1 :

De façon similaire au moteur, pour des fautes importantes, c'est-à-dire des fautes pour lesquelles le contrôleur n'arrive pas à ramener à l'état à son état d'équilibre, la méthode FDD ne parvient plus à estimer correctement les paramètres de faute et l'état n'est plus intègre. C'est le cas pour les blocages à des valeurs inférieures à $-0.2rad$.

Le but est aussi de voir l'impact d'une modélisation simplifiée des actionneurs sur l'estimation des paramètres de fautes. Les erreurs sont, quel que soit la panne, plus importantes lorsque les mesures sont générées à partir du modèle prenant en compte une modélisation plus complexe. Par exemple, la Figure 6.27 montre les erreurs obtenues avec les deux types de modélisation pour les mesures (\circ pour le modèle complexe et \times pour le modèle simplifié).

Les erreurs sont supérieures et assez importantes lorsque les mesures sont générées à partir du modèle prenant en compte une modélisation plus complexe. Au vu des résultats obtenus, il est donc important de modéliser le retard ainsi que la saturation en vitesse au niveau du modèle linéaire pour avoir une estimation précise des paramètres de fautes élevo.

De plus, comme on peut le voir sur la Figure 6.27, il y a des MD pour des pertes d'efficacité inférieures à 60% avec le modèle complexe alors qu'il n'y en a pas avec le modèle simplifié. En effet, il est difficile de détecter les pannes car la covariance

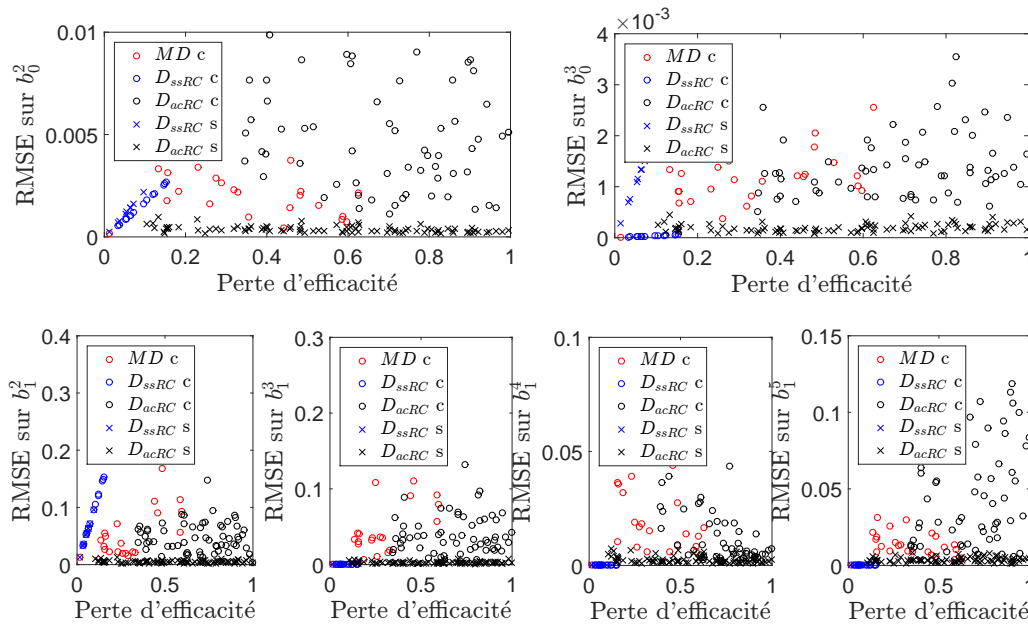


FIGURE 6.27 – Erreurs sur les paramètres de fautes associés aux éleveurs en cas de perte d'efficacité de l'élevon 1 - c : modèle complexe - s : modèle simplifié

est très grande, ce qui traduit un manque de confiance en la valeur estimée. Cette covariance est grande car la matrice Q^x est importante pour les termes p et δ_a .

“ce qu'il faut retenir”

Deux faits sont à retenir des simulations précédentes. Premièrement, le filtre n'est pas adapté si l'on s'éloigne de la valeur d'équilibre. Or en se plaçant dans des cas pannes, le drone est amené à s'éloigner de l'équilibre. Pour pallier ce problème, il faut utiliser un filtre étendu. Deuxièmement, la modélisation de la dynamique actionneur utilisée dans le filtre doit être représentative de la réalité pour avoir une bonne précision au niveau de l'estimation.

6.4.1.2 Utilisation du filtre EKF séquentiel

Le filtre EKF utilise les équations de l'Annexe A avec le modèle simplifié au niveau des éleveurs, c'est-à-dire avec un premier ordre. Ces équations doivent être discrétisées dans le temps pour pouvoir être utilisées. On fait le choix de discrétiser les équations différentielles avec la méthode d'Euler au premier ordre avec une fréquence de $200Hz$. Au niveau du filtre EKF, ces équations discrétisées sont bouclées 4 fois pour se ramener à $50Hz$. Cette étape est nécessaire pour pouvoir capter l'effet immédiat des éleveurs sur les autres états du système.

Remarque 6.6

¹ La méthode d'Euler au premier ordre est utilisée de façon à limiter la taille des

6.4. Validation sur mesures générées à partir du modèle non linéaire 41

équations. Néanmoins il serait nécessaire d'avoir une discrétisation avec un ordre plus élevé -ou avec une autre méthode telle que Runge-Kutta- pour avoir une meilleure précision.

Simulations de pannes moteur :

Contrairement au filtre AKF, le filtre EKF arrive à converger vers une valeur lorsque le drone est sujet à des pannes importantes. La Figure 6.28 montre les erreurs sur les paramètres de fautes associés au moteur en cas de pannes moteur.

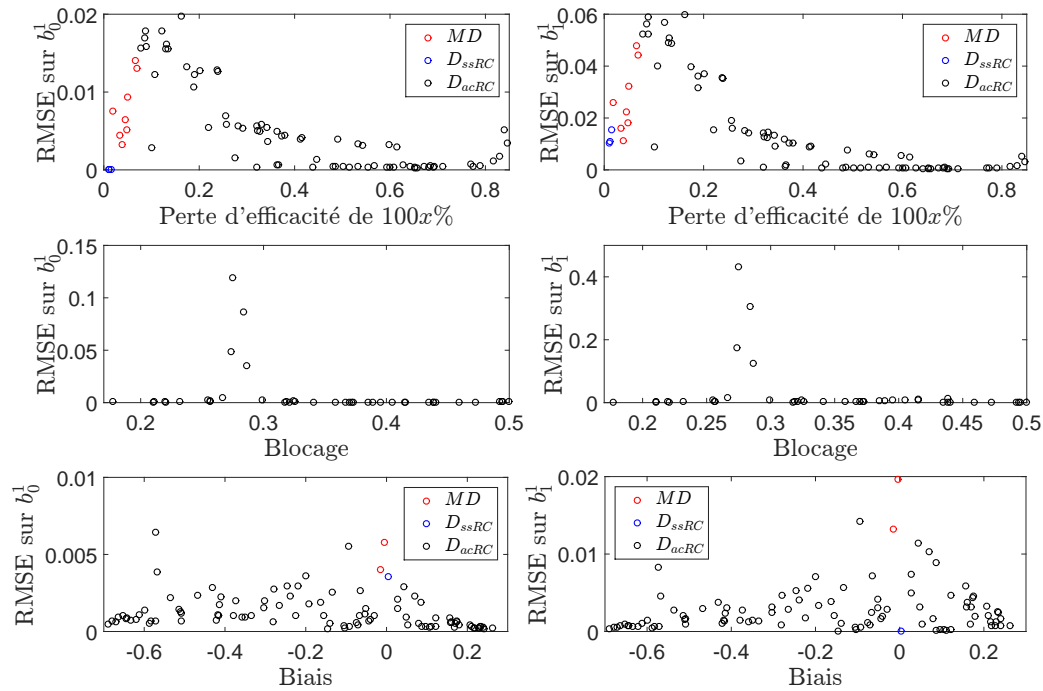


FIGURE 6.28 – Erreurs sur les paramètres de fautes associés au moteur en cas de pannes moteur

Le filtre EKF permet non seulement de converger vers une valeur mais aussi d'obtenir une meilleure précision au niveau des paramètres de fautes. De façon similaire aux résultats obtenus avec le modèle linéaire, il y a quelques MD pour des pannes faibles, c'est-à-dire pour de faibles pertes d'efficacité et pour de faibles biais. Le temps de détection est lui aussi semblable à celui obtenu avec le modèle linéaire.

Remarque 6.7

Pour des pertes d'efficacité supérieures à 85%, le système, représenté ici par le modèle non linéaire, devient instable. Néanmoins, avant de devenir instable, la méthode FDD détecte la panne et arrive à converger vers une valeur précise pour les paramètres de fautes. C'est aussi le cas pour des blocages à une valeur supérieure à 0.5.

Simulations de pannes éleवन 1 :

Contrairement à l'actionneur moteur qui a une dynamique lente $\tau = 0.4$, les actionneurs éleवन ont une dynamique rapide $\tau = 0.013$. La discrétisation des équations différentielles au premier ordre a peu d'impact au niveau de l'effet du moteur sur les autres états étant donné sa dynamique lente. Au contraire, les éleवन ayant une dynamique rapide, une discrétisation des équations différentielles au premier ordre peut entraîner des erreurs au niveau de la modélisation des effets immédiats des éleवन sur les autres états. Le but ici est de comparer les résultats obtenus avec l'AKF, qui se base sur un modèle qui discrétise les équations différentielles avec la méthode Runge-Kutta d'ordre 4, et les résultats obtenus avec l'EKF. La Figure 6.29 montre les erreurs sur les paramètres de fautes associés aux éleवन en cas de pertes d'efficacité sur l'éleवन 1 avec le filtre AKF et avec le filtre EKF.

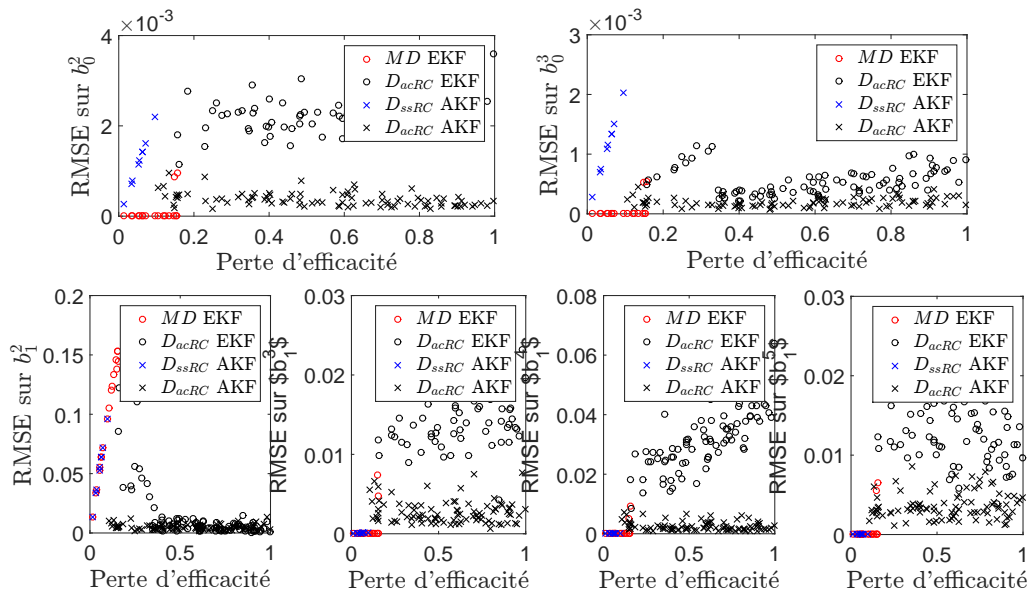


FIGURE 6.29 – Erreurs sur les paramètres de fautes associés aux éleवन en cas de pertes d'efficacité sur l'éleवन 1

Comme le montre la Figure, le filtre AKF est plus précis. Par conséquent, la méthode de discrétisation utilisée pour les équations différentielles n'est pas adaptée à la dynamique des éleवन.

“ce qu'il faut retenir”

Le filtre EKF permet bien d'estimer les paramètres de fautes, même lorsque l'on s'éloigne de l'équilibre. Cependant, la méthode de discrétisation des équations différentielles représentant le système doit être adaptée au dynamique rapide des actionneurs pour pouvoir représenter le plus fidèlement possible les effets immédiats sur les autres états du système.

6.4.2 Cas où l'on considère uniquement les pannes capteurs

Les résultats de cette section proviennent de simulations pour lesquelles les mesures sont générées avec le modèle plus réaliste au niveau des éleveurs, c'est-à-dire que les éleveurs sont modélisés par la combinaison d'un retard, d'une saturation en vitesse et d'un premier ordre.

Les résultats des simulations avec l'AKF ou l'EKF sont assez semblables à ceux obtenus avec le modèle linéaire excepté pour le capteur p . Pour ce capteur, le temps de détection est beaucoup plus lent. Le Tableau 6.15 montre les temps de détection obtenus avec le modèle linéaire F_{lin} et ceux obtenus avec le modèle non linéaire F_{nlin} .

	$MD\%$	$TD > 5s$ (%)	$TD \in]1, 5]s$ (%)	$TD \in]0, 1]s$ (%)	$TD = 0s$ (%)
F_{lin}	0%	12%	26.33%	1.33%	60.33%
F_{nlin}	3.33%	41.67%	1.67%	51.33%	1.2%

Tableau 6.15 – Taux de détections en cas de pannes du capteur p sur 300 simulations

Bien que le temps de détection soit moins bon, le filtre reste intègre. Le temps de détection est moins bon à cause de la matrice Q^x . En effet, lorsque l'on utilise les mesures générées avec le modèle plus réaliste au niveau des éleveurs, la valeur associée à l'état p est importante 0.0062, contre 2×10^{-8} avec le modèle linéaire. Ainsi, lorsque la covariance du paramètre de faute associé à p est réouverte, il est difficile par la suite de diminuer cette valeur étant donné le manque de confiance que l'on a dans l'état p . Ainsi, malgré une bonne estimation du paramètre de faute associé à p , il est plus difficile de détecter la panne à cause d'un manque de confiance, c'est-à-dire à cause d'une covariance élevée.

“ce qu'il faut retenir”

S'il y a beaucoup d'incertitudes sur l'état entre le modèle utilisé pour l'estimation et le modèle réel, le temps de détection est dégradé au niveau des capteurs. Une valeur de Q^x grande traduit un manque de confiance qui empêche une diminution rapide de la covariance associées aux paramètres de fautes.

6.5 Conclusion

Lors de cette section, deux types de résultats ont été établis ; ceux pour lesquels les mesures sont générées avec le modèle linéaire et ceux pour lesquels les mesures sont générées avec le modèle non linéaire.

Dans le premier cas, c'est-à-dire lorsque l'on utilise le modèle linéaire pour générer les mesures, on se place dans un cas où le modèle du drone disponible est très

proche de la réalité. Les simulations donnent de très bons résultats ; l'estimation est précise et la détection rapide.

Dans le second cas, c'est-à-dire lorsque l'on utilise le modèle non linéaire, on se place dans un cas où le modèle disponible est moins précis et plus éloigné de la réalité. Les simulations réalisées permettent d'évaluer la méthode face à différents types d'incertitudes qui peuvent alors nuire au bon fonctionnement de la méthode FDD développée. Dans un premier temps, il est important de noter qu'un filtre de Kalman étendu est nécessaire pour détecter les pannes importantes. En effet, en cas de pannes importantes, le drone s'éloigne de son état d'équilibre, par conséquent le modèle linéarisé n'est plus suffisamment représentatif de la réalité et le filtre AKF n'arrive plus à converger. Dans un second temps, les simulations réalisées avec l'AKF ont montré qu'il était important d'avoir un modèle représentant précisément la dynamique des élevons. En cas d'incertitudes au niveau des élevons dues à une simplification de la dynamique des élevons, l'estimation des paramètres de fautes associés aux élevons n'est pas précise. Dans un troisième temps, les simulations réalisées avec l'EKF ont montré qu'il était important de choisir une méthode de discrétisation adaptée aux dynamiques rapides de façon à représenter fidèlement l'effet immédiat des élevons sur les états de l'avion. Par conséquent, les incertitudes de modèle impactent la précision de l'estimation. L'impact d'une estimation moins précise sur la trajectoire du drone est évalué au Chapitre 8.

Cette méthode FDD a été améliorée au fur et à mesure de la complexification du modèle et a fait l'objet de publications :

- la méthode a dans un premier temps été appliquée en faisant l'hypothèse qu'un seul actionneur pouvait tomber en panne [Boche 2018],
- la méthode a ensuite été appliquée en faisant l'hypothèse que les deux actionneurs redondants pouvaient tomber en panne [Boche 2017a],
- la méthode a ensuite été appliquée en faisant l'hypothèse que tous les actionneurs pouvaient tomber en panne [Boche 2017b],
- finalement la méthode a été appliquée en faisant l'hypothèse que tous les actionneurs pouvaient tomber en panne ainsi que certains capteurs.

Amélioration de la performance par reconfiguration du système de commande

Sommaire

7.1	Principe du système de reconfiguration	147
7.2	Définition de l'indicateur de performance	147
7.3	Conception des contrôleurs	148
7.3.1	Conception du contrôleur nominal	148
7.3.2	Conception d'un contrôleur adaptatif pour les fautes d'actionneurs redondants	149
7.3.3	Conception d'un contrôleur adaptatif pour les fautes d'actionneurs non redondants	152
7.4	Utilisation de méthodes d'optimisation pour sélectionner le contrôleur	153
7.4.1	Hypothèses du processus de prise de décision	153
7.4.2	Utilisation du formalisme POMDP	155
7.4.3	Utilisation du formalisme MPC	160
7.4.4	Utilisation du formalisme SMPC	160

Résumé

CE chapitre vise à présenter le système de reconfiguration de la méthode proposée. Dans un premier temps, les différents contrôleurs conçus pour cette méthode sont présentés. Un contrôleur nominal et des contrôleurs adaptatifs pour les modes dégradés sont développés en se basant sur la méthode LQR. Par la suite, la logique permettant de changer de contrôleur est détaillée. Différentes variantes sont présentées.

7.1 Principe du système de reconfiguration

Le but d'un système de reconfiguration est d'améliorer la performance du système en cas de panne tout en conservant une performance acceptable quand le système est nominal. Un ensemble de contrôleurs par retour d'état estimé est donc développé pour couvrir une bonne performance pour les différents modes. Pour le mode nominal, le contrôleur est conçu de façon à optimiser un certain critère. Pour les modes de pannes, trois cas sont considérés :

- les modes de pannes qui correspondent à des pannes capteurs. Dans ce cas, il est supposé que, malgré la présence d'une faute, le filtre de Kalman utilisé pour la détection et le diagnostic de faute est suffisamment robuste pour fournir une estimation de l'état non biaisée. Dans ce cas, le contrôleur nominal reste valide pour cette situation.
- les modes de pannes qui correspondent à des pannes d'actionneurs redondants. Ici, le gain du contrôleur nominal peut être utilisé. Cependant, la distribution du signal de commande doit considérer les paramètres de fautes de l'actionneur dans le but de compenser l'action de l'actionneur en panne en utilisant le ou les actionneurs redondants.
- les modes de pannes qui correspondent à des pannes d'actionneurs non-redondants. Dans ce cas, un nouveau contrôleur doit être conçu.

7.2 Définition de l'indicateur de performance

Un des objectifs d'une FTC est de maintenir la performance au meilleur niveau malgré les pannes. Cette performance peut être définie de plusieurs façon ; elle peut prendre en compte par exemple la stabilité, le suivi de consigne, le rejet de perturbations ou encore le temps de réponse. Une façon de la définir est le critère J_0^∞ avec :

$$J_k^{k''} = \sum_{k'=k}^{k'=k''} \alpha^{k'} (\mathbf{x}_{k'}^T Q_p \mathbf{x}_{k'} + \mathbf{u}_{k'}^T R_p \mathbf{u}_{k'}) \quad (7.1)$$

L'équation 7.1 correspond à un critère mixte état-commande avec un facteur de réduction $\alpha \in]0, 1[$.

Un facteur de réduction est utilisé de manière à avoir un indicateur fini sur un horizon infini même dans les cas de pannes. En effet, pour les pannes les moins graves on peut s'attendre à arriver à maintenir l'état \mathbf{x}_k à zéro au prix d'une commande \mathbf{u}_k non nulle. C'est-à-dire que le terme $\mathbf{u}_{k'}^T R_p \mathbf{u}_{k'}$, serait de toute manière supérieur à une constante. De plus, pour les pannes les plus graves, on peut s'attendre à ne pas pouvoir maintenir l'état \mathbf{x}_k à zéro.

Pour l'atterrissage de l'Altimum 4, le but en cas de pannes actionneurs est d'adapter la commande de façon à ce que le drone suive toujours sa trajectoire de référence et donc la pente γ . La position horizontale x and la position verticale z sont liées par la relation suivante : $z = \tan(\gamma)x$. En cas d'incapacité à maintenir

tous les états à zéro, la priorité est donnée au maintien de la pente et donc à des pondérations nulles pour tous les états à l'exception de x et z et à des pondérations proportionnelles entre x et z . Donc, pour le modèle longitudinal, la matrice Q_p est choisie de la façon suivante : $Q_p = \text{diag}\{0, 0, 0, \tan^2(\gamma), 1, 0, 0, 0, 0\}$. Pour le modèle complet, il est aussi souhaitable de ne pas dévier de la trajectoire latérale. Une pondération est donc rajoutée à l'état y . La matrice Q_p est choisie de la façon suivante : $Q_p = \text{diag}\{0, 0, 0, 0, 0, \tan^2(\gamma), 1, 1, 0, 0, 0, 0, 0\}$.

La matrice R_p est choisie de la façon suivante : $R_p = 0.1\mathbb{I}_l$. En choisissant des valeurs trop faibles pour R_p , le système est moins excité. Or le système est partiellement observable, il est donc favorable d'avoir une légère excitation permanente au niveau des commandes. En choisissant des valeurs trop élevées R_p , on prend le risque d'abimer les actionneurs sur le long terme. Ainsi, on choisit des valeurs intermédiaires pour la matrice R_p .

7.3 Conception des contrôleurs

Il faut construire le signal de commande permettant d'avoir la meilleure performance. Dans cette section, les signaux \mathbf{u}_k^c et \mathbf{u}_k , décrit sur la Figure 5.2, sont définis.

7.3.1 Conception du contrôleur nominal

D'après le critère de Kalman (cf. Annexe B), le système est commandable. Le gain K est calculé en minimisant le critère de performance J_0^∞ . Cette optimisation revient à appliquer la méthode LQR avec un facteur de réduction α (cf. Annexe F). Dans ce cas, le gain K est calculé avec l'équation classique de Riccati appliquée à la paire $(\sqrt{\alpha}A, \sqrt{\alpha}B)$ et la commande est la suivante :

$$\mathbf{u}_k^c = -K\hat{\mathbf{x}}_k \quad (7.2)$$

L'estimation de l'état $\hat{\mathbf{x}}_k$ est utilisée à la place de l'état \mathbf{x}_k car celui-ci n'est pas entièrement mesurable.

Ainsi, le contrôleur nominal du modèle longitudinal est le suivant :

$$\mathbf{u}_k = \begin{bmatrix} f(-K_1.\hat{\mathbf{x}}_k, 0, 0) \\ f(-K_2.\hat{\mathbf{x}}_k, S_{3211}, A_k) \\ f(-K_2.\hat{\mathbf{x}}_k, -S_{3211}, A_k) \end{bmatrix} \quad (7.3)$$

Où $K^T = [K_1^T K_2^T K_2^T]$, S_{3211} est le signal d'entrée 3211 qui est ajouté quand une faute élevo est suggérée.

Pour le contrôleur nominal du modèle complet, la commande est la suivante :

$$\mathbf{u}_k = \begin{bmatrix} f(-K_1 \hat{\mathbf{x}}_k, 0, 0) \\ f(-K_2 \hat{\mathbf{x}}_k, 0.6S_{3211}, A_k) \\ f(-K_3 \hat{\mathbf{x}}_k, -0.4S_{3211}, A_k) \\ f(-K_4 \hat{\mathbf{x}}_k, -0.6S_{3211}, A_k) \\ f(-K_5 \hat{\mathbf{x}}_k, 0.4S_{3211}, A_k) \end{bmatrix} \quad (7.4)$$

Où $K^T = [K_1^T K_2^T K_3^T K_4^T K_5^T]$, S_{3211} est le signal d'entrée 3211 qui est ajouté quand une faute élévon est suggérée.

7.3.2 Conception d'un contrôleur adaptatif pour les fautes d'actionneurs redondants

Dans le cas où l'actionneur en panne est redondant, le système reste contrôlable. Une panne d'un actionneur va être compensée par le ou les autre(s) actionneur(s) avec lesquels il est redondant. Un seul contrôleur est conçu pour traiter les pannes d'actionneurs redondants. Ce contrôleur permet d'améliorer les performances pour les pannes de types blocage, perte d'efficacité ou biais. Le contrôleur est conçu de façon à approcher l'équation suivante :

$$\mathbf{u}_k^f = -K \hat{\mathbf{x}}_k \quad (7.5)$$

Où \mathbf{u}_k^f est défini par l'équation 4.7.

En effet, dans chaque cas de panne, il est souhaitable d'avoir un comportement similaire à la situation nominale. Le rôle du contrôleur est alors de mimer la situation nominale. Ainsi pour la construction des contrôleurs, le gain du contrôleur nominal K est utilisé et l'on cherche à compenser la faute en ajoutant des termes adaptatifs.

7.3.2.1 Cas où l'actionneur en panne a un effet proportionnel à un ou plusieurs autre(s) actionneur(s)

Pour le modèle longitudinal, les deux élévons ont strictement le même effet, c'est-à-dire que $B_2 = B_3$. Les gains associés aux élévons sont donc les mêmes, $K_2 = K_3$. La démonstration du contrôleur est faite pour un cas plus général, dans le cas où les actionneurs ne sont pas identiques mais proportionnels et que l'actionneur en panne est proportionnel avec plusieurs autres actionneurs.

Supposons u_k^{fj} en panne et I l'ensemble des commandes redondantes avec u_k^{fj} : $I = \{i/\exists \eta_i, K_{i.} = \eta_i K_{j.}\}$.

D'un côté, avoir le même retour qu'en l'absence de fautes implique :

$$\begin{aligned} u_k^{fj} + \sum_{i \in I} u_k^{fi} &= -K_j \hat{\mathbf{x}}_k - \left(\sum_{i \in I} K_{i.} \right) \hat{\mathbf{x}}_k \\ &= -(1 + \sum_{i \in I} \eta_i) K_j \hat{\mathbf{x}}_k \end{aligned} \quad (7.6)$$

D'un autre côté, la définition de la commande implique :

$$u_k^{fj} + \sum_{i \in I} u_k^{fi} = b_0^j + (1 - b_1^j)u_k^j - b_1^j u_0^j + \sum_{i \in I} (b_0^i + (1 - b_1^i)u_k^i - b_1^i u_0^i) \quad (7.7)$$

En combinant (7.6) et (7.7), on définit Φ par l'expression suivante qui doit être nulle :

$$\Phi = (1 + \sum_{i \in I} \eta_i)K_j \widehat{\mathbf{x}}_k + b_0^j + (1 - b_1^j)u_k^j - b_1^j u_0^j + \sum_{i \in I} (b_0^i + (1 - b_1^i)u_k^i - b_1^i u_0^i) \quad (7.8)$$

Dès lors, le problème d'optimisation à résoudre pour maintenir l'action de commande sous régime de panne tout en minimisant l'énergie de commande, est le suivant :

$$\begin{aligned} \min_{u_k^j, u_k^i} \quad & \frac{1}{2}((u_k^j)^2 + \sum_{i \in I} (u_k^i)^2) \\ \text{s.t.} \quad & \Phi = 0 \end{aligned} \quad (7.9)$$

La commande optimisant (7.9) est donc :

$$u_k^j = -\lambda(1 - b_1^j) \quad u_k^i = -\lambda(1 - b_1^i) \quad (7.10)$$

Où λ vaut :

$$\lambda = \frac{(b_0^j - b_1^j u_0^j + \sum_{i \in I} b_0^i - \sum_{i \in I} (b_1^i u_0^i) + (1 + \sum_{i \in I} \eta_i)K_j \widehat{\mathbf{x}}_k)}{((1 - b_1^j)^2 + \sum_{i \in I} (1 - b_1^i)^2)} \quad (7.11)$$

Les paramètres de fautes n'étant pas connus, leur estimation est utilisée dans le système de commande. Dans le cas du modèle longitudinal, les élevons u_k^2 et u_k^3 sont redondants. En considérant l'ajout des signaux excitants définis à la Section 5.2.1, le contrôleur pour les pannes élevons est :

$$\mathbf{u}_k = \begin{bmatrix} f(-K_1 \widehat{\mathbf{x}}_k, 0, 0) \\ f(-\lambda(1 - \widehat{b}_1^2), (1 - \widehat{b}_1^3)S_{3211}, A_k) \\ f(-\lambda(1 - \widehat{b}_1^3), -(1 - \widehat{b}_1^2)S_{3211}, A_k) \end{bmatrix} \quad (7.12)$$

Avec λ :

$$\lambda = \frac{(\widehat{b}_0^2 - \widehat{b}_1^2 u_0^2 - \widehat{b}_1^3 u_0^3 + 2K_2 \widehat{\mathbf{x}}_k)}{((1 - \widehat{b}_1^2)^2 + (1 - \widehat{b}_1^3)^2)} \quad (7.13)$$

Il est important de noter que, vu la forme de l'équation 7.13, la commande \mathbf{u}_k comprend non seulement une contre réaction de l'état estimé mais aussi un terme constant.

7.3.2.2 Cas où l'actionneur en panne a un effet qui est une combinaison linéaire avec plusieurs autres actionneurs

Pour le modèle complet, les quatre élevons sont redondants et deux équations permettent d'établir les relations entre ces élevons 2.1. La démonstration du contrôleur est faite pour un cas plus général dont l'hypothèse est la suivante :

$$\delta_a = \sum_{i \in I} a_i u_k^{f^i} \quad \delta_e = \sum_{i \in I} e_i u_k^{f^i} \quad (7.14)$$

Où l'ensemble I correspond aux actionneurs redondants.

D'un côté, avoir le même retour qu'en l'absence de fautes implique :

$$\delta_a = \sum_{i \in I} a_i u_k^{f^i} = - \sum_{i \in I} a_i K_i \hat{\mathbf{x}} \quad (7.15)$$

$$\delta_e = \sum_{i \in I} e_i u_k^{f^i} = - \sum_{i \in I} e_i K_i \hat{\mathbf{x}} \quad (7.16)$$

D'un autre côté, la définition de la commande implique :

$$\delta_a = \sum_{i \in I} a_i u_k^i = \sum_{i \in I} a_i (b_0^i + (1 - b_1^i) u_k^i - b_1^i u_0^i) \quad (7.17)$$

$$\delta_e = \sum_{i \in I} e_i u_k^i = \sum_{i \in I} e_i (b_0^i + (1 - b_1^i) u_k^i - b_1^i u_0^i) \quad (7.18)$$

En combinant les équations 7.15 et 7.17, respectivement 7.16 et 7.18, Φ_a , respectivement Φ_e , sont définis comme suit :

$$\Phi_a = \sum_{i \in I} a_i (b_0^i + (1 - b_1^i) u_k^i - b_1^i u_0^i) + \sum_{i \in I} a_i K_i \hat{\mathbf{x}} \quad (7.19)$$

$$\Phi_e = \sum_{i \in I} e_i (b_0^i + (1 - b_1^i) u_k^i - b_1^i u_0^i) + \sum_{i \in I} e_i K_i \hat{\mathbf{x}} \quad (7.20)$$

Le problème d'optimisation à résoudre est :

$$\begin{aligned} \min_{u_k^i} \quad & \frac{1}{2} \left(\sum_{i \in I} (u_k^i)^2 \right) \\ \text{s.t.} \quad & \Phi^e = 0 \\ & \Phi^a = 0 \end{aligned} \quad (7.21)$$

La commande optimisant le problème est donc :

$$u_k^i = -(e_i \lambda + a_i \mu) (1 - b_1^i) \quad (7.22)$$

Où λ et μ valent :

$$\lambda = \frac{C^e \sum_{i \in I} a_i^2 (1 - b_1^i)^2 - C^a \sum_{i \in I} a_i e_i (1 - b_1^i)^2}{\sum_{i \in I} a_i^2 (1 - b_1^i)^2 \sum_{i \in I} e_i^2 (1 - b_1^i)^2 - [\sum_{i \in I} a_i e_i (1 - b_1^i)^2]^2} \quad (7.23)$$

$$\mu = \frac{C^a \sum_{i \in I} e_i^2 (1 - b_1^i)^2 - C^e \sum_{i \in I} a_i e_i (1 - b_1^i)^2}{\sum_{i \in I} a_i^2 (1 - b_1^i)^2 \sum_{i \in I} e_i^2 (1 - b_1^i)^2 - [\sum_{i \in I} a_i e_i (1 - b_1^i)^2]^2} \quad (7.24)$$

$$\text{Avec } C^e \text{ et } C^a : \begin{cases} C^e = \sum_{i \in I} e_i (K_i \hat{\mathbf{x}} + b_0^i - b_1^i u_0^i) \\ C^a = \sum_{i \in I} a_i (K_i \hat{\mathbf{x}} + b_0^i - b_1^i u_0^i) \end{cases}$$

Ici aussi, il est possible de remarquer que, vu la forme de C^e et C^a , la commande présente en plus de la contre réaction de l'état estimé un terme constant.

Dans le cas du modèle complet, les élevons u_k^2 , u_k^3 , u_k^4 et u_k^5 sont redondants. Les coefficients e_i ont la même valeur -0.25 , alors que les coefficients a_i ont des valeurs différentes, respectivement 0.2 , 0.3 , -0.2 et -0.3 (cf. Équation 2.1). En considérant l'ajout des signaux excitants définis à la Section 5.2.1, le contrôleur pour les pannes élevons est :

$$\mathbf{u}_k = \begin{bmatrix} f(-K_1 \hat{\mathbf{x}}_k, 0, 0) \\ f(-(-0.25\lambda + 0.2\mu)(1 - \hat{b}_1^2), 0.6(1 - \hat{b}_1^3)(1 - \hat{b}_1^4)(1 - \hat{b}_1^5)S_{3211}, A_k) \\ f(-(-0.25\lambda + 0.3\mu)(1 - \hat{b}_1^3), -0.4(1 - \hat{b}_1^2)(1 - \hat{b}_1^4)(1 - \hat{b}_1^5)S_{3211}, A_k) \\ f(-(-0.25\lambda - 0.2\mu)(1 - \hat{b}_1^4), -0.6(1 - \hat{b}_1^2)(1 - \hat{b}_1^3)(1 - \hat{b}_1^5)S_{3211}, A_k) \\ f(-(-0.25\lambda - 0.3\mu)(1 - \hat{b}_1^5), 0.4(1 - \hat{b}_1^2)(1 - \hat{b}_1^3)(1 - \hat{b}_1^4)S_{3211}, A_k) \end{bmatrix} \quad (7.25)$$

7.3.3 Conception d'un contrôleur adaptatif pour les fautes d'actionneurs non redondants

Dans le cas où l'actionneur en panne n'est pas redondant, le système n'est pas forcément contrôlable. Si l'actionneur est en perte d'efficacité, le système est toujours contrôlable. Cependant, si l'actionneur est bloqué à une certaine position, le système n'est plus contrôlable. Cependant, il est possible d'améliorer la trajectoire en utilisant les autres actionneurs. Le système est toujours stabilisable (cf. Annexe B) et donc la méthode LQR peut toujours être appliquée.

Supposons la paire (A, B) stabilisable. Supposons u_k^{fj} un actionneur non redondant en panne qui a la forme suivante : $b_0 + (1 - b_1)u_k^j - b_1 u_0^j$ et supposons les autres actionneurs en cas nominal.

Donc l'équation d'état s'écrit sous la forme suivante :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + [B_{.,1} \quad \cdots \quad (1 - b_1)B_{.,j} \quad \cdots \quad B_{.,p}] \mathbf{u}_k + B_{.,j}(b_0 - b_1 u_0^j) + \mathbf{w}_k \quad (7.26)$$

Le terme $B_{.,j}(b_0 - b_1 u_0^j)$ n'est pas considéré dans la théorie du LQR réduit. Dans le théorème suivant, il est montré que ce terme additionnel implique un signal \mathbf{v}_k qui s'ajoute au retour d'état.

Théorème 7.1 (LQR réduit avec un terme additionnel)

Soit le système à contrôler décrit par l'équation linéaire suivante :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + E\mathbf{u}_k + F \quad (7.27)$$

L'indice de performance associé est la fonction quadratique suivante :

$$J_0^\infty = \sum_{k=0}^{k=\infty} \alpha^k (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) \quad (7.28)$$

La commande optimisant l'indice de performance est la suivante :

$$\mathbf{u}_k = -\alpha(R + \alpha E^T P E)^{-1} E^T P A \mathbf{x}_k - \alpha(R + \alpha E^T P E)^{-1} E^T [P F - \mathbf{v}] \quad (7.29)$$

Avec : $P = Q + \alpha A^T [P - \alpha P E (\alpha E^T P E + R)^{-1} E^T P] A$ et $\mathbf{v} = \alpha A^T (Id + \alpha P E R^{-1} E^T)^{-1} (\mathbf{v} - P F)$

La preuve de ce théorème est donnée en Annexe F.

Dans le cas du modèle longitudinal, le contrôleur pour les pannes moteur est le suivant :

$$\mathbf{u}_k = \begin{bmatrix} f(-K_1 \hat{\mathbf{x}}_k + v^1, 0, 0) \\ f(-K_2 \hat{\mathbf{x}}_k + v^2, S_{3211}, A_k) \\ f(-K_2 \hat{\mathbf{x}}_k + v^3, -S_{3211}, A_k) \end{bmatrix} \quad (7.30)$$

Dans le cas du modèle complet, le contrôleur pour les pannes moteur est le suivant :

$$\mathbf{u}_k = \begin{bmatrix} f(-K_1 \hat{\mathbf{x}}_k + v^1, 0, 0) \\ f(-K_2 \hat{\mathbf{x}}_k + v^2, 0.6 S_{3211}, A_k) \\ f(-K_3 \hat{\mathbf{x}}_k + v^3, -0.4 S_{3211}, A_k) \\ f(-K_4 \hat{\mathbf{x}}_k + v^4, -0.6 S_{3211}, A_k) \\ f(-K_5 \hat{\mathbf{x}}_k + v^5, 0.4 S_{3211}, A_k) \end{bmatrix} \quad (7.31)$$

Hors-ligne, le gain K et le terme additionnel \mathbf{v} sont calculés pour différentes valeurs de b_0^1 et b_1^1 . En effet, pour calculer K et \mathbf{v} , l'équation de Riccati doit être résolue et celle-ci dépend des paramètres de fautes b_0^1 and b_1^1 . Pour éviter un temps de calcul en ligne trop long, ces valeurs sont interpolées en ligne suivant l'estimation de b_0^1 and b_1^1 .

7.4 Utilisation de méthodes d'optimisation pour sélectionner le contrôleur

7.4.1 Hypothèses du processus de prise de décision

Les méthodes de prise de décision nécessitent une puissance calculatoire importante, il faut donc limiter ce temps de calcul de façon à pouvoir embarquer l'algorithme sur le drone.

Comme précisé au Chapitre 4, le mécanisme de reconfiguration permettant de sélectionner le contrôleur est échantillonné avec une valeur T_r plus importante que pour l'échantillonnage de l'estimateur ou du contrôleur T_s . Il faut prendre T_r plus grand que le temps nécessaire pour effectuer le code associé au mécanisme de reconfiguration. Néanmoins, il ne faut pas choisir T_r trop grand de façon à ne pas dégrader la stabilité du système. En effet, si une faute intervient, il faut être capable de la compenser le plus rapidement possible et donc de sélectionner rapidement le meilleur contrôleur.

La prise de décision se base sur une prédiction de l'évolution du système sur un horizon H . Un exemple de l'ensemble des solutions possibles pour cette prédiction est montré sur la Figure 7.1.

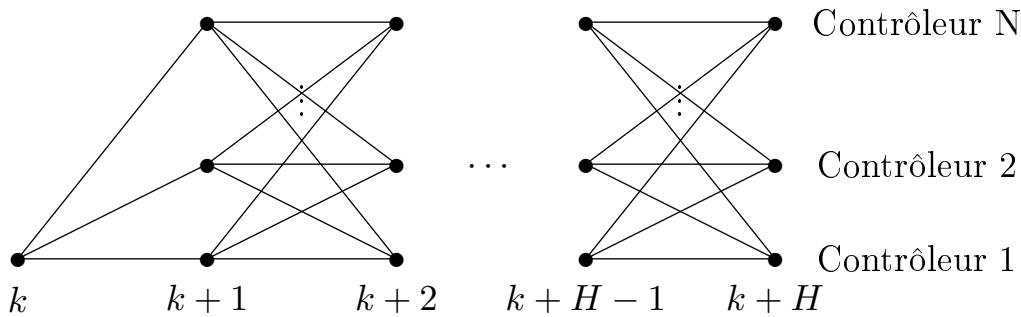


FIGURE 7.1 – Espace de recherche pour N contrôleurs, sachant que le contrôleur courant est le contrôleur 1

À l'instant $k+i$, il est possible d'estimer l'état du système $\mathbf{x}_{k+i|k}$, les paramètres de fautes $\mathbf{b}_{k+i|k}$ ainsi que l'état de croyance $\boldsymbol{\rho}_{k+i}$ grâce au modèle dynamique du système. Le système étant modélisé avec un modèle échantillonné à T_s , la prédiction est réalisée avec l'échantillonnage T_s . Néanmoins la prise de décision est réalisée avec un échantillonnage T_r . Il faut donc que l'indice k soit un multiple de l'indice κ , c'est-à-dire $\kappa = kT_s/T_r$.

L'arbre correspondant à la Figure 7.1 possède N^H feuilles et un raisonnement de type Programmation Dynamique où l'état serait uniquement le contrôleur demande $H \times N^2$ calculs. Néanmoins, si l'on considère les modes, l'état ne se réduit pas au contrôler, ce qui peut entraîner une puissance calculatoire très importante. Deux solutions peuvent réduire le cout calculatoire :

- avoir un horizon H faible ou
- réduire l'espace de recherche.

Il est possible de réduire l'espace de recherche comme montré sur la Figure 7.2. On suppose que le contrôleur est changé à l'instant $k+1$ uniquement.

En effet, le vecteur des paramètres de fautes \mathbf{b} étant modélisé comme une marche aléatoire, la valeur de la prédiction à l'instant $k+i$ est la même que celle à l'instant k , $\mathbf{b}_{k+i|k} = \mathbf{b}_{k|k}$. Ainsi sans information sur l'évolution future des paramètres de fautes, il est peu probable qu'il soit intéressant de changer plusieurs fois de contrôleurs. Notons que cela n'empêche pas que, suite à de nouvelles informations, le contrôleur

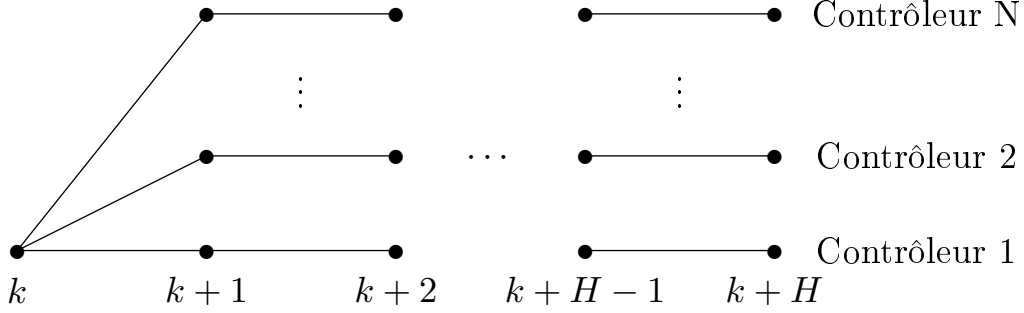


FIGURE 7.2 – Espace de recherche simplifié pour N contrôleurs, sachant que le contrôleur courant est le contrôleur 1

soit changé de nouveau plus tard.

7.4.2 Utilisation du formalisme POMDP

7.4.2.1 Principe

Un POMDP est un processus de décision. La partie discrète étant modélisée avec le formalisme du POMDP, il est intéressant d'utiliser ce processus pour la tâche de prise de décision quant au choix du contrôleur.

Dans le but d'utiliser le cadre du POMDP classique, il est nécessaire de discrétiser l'espace d'observation continue pour retrouver une observation discrète. Soit $\{\Omega^1, \dots, \Omega^{|\Omega|}\}$ une partition de l'espace continu dans lequel le paramètre ω évolue. La probabilité d'avoir le paramètre dans un ensemble Ω^i est donnée par :

$$p(\Omega^i|m) = \int_{\omega \in \Omega^i} p(\omega|m)d\omega \quad (7.32)$$

Le but est de construire une politique qui minimise le coût. L'équation C.3, représentant le problème à résoudre pour un POMDP classique, est adapté à notre problème, où l'on souhaite minimiser le coût et où les actions n'ont aucune influence sur les probabilités de modes. Le problème à résoudre pour un espace de recherche complet (cf. Figure 7.1) devient :

$$V_{k'}(\boldsymbol{\rho}, c) = \min_{a \in A} r_{k'}(\boldsymbol{\rho}, c, a) + \gamma \sum_{i=1}^{|\Omega|} p(\Omega^i|\boldsymbol{\rho}) V_{k'+1}(\boldsymbol{\rho}^{\Omega^i}, a) \quad k' \geq k \quad (7.33)$$

Où $\gamma \in [0, 1[$ est un facteur de réduction, et

$$\begin{cases} r_{k'}(\boldsymbol{\rho}, c, a) = \sum_m [R_{k'}(m, a) + P(c, a)] \rho(m) \\ p(\Omega^i|\boldsymbol{\rho}) = \sum_{m'} p(\Omega^i|m') \sum_m p(m'|m) \rho(m) \\ \rho^{\Omega^i}(m') = \frac{p(\Omega^i|m') \sum_m p(m'|m) \rho(m)}{p(\Omega^i|\boldsymbol{\rho})} \end{cases}$$

La fonction de coût prend en compte deux termes ; un coût $R_{k'}(m, a)$ associé au fait d'être dans un mode m avec un contrôleur donné par l'action a et une pénalité

$P(c, a)$ due aux changements de contrôleurs, c'est-à-dire de passer du contrôleur c de l'état s au contrôleur sélectionné par l'action a . Si l'action a sélectionne le contrôleur courant, la pénalité est nulle.

L'équation 7.33 est initialisé à $V_{k+H}(\boldsymbol{\rho}, c) = \min_{a \in A} r_{k+H}(\boldsymbol{\rho}, c, a)$, ce qui correspond au coût immédiat. Le problème est ensuite résolu par programmation dynamique et nécessite un coût calculatoire important, ce qui implique un horizon H assez faible.

Comme expliqué à la Section 7.4.1, l'espace de recherche est simplifié selon la Figure 7.2. Avec cette espace de recherche, l'action a choisie est la même sur l'horizon H . Il est donc possible de construire une fonction de valeur pour chaque action a . Notons V^a la fonction de valeur associée à l'action a :

$$V_{k'}^a(\boldsymbol{\rho}) = r_{k'}(\boldsymbol{\rho}, c, a) + \gamma \sum_{i=1}^{|\Omega|} p(\Omega^i | \boldsymbol{\rho}) V_{k'+1}^a(\boldsymbol{\rho}^{\Omega^i}) \quad k' \geq k \quad (7.34)$$

La fonction de coût prend en compte le terme de pénalité uniquement à l'instant k , c'est-à-dire au seul instant où il est possible de changer de contrôleur. Pour les instants $k' > k$, la pénalité $P(c, a)$ est nulle car $c = a$. L'équation récursive 7.34 est initialisée par $V_{k+H}^a = r_{k+H}(\boldsymbol{\rho}, a, a)$. Le problème à résoudre devient plus simple ; l'action a minimisant le coût est celle donnant la plus petite valeur pour la fonction de valeur $V_k^a(\boldsymbol{\rho})$:

$$\min_{a \in A} V_k^a(\boldsymbol{\rho}) \quad (7.35)$$

Le calcul récursif décrit sur l'équation 7.34 a la propriété de générer des fonctions convexes et linéaires pour chaque horizon. Chaque fonction est ainsi déterminée par un hyperplan. La fonction de valeur peut alors s'écrire sous la forme $V_{k'}^a(\boldsymbol{\rho}) = \sum_m \rho(m) \alpha_{k'}^a(m)$. Où α_k^a est l'hyperplan associé à l'action a à l'horizon k . La fonction de valeur peut alors s'écrire sous la forme :

$$V_{k'}^a(\boldsymbol{\rho}) = r_{k'}(\boldsymbol{\rho}, c, a) + \gamma \sum_{i=1}^{|\Omega|} p(\Omega^i | \boldsymbol{\rho}) \sum_m \rho^{\Omega^i}(m) \alpha_{k'+1}^a(m) \quad k' \geq k \quad (7.36)$$

Par définition, $\rho^{\Omega^i}(m) = \frac{p(\Omega^i | m) p(m | \boldsymbol{\rho})}{p(\Omega^i | \boldsymbol{\rho})}$ et $\sum_{i=1}^{|\Omega|} p(\Omega^i | m) = 1$, la fonction de valeur vaut :

$$V_{k'}^a(\boldsymbol{\rho}) = r_{k'}(\boldsymbol{\rho}, c, a) + \gamma \sum_{i=1}^{|\Omega|} p(\Omega^i | \boldsymbol{\rho}) \frac{\sum_m p(\Omega^i | m) p(m | \boldsymbol{\rho}) \alpha_{k'+1}^a(m)}{p(\Omega^i | \boldsymbol{\rho})} \quad (7.37)$$

$$= r_{k'}(\boldsymbol{\rho}, c, a) + \gamma \sum_{i=1}^{|\Omega|} \sum_m p(\Omega^i | m) p(m | \boldsymbol{\rho}) \alpha_{k'+1}^a(m) \quad (7.38)$$

$$= r_{k'}(\boldsymbol{\rho}, c, a) + \gamma \sum_m p(m | \boldsymbol{\rho}) \alpha_{k'+1}^a(m) \quad (7.39)$$

7.4.2.2 Partitionnement de l'espace

L'espace continu doit être partitionné si l'on souhaite faire une recherche sur l'espace complet et utiliser l'équation 7.33. Deux choix sont possibles :

- faire une discrétisation *a priori* de l'espace d'observation, ou
- partitionner l'espace continu de façon à regrouper les observations entraînant la même prise de décision.

Une action est nécessaire lorsqu'une panne actionneur apparaît. Lorsqu'une panne capteur apparaît, on suppose que le filtre de Kalman est suffisamment robuste pour gérer la panne, aucune action n'est donc nécessaire. Les observations qui nous intéressent sont donc les paramètres de fautes associés aux actionneurs, b_0^i et b_1^i . On veut alors partitionner en trois zones d'observations :

- une zone pour laquelle l'action à effectuer est a_1 , c'est-à-dire de choisir le contrôleur c_1 , le contrôleur nominal ;
- une zone pour laquelle l'action à effectuer est a_2 , c'est-à-dire de choisir le contrôleur c_2 , le contrôleur pour les pannes moteur et ;
- une zone pour laquelle l'action à effectuer est a_3 , c'est-à-dire de choisir le contrôleur c_3 , le contrôleur pour les pannes élevons.

7.4.2.3 Construction de récompenses hors-ligne

Contrairement au POMDP classique, la fonction de coût n'est pas stationnaire. A chaque instant k , la fonction de coût prend en compte deux termes, $R_k(m, a)$ et $P(c, a)$. La pénalité de changement de contrôleur $P(c, a)$ est ajouté dans le but d'éviter les changements de contrôleur sans aucune raison sérieuse et pour prendre en compte le fait qu'un changement de contrôleur présente toujours un risque. Le coût immédiat $R_k(m, a)$ est calculé hors ligne grâce à l'indice de performance de la façon suivante :

$$R_k(m, a) = J_k^k \quad (7.40)$$

Pour compenser un horizon H faible, un coût "longue" est utilisée à l'instant $k + H$:

$$R_{k+H}(m, a) = J_{k+H}^N \quad (7.41)$$

La pénalité $P(c, a)$ est choisie de façon arbitraire ; $P(c_i, a_i) = 0$ et $P(c_i, a_j) = 0.1$, $\forall i \neq j$

La Figure 7.3 et la Figure 7.4 montre l'évolution des récompenses instantanées $R_k(m, a)$ et des récompenses cumulées $R_{k+H}(m, a)$.

Ces récompenses sont obtenues en faisant une moyenne sur différentes simulations. Dans le cas où on est en mode nominal m_0 , 100 simulations nominal ont été réalisées pour chaque contrôleur. Dans le cas où le moteur est en panne, c'est-à-dire en mode m_1 , 200 simulations -100 simulations de biais et 100 simulations de perte d'efficacité- ont été réalisées pour chaque contrôleur. Dans le cas où l'un des élevons est en panne, c'est-à-dire en mode m_2 , 300 simulations -100 simulations de biais, 100 simulations de perte d'efficacité et 100 simulations de blocage- ont été réalisées pour chaque contrôleur.

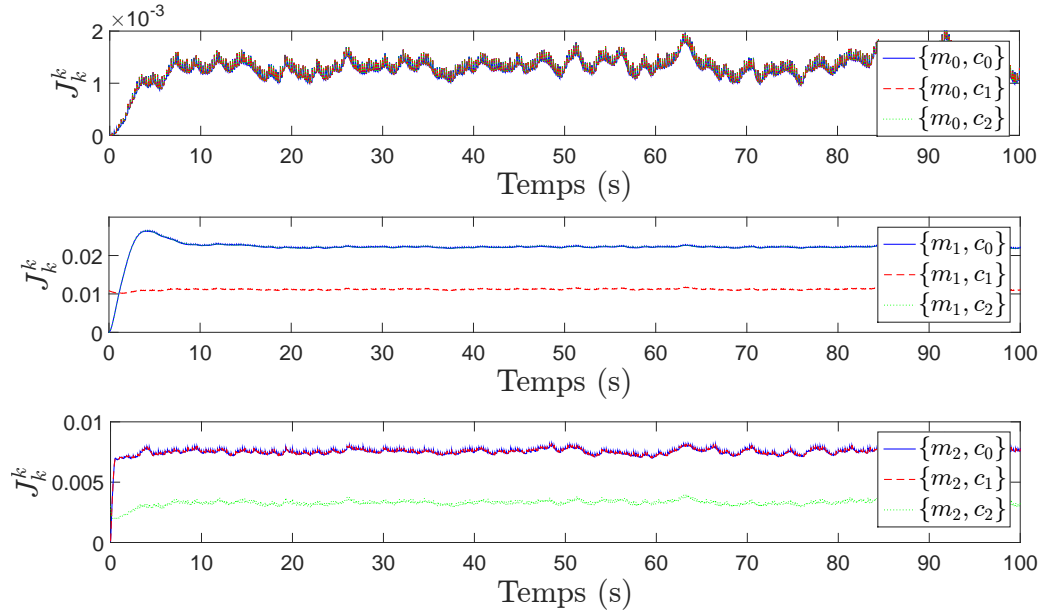


FIGURE 7.3 – Courbes des récompenses instantanées en fonction du mode et du contrôleur

Remarque 7.1

Pour le calcul des récompenses, les indicateurs de performance en cas de blocage et en cas de pertes d'efficacité supérieures à 70% sur le moteur ne sont pas pris en compte. En effet, pour ces cas, le drone n'est plus contrôlable et les valeurs des indicateurs de performance deviennent importantes. Malgré tout le contrôleur conçu pour les pannes moteur fournit de meilleurs résultats pour ces cas de pannes aussi.

Les récompenses immédiates (cf. Figure 7.3) sont quasiment stationnaires. Le début de la simulation correspond en réalité à une période de transition due à une initialisation de la commande \mathbf{u} à zéro. Il est donc possible de remplacer ces récompenses non stationnaires par des récompenses stationnaires qui correspondent aux valeurs en régime établi. Le Tableau 7.1 récapitule les valeurs stationnaires des récompenses instantanées.

Dans le cas nominal, les trois récompenses sont équivalentes. En effet, avec une bonne estimation des paramètres de fautes, c'est-à-dire proche de zéro en mode nominal, les trois contrôleurs sont équivalents. Néanmoins, il est préférable d'avoir le contrôleur nominal en mode nominal. En effet, celui-ci ne dépend pas de l'estimation des paramètres de fautes, qui peut être source d'erreurs.

7.4.2.4 Limite de la méthode

Un exemple pour un problème à deux dimensions est exposé en Annexe G. Cette annexe montre comment calculer les α -vecteurs avec l'espace de recherche

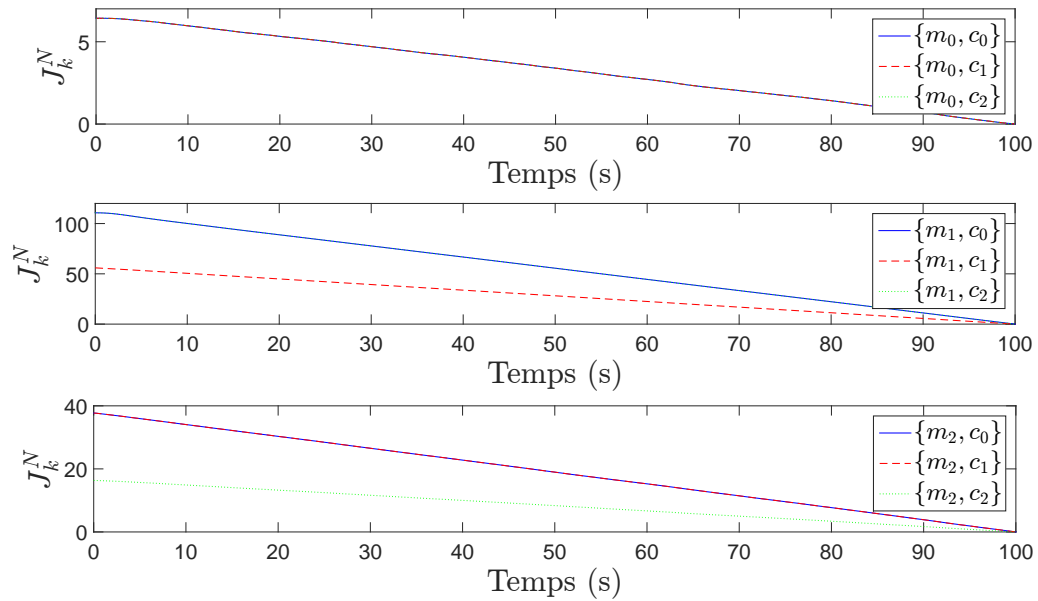


FIGURE 7.4 – Courbes des récompenses cumulées en fonction du mode et du contrôleur

Mode/Contrôleur	$\{m_0, c_0\}$	$\{m_0, c_1\}$	$\{m_0, c_2\}$
$R_k(m, a)$	9.2×10^{-4}	9.2×10^{-4}	9.2×10^{-4}
Mode/Contrôleur	$\{m_1, c_0\}$	$\{m_1, c_1\}$	$\{m_1, c_2\}$
$R_k(m, a)$	2.1×10^{-2}	1.1×10^{-2}	2.1×10^{-2}
Mode/Contrôleur	$\{m_2, c_0\}$	$\{m_2, c_1\}$	$\{m_2, c_2\}$
$R_k(m, a)$	7.2×10^{-3}	7.2×10^{-3}	2.9×10^{-3}

Tableau 7.1 – Valeurs stationnaires des récompenses instantanées

simplifié et l'espace de recherche complet. L'utilisation du processus POMDP avec l'espace de recherche complet (cf. Figure 7.1) présente un désavantage important : ce processus nécessite une puissance calculatoire importante par conséquent l'horizon d'optimisation est assez faible. Deux processus sont coûteux :

- le calcul des probabilités $p(\Omega^i|m)$ et
- l'optimisation de l'équation 7.33.

L'utilisation du processus POMDP avec l'espace de recherche simplifié (cf. Figure 7.2) est beaucoup moins coûteux. Premièrement, le calcul des probabilités $p(\Omega^i|m)$ n'est pas nécessaire et l'optimisation de l'équation 7.35 est moins coûteuse.

Néanmoins cette méthode présente certaines limites, notamment au niveau des récompenses. Les récompenses sont basées sur des calculs hors-ligne, en faisant une récompense moyenne sur l'ensemble des situations envisagées. De plus, la pénalité de changement de contrôleur est arbitraire et ne représente pas parfaitement la réalité.

Une solution permettant de pallier ces problèmes est d'utiliser le modèle dynamique du système en ligne pour prédire le comportement futur du drone.

7.4.3 Utilisation du formalisme MPC

7.4.3.1 Principe

Le formalisme MPC permet de prédire le comportement futur du système grâce au modèle dynamique du drone. Ici, le formalisme MPC est utilisé pour prédire l'évolution de l'état en fonction de l'action choisie. De façon à limiter le coût calculatoire, l'espace de recherche simplifié montré sur la Figure 7.2 est utilisé.

À chaque instant, $|A|$, c'est-à-dire le nombre d'actions possibles, coûts sont calculés en ligne, un pour chaque action :

$$I_k^{k+H}(a) = \sum_{k'=1}^H \alpha^{k+k'} (\mathbf{x}_{k+k'|k}(a))^T Q_{lqr} \mathbf{x}_{k+k'|k}(a) + \mathbf{u}_{k+k'}(a)^T R_{lqr} \mathbf{u}_{k+k'}(a) \quad (7.42)$$

Où a indique le contrôleur pour lequel le coût est calculé.

Par ailleurs la prédiction est réalisée sur un horizon H . Pour compenser le fait que la prédiction est réalisée sur un horizon fini, un coût final est ajouté. Pour cela, les récompenses longues calculées hors ligne sont utilisées (cf. Section 7.4.2.3). Le coût final est le suivant :

$$R_{k+H}(m, c, a) = J_{k+H}^N + P(c, a) \quad (7.43)$$

Le problème à résoudre est le suivant :

$$\min_{a \in A} \left[\sum_m \rho_{k+H|k+H}(m) R_{k+H}(m, a) + I_k^{k+H}(a) \right] \quad (7.44)$$

L'action a minimisant le problème est choisie.

7.4.3.2 Limite de la méthode

Les paramètres de fautes étant modélisés comme des marches aléatoires, la prédiction de ceux-ci ne change pas : $\mathbf{b}_{k+1} = \mathbf{b}_k$. Cette méthode ne considère donc pas l'évolution future des paramètres de fautes.

De façon similaire, l'estimation de l'état de croyance $\rho_{k+H|k+H}(m)$ ne considère pas l'évolution future des paramètres de fautes.

7.4.4 Utilisation du formalisme SMPC

7.4.4.1 Principe

Pour pallier la limite de la méthode précédente, il peut être intéressant d'utiliser le formalisme stochastique MPC.

Soit $\{\Omega^1, \dots, \Omega^{|\Omega|}\}$ une partition de l'espace continu dans lequel le paramètre ω évolue. La probabilité d'avoir le paramètre dans un ensemble Ω^i est donnée par :

$$p(\Omega^i|m) = \int_{\omega \in \Omega^i} p(\omega|m) d\omega \quad (7.45)$$

Pour le formalisme SMPC, l'équation d'état peut être réécrite de la façon suivante :

$$\mathbf{x}_{k+1}(\Omega_i) = \mathbf{x}_{k+1}^{\Omega_i} = A\mathbf{x}_k + B\mathbf{u}_k + F(\Omega_i)\mathbf{b}_k \quad (7.46)$$

Où $F(\Omega_i)$ est une matrice dont les valeurs dépendent de l'ensemble Ω^i .

Le problème SMPC est formulé à chaque pas de temps de la façon suivante :

$$V(a) = \sum_{j=1}^H \sum_{i=1}^{|\Omega|} \pi_{k+j}^i \alpha^{k+j} (\mathbf{x}_{k+j}^{\Omega_i})^T (a) Q_{lqr} \mathbf{x}_{k+j}^{\Omega_i} (a) + u_{k+j}^T(a) R_{lqr} u_{k+j}(a) \quad (7.47)$$

Où π_{k+j}^i est la probabilité d'avoir les paramètres de fautes dans un ensemble donné Ω_i , $\pi_{k+j}^i = \sum_m p(\Omega^i|m) \rho_{k+j|k+j}(m)$, $u_{k+j}(a)$ est la commande qui est issue du contrôleur sélectionné par l'action a et $\mathbf{x}_{k+j}^{\Omega_i}(a)$ est son état associé.

De façon similaire à la méthode précédente, un coût final est ajouté $R_{k+H}(m, c, a)$. Ainsi, le problème à résoudre est le suivant :

$$\min_{a \in A} [r(\rho_{k+H|k+H}, a) + V(a)] \quad (7.48)$$

Où $r(\rho_{k+H|k+H}, a) = \sum_m R_{k+H}(m, a) \rho_{k+H|k+H}$.

7.4.4.2 Application à l'Altimum 4

Pour le modèle complet de l'Altimum 4, trois ensemble sont choisis permettant de définir la fonction $F(\Omega_i)$ de la façon suivante :

$$\begin{cases} F(\Omega_1) = \mathbb{O} \\ F(\Omega_2) = [B_{.1} \quad \mathbb{O} \quad D_{k.1} \quad \mathbb{O} \quad \mathbb{O}] \\ F(\Omega_3) = [\mathbb{O} \quad B_e \quad B_a \quad \mathbb{O} \quad D_{k.2:5} \quad \mathbb{O}] \end{cases}$$

Le partitionnement de l'espace continu est fait de façon à avoir :

- Ω_1 l'ensemble pour lequel il n'y a pas de faute ;
- Ω_2 l'ensemble pour lequel le moteur est en panne ;
- Ω_3 l'ensemble pour lequel l'un des éleveurs est en panne.

Évaluation de la capacité de la méthode à suivre une trajectoire d'atterrissage en cas de pannes

Sommaire

8.1 Critère d'évaluation	165
8.1.1 Critère d'évaluation quantitative pour la reconfiguration . . .	165
8.1.2 Critère d'évaluation quantitative pour le suivi de trajectoire .	165
8.2 Étude comparative entre la méthode FTC développée et une loi de commande classique	165
8.2.1 Comportement en mode nominal	166
8.2.2 Comportement en cas de pannes actionneurs	167
8.2.3 Comportement en cas de pannes capteurs	171
8.3 Impact d'un système de commande basé sur une méthode linéaire	176
8.3.1 Comportement en cas de pannes élevons	177
8.3.2 Comportement en cas de pannes moteur	177
8.4 Impact d'une mauvaise estimation sur le suivi de trajectoire	178
8.5 Conclusion	182

Résumé

Ce chapitre vise à évaluer la capacité de la méthode à suivre une trajectoire d'atterrissage. Dans un premier temps, les performances obtenues avec cette méthode sont mise en avant. Pour cela, la performance de la méthode est comparée à celle d'une commande classique LQR. Dans un second temps, l'impact d'une commande linéaire pour un système non-linéaire est analysé. Finalement, l'impact d'une mauvaise estimation des paramètres de fautes sur le suivi de trajectoire est analysé.

8.1 Critère d'évaluation

8.1.1 Critère d'évaluation quantitative pour la reconfiguration

L'évaluation quantitative est définie à partir des critères suivants :

1. Le temps de reconfiguration T_R définie par le temps nécessaire à la reconfiguration du système de commande une fois la panne apparue.
2. Le temps de reconfiguration $T_{D \rightarrow R}$ définie par le temps nécessaire à la reconfiguration du système de commande une fois la panne détectée. Ce temps peut être négatif si le système de commande est reconfiguré avant la détection de la panne.
3. Le taux de mauvaise sélection du contrôleur, c'est à dire la sélection d'un contrôleur dégradé alors que le système est nominal ou la sélection du mauvais contrôleur en cas de panne.

De plus, il faut vérifier que le temps de calcul du mécanisme de reconfiguration est bien inférieur à T_r , c'est-à-dire à 1s.

8.1.2 Critère d'évaluation quantitative pour le suivi de trajectoire

L'évaluation quantitative est faite à partir de l'indicateur de performance ayant permis la construction des contrôleurs 7.1. On définit deux indicateurs :

- l'indicateur d'énergie au niveau de la commande : $J_u = \sum_{k=0}^N \alpha^k (\mathbf{u}_k^T R_p \mathbf{u}_k)$
- l'indicateur de suivi de trajectoire : $J_x = \sum_{k=0}^N \alpha^k (\mathbf{x}_k^T Q_p \mathbf{x}_k)$

Dans un premier temps, il faut vérifier que la méthode FTC ne dégrade pas la performance en cas de fonctionnement nominal par rapport à une loi de commande classique. Les contrôleurs de la méthode FTC étant construits à partir de la méthode LQR, la méthode FTC est comparée à une loi de commande LQR classique avec facteur de réduction (cf. Annexe F). Cette loi de commande nécessite l'estimation de l'état : celle-ci est calculée en parallèle par un filtre de Kalman classique.

Dans un second temps, il faut analyser le comportement du drone en cas de panne. Deux points sont à analyser :

- Il faut vérifier que la méthode FTC améliore bien la performance par rapport à une loi de commande classique.
- Il faut vérifier qu'en cas de panne la performance est équivalente à celle obtenue en cas nominal.

8.2 Étude comparative entre la méthode FTC développée et une loi de commande classique

Le but de cette section est de comparer :

- Une loi de commande classique prenant en compte un filtre de Kalman et une commande LQR et

- la méthode FTC développée prenant en compte la méthode FDD développée ainsi la reconfiguration du système de commande.

Les simulations de cette section sont réalisées avec les mesures générées à partir du modèle linéaire. De plus, le filtre AKF séquentiel est utilisé avec le modèle complet prenant en compte les fautes actionneurs et capteurs.

8.2.1 Comportement en mode nominal

La commande tolérante aux fautes doit améliorer les performances en cas de pannes actionneurs et capteurs mais elle ne doit pas dégrader pour autant la performance en cas de fonctionnement nominal. Ainsi, 500 simulations ont été réalisées avec 500 séquences de bruit différentes. La Figure 8.1 montre les valeurs des indicateurs de performances J_x et J_u obtenues avec une commande LQR classique et celles obtenues avec notre méthode FTC.

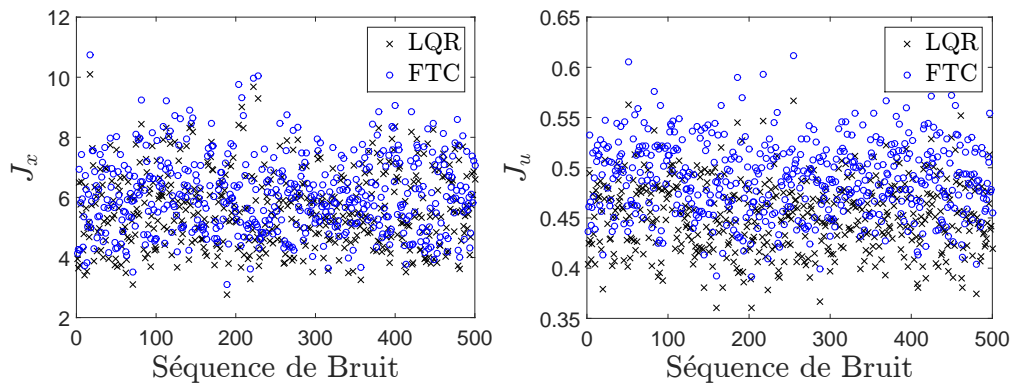


FIGURE 8.1 – Indicateur de performance en cas de simulations sans panne

Les valeurs des indicateurs de performance J_x et J_u sont semblables pour chaque séquence de bruit. Le Tableau 8.1 donne la valeur moyenne des indicateurs de performance J_x et J_u en fonction de la méthode utilisée.

	LQR	FTC
J_x	5.5917	6.0633
J_u	0.4512	0.4889

Tableau 8.1 – Valeurs moyennes sur 500 simulations des indicateurs de performance

La commande FTC dégrade légèrement la performance en cas de fonctionnement nominal. En effet, la commande FTC prend en compte l'état du système augmenté avec les paramètres de fautes ce qui entraîne plus d'incertitudes au niveau de l'estimation de l'état. Par conséquent, la commande u est plus bruitée et donc l'état réel est plus bruité.

Par ailleurs, avec la méthode FTC, le contrôleur nominal est bien conservé pour l'ensemble des simulations, quel que soit l'outil de décision utilisé. En effet, il n'y a pas de *FA* au niveau des actionneurs, par conséquent il n'y a aucune raison de changer le contrôleur.

Remarque 8.1

En fonctionnement nominal, les valeurs des indicateurs de performances J_x et J_u sont les mêmes quel que soit l'outil de décision utilisé (POMDP, MPC ou SMPC) et quel que soit l'horizon utilisé.

8.2.2 Comportement en cas de pannes actionneurs

8.2.2.1 Indicateurs de performances en cas de pannes actionneurs

Lors de cette section, les simulations ont été réalisées en utilisant la méthode de décision SMPC avec un horizon de 5. La Figure 8.2 montre les valeurs des indicateurs de performances en cas de perte d'efficacité du moteur inférieures à 70% et en cas de biais sur le moteur.

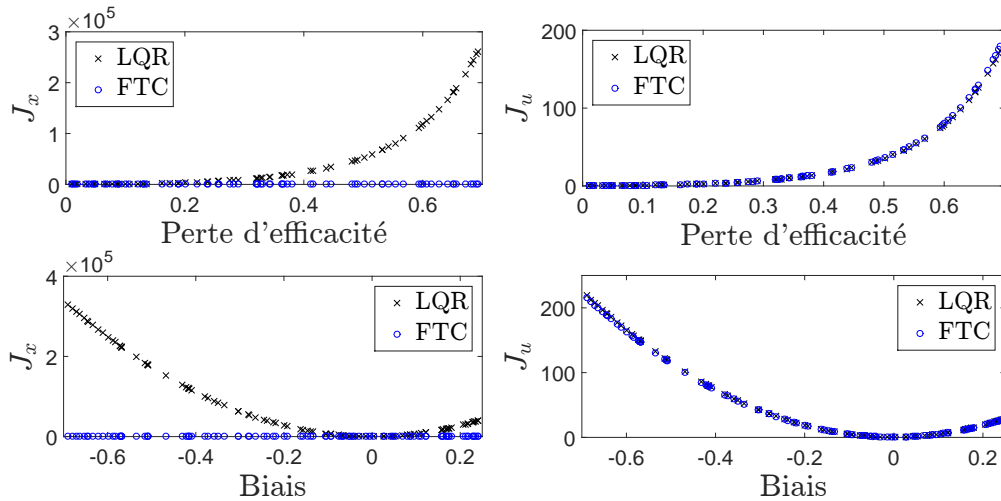


FIGURE 8.2 – Indicateur de performance en cas de simulations avec panne moteur

L'indicateur de performance J_u augmente avec la perte d'efficacité, que ce soit avec la commande LQR ou avec la commande FTC. En effet, il faut dans les deux cas fournir plus d'énergie au niveau des actionneurs pour compenser la panne. Avec la commande LQR, bien que l'actionneur soit sollicité pour compenser la panne, l'indicateur de performance J_x augmente avec la perte d'efficacité ce qui montre que la commande LQR n'arrive pas à compenser entièrement la panne. Au contraire, avec la commande FTC, l'indicateur J_x reste à peu près constant pour toutes pertes d'efficacité inférieures à 70%.

Les résultats obtenus en cas de biais sont similaires. Avec la commande LQR, bien que l'actionneur soit sollicité pour compenser la panne, l'indicateur de perfor-

mance J_x augmente avec l'intensité du biais ce qui montre que la commande LQR n'arrive pas à compenser entièrement la panne. Au contraire, avec la commande FTC, l'indicateur J_x est constant pour tous biais compris entre -0.55 et 0.27 .

Pour les pertes d'efficacité supérieures à 70%, le drone n'est plus contrôlable ; la commande du moteur est en saturation par conséquent il n'est plus possible d'améliorer le suivi de trajectoire du drone. Ainsi les valeurs obtenus pour les indicateurs de performance J_x augmentent à partir d'une perte d'efficacité de 70%. En cas de blocage moteur, le drone n'est plus contrôlable. Les valeurs obtenus pour les indicateurs de performance J_x sont très grandes et augmentent lorsque la valeur de blocage s'éloigne de celle d'équilibre. Néanmoins, l'indicateur J_x associé à la méthode FTC reste inférieur à l'indicateur J_x associé au contrôleur LQR.

La Figure 8.3 montre les valeurs des indicateurs de performances en cas de pannes de l'élevon1.

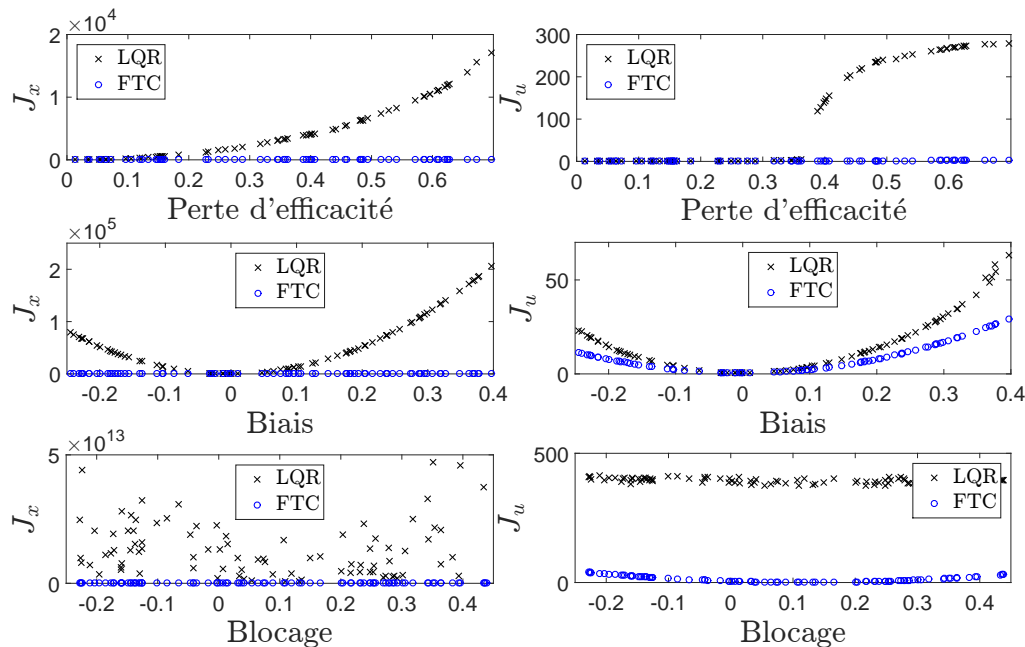


FIGURE 8.3 – Indicateur de performance en cas de simulations avec panne elevon

Quel que soit le type de panne, la FTC améliore considérablement l'indicateur de performance J_x . Pour la FTC, sa valeur est à peu près constante quel que soit la panne. Avec la commande LQR, l'indicateur J_x augmente avec la perte d'efficacité, ainsi qu'avec l'intensité du biais. Quel que soit le blocage, la commande LQR n'arrive pas à gérer la panne. De plus, contrairement à la panne moteur, l'indicateur de performance J_u est lui aussi considérablement réduit. En effet, avec la commande LQR, le contrôleur va chercher à compenser la panne en grande partie avec l'élevon en panne, alors qu'avec la commande FTC, le contrôleur est conçu de façon à minimiser l'énergie de l'ensemble des élevons.

Remarque 8.2

On avait vu à la Section 6.3.1.2 que l'estimateur était moins précis et que la détection était moins rapide pour des pannes faibles. On voit ici que ces erreurs n'ont pas d'impact sur la trajectoire. Le contrôleur nominal est robuste face aux pannes faibles.

8.2.2.2 Comparaison du temps de reconfiguration entre les trois méthodes de décision

Les simulations ont été réalisées avec les trois méthodes de décision et avec différents horizon :

- la méthode POMDP avec un horizon de 5 ;
- la méthode MPC avec un horizon de 5, de 10, de 20, de 50 et de 100 ;
- la méthode SMPC avec un horizon de 5, de 10 et de 20.

Remarque 8.3

Un horizon plus important pour l'outil POMDP ne changera pas le résultat. La récompense instantanée est stationnaire par conséquent, en utilisant l'espace de recherche réduit, cela revient à déplacer les α -vecteurs vers le haut sans changer le point d'intersection.

Comparaison des valeurs de l'indicateur de performance J_x :

Le Tableau 8.2 indique les valeurs moyennes de l'indicateur de performance J_x . La valeur moyenne est calculée pour des cas où l'on peut toujours compenser la panne, c'est-à-dire lorsque le système est toujours contrôlable et que la commande n'est pas en saturation ; pour des pertes d'efficacité moteur inférieures à 70%, pour des biais moteur compris entre -0.55 et 0.27 , pour des pertes d'efficacité élevoon, pour des blocages élevoon compris entre -0.2 et 0.44 et pour des biais élevoon compris entre -0.3 et 0.5 .

J_x	Efficacité Moteur	Biais Moteur	Efficacité Elevon1	Blocage Elevon1	Biais Elevon1
$POMDP_{h=5}$	8.2499	9.0785	7.6913	8.6552	7.9152
$MPC_{h=5}$	8.2461	9.0759	7.6928	8.6615	7.9167
$MPC_{h=10}$	8.2459	9.0695	7.6951	8.6672	7.9166
$MPC_{h=20}$	8.2316	9.0682	7.6986	8.6762	7.9166
$MPC_{h=50}$	8.2154	9.0645	7.7046	8.7186	7.9187
$SMPC_{h=5}$	8.1162	8.9710	7.6874	7.7595	7.3591
$SMPC_{h=10}$	8.1162	8.9710	7.6895	8.0196	7.5088

Tableau 8.2 – Valeur moyenne de l'indicateur de performance J_x

Quel que soit les méthodes ou les horizons, les valeurs sont à peu près semblables. Avec l'outil SMPC et un horizon de 5, on obtient les meilleurs valeurs pour l'indica-

teur de performance. Les valeurs de J_x sont plus élevées que dans le cas nominal. En effet, même si le contrôleur réussit à ramener le drone sur sa trajectoire de référence, il y a une période de transition pendant laquelle le drone va légèrement dévier de sa trajectoire. Cette période de transition peut être divisée en deux parties : la première partie qui correspond au temps de reconfiguration pendant laquelle le drone va s'éloigner de sa trajectoire de référence et la deuxième partie qui correspond au temps nécessaire pour ramener le drone sur sa trajectoire de référence.

Temps de Reconfiguration T_R :

Pour la plupart des simulations, le contrôleur est changé dès que la panne détecté et que le mécanisme de reconfiguration est appelé. Par exemple si la panne est détecté à $10.8s$, le contrôleur est changé à $11s$. Pour quelques simulations, le mécanisme de reconfiguration ne propose pas la même reconfiguration en fonction des méthodes de décision et en fonction des horizons utilisés. D'où les valeurs moyennes différentes obtenues pour J_x dans le Tableau 8.2. Les outils basés sur le formalisme MPC et SMPC prennent en compte les valeurs courantes des paramètres de fautes ce qui permet parfois de reconfigurer le système avant même d'avoir détecté la panne. Par ailleurs, l'outil SMPC, avec un horizon de 5 ou de 10, permet de reconfigurer le système en cas de MD de la panne. En cas de perte d'efficacité moteur, il y a 11 MD et pour ces 11 MD le système est reconfiguré. En cas de biais moteur, il y a 1 MD et pour cette MD le système est reconfiguré. C'est pourquoi on obtient la meilleure valeur moyenne pour l'indicateur J_x .

Cas où le mauvais contrôleur est sélectionné :

Comme montré à la Section 6.3.3.2, il y a des MI temporaires des élevons en cas de panne moteur. Le problème est donc de savoir si le contrôleur c_2 associé aux élevons est sélectionné temporairement ou non et si cela a un impact sur les indicateurs de performance. En cas de perte d'efficacité moteur, il y a une MI temporaire. Quel que soit l'outil choisi, le contrôleur c_2 est sélectionné pendant $1s$. Cependant, cette mauvaise sélection n'a pas d'impact important sur la trajectoire du drone. En cas de blocage du moteur, il y a une MI temporaire. Cette MI étant très courte, quel que soit l'outil choisi, le contrôleur c_2 n'est pas sélectionné. En cas de biais sur le moteur, il y a 9 MI temporaires. Là aussi, les MI sont très courtes, ainsi quel que soit l'outil choisi, le contrôleur c_2 n'est pas sélectionné.

“ce qu'il faut retenir”

La FTC permet bien d'obtenir des performances acceptables sur le critère J_x et cela quel que soit la panne jusqu'à la limite de la saturation de l'actionneur. A l'opposé, le critère J_u augmente avec l'intensité de la panne, cela pour obtenir des performances acceptables sur J_x .

8.2.3 Comportement en cas de pannes capteurs

8.2.3.1 Indicateurs de performances en cas de pannes capteurs

Les résultats montrés lors de cette section utilisent l’outil de décision POMDP avec un horizon de 5.

Simulations de perte d’efficacité :

Comme expliqué à la Section 6.3.2.2, il est difficile de détecter les pertes d’efficacité des capteurs ayant une valeur d’équilibre nulle. Deux cas sont donc différenciés :

- les pertes d’efficacité des capteurs p , q , r , a_{sy} , ψ , ϕ et β , ayant une valeur d’équilibre nulle.
- les pertes d’efficacité des capteurs a_{sx} , a_{sz} , θ , α et z_B , ayant une valeur d’équilibre différente de zéro.

Dans le premier cas, les valeurs des indicateurs de performances J_x et J_u obtenues avec le LQR sont semblables à celles obtenues avec la méthode FTC. De plus, les valeurs moyennes de J_x et J_u sont équivalentes à celles obtenues en cas nominal. En effet, une perte d’efficacité sur une valeur d’équilibre nulle ne change pas le comportement du drone. Dans le deuxième cas, les valeurs des indicateurs de performances sont plus élevées avec le LQR. La Figure 8.4 montre la valeur des indicateurs de performances en cas de perte d’efficacité sur a_{sx} .

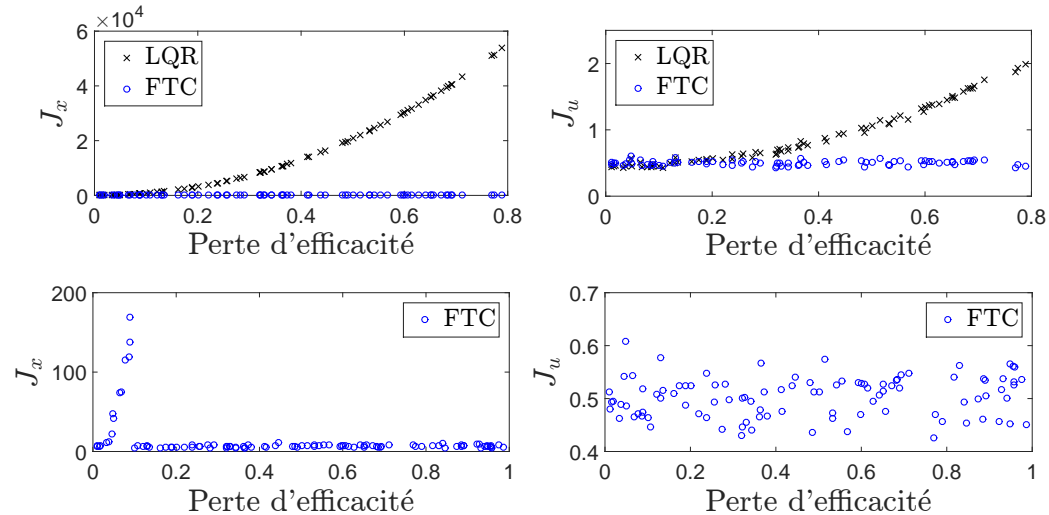


FIGURE 8.4 – Indicateur de performance en cas de simulations avec perte d’efficacité sur a_{sx}

Avec la méthode LQR, les valeurs des indicateurs de performance J_x et J_u augmentent avec la perte d’efficacité. Avec la méthode FTC, la valeur de J_u est constante avec une valeur moyenne de 0.5 -équivalente celle obtenue dans le cas nominal- et la valeur de J_x augmente avec des petites pertes d’efficacité. Cette zone correspond au cas où la covariance du paramètres de fautes associé au capteur a_{sx} n’est pas réouverte. Comme la covariance n’est pas réouverte, la panne est compensée par le

paramètre de faute et par l'état x , d'où une valeur plus grande pour J_x . Pour les cas où la covariance est réouverte, la valeur moyenne de J_x est équivalente à celle obtenue dans le cas nominal.

La Figure 8.5 montre la valeur des indicateurs de performances en cas de perte d'efficacité sur z_B .

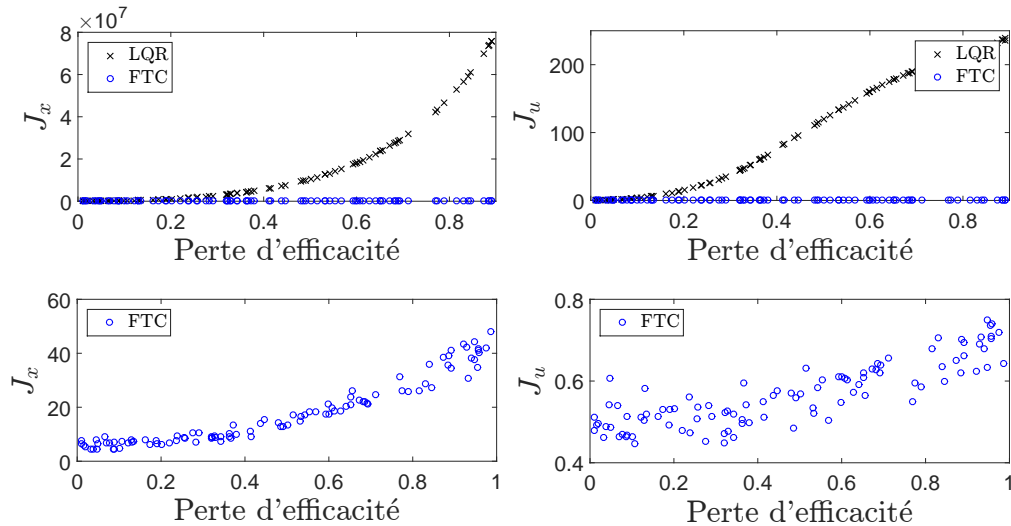


FIGURE 8.5 – Indicateur de performance en cas de simulations avec perte d'efficacité sur z_B

Avec la méthode LQR et la méthode FTC, les valeurs des indicateurs de performance J_x et J_u augmentent avec la perte d'efficacité. Cependant, avec la méthode FTC, J_x et J_u augmentent beaucoup moins fortement.

Pour les autres capteurs α , a_{sz} et θ , avec le LQR les valeurs des indicateurs de performance J_x et J_u augmentent avec la perte d'efficacité, alors qu'avec la méthode FTC développée, les valeurs moyennes des indicateurs de performance J_x et J_u sont équivalentes à celles obtenues dans le cas nominal.

Simulations de biais :

Les biais sur les capteurs donnent des résultats similaires quel que soit le capteur en panne. La Figure 8.6 montre la valeur des indicateurs de performances en cas de biais sur ϕ .

De façon générale, c'est-à-dire quel que soit le capteur, avec le LQR les valeurs des indicateurs de performance J_x et J_u augmentent avec l'amplitude du biais, alors qu'avec la méthode FTC développée, les valeurs restent à peu près constantes. Les valeurs moyennes des indicateurs de performance J_x et J_u sont légèrement supérieures à celles obtenues en mode nominal. Le Tableau 8.3 répertorie les valeurs moyennes de J_x et J_u pour les différents capteurs.

Simulations de blocage :

Les blocages sur les capteurs donnent des résultats similaires quel que soit le capteur en panne. La Figure 8.7 montre la valeur des indicateurs de performances

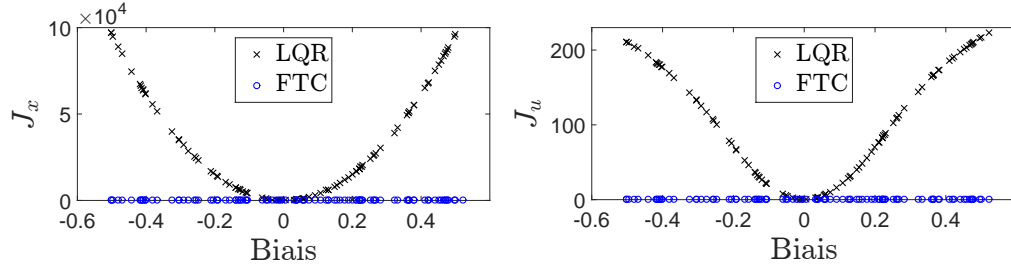


FIGURE 8.6 – Indicateur de performance en cas de simulations avec biais sur ϕ

	p	q	r	a_{sx}	a_{sy}	a_{sz}	ψ	ϕ	θ	α	β	z_B
J_x	5.9	5.9	6.0	6.2	5.9	6.0	6.2	6.0	5.9	5.9	5.9	6.2
J_u	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Tableau 8.3 – Valeur moyenne des indicateurs de performance sur 100 simulations de biais

en cas de blocage sur le baromètre z_B .

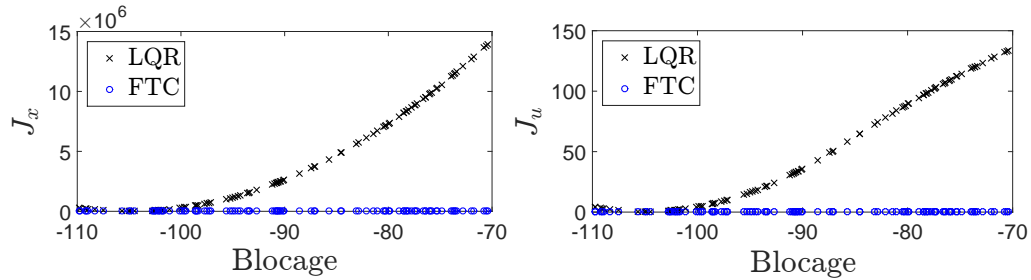


FIGURE 8.7 – Indicateur de performance en cas de simulations avec blocage sur z_B

De façon générale, c'est-à-dire quel que soit le capteur, avec le LQR les valeurs des indicateurs de performance J_x et J_u augmentent lorsque la valeur de blocage s'éloigne de la valeur d'équilibre, alors qu'avec la méthode FTC développée, les valeurs des indicateurs de performance J_x et J_u restent constantes. Pour la plupart des capteurs, les valeurs moyennes de J_x et de J_u sont équivalentes à celles obtenues dans le cas nominal. La valeur moyenne de J_x est légèrement plus élevée pour les capteurs a_{sx} , ψ , θ et z_B . Le Tableau 8.4 répertorie les valeurs moyennes de J_x et J_u pour ces capteurs.

En cas de blocage sur z_B , l'indicateur de performance J_x augmente légèrement pour des blocages proches de la valeur d'équilibre. De plus, il y a une simulation pour laquelle le drone ne suit pas tout à fait la trajectoire et pour laquelle l'indicateur J_x est élevé ($J_x = 666$). Pour ce blocage, la valeur est très proche de celle d'équilibre, l'erreur du capteur est partagé sur l'estimation du paramètre de faute associé au

	p	q	r	a_{sx}	a_{sy}	a_{sz}	ψ	ϕ	θ	α	β	z_B
J_x	5.9	5.9	6.0	6.8	5.9	6.2	7.3	6.2	6.5	5.9	5.9	7.9
J_u	0.5	0.5	0.5	0.5	0.5	0.5	0.4	0.6	0.5	0.5	0.5	0.5

Tableau 8.4 – Valeur moyenne des indicateurs de performance sur 100 simulations de blocage

capteur z_B et l'état z . Par conséquent, il y a une erreur au niveau de la trajectoire. L'erreur finale est de $-0.45m$ ce qui reste acceptable.

Simulations de dérive :

Les simulations de dérives sur les capteurs α , β , q et a_{sy} donnent des résultats similaires. Avec la méthode LQR, les valeurs de J_x et J_u augmentent avec la pente de la dérive. Avec la méthode FTC, les indicateurs de performance J_x et J_u sont constants ; les valeurs moyennes sont proches de celles obtenues dans le cas nominal. Pour les simulations de dérives sur les capteurs p , r , ψ , θ et z_B , les valeurs de l'indicateur de performance J_x sont plus importantes mais toujours à peu près constante en fonction de la dérive. Le Tableau 8.5 répertorie les valeurs moyennes de J_x et J_u pour ces capteurs.

	p	q	r	a_{sy}	ψ	θ	α	β	z_B
J_x	10	6.0	13	5.9	36	23	5.9	5.9	17
J_u	0.6	0.5	0.5	0.5	0.6	0.5	0.5	0.5	0.5

Tableau 8.5 – Valeur moyenne des indicateurs de performance sur 100 simulations de dérive

Remarque 8.4

On a vu à la Section 6.3.3.2, qu'il y avait 78 MI temporaires des élevons en cas de dérives sur p . Sur 68 simulations, le contrôleur c_2 associé aux élevons est sélectionné de façon permanente. Cependant au vu de la valeur moyenne obtenue pour J_x et J_u , on peut voir que cela a peu d'impact sur la trajectoire du drone. En effet, les paramètres de fautes associés aux élevons sont faibles et par conséquent, le contrôleur c_2 est quasiment équivalent au contrôleur nominal.

Les simulations de dérives sur les capteurs a_{sx} et a_{sz} sont semblables. L'erreur est plus importante pour des dérives faibles. La Figure 8.8 montre la valeur des indicateurs de performances en cas de dérive sur le capteur a_{sz} .

Les valeurs de J_x et J_u obtenues avec la méthode LQR augmentent avec la pente de la dérive et sont toujours plus grandes que celles obtenues avec la méthode FTC. Avec la méthode FTC, les valeurs de J_x sont plus importantes pour de faibles dérives. De plus, on a vu à la Section 6.3.3.2, qu'il y avait 6 MI temporaires des élevons en cas de dérive sur a_{sz} . Parmi ces 6 simulations qui correspondent à des dérives très faibles, 5 sélectionnent le contrôleur c_2 et cela de façon permanente après

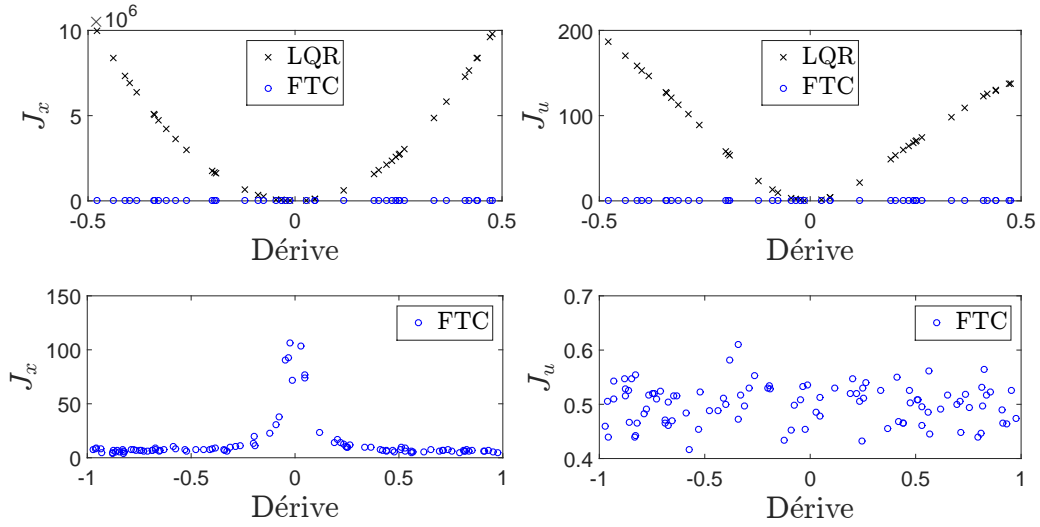


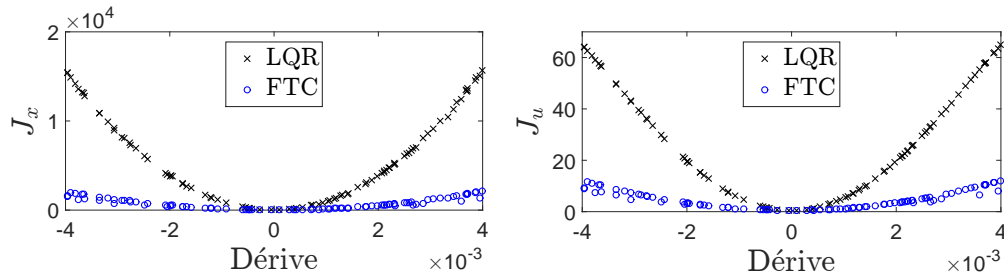
FIGURE 8.8 – Indicateur de performance en cas de simulations avec dérive sur a_{sz}

l'apparition de la panne. Ces simulations ont des valeurs de J_x plus élevées, entre 70 et 103. L'erreur finale sur la trajectoire reste faible, il y a une erreur maximum de $0.15m$ sur x et sur z .

Pour les dérives sur a_{sx} , on a vu à la Section 6.3.3.2, qu'il y avait une MI permanente et 3 MI temporaires du moteur en cas de faibles dérives sur a_{sx} . Pour ces 4 simulations le contrôleur c_1 associé au moteur est sélectionné de façon permanente après l'apparition de la panne. Parmi ces 4 simulations, 2 ont du mal à converger et ont une valeur de J_x élevée ($J_x = 13 \times 10^3$ et $J_x = 48 \times 10^3$). L'erreur finale de la position est d'environ $-2m$ sur x et de $5m$ sur z . Cette erreur n'est pas excessive, cependant pour une simulation plus longue, l'erreur continuerait à augmenter ce qui n'est pas acceptable sur du long terme. Pour les deux autres simulations, l'erreur est beaucoup plus faible $J_x = 75$ et $J_x = 134$, ce qui correspond à une erreur finale de $0.1m$ sur z . Hormis ces 4 simulations, les indicateurs de performance sont bons, la valeur moyenne de J_x est de 8.6 et la valeur moyenne de J_u est de 0.5.

Les dérives sur ϕ donnent des résultats différents. La Figure 8.9 montre la valeur des indicateurs de performances en cas de dérive sur le capteur ϕ .

Les valeurs de J_x et J_u obtenues avec la méthode LQR augmentent avec la pente de la dérive et sont toujours plus grandes que celles obtenues avec la méthode FTC. Avec la méthode FTC, les valeurs de J_x et J_z augmentent elles aussi avec la pente de la dérive mais moins fortement. Pour une dérive de $-0.25^\circ/s$, l'indicateur J_x est de 2500. L'erreur finale est de $0.9m$ sur la position verticale y , ce qui reste acceptable.

FIGURE 8.9 – Indicateur de performance en cas de simulations avec dérive sur ϕ **“ce qu’il faut retenir”**

La FTC donnent bien de meilleures performances qu’avec la méthode LQR quel que soit le capteur et quel que soit le type de panne. Pour la plupart des cas, les critères J_x et J_u sont proches de ceux obtenus dans le cas nominal, ce qui montre bien un recouvrement total malgré la panne. La méthode FTC a néanmoins plus de difficulté avec les cas de dérive, ce qui est cohérent avec les problèmes d’intégrité rencontrés lors de l’estimation. Pour les cas de blocage, de biais et de perte d’efficacité, l’estimateur est suffisamment précis pour ne nécessiter aucune action en cas de pannes capteurs. La méthode pourrait être améliorée pour les cas de dérive.

8.3 Impact d’un système de commande basé sur une méthode linéaire

Les simulations de la section précédente utilisent le modèle linéaire du drone pour générer les mesures et le filtre AKF est utilisé. Ici, les mesures sont générées avec le modèle non linéaire du drone modélisant les élévons avec un unique premier ordre et le filtre EKF est utilisé. On se place donc dans un cadre non linéaire. Cependant, le système de commande est basé sur un modèle linéaire. Le but de cette section est donc de voir l’impact de l’utilisation de contrôleurs linéaires sur la trajectoire du drone en cas de panne.

Le système de commande développé au Chapitre 7 se décompose en deux parties : les contrôleurs et la tâche de décision. Les contrôleurs sont basés sur la méthode LQR et nécessitent par conséquent un modèle linéaire. Ainsi, les matrices A et B utilisées proviennent du modèle linéaire du drone (cf. Annexe A).

Concernant la tâche de décision sur le choix du contrôleur, trois outils ont été développés :

- l’outil basé sur le formalisme POMDP qui nécessite seulement la connaissance des probabilités de modes ;
- l’outil basé sur le formalisme MPC qui nécessite la connaissance de l’équation d’état et qui peut par conséquent être adapté à un modèle non linéaire ;

8.3. Impact d'un système de commande basé sur une méthode linéaire 77

- l'outil basé sur le formalisme SMPC qui nécessite la connaissance de l'équation d'état et qui peut lui aussi être adapté à un modèle non linéaire.

Les simulations montrées lors de cette section utilisent l'outil basé sur le formalisme POMDP. Dans le cas nominal, les valeurs moyennes des indicateurs de performances J_x et J_u sont de respectivement 6.0684 et 0.1756.

8.3.1 Comportement en cas de pannes élevons

La Figure 8.10 montre les valeurs des indicateurs de performance en cas de pannes élevons.

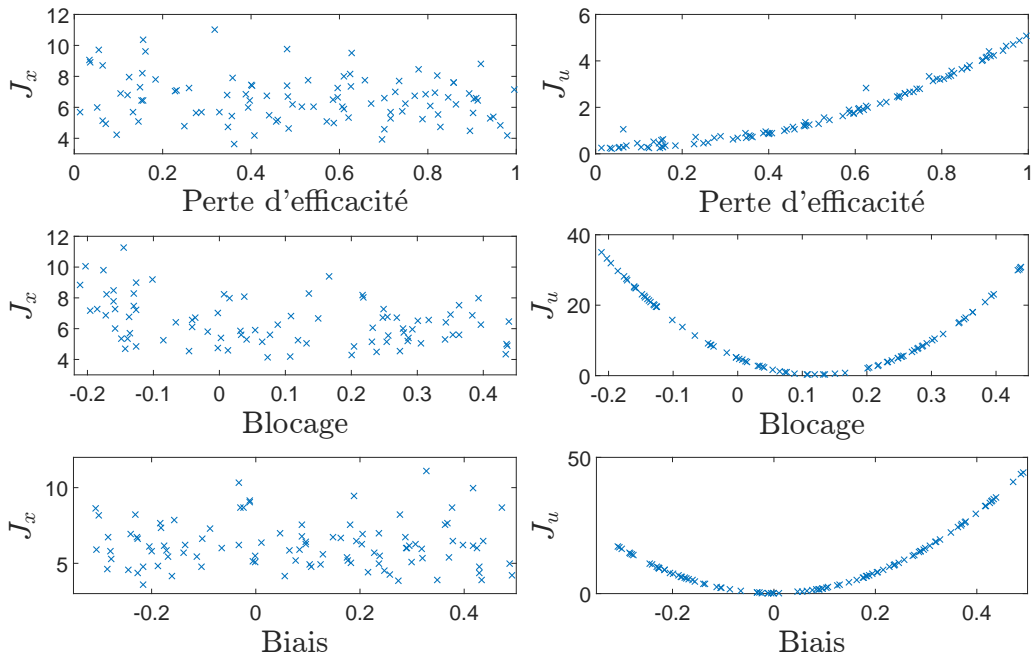


FIGURE 8.10 – Indicateur de performance en cas de panne sur l'élevon 1

La performance J_x est légèrement supérieure au cas nominal. En effet, il faut prendre en compte la période de transition après l'apparition de la panne. En cas de perte d'efficacité, la valeur moyenne de J_x est de 6.5532, en cas de blocage, la valeur est de 6.2075 et en cas de biais, la valeur est de 6.2192. Le contrôleur est donc bien adapté au système non linéaire malgré l'apparition de panne lors de sa trajectoire de descente.

8.3.2 Comportement en cas de pannes moteur

Contrairement au contrôleur développé pour les pannes élevons, le contrôleur développé pour les pannes moteur dégrade la performance en cas de pannes moteur. La Figure 8.11 montre les valeurs des indicateurs de performance en cas de pannes moteur.

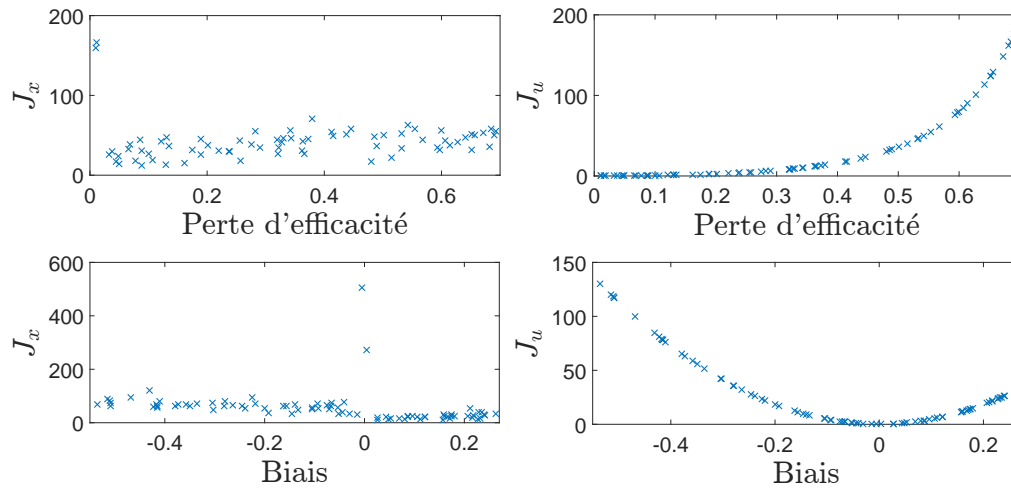


FIGURE 8.11 – Indicateur de performance en cas de panne sur le moteur

D'après la Figure 8.11, l'indicateur de performance J_x est plus élevé que dans le cas nominal. De plus, on remarque que la performance J_x est plus élevée pour des très faibles pertes d'efficacité et pour des biais très faibles. Excepté ces 4 simulations, les valeurs de J_x sont à peu près constantes ; la moyenne en cas de perte d'efficacité est de 51.4622 et la moyenne en cas de biais est de 55.2396. Ces valeurs plus importantes sont dues à une période de transition plus longue. La Figure 8.12 montre l'erreur entre la position et la position de référence en cas de biais moteur de -0.3 .

Sur la Figure 8.12 on peut voir que le drone met 40s avant de rattraper sa trajectoire de référence, d'où une valeur plus élevée pour la performance J_x .

“ce qu'il faut retenir”

Le contrôleur développé pour les pannes moteur réussit plus difficilement à ramener le drone sur sa trajectoire.

8.4 Impact d'une mauvaise estimation sur le suivi de trajectoire

A la Section 6.4.1.1, on a vu qu'une mauvaise modélisation de la dynamique des élevons entraînait des erreurs d'estimation. Le but de cette section est de savoir si le drone arrive à suivre sa trajectoire malgré l'apparition d'une panne élevon et d'une mauvaise estimation des paramètres de faute élevons.

Lors de cette section, les mesures sont générées à partir du modèle non linéaire modélisant les élevons avec le modèle complexe. Le filtre utilisé est le même qu'à la Section 6.4.1.1, c'est-à-dire le filtre AKF séquentiel.

Cas de perte d'efficacité :

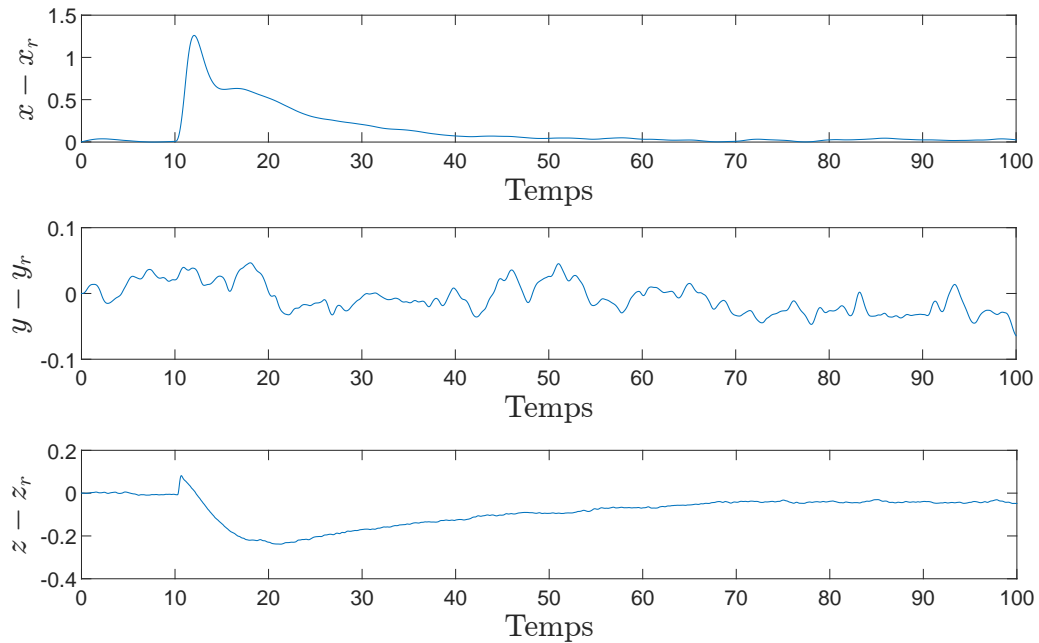


FIGURE 8.12 – Erreur sur la trajectoire en cas de biais de -0.3 sur le moteur

La Figure 8.13 montre les erreurs d'estimation des paramètres de faute associés aux élevons en cas de perte d'efficacité sur l'élevon 1.

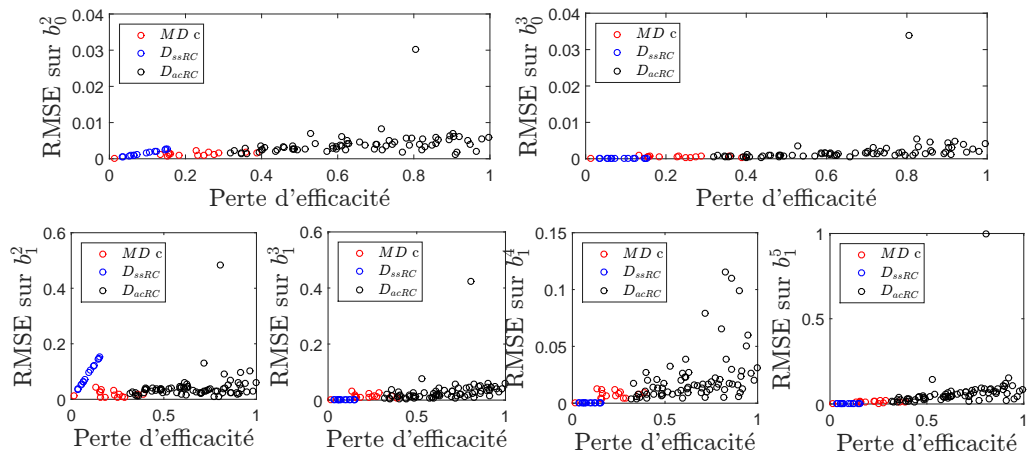


FIGURE 8.13 – Erreur sur l'estimation des paramètres de faute élevons en cas de perte d'efficacité sur l'élevon 1

La Figure 8.14 montre la valeur des indicateurs de performance en fonction de la perte d'efficacité sur l'élevon 1.

Bien que les erreurs soient importantes, l'indicateur de performance J_x est ac-

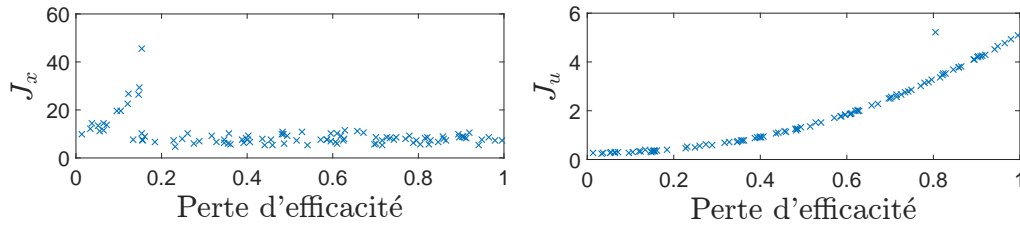


FIGURE 8.14 – Indicateur de performance en cas de perte d'efficacité sur l'élevon 1

ceptable. Pour des pertes d'efficacité supérieures à 20%, l'indicateur est à peu près constant et a une valeur moyenne de 7.8423 ce qui est équivalent au cas nominal. On remarque notamment une simulation sur la Figure 8.13, pour une perte d'efficacité de 80.46%, pour laquelle l'estimation est très éloignée de la réalité. Pour cette simulation, l'indicateur de performance vaut 5.666.

Cas de blocage :

La Figure 8.15 montre les erreurs d'estimation des paramètres de faute associés aux élevons en cas de blocage sur l'élevon 1.

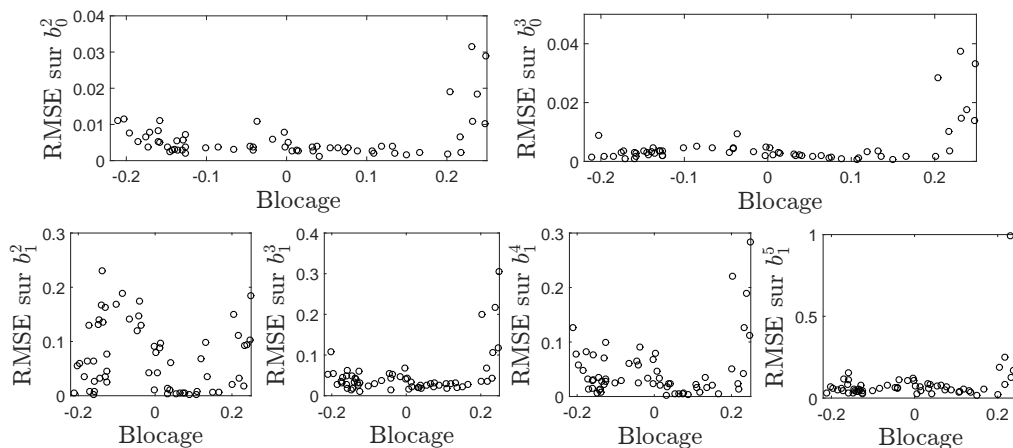


FIGURE 8.15 – Erreur sur l'estimation des paramètres de faute élevons en cas de blocage sur l'élevon 1

La Figure 8.16 montre la valeur des indicateurs de performance en fonction de la valeur du blocage sur l'élevon 1.

Sur la Figure 8.16, on remarque 3 simulations qui ont une valeur de J_x beaucoup plus élevée que pour les autres simulations. Pour ces simulations la période de transition est assez longue. La Figure 8.17 montre l'erreur sur la trajectoire pour la simulation ayant un indicateur de performance de 781.

Lorsque la panne apparaît à 10s, le drone s'écarte de sa trajectoire. La transition est de 40s avant que le drone arrive à rattraper sa trajectoire de référence, d'où une valeur importante pour l'indicateur J_x . Pour les autres simulations, les valeurs de

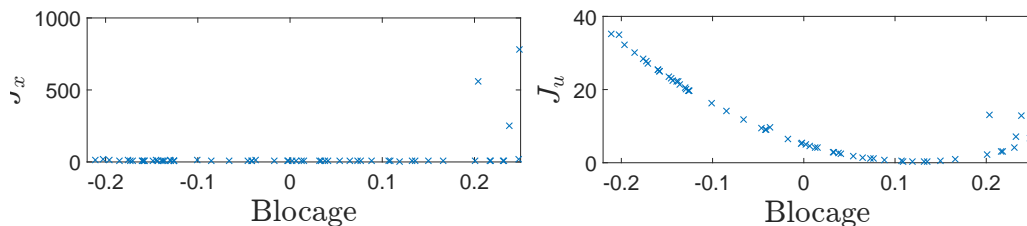


FIGURE 8.16 – Indicateur de performance en cas de blocage sur l'élevon 1

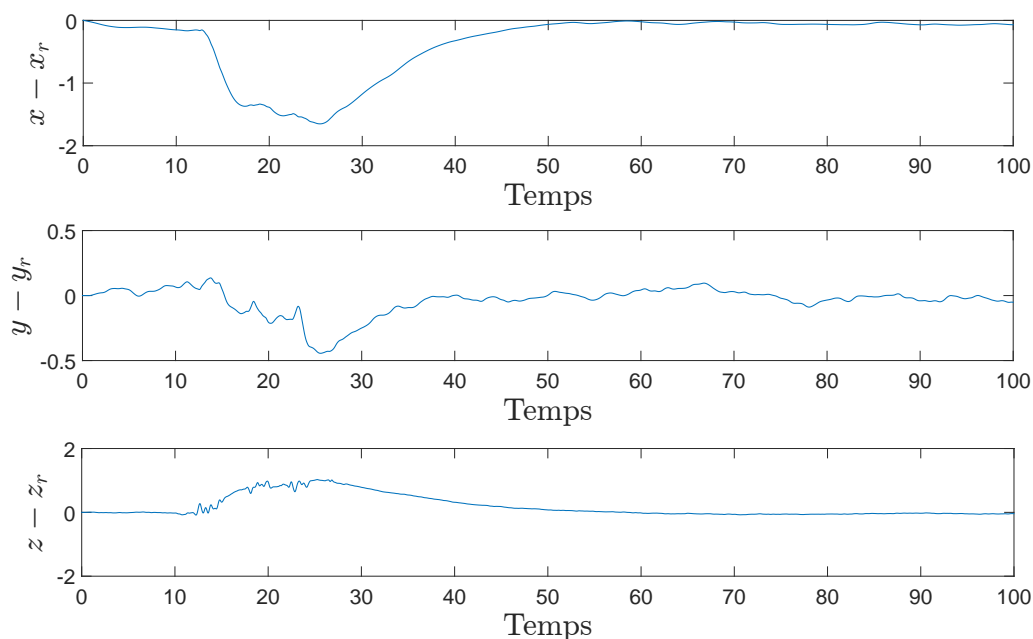


FIGURE 8.17 – Erreur sur la trajectoire en cas de blocage de l'élevon 1 à 0.2485

J_x sont proches de celles du cas nominal.

Cas de biais :

La Figure 8.18 montre les erreurs d'estimation des paramètres de faute associés aux élevons en cas de biais sur l'élevon 1.

La Figure 8.19 montre la valeur des indicateurs de performance en fonction du biais sur l'élevon 1.

De façon similaire au blocage, on remarque 5 simulations pour lesquelles l'indicateur de performance J_x est plus élevé. Pour ces simulations, le temps de transition est assez long, d'environ 40s avant que le drone n'arrive à rattraper sa trajectoire. Pour les autres simulations, les valeurs de J_x sont proches de celles du cas nominal.

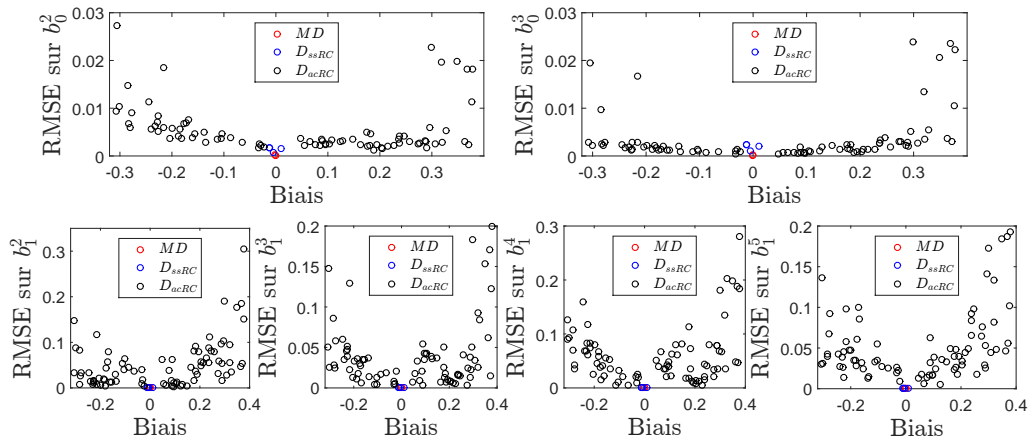


FIGURE 8.18 – Erreur sur l’estimation des paramètres de faute élevons en cas de biais sur l’élevon 1

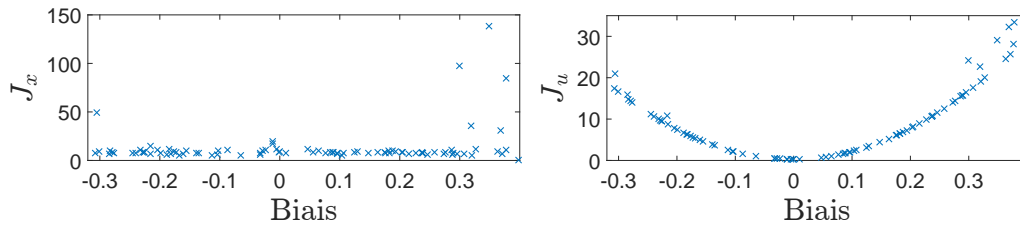


FIGURE 8.19 – Indicateur de performance en cas de biais sur l’élevon 1

“ce qu’il faut retenir”

Bien que l’estimation ne soit pas très précise, on obtient des bonnes valeurs pour le critère J_x . Ce phénomène est dû à la redondance des élevons. Les paramètres de fautes des élevons sont partiellement identifiables, par conséquent il est possible que les paramètres de faute convergent vers une valeur différente de celle attendue. L’effet dynamique sur le drone est correctement estimé, ce qui permet d’avoir un contrôleur qui offre de bonnes performances, même si l’imputation des erreurs aux actionneurs peut être erronée. En effet, le contrôleur associé aux pannes élevons prend en considération la redondance et la combinaison linéaire existant entre les élevons.

8.5 Conclusion

Ce chapitre permet de mettre en avant les avantages de la méthode FTC proposée. Dans un premier temps, l’avantage d’une méthode FTC sur une méthode LQR est mis en avant. Contrairement à une commande classique LQR, la méthode per-

met d'obtenir, quel que soit le type de panne, des performances acceptables sur le critère J_x . Pour les pannes capteurs, le critère J_u reste lui aussi acceptable. Pour les pannes actionneurs, le critère J_u augmente avec l'intensité de la panne, mais cette détérioration du critère J_u est nécessaire pour obtenir des performances acceptables sur le critère J_x . La méthode est cependant peu efficace pour certaines pannes actionneurs qui sont très importantes et qui sont difficile à améliorer à cause de la saturation en commande.

Dans un second temps, la méthode a été testée face à des incertitudes de linéarisation. En effet, les contrôleurs ont été construits à partir d'un modèle linéaire de l'Altium 4, alors que le système est non linéaire. Le contrôleur nominal et le contrôleur conçu pour les pannes élevons ne dégradent pas la performance du critère J_x . Au contraire, le contrôleur conçu pour les pannes moteur est légèrement impacté par les effets de linéarisation ; celui-ci est plus lent à rattraper la trajectoire du drone après l'apparition d'une faute.

Finalement, la méthode a été testée face à des incertitudes d'estimation au niveau des paramètres de fautes associés aux élevons. Malgré une mauvaise estimation, le critère J_x fournit de bonnes performances. En réalité, l'estimation n'est pas la bonne mais l'effet dynamique du drone est correctement estimée. Le contrôleur conçu pour les élevons prend en compte la redondance des actionneurs, ce qui permet d'obtenir de bonnes performances même si l'imputation des erreurs aux actionneurs peut être erronée.

La méthode POMDP a été appliqué dans deux articles ayant fait l'objet de publications [Boche 2018], [Boche 2017a] et la méthode MPC a été appliqué dans un article ayant fait l'objet d'une publication [Boche 2017b].

Conclusion

Sommaire

9.1	Conclusions	185
9.2	Perspectives	186
9.2.1	Amélioration du modèle	186
9.2.2	Amélioration de la méthode	186

9.1 Conclusions

Dans ce mémoire, une méthode générique de commande tolérante aux fautes active et indirecte est développée. Celle-ci se base sur des cadres de modélisation discret et continu. La modélisation continue proposée permet non seulement d'estimer l'état et les paramètres de fautes mais aussi de calculer les contrôleurs. L'objectif de la modélisation discrète est de prendre une décision en ligne quant au contrôleur à utiliser. L'estimation continu permet d'avoir plus d'informations sur la faute qu'avec une modélisation discrète, alors que celle-ci permet de prendre en compte des probabilités de panne et de mettre en œuvre des techniques d'optimisation qui sont plus adaptées à la tâche de décision. La méthode développée nécessite un module de détection et de diagnostic actif. Ce module doit prendre en compte un estimateur de l'état et des paramètres de faute, ainsi qu'un estimateur des probabilités de modes pour détecter et identifier la panne. L'estimateur des probabilités de modes nécessite l'estimation des moyennes des paramètres de fautes, ainsi que de leur matrice de covariance. Par conséquent, l'estimateur de l'état et des paramètres de fautes doit être adapté.

L'estimation de l'état et des paramètres de fautes se base sur un filtre de Kalman. Celui-ci a été adapté pour une prise en compte des changements abrupts que peuvent subir les paramètres de fautes lorsque les pannes apparaissent. Les filtres AKF, TSKF et EKF ont été utilisés. La précision de l'estimateur ainsi que la capacité de détection et d'identification du module FDD ont été évalués. L'évaluation en simulation montre que, lorsque l'écart entre le modèle utilisé par la FTC et la réalité est faible, la méthode proposée donne de très bons résultats. Le fait d'utiliser un modèle linéaire pour un processus non linéaire ne dégrade pas ces résultats. Par contre une dynamique des actionneurs trop grossièrement modélisée, retards et saturations négligés, conduit à des dégradations importantes.

Le système de reconfiguration développé utilise plusieurs contrôleurs basés sur la méthode LQR, un pour le fonctionnement normal et les cas de pannes capteurs, un pour les pannes moteur et un pour les pannes élevons. De plus, les contrôleurs développés pour les modes dégradés sont adaptatifs en fonction de l'estimation des paramètres de fautes. Une optimisation en ligne permet alors de choisir le contrôleur le plus adapté.

L'évaluation en simulation de l'aptitude de la méthode à suivre une trajectoire de descente malgré l'apparition d'une panne sur un drone à voilure fixe, indique que lorsqu'il y a peu d'incertitudes au niveau du modèle et de l'estimation les fautes sont corrigées efficacement. Finalement, en ce qui concerne les fautes élevons, le fait d'utiliser une dynamique des actionneurs grossièrement modélisée et donc d'avoir une mauvaise estimation des paramètres de fautes n'empêche pas de suivre la trajectoire désirée.

Bien que la méthode ait été développée pour un drone à voilure fixe, celle-ci pourrait être utilisable pour tous autres systèmes complexes, tels qu'un octorotor ou un avion, tant qu'un modèle est disponible pour ce système. La méthode pourrait par ailleurs être plus efficace si des capteurs étaient disponibles au niveau des actionneurs. Une signature de panne unique pourrait alors être disponible pour le moteur, ce qui permettrait de différencier immédiatement les pannes moteur, des pannes capteur de l'accélération spécifique horizontale.

9.2 Perspectives

Avant de proposer des améliorations à la méthode développée, nous allons discuter des étapes restantes avant de pouvoir implémenter la méthode sur le drone.

9.2.1 Amélioration du modèle

Comme discuté au Chapitre 6, le modèle doit être suffisamment proche de la réalité pour avoir une bonne estimation. Le modèle du drone utilisé pour appliquer notre méthode ne constitue pas un modèle dynamique complet. Il serait intéressant de modéliser le retard, ainsi que la saturation en vitesse des élevons pour avoir de meilleurs résultats au niveau de la méthode d'estimation.

De plus, les simulations réalisées ne prennent pas en compte la possibilité de vent. Une solution serait d'augmenter le système du drone dans l'estimateur avec des états "vent" pour avoir une estimation en ligne du vent. Ceci peut néanmoins poser un problème d'observabilité; par exemple être capable de distinguer un biais moteur d'un vent contraire ou favorable à la trajectoire du drone.

9.2.2 Amélioration de la méthode

Dans la suite de cette étude, de nombreux développements sont envisageables. Premièrement, des choix ont été faits pour la méthode développée. Pour l'estimation de l'état et des paramètres de fautes, on a choisi d'utiliser les filtres AKF, TSKF

et EKF. D'autres estimateurs pourraient néanmoins être envisagés tels que le filtre UKF ou le filtre particulaire. De façon similaire, on a fait le choix de baser les contrôleurs sur la méthode LQR. Néanmoins d'autres types de commande peuvent être envisagés tels que la méthode EA ou la méthode PI. Par ailleurs, pour obtenir de meilleures performances, il est possible de construire des contrôleurs prenant en compte les saturations des actionneurs, par exemple en utilisant une méthode anti wind up.

Deuxièmement, la stabilité des contrôleurs dégradés n'a pas été prouvée mathématiquement. Expérimentalement, aucune simulation n'a fait preuve d'instabilité sur le système. Il faudrait néanmoins prouver la stabilité de ces contrôleurs, par exemple à l'aide d'une fonction de Lyapunov [Branicky 1998].

Troisièmement, le GPS est supposé fonctionnel lors de cette étude pour une raison d'observabilité. Une solution pour pallier au manque d'observabilité consiste à rajouter des mesures de la position. Ces mesures peuvent venir d'un autre GPS ou d'une caméra mais l'ajout d'un autre capteur sur le drone n'est pas toujours envisageable. L'intégration de la mesure de l'accélération de l'IMU peut être une solution en cas de pannes GPS temporaires. En effet, celle-ci se dégrade au cours du temps sans recalage. Une autre solution consiste à utiliser la méthode du GLR [Lundin 2017]. Celle-ci pourrait être utilisée en parallèle de notre méthode.

Quatrièmement, dans notre étude, nous avons considéré seulement les pannes uniques. Des scénarios de pannes multiples peuvent être envisagés. L'ajout des pannes multiples entraînerait l'ajout d'autres modes dégradés, tel qu'une panne élevation avec une panne capteur, mais aussi un mode de "crash", pour lequel le drone n'est plus contrôlable et pour lequel la meilleure solution serait un "crash" en douceur, par exemple avec l'arrêt du moteur et l'ouverture d'un parachute. La Figure 9.1 montre comment le mode "crash" pourrait être inséré.

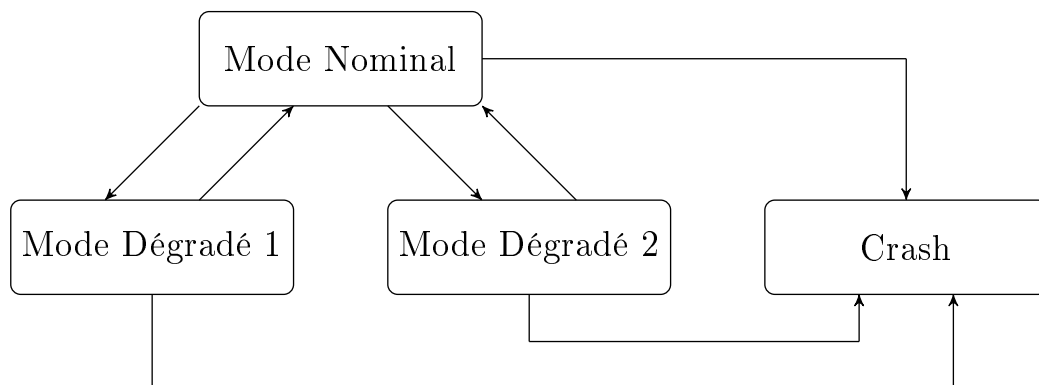


FIGURE 9.1 – Exemple de système prenant en compte un mode "Crash"

Finalement, le cadre POMDP conduit à discrétiser de manière relativement arbitraire l'espace des paramètres de fautes et à calculer des probabilités pour chaque élément de cette discrétisation. L'apprentissage d'un classificateur basé sur un réseau de neurones présentant les moyennes et covariances des paramètres de fautes

en entrée et le mode estimé en sortie pourrait être une alternative intéressante à la méthode actuelle. Par ailleurs, un coût élémentaire analytique plutôt que numérique rendrait la méthode plus générique et moins dépendante de simulations hors ligne.

Modèle dynamique de l'Altimum 4

Sommaire

A.1 Équations générales	189
A.1.1 Équations dynamiques et cinématiques	189
A.1.2 Moments et Forces	190
A.1.3 Coefficients aérodynamiques	191
A.1.4 Élevons	191
A.1.5 Moteur et hélice	191
A.2 Modèle complet	191
A.3 Modèle longitudinal	192

Cette annexe fournit dans un premier temps les équations ayant permis de construire le modèle Simulink de l'Altimum 4. Dans un second temps, elle présente le modèle linéarisé continu, et les paramètres associés, utilisés lors des simulations.

A.1 Équations générales

Les équations présentées ici sont issues d'une modélisation de mécanique du vol assumant certaines hypothèses [Lesprier 2015b].

A.1.1 Équations dynamiques et cinématiques

Soit $\mathbf{V} = [u \ v \ w]^T$, $\mathbf{\Omega} = [p \ q \ r]^T$, $\mathbf{X} = [x \ y \ z]^T$ et $\mathbf{\Phi} = [\phi \ \theta \ \psi]^T$.

$$\dot{\mathbf{V}} = \frac{\mathbf{F}}{m_a} - \mathbf{\Omega} \times \mathbf{V} \quad (\text{A.1})$$

$$\dot{\mathbf{\Omega}} = I^{-1}(\mathbf{M} - \mathbf{\Omega} \times I\mathbf{\Omega}) \quad (\text{A.2})$$

$$\dot{\mathbf{X}} = R_{rot}^{-1}\mathbf{V} \quad (\text{A.3})$$

$$\dot{\mathbf{\Phi}} = T_{rot}\mathbf{\Omega} \quad (\text{A.4})$$

Où m_a correspond à la masse du drone, I à la matrice d'inertie, R_{rot}^{-1} et T_{rot} aux matrices de rotation, et \times au produit vectoriel.

On a :

$$R_{rot}^{-1} = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}$$

et

$$T_{rot} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi (\cos \theta)^{-1} & \cos \phi (\cos \theta)^{-1} \end{bmatrix}$$

A.1.2 Moments et Forces

Le moment au centre de gravité résulte de la poussée et des actions aérodynamiques :

$$\mathbf{M} = \begin{bmatrix} 0 \\ z_{eng} F_x \\ 0 \end{bmatrix} + q_d S L \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} \quad (\text{A.5})$$

Où z_{eng} est la distance verticale entre le centre de gravité et la position du moteur, F_x est la poussée, $q_d = 0.5\rho V_a$ est la pression dynamique, S est la surface de référence, $L = \text{diag}(L_a, L_o, L_a)$ prend en compte les longueurs latérales L_a et longitudinales L_o de référence, C_l , C_m et C_n correspondent aux coefficients aérodynamiques de moment.

La force \mathbf{F} résulte de la poussée \mathbf{F}_{eng} , des actions aérodynamiques \mathbf{F}_a et du poids \mathbf{F}_g :

$$\mathbf{F} = \mathbf{F}_{eng} + \mathbf{F}_g + \mathbf{F}_a \quad (\text{A.6})$$

$$\mathbf{F}_{eng} = \begin{bmatrix} F_x \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.7})$$

$$\mathbf{F}_g = m_a g \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \quad (\text{A.8})$$

$$\mathbf{F}_a = q_d S \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix} \quad (\text{A.9})$$

Où C_D est le coefficient de traînée, C_Y est le coefficient latéral et C_L est le coefficient de portance.

A.1.3 Coefficients aérodynamiques

Les coefficients aérodynamiques sont donnés par les expressions suivantes :

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} \frac{q}{V_a} + C_{L_{\delta_e}} \delta_e \quad (\text{A.10})$$

$$C_Y = C_{Y_\beta} \beta + C_{L_p} \frac{p}{V_a} + C_{L_r} \frac{r}{V_a} + C_{L_{\delta_a}} \delta_a \quad (\text{A.11})$$

$$C_D = C_{D_0} + C_{D_{C_L}} C_L^2 \quad (\text{A.12})$$

$$C_l = C_{l_\beta} \beta + (C_{l_p} p + C_{l_r} r) \frac{L_a}{V_a} + C_{l_{\delta_a}} \delta_a \quad (\text{A.13})$$

$$C_m = C_{m_0} + C_{m_\alpha} \alpha + C_{m_q} q \frac{L_o}{V_a} + C_{m_{\delta_e}} \delta_e \quad (\text{A.14})$$

$$C_n = C_{n_\beta} \beta + (C_{n_p} p + C_{n_r} r) \frac{L_a}{V_a} + C_{n_{\delta_a}} \delta_a \quad (\text{A.15})$$

où δ_e et δ_a sont les déflexions de profondeur et d'ailerons virtuels équivalentes aux déflexions élévons :

$$\delta_a = \frac{0.4\delta_l^1 + 0.6\delta_l^2 - 0.4\delta_r^1 - 0.6\delta_r^2}{2} \quad \delta_e = -\frac{0.5\delta_l^1 + 0.5\delta_l^2 + 0.5\delta_r^1 + 0.5\delta_r^2}{2} \quad (\text{A.16})$$

et α et β sont respectivement l'incidence et le dérapage.

A.1.4 Élévons

Les commandes des élévons prennent en compte :

- un retard pur t_d
- une dynamique du premier ordre τ_δ
- des saturations en vitesse et en position.

A.1.5 Moteur et hélice

La commande de poussée δ_x présente une dynamique du premier ordre :

$$\dot{\delta}_{xf} = \frac{1}{\tau_{\delta_x}} (\delta_x - \delta_{xf}) \quad (\text{A.17})$$

La vitesse du moteur R_m et la force F_x sont définies par une approximation polynomiale :

$$R_m = r_{00} + r_{10} \delta_{xf} + r_{01} V_a + r_{20} \delta_{xf}^2 + r_{11} \delta_{xf} V_a + r_{02} V_a^2 \quad (\text{A.18})$$

$$F_x = f_{00} + f_{20} R_m^2 + f_{11} R_m V_a + f_{02} V_a^2 \quad (\text{A.19})$$

A.2 Modèle complet

Le modèle linéaire utilisé lors de cette thèse est linéarisé pour le point de vol suivant : une vitesse air de $20m/s$ et une pente de -3° . Les matrices A , B , et C

continues sont les suivantes.

$$x = [u \ v \ w \ p \ q \ r \ x \ y \ z \ \phi \ \theta \ \psi \ E \ \delta_e \ \delta_a]^T,$$

$$u = [\delta_x \ \delta_l^1 \ \delta_l^2 \ \delta_r^1 \ \delta_r^2]^T,$$

$$y = [p \ q \ r \ a_{sx} \ a_{sy} \ a_{sz} \ \psi \ \phi \ \theta \ \alpha \ \beta \ z_B \ x \ y \ z \ V_x \ V_y \ V_z]^T$$

$$A = \begin{bmatrix} -0.18 & 0 & 0.67 & 0 & -1.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -9.8 & 0 & 11 & 0.59 & 0 \\ 0 & -0.24 & 0 & 1.6 & 0 & -20 & 0 & 0 & 0 & 9.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.6 \\ -0.57 & 0 & -5.1 & 0 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.27 & 0 & 0 & -16 & 0 \\ 0 & -0.59 & 0 & -47 & 0 & 2.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 187 \\ 0.5 & 0 & -5.4 & 0 & -5.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4.4 & -68 & 0 \\ 0 & 0.65 & 0 & -0.77 & 0 & -0.69 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.6 \\ 1 & 0 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.6 & 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.03 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -2.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -77 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -77 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2.5 & 0 & 0 & 0 & 0 \\ 0 & -19 & -19 & -19 & -19 \\ 0 & 15 & 23 & -15 & -23 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.18 & 0 & -0.67 & 0 & -0.32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -11 & -0.59 & 0 & 0 & 0 \\ 0 & 0.24 & 0 & 0.04 & 0 & -0.06 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 \\ 0.57 & 0 & 5.1 & 0 & 8.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.004 & 0 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.6 & 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.03 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A.3 Modèle longitudinal

Pour le modèle longitudinal, les matrices suivantes sont utilisées

$$x = [u \ w \ q \ x \ z \ \theta \ E \ \delta_e]^T,$$

$$u = [\delta_x \ \delta_l \ \delta_r]^T,$$

$$y = [q \ a_{sx} \ a_{sz} \ \theta \ \alpha \ z_B \ x \ z \ V_x \ V_z]^T$$

$$A = \begin{bmatrix} -0.18 & 0.67 & -1.3 & 0 & 0 & -9.8 & 11 & 0.59 \\ -0.57 & -5.1 & 11 & 0 & 0 & -0.27 & 0 & -16 \\ 0.5 & -5.4 & -5.1 & 0 & 0 & 0 & -4.4 & -68 \\ 1 & 0.03 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.03 & 1 & 0 & 0 & 0 & -20 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -77 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.5 & 0 & 0 \\ 0 & -38 & -38 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.18 & -0.67 & -0.32 & 0 & 0 & 0 & -11 & -0.59 \\ 0.57 & 5.1 & 8.6 & 0 & 0 & 0 & 0 & 16 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.004 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0.03 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.03 & 1 & 0 & 0 & 0 & -20 & 0 & 0 \end{bmatrix}$$

Représentations d'état

Sommaire

B.1 Modèles	196
B.1.1 Représentation d'état des systèmes non linéaires	196
B.1.2 Représentation d'état des systèmes linéaires	196
B.2 Stabilité, observabilité et contrôlabilité d'un système	197
B.2.1 Stabilité	197
B.2.2 Définitions de la contrôlabilité et de l'observabilité	197
B.2.3 Contrôlabilité et observabilité des systèmes discrets linéaire à paramètres invariants	198
B.2.4 Contrôlabilité et observabilité des systèmes discrets linéaire à paramètres variants	198
B.2.5 Contrôlabilité et observabilité des systèmes discrets non-linéaires	199

Cette annexe détaille dans une première section différentes représentations d'état pour des systèmes à dynamique continue. Pour les différentes représentations, plusieurs variables interviennent :

- $\mathbf{x} \in \mathbb{R}^n$ le vecteur d'état ;
- $\mathbf{u} \in \mathbb{R}^l$ le vecteur d'entrée ;
- $\mathbf{z} \in \mathbb{R}^m$ le vecteur de sortie ;
- $\mathbf{w} \in \mathbb{R}^n$ le vecteur du bruit d'état, lié aux incertitudes du modèle d'évolution ;
- $\mathbf{v} \in \mathbb{R}^m$ est le vecteur du bruit de mesures.

Certaines représentations peuvent ne pas prendre en compte des entrées, du bruit d'état ou du bruit de mesures. Par la suite, les représentations sont exposées avec l'ensemble des variables sachant que l'on peut considérer certaines variables comme nulles.

La deuxième section de cette annexe rappelle les bases de l'étude de la stabilité, de l'observabilité et de la contrôlabilité d'un système. Pour plus d'informations sur la contrôlabilité et l'observabilité, le lecteur peut lire des publications de référence [Chen 1995] [Hermann 1977].

B.1 Modèles

B.1.1 Représentation d'état des systèmes non linéaires

La représentation d'un système non linéaire est de la forme :

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \quad (\text{B.1})$$

$$\mathbf{z}(t) = h(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{v}(t) \quad (\text{B.2})$$

B.1.2 Représentation d'état des systèmes linéaires

Pour les systèmes linéaires, la représentation d'état est matricielle. Quatre matrices sont présentes :

- $A \in \mathbb{R}^{n \times n}$ la matrice d'état ;
- $B \in \mathbb{R}^{n \times l}$ la matrice de commande ;
- $C \in \mathbb{R}^{m \times n}$ la matrice d'observation ou de sortie ;
- $D \in \mathbb{R}^{m \times l}$ la matrice d'action directe.

Certaines représentations peuvent considérer soit la matrice B , soit la matrice D , soit les deux matrices nulles.

B.1.2.1 Représentation d'état à temps continu

Il existe deux types de représentation, les représentations stationnaires et non-stationnaires. Dans le premier cas, les paramètres sont invariants dans le temps, c'est-à-dire que les matrices A , B , C et D sont invariantes. Dans le cas d'une représentation non-stationnaire, les paramètres varient dans le temps, c'est à dire que les matrices $A(t)$, $B(t)$, $C(t)$ et $D(t)$ dépendent du temps.

La représentation d'un système linéaire stationnaire est de la forme :

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{w}(t) \quad (\text{B.3})$$

$$\mathbf{z}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) + \mathbf{v}(t) \quad (\text{B.4})$$

La représentation d'un système linéaire non-stationnaire est de la forme :

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + \mathbf{w}(t) \quad (\text{B.5})$$

$$\mathbf{z}(t) = C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t) + \mathbf{v}(t) \quad (\text{B.6})$$

B.1.2.2 Représentation d'état à temps discret

Il est possible de discrétiser avec une période d'échantillonnage T_s un système linéaire à temps continu. Les matrices deviennent les suivantes :

$$A_d = \exp^{AT_s} \quad (\text{B.7})$$

$$B_d = \int_0^{T_s} \exp^{At} B dt \quad (\text{B.8})$$

$$C_d = C \quad (\text{B.9})$$

$$D_d = D \quad (\text{B.10})$$

La représentation d'un système linéaire stationnaire est de la forme :

$$\mathbf{x}_{k+1} = A_d \mathbf{x}_k + B_d \mathbf{u}_k + \mathbf{w}_k \quad (\text{B.11})$$

$$\mathbf{z}_k = C_d \mathbf{x}_k + D_d \mathbf{u}_k + \mathbf{v}_k \quad (\text{B.12})$$

où $k \in \mathbb{N}$.

La représentation d'un système linéaire non-stationnaire est de la forme :

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{w}_k \quad (\text{B.13})$$

$$\mathbf{z}_k = C_k \mathbf{x}_k + D_k \mathbf{u}_k + \mathbf{v}_k \quad (\text{B.14})$$

B.2 Stabilité, observabilité et contrôlabilité d'un système

B.2.1 Stabilité

Pour des systèmes linéaires, l'analyse des pôles permet de conclure sur la stabilité entrée-sortie du système.

- Un système à temps continu est stable si, et seulement si tous ses pôles sont à partie réelle strictement négative.
- Un système à temps discret est stable si, et seulement si tous ses pôles ont un module strictement inférieur à 1.

B.2.2 Définitions de la contrôlabilité et de l'observabilité

Définition B.1 (*Contrôlabilité d'un système discret*)

Un système à temps discret est contrôlable si pour tout état initial \mathbf{x}_0 et tout état final \mathbf{x}_f il existe un entier fini $k_1 > 0$, une séquence d'entrées $\mathbf{u}_0, \dots, \mathbf{u}_{k_1-1}$ telles que $\mathbf{x}_{k_1} = \mathbf{x}_f$.

Définition B.2 (*Observabilité d'un système discret*)

Un système à temps discret est observable si pour un état initial inconnu \mathbf{x}_0 , il existe un entier fini $k_1 > 0$ de façon à ce que la connaissance de la séquence d'entrée \mathbf{u}_k et de la séquence de sortie \mathbf{z}_k de $k = 0$ à k_1 suffise à déterminer l'état initial \mathbf{x}_0 .

B.2.3 Contrôlabilité et observabilité des systèmes discrets linéaire à paramètres invariants

Soit le système discret linéaire à paramètres invariants dans le temps :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \quad (\text{B.15})$$

$$\mathbf{z}_{k+1} = C\mathbf{x}_{k+1} \quad (\text{B.16})$$

Théorème B.1 (Contrôlabilité : Critère de Kalman)

Le système discret, ou la paire (A, B) est contrôlable si et seulement si la matrice de contrôlabilité C est de rang plein.

$$\text{rang}C = \text{rang} \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} = n \quad (\text{B.17})$$

Théorème B.2 (Observabilité : Critère de Kalman)

Le système discret, ou la paire (A, C) est observable si et seulement si la matrice d'observabilité O est de rang plein.

$$\text{rang}O = \text{rang} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n \quad (\text{B.18})$$

B.2.4 Contrôlabilité et observabilité des systèmes discrets linéaire à paramètres variants

Soit le système discret linéaire à paramètres variants dans le temps :

$$\mathbf{x}_{k+1} = A_k\mathbf{x}_k + B_k\mathbf{u}_k \quad (\text{B.19})$$

$$\mathbf{z}_{k+1} = C_k\mathbf{x}_{k+1} \quad (\text{B.20})$$

L'évaluation de l'observabilité du système se base sur les concepts de matrice de transition et de grammien d'observabilité.

Définition B.3 (Matrice de transition)

La matrice de transition du système $x_{k+1} = A_k x_k$ est solution de

$$\Phi(k+1, k_0) = A_k \Phi(k, k_0) \quad (\text{B.21})$$

avec $\Phi(k_0, k_0) = \mathbb{I}$

Pour $k = k_0 + s, s > 0$, la solution peut être obtenue directement avec :

$$\Phi(k_0 + s, k_0) = A_{k_0+s-1} A_{k_0+s-2} \cdots A_{k_0} \quad (\text{B.22})$$

Théorème B.3 (Contrôlabilité d'un système discret à paramètres variants)

La paire (A_k, B_k) est contrôlable pour tout entier fini k_0 si et seulement si il existe un entier fini $k_1 > k_0$ de façon à ce que le grammien de contrôlabilité, matrice $n \times n$

$$W_c(k_0, k_1) = \sum_{i=k_0}^{k_1} \Phi(k_1, i) B_i B_i^T \Phi^T(k_1, i) \quad (\text{B.23})$$

où $\Phi(i, k_1)$ est la matrice de transition de $x_{k+1} = A_k x_k$, soit non singulier.

Pour $k_1 = k_0 + s, s > 0$, le grammien de contrôlabilité peut être obtenu directement avec :

$$W_c(k_0, k_0 + s) = \sum_{i=0}^s \Phi(k_1, k_0 + i) B_i B_i^T \Phi^T(k_1, k_0 + i) \quad (\text{B.24})$$

Théorème B.4 (Observabilité d'un système discret à paramètres variants)

La paire (A_k, C_k) est observable à tout entier fini k_0 si et seulement si il existe un entier fini $k_1 > k_0$ de façon à ce que le grammien d'observabilité, matrice $n \times n$

$$W_o(k_0, k_1) = \sum_{i=k_0}^{k_1} \Phi^T(i, k_0) C_i^T C_i \Phi(i, k_0) \quad (\text{B.25})$$

où $\Phi(i, k_0)$ est la matrice de transition de $x_{k+1} = A_k x_k$, est non singulier.

Pour $k_1 = k_0 + s, s > 0$, le grammien d'observabilité peut être obtenue directement avec :

$$W_o(k_0, k_0 + s) = \sum_{i=0}^s \Phi^T(k_0 + i, k_0) C_i^T C_i \Phi(k_0 + i, k_0) \quad (\text{B.26})$$

Remarque B.1

Il existe une autre méthode qui consiste à modéliser le système comme un système linéaire par morceaux [Goshen-Meskin 1992].

B.2.5 Contrôlabilité et observabilité des systèmes discrets non-linéaires

Dans le cas des systèmes non-linéaires à plusieurs entrées et plusieurs sorties, aucune solution générale aux problèmes d'observabilité et de contrôlabilité n'existe. Il est possible de regarder localement l'observabilité et la contrôlabilité.

Soit le système discret linéaire à paramètres variants dans le temps :

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad (\text{B.27})$$

$$\mathbf{z}_{k+1} = h(\mathbf{x}_{k+1}) \quad (\text{B.28})$$

Théorème B.5 (Contrôlabilité local d'un système non-linéaire)

Le système discret est localement contrôlable en \mathbf{x}_0 si et seulement si la matrice de contrôlabilité $C(\mathbf{x}_0)$ est de rang plein.

$$\text{rang}C(\mathbf{x}_0) = \text{rang} [f_1(\mathbf{x}_0) \quad f_1f_2(\mathbf{x}_0) \quad \cdots \quad f_1 \cdots f_n(\mathbf{x}_0)] = n \quad (\text{B.29})$$

Où $f_i(\mathbf{x}_k) = f(\mathbf{x}_k, \mathbf{u}_i)$

Théorème B.6 (Observabilité local d'un système non-linéaire)

Le système discret est localement observable en \mathbf{x}_0 si et seulement si la matrice d'observabilité $O(\mathbf{x}_0)$ est de rang plein.

$$\text{rang}O(\mathbf{x}_0) = \text{rang} \begin{bmatrix} dh(\mathbf{x}_0) \\ dL_f h(\mathbf{x}_0) \\ \vdots \\ dL_f^{n-1} h(\mathbf{x}_0) \end{bmatrix} = n \quad (\text{B.30})$$

Où $L_f h = dhf$ est la dérivée de Lie et $L_f^{n-1} h = (L_f^{n-2} h)f$

Systemes à événements discrets

Sommaire

C.1 Les automates	201
C.2 Les modèles de Markov	201
C.3 Les processus de décision Markovien	202
C.3.1 Processus MDP	202
C.3.2 Processus POMDP	203

C.1 Les automates

Un automate est un dispositif reproduisant en autonomie une séquence d'actions prédéterminées sans l'intervention humaine, le système fait toujours la même chose, ou s'adapte à des conditions environnementales perçues par l'intermédiaire de capteurs.

Un automate fini $A = \{S, E, T, s_0\}$ est composé de [Sampath 1995] :

- S , un ensemble fini d'états,
- E , un ensemble fini d'événements,
- $T : S \times E \rightarrow S$, la fonction de transition d'états, et
- s_0 , l'état initial, élément de S .

C.2 Les modèles de Markov

Les modèles de Markov sont des automates probabilistes qui se basent sur l'hypothèse de Markov : "l'information utile pour la prédiction du futur est entièrement contenue dans l'état présent du processus et n'est pas dépendante des états antérieurs". Différents formalismes existent pour les modèles de Markov. Les modèles de cette famille sont montrés dans le Tableau C.1.

Une chaîne de Markov est un processus à temps discret, ou à temps discret et à espace d'états discret. On s'intéresse plus particulièrement aux chaînes de Markov à espace d'états discret. C'est un processus stochastique possédant la propriété suivante :

Définition C.1 (*Chaîne de Markov*)

Le processus stochastique $(X_n)_{n \geq 0}$ à valeurs dans S ensemble fini ou dénombrable est une chaîne de Markov si $\forall n \in \mathbb{N}, \forall x_0, \dots, y \in S$,

	Aucune action au niveau des transitions	Action au niveau des transitions
États observables	Chaînes de Markov	MDP
États partiellement observables	HMM	POMDP

Tableau C.1 – Formalismes Markoviens : *Légende : Processus de décision markovien, en anglais Markov Decision Process (MDP), Modèle de Markov caché, en anglais Hidden Markov Model (HMM), Processus de décision markovien partiellement observable, en anglais Partially Observable Markov Decision Process (POMDP)*

$$\mathbb{P}(X_{n+1} = y | X_0 = x_0, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = y | X_n = x_n)$$

Les chaînes de Markov permettent de modéliser des systèmes dont l'évolution est incertaine. La fonction probabiliste permet donc d'exprimer la loi d'évolution du modèle, sous la forme d'une matrice de probabilité. Les transitions à venir sont incertaines de l'utilisateur mais les états d'une exécution sont connus.

Les systèmes pour lesquels les états d'une exécution sont inconnus de l'utilisateur et seul certains paramètres sont connus de l'utilisateur peuvent être modélisés par des modèles de Markov caché, *Hidden Markov Model* (HMM). Un HMM est un quadruplet $\{S, T, \Omega, O\}$ où :

- $S = \{s_0, \dots, s_{|S|-1}\}$ est l'ensemble fini des états possibles du système,
- $T : S \times S \rightarrow [0; 1]$ est la fonction de transition du système,
- $\Omega = \{\omega_0, \dots, \omega_{|\Omega|-1}\}$ est un ensemble de symboles observables, et
- $O : S \times \Omega \rightarrow [0; 1]$ une fonction d'observation qui à un état donné associe la probabilité d'observer un symbole donné.

Les chaînes de Markov ont été étendues pour des systèmes dépendants d'actions au cours du temps. On retrouve les processus de décision markovien, en anglais *Markov Decision Process* (MDP) dérivés des chaînes de Markov, et les processus de décision markovien partiellement observable, en anglais *Partially Observable Markov Decision Process* (POMDP) dérivés des HMM [Puterman 2014].

C.3 Les processus de décision Markovien

C.3.1 Processus MDP

Un MDP est un modèle stochastique où un agent prend des décisions et où les résultats de ses actions sont aléatoires. C'est un quadruplet $\{S, A, T, R\}$ où :

- $S = \{s_0, \dots, s_{|S|-1}\}$ est l'ensemble fini des états possibles du système,
- $A = \{a_0, \dots, a_{|A|-1}\}$ est l'ensemble fini des actions,
- $T : S \times A \times S \rightarrow [0; 1]$ est la fonction de transition du système en réaction aux actions, et
- $R : S \times A \times S \rightarrow \mathbb{R}$ est la fonction de récompense.

À chaque étape, le processus est dans un certain état s , et l'agent choisit une action a . La probabilité que le processus arrive à l'état s' est décrite par la fonction

de transition d'états T qui satisfait la propriété de Markov. Finalement, quand le processus passe de l'état s à l'état s' avec l'action a , l'agent gagne une récompense R . La différence par rapport aux chaînes de Markov est l'addition des actions choisies par l'agent et des récompenses gagnées par l'agent. S'il n'y a qu'une seule action à tirer dans chaque état et que les récompenses sont égales, le processus de décision markovien est une chaîne de Markov.

L'objectif est d'amener le système dans un état considéré comme bénéfique, en évitant de lui faire traverser des états néfastes, à partir d'un ensemble d'actions possibles sur le système. Il faut donc trouver à chaque pas la meilleure action. On se retrouve face à un problème d'optimisation. Résoudre le problème d'optimisation pour un MDP consiste à trouver une stratégie maximisant la somme des récompenses attendue. L'équation suivante représente le principe de résolution du problème par Programmation Dynamique :

$$V(s) = \max_{a \in A} \left[\sum_{s'} T(s'|s, a) (R(s, a, s) + \gamma V(s')) \right] \quad (\text{C.1})$$

où γ est un facteur de réduction et la fonction $V(\cdot)$ représente le gain espéré total pour chaque état.

C.3.2 Processus POMDP

Les POMDP sont des HMM particuliers, dans lesquels on dispose d'actions probabilistes. Un POMDP est un tuple $\{S, A, T, R, \Omega, O\}$ où :

- $S = \{s_0, \dots, s_{|S|-1}\}$ est l'ensemble fini des états possibles du système,
- $A = \{a_0, \dots, a_{|A|-1}\}$ est l'ensemble fini des actions,
- $T : S \times A \times S \rightarrow [0; 1]$ est la fonction de transition du système en réaction aux actions,
- $R : S \times A \times S \rightarrow \mathbb{R}$ est la fonction de récompense,
- $\Omega = \{\omega_0, \dots, \omega_{|\Omega|-1}\}$ est un ensemble de symboles observables, et
- $O : S \times \Omega \rightarrow [0; 1]$ une fonction d'observation qui à un état donné associe la probabilité d'observer un symbole donné.

Un MDP n'inclut pas d'observations car l'agent connaît toujours l'état courant du système. Un POMDP ne connaît pas l'état courant mais a accès à des observations lui permettant d'estimer l'état courant. Ainsi, à chaque pas de temps, un état de croyance est mis à jour de la façon suivante :

$$\rho^{a, \omega}(s') = \frac{O(\omega|s') Pr(s'|\rho, a)}{O(\omega|\rho, a)} \quad (\text{C.2})$$

$$\text{Où} \begin{cases} Pr(s'|\rho, a) = \sum_s T(s'|s, a) \rho(s) \\ O(\omega|\rho, a) = \sum_{s''} O(\omega|s'') Pr(s''|\rho, a) \end{cases}$$

De façon similaire au MDP, le but est de construire une politique qui maximise le coût R . Un POMDP peut être formulé comme un MDP où chaque état de croyance est un état.

L'équation suivante représente le principe de résolution du problème par Programmation Dynamique :

$$V_k(\rho) = \max_{a \in A} [r(\rho, a) + \gamma \sum_{\omega} O(\omega | \rho, a) V_{k+1}(\rho^{a, \omega})] \quad (\text{C.3})$$

Où $\gamma \in [0, 1[$ est un facteur de réduction, $r(\rho, a) = \sum_s \sum_{s'} R(s, a, s') p(s' | s, a) \rho(s)$ et $V(\cdot)$ est le gain espéré total fonction de la croyance.

Filtre de Kalman

Sommaire

D.1 Le filtre de Kalman pour les systèmes linéaires	205
D.1.1 L'approche classique	205
D.1.2 Une approche séquentielle pour le traitement des observations	206
D.2 Le filtre de Kalman pour les systèmes non-linéaires	207
D.2.1 Le filtre de Kalman étendu	207
D.2.2 Le filtre de Kalman non parfumé	208

Cette annexe détaille le principe du Filtre de Kalman. Les filtres de Kalman sont construits sur la base d'un algorithme récursif en deux étapes : une étape de prédiction et une étape de mise à jour (ou correction). La phase de prédiction utilise l'état estimé de l'instant précédent pour produire une estimation de l'état courant. Dans l'étape de mise à jour, les mesures de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise. Les équations du filtre de Kalman sont présentées lors de la première section. Les équations du filtre étendu sont présentées lors de la deuxième section.

D.1 Le filtre de Kalman pour les systèmes linéaires

D.1.1 L'approche classique

Le filtre de Kalman permet d'estimer l'état d'un système dynamique linéaire à partir d'une série de mesures bruitées. C'est un filtre récursif, c'est à dire que pour estimer l'état courant, seule l'estimation de l'état précédent et les mesures actuelles sont nécessaires. Le filtre de Kalman permet de prendre en compte les incertitudes liées aux états et aux mesures. Il permet de résoudre le problème d'estimation en temps continu et discret. Seule la synthèse discrète est considérée ici.

On considère la représentation d'équations linéaires à temps discret (cf. Annexe B), en considérant D nulle. Il est supposé que les vecteurs de bruits \mathbf{w}_k et \mathbf{v}_k sont stationnaires et correspondent à des processus gaussiens à moyenne nulle ayant pour covariance :

$$\mathbb{E} \left\{ \mathbf{w}_k \mathbf{w}_j^T \right\} = \delta_{kj} \mathbf{Q}_k \quad \mathbb{E} \left\{ \mathbf{v}_k \mathbf{v}_j^T \right\} = \delta_{kj} \mathbf{R}_k, \quad \mathbb{E} \left\{ \mathbf{v}_k \mathbf{w}_j^T \right\} = 0, \quad \forall k, j \quad (\text{D.1})$$

Où δ_{kj} est le symbole de Kronecker.

Les erreurs liées au système et celles liées aux observations sont non-corrélées. La densité de probabilité de la variable aléatoire représentant l'état initial est une Gaussienne de covariance $P_{0|0}$ centrée en $\hat{\mathbf{x}}_{0|0}$.

La phase de prédiction est définie par les deux équations suivantes :

$$\hat{\mathbf{x}}_{k+1|k} = A\hat{\mathbf{x}}_{k|k} + B\mathbf{u}_k \quad (\text{D.2})$$

$$P_{k+1|k} = AP_{k|k}A^T + Q_k \quad (\text{D.3})$$

D.2 et D.3 correspondent respectivement à la moyenne et à la covariance de la prédiction de l'état.

La phase de mise à jour est définie par les équations suivantes :

$$\boldsymbol{\mu}_{k+1} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k} = \mathbf{z}_{k+1} - C\hat{\mathbf{x}}_{k+1|k} \quad (\text{D.4})$$

$$K_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + R_{k+1})^{-1} \quad (\text{D.5})$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1}\boldsymbol{\mu}_{k+1} \quad (\text{D.6})$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1}CP_{k+1|k} \quad (\text{D.7})$$

D.4 correspond au vecteur d'innovation, c'est à dire à la différence entre les observations mesurées et prédites D.5 correspond au gain de Kalman optimal, D.6 et D.7 correspondent respectivement à la moyenne et à la covariance de l'estimation mise à jour de l'état.

Remarque D.1

Le terme $CP_{k+1|k}C^T + R_{k+1}$ correspond à la matrice de covariance associée au vecteur d'innovation.

D.1.2 Une approche séquentielle pour le traitement des observations

Pour beaucoup de systèmes, l'ensemble des capteurs ne fournissent pas les mesures au même instant. La fréquence des mesures dépend du capteur utilisé. Pour pallier ce problème, il est possible d'utiliser une approche séquentielle lors de l'étape de correction. Cette approche permet aussi de réduire le coût calculatoire du filtre de Kalman en s'affranchissant de l'étape d'inversion matricielle présente dans l'équation D.5. Néanmoins, la covariance du bruit R doit être diagonale pour pouvoir utiliser cette approche.

Les différentes composantes de l'observation courante \mathbf{z}_k sont traitées selon la procédure suivante :

- l'étape d'initialisation : $\hat{\mathbf{x}}_{k+1}^0 = \hat{\mathbf{x}}_{k+1|k}$ et $P_{k+1}^0 = P_{k+1|k}$
- l'itération de i allant de 1 à m mesures disponible à l'instant $k+1$:

$$K_{k+1}^i = P_{k+1}^{i-1}C_i^T(C_iP_{k+1}^{i-1}C_i^T + R_{ii,k+1})^{-1} \quad (\text{D.8})$$

$$\hat{\mathbf{x}}_{k+1}^i = \hat{\mathbf{x}}_{k+1}^{i-1} + K_{k+1}^i(\mathbf{z}_{k+1}^i - C_i\hat{\mathbf{x}}_{k+1}^{i-1}) \quad (\text{D.9})$$

$$P_{k+1}^i = P_{k+1}^{i-1} - K_{k+1}^iC_iP_{k+1}^{i-1} \quad (\text{D.10})$$

— la correction finale : $\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1}^m$ et $P_{k+1|k+1} = P_{k+1}^m$

D.2 Le filtre de Kalman pour les systèmes non-linéaires

D.2.1 Le filtre de Kalman étendu

La première variante correspond à l'EKF qui permet d'estimer l'état d'un système non-linéaire. Cette approche a été généralisée par Breakwell et Jazwinski [Jazwinski 2007] en 1970 et consiste à recourir à un développement en série de Taylor, limité au premier ordre, du système non-linéaire d'origine autour de l'état estimé. Le filtrage est sous-optimal mais de très bonnes performances sont tout de même obtenues sous l'hypothèse de bruits gaussiens et en présence de faibles non-linéarités.

Pour le filtre EKF, les modèles d'évolution et d'observation n'ont pas besoin d'être des fonctions linéaires de l'état mais peuvent être des fonctions quelconques. On considère le système d'équations non-linéaires à temps discret (cf. Annexe B) avec $u = 0$ dans l'équation de sortie. Là encore, les bruits d'états et de mesures \mathbf{w} et \mathbf{v} qui entrent dans le modèle sont supposés gaussiens, centrés et décorrélés.

La fonction f peut être utilisée pour calculer l'état prédit à partir de l'état estimé précédent et, semblablement, la fonction h peut être employée pour calculer l'observation prédite de l'état prédit. Cependant, f et h ne peuvent pas être appliqués directement au calcul de la covariance : des matrices de dérivées partielles, les Jacobiennes F_k et H_k , sont calculées.

À chaque instant, les Jacobiennes sont évaluées pour les commandes courantes et les états estimés courants. Ces matrices peuvent être employées dans les équations du filtre de Kalman. Ce processus linéarise essentiellement la fonction non linéaire autour de l'estimation courante.

La phase de prédiction est définie par les deux équations suivantes :

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k) \quad (\text{D.11})$$

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \quad (\text{D.12})$$

La phase de mise à jour est définie par les équations suivantes :

$$\boldsymbol{\mu}_{k+1} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k} = \mathbf{z}_{k+1} - h(\hat{\mathbf{x}}_{k+1|k}) \quad (\text{D.13})$$

$$K_{k+1} = P_{k+1|k} H_k^T (H_k P_{k+1|k} H_k^T + R_{k+1})^{-1} \quad (\text{D.14})$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1} \boldsymbol{\mu}_{k+1} \quad (\text{D.15})$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} H_k P_{k+1|k} \quad (\text{D.16})$$

Où F_k et H_k sont définies comme étant les Jacobiennes respectivement associées à f et g autour des vecteurs d'état estimés $\hat{\mathbf{x}}_{k|k}$ et $\hat{\mathbf{x}}_{k+1|k}$:

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k} \quad (\text{D.17})$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{\hat{\mathbf{x}}_{k+1|k}} \quad (\text{D.18})$$

Les limites associées au filtre EKF proviennent de son étape de linéarisation recourant à un développement en série de Taylor au premier ordre. Dans le cas de fortes non linéarités, cette approximation peut s'avérer insuffisante et conduire à la propagation d'erreurs pouvant impacter la stabilité du filtre.

D.2.2 Le filtre de Kalman non parfumé

Des outils mathématiques ont vu le jour pour adapter le filtrage de Kalman aux systèmes présentant de fortes non-linéarités. On retrouve le filtre de Kalman non parfumé, en anglais *Unscented Kalman Filter* (UKF) [Julier 2004]. Le principe est le même que celui du filtre EKF, néanmoins la méthode de linéarisation est modifiée. Ce filtre est basé sur la transformée non parfumée, une fonction mathématique permettant d'estimer le résultat de l'application d'une transformation non-linéaire à une distribution caractérisée par un ensemble fini de points d'échantillonnage déterministes pour calculer sa valeur moyenne et sa covariance. Avec l'UKF, il n'est pas nécessaire de calculer les jacobiniennes.

L'UKF donne en général de meilleurs résultats que l'EKF cependant, la différence est faible [Chowdhary 2010], [Rhudy 2013]. Cet avantage de l'UKF sur l'EKF est restreint à des cas d'applications fortement non linéaires. Dans le cas de faibles non linéarités, les erreurs commises par l'EKF et l'UKF sont quasiment similaires.

Loi normale et tests associés

Sommaire

E.1	Loi normale	209
E.2	Loi normale multivariée	209
E.3	Le test de Student et le test du χ^2	210

Cette annexe détaille la loi normale, la loi normale multidimensionnelle, ainsi que les tests de décision associés à la loi normale.

E.1 Loi normale

La loi normale est une loi de probabilité continue qui dépend de deux paramètres : son espérance, ou sa moyenne, μ et son écart type, σ . La densité de probabilité de la loi normale est donnée par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (\text{E.1})$$

Lorsqu'une variable aléatoire X suit la loi normale, elle est dite gaussienne ou normale et il est habituel d'utiliser la notation avec la variance σ^2 : $X \sim \mathcal{N}(\mu, \sigma^2)$

E.2 Loi normale multivariée

La loi normale multi-variée est une généralisation multidimensionnelle de la loi normale. La loi multi-normale est paramétrée par un vecteur $\boldsymbol{\mu} \in \mathbb{R}^N$ représentant son centre et une matrice semi-définie positive $\Sigma \in \mathbb{R}^{N \times N}$. La densité de probabilité de la loi multi-normale est donnée par :

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp^{-\frac{1}{2}((\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu}))} \quad (\text{E.2})$$

Où $|\Sigma|$ est le déterminant de Σ .

Cette loi est habituellement notée $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

E.3 Le test de Student et le test du χ^2

Le test de Student, ou test t , s'applique lorsque que le signal étudié suit une loi de distribution gaussienne et que l'on cherche un niveau de confiance fixé. Le principe est de déterminer si la valeur η d'une population de distribution Normale et d'écart type σ non connu est égale à la valeur η_0 de l'hypothèse nulle H_0 .

Soit X une série de valeurs x_0, \dots, x_{n-1} de taille n de la population, de moyenne \bar{x} et d'écart-type s . La comparaison de la moyenne observée \bar{x} à la valeur d'hypothèse η_0 est donnée par la variable t :

$$t = \frac{\bar{x} - \eta_0}{s/\sqrt{n}} \quad (\text{E.3})$$

avec $s = \sqrt{\frac{1}{n-1} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}$

Selon l'hypothèse nulle H_0 , la variable t suit une loi de Student à $n - 1$ degrés de liberté. Pour rejeter ou non l'hypothèse nulle H_0 , on peut ensuite utiliser une table des valeurs de la distribution de loi de Student pour voir si la valeur de la statistique est supérieure au quantile à 95%, ce qui correspondant à un risque de 5%.

Contrairement au test du χ^2 , ce test est applicable uniquement si la série de valeurs X suit une loi Normale d'écart-type inconnu. Le principe du test du χ^2 est similaire à celui du test de Student. Il permet de vérifier si un échantillon d'une variable aléatoire X donne des observations comparables à celle d'une loi de probabilité définie *a priori* selon l'hypothèse nulle H_0 . Pour ce test, l'ensemble des valeurs que peut prendre x est découpé en N classes. Ainsi, la probabilité empirique que X appartienne à la classe $j < N$ est comparée à la probabilité théorique d'y appartenir.

Méthode LQR

Sommaire

F.1	Théorème LQR	211
F.2	Théorème LQR réduit	212
F.3	Théorème LQR réduit avec terme additionnel	213

Cette annexe détaille la méthode LQR ; les deux premières sections présentent le théorème LQR et le théorème LQR réduit [Bertsekas 1995] et la troisième section présente le théorème LQR réduit avec terme additionnel développé dans cette thèse.

F.1 Théorème LQR

Théorème F.1 (LQR)

Soit le système à contrôler décrit par l'équation linéaire suivante :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \quad (\text{F.1})$$

Considérons le critère quadratique suivant, qui prend en compte l'objectif conjoint de réduire l'état et la commande au cours du temps :

$$J_0^\infty = \sum_{k=0}^{k=\infty} (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) \quad (\text{F.2})$$

La commande linéaire minimisant l'indice de performance est :

$$\mathbf{u}_k = -(R + B^T P B)^{-1} B^T P A \mathbf{x}_k \quad (\text{F.3})$$

Avec : $P = Q + A^T [P - P B (B^T P B + R)^{-1} B^T P] A$

Remarque F.1

La démonstration du LQR est équivalente à celle du LQR réduit, montrée dans la section suivante, en prenant $\alpha = 1$.

F.2 Théoreme LQR réduit

Théorème F.2 (*LQR réduit*)

Soit le système à contrôler décrit par l'équation linéaire suivante :

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \quad (\text{F.4})$$

Considérons le critère quadratique suivant, qui prend en compte l'objectif conjoint de réduire l'état et la commande au cours du temps :

$$J_0^\infty = \sum_{k=0}^{k=\infty} \alpha^k (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) \quad (\text{F.5})$$

La commande linéaire minimisant l'indice de performance est :

$$\mathbf{u}_k = -\alpha(R + \alpha B^T P B)^{-1} B^T P A \mathbf{x}_k \quad (\text{F.6})$$

Avec : $P = Q + \alpha A^T [P - \alpha P B (\alpha B^T P B + R)^{-1} B^T P] A$

Démonstration : La fonction Hamiltonienne associée est :

$$H_k = \frac{1}{2} \alpha^k (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) + \boldsymbol{\mu}_{k+1}^T (A\mathbf{x}_k + B\mathbf{u}_k) \quad (\text{F.7})$$

Où $\boldsymbol{\mu}_k$ est le vecteur adjoint. L'équation récurrente est la suivante :

$$\boldsymbol{\mu}_k = \frac{\partial H_k}{\partial \mathbf{x}_k} = \alpha^k Q \mathbf{x}_k + A^T \boldsymbol{\mu}_{k+1} \quad (\text{F.8})$$

La condition de stationnarité est :

$$0 = \frac{\partial H_k}{\partial \mathbf{u}_k} = \alpha^k R \mathbf{u}_k + B^T \boldsymbol{\mu}_{k+1} \quad (\text{F.9})$$

Selon l'équation F.9,

$$\mathbf{u}_k = -\alpha^{-k} R^{-1} B^T \boldsymbol{\mu}_{k+1} \quad (\text{F.10})$$

Le système Hamiltonien est obtenu en combinant les équations (F.4), (F.9) et (F.10) :

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu}_k \end{bmatrix} = \begin{bmatrix} A & -\alpha^{-k} B R^{-1} B^T \\ \alpha^k Q & A^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\mu}_{k+1} \end{bmatrix} \quad (\text{F.11})$$

Supposons que $\boldsymbol{\mu}_k$ soit de la forme :

$$\boldsymbol{\mu}_k = \alpha^k P_k \mathbf{x}_k \quad (\text{F.12})$$

Remplacer (F.12) dans la première équation de (F.11) donne :

$$\mathbf{x}_{k+1} = (Id + \alpha B R^{-1} B^T P_{k+1})^{-1} A \mathbf{x}_k \quad (\text{F.13})$$

Remplacer (F.12) et (F.13) dans la seconde équation de F.11 donne :

$$[-P_k + Q + \alpha A^T P_{k+1} (Id + \alpha B R^{-1} B^T P_{k+1})^{-1} A] \mathbf{x}_k = 0 \quad (\text{F.14})$$

Cette équation doit être valable pour toutes les valeurs de l'état \mathbf{x}_k , donc le terme entre crochets doit s'annuler. En utilisant le lemme d'inversion matriciel :

$$P_k = Q + \alpha A^T [P_{k+1} - \alpha P_{k+1} B (\alpha B^T P_{k+1} B + R)^{-1} B^T P_{k+1}] A \quad (\text{F.15})$$

La solution pour un horizon infini de P est :

$$P = Q + \alpha A^T [P - \alpha P B (\alpha B^T P B + R)^{-1} B^T P] A \quad (\text{F.16})$$

Finalement, à partir des équations (F.10), (F.12) et (F.4) avec $P_{k+1} = P$:

$$\mathbf{u}_k = -\alpha (R + \alpha B^T P B)^{-1} B^T P A \mathbf{x}_k \quad (\text{F.17})$$

En définissant $\bar{A} = \sqrt{\alpha} A$ et $\bar{B} = \sqrt{\alpha} B$,

$$P = Q - \bar{A}^T P \bar{B} [\bar{B}^T P \bar{B} + R]^{-1} \bar{B}^T P \bar{A} + \bar{A}^T P \bar{A} \quad (\text{F.18})$$

L'équation (F.18) correspond à l'équation classique de Riccati appliquée à la paire (\bar{A}, \bar{B}) . ■

Théorème F.3 (*Inversion Matricielle*)

Soit A, B, C et D des matrices. La relation suivante est vraie :

$$(A^{-1} + BCD)^{-1} = A - AB(DAB + C^{-1})^{-1}DA \quad (\text{F.19})$$

F.3 Théorème LQR réduit avec terme additionnel

Théorème F.4 (*LQR réduit avec un terme additionnel*)

Soit le système à contrôler décrit par l'équation linéaire suivante :

$$\mathbf{x}_{k+1} = A \mathbf{x}_k + E \mathbf{u}_k + F \quad (\text{F.20})$$

L'indice de performance associé est la fonction quadratique suivante :

$$J_0^\infty = \sum_{k=0}^{k=\infty} \alpha^k (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) \quad (\text{F.21})$$

La commande optimisant l'indice de performance est la suivante :

$$\mathbf{u}_k = -\alpha (R + \alpha E^T P E)^{-1} E^T P A \mathbf{x}_k - \alpha (R + \alpha E^T P E)^{-1} E^T [P F - \mathbf{v}] \quad (\text{F.22})$$

Avec : $P = Q + \alpha A^T [P - \alpha P E (\alpha E^T P E + R)^{-1} E^T P] A$ et $\mathbf{v} = \alpha A^T (Id +$

$$\left[\alpha P E R^{-1} E^T \right]^{-1} (\mathbf{v} - P F)$$

Démonstration : La fonction Hamiltonienne associée est :

$$H_k = \frac{1}{2} \alpha^k (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) + \boldsymbol{\mu}_{k+1}^T (A \mathbf{x}_k + E \mathbf{u}_k + F) \quad (\text{F.23})$$

Où $\boldsymbol{\mu}_k$ est le vecteur adjoint. L'équation récurrente est la suivante :

$$\boldsymbol{\mu}_k = \frac{\partial H_k}{\partial \mathbf{x}_k} = \alpha^k Q \mathbf{x}_k + A^T \boldsymbol{\mu}_{k+1} \quad (\text{F.24})$$

La condition de stationnarité est :

$$0 = \frac{\partial H_k}{\partial \mathbf{u}_k} = \alpha^k R \mathbf{u}_k + E^T \boldsymbol{\mu}_{k+1} \quad (\text{F.25})$$

Selon l'équation F.25,

$$\mathbf{u}_k = -\alpha^{-k} R^{-1} E^T \boldsymbol{\mu}_{k+1} \quad (\text{F.26})$$

Le système Hamiltonien est obtenu en combinant les équations (F.20), (F.25) et (F.26) :

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu}_k \end{bmatrix} = \begin{bmatrix} A & -\alpha^{-k} E R^{-1} E^T \\ \alpha^k Q & A^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\mu}_{k+1} \end{bmatrix} + \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (\text{F.27})$$

Supposons que $\boldsymbol{\mu}_k$ soit de la forme :

$$\boldsymbol{\mu}_k = \alpha^k (P_k \mathbf{x}_k - \mathbf{v}_k) \quad (\text{F.28})$$

Remplacer (F.28) dans la première équation de (F.27) donne :

$$\mathbf{x}_{k+1} = (Id + \alpha E R^{-1} E^T P_{k+1})^{-1} A \mathbf{x}_k + (Id + \alpha E R^{-1} E^T P_{k+1})^{-1} [\alpha E R^{-1} E^T \mathbf{v}_{k+1} + F] \quad (\text{F.29})$$

Remplacer (F.28) et (F.29) dans la second équation de F.27 donne :

$$\begin{aligned} & [-P_k + Q + \alpha A^T P_{k+1} (Id + \alpha E R^{-1} E^T P_{k+1})^{-1} A] \mathbf{x}_k \\ & + [\alpha A^T P_{k+1} (Id + \alpha E R^{-1} E^T P_{k+1})^{-1} \alpha E R^{-1} E^T \mathbf{v}_{k+1} \\ & + \alpha A^T P_{k+1} (Id + \alpha E R^{-1} E^T P_{k+1})^{-1} F + \mathbf{v}_k - \alpha A^T \mathbf{v}_{k+1}] = 0 \end{aligned} \quad (\text{F.30})$$

Cette équation doit être valable pour toutes les valeurs de l'état \mathbf{x}_k , donc les termes entre crochets doivent s'annuler individuellement. En utilisant le lemme d'inversion matriciel (cf. Annexe F) :

$$P_k = Q + \alpha A^T [P_{k+1} - \alpha P_{k+1} E (\alpha E^T P_{k+1} E + R)^{-1} E^T P_{k+1}] A \quad (\text{F.31})$$

et

$$\begin{aligned} \mathbf{v}_k &= \alpha A^T \mathbf{v}_{k+1} - \alpha A^T P_{k+1} (Id + \alpha E R^{-1} E^T P_{k+1})^{-1} \alpha E R^{-1} E^T \mathbf{v}_{k+1} \\ &\quad - \alpha A^T P_{k+1} (Id + \alpha E R^{-1} E^T P_{k+1})^{-1} F \end{aligned} \quad (\text{F.32})$$

La solution pour un horizon infini de P et de \mathbf{v} sont :

$$P = Q + \alpha A^T [P - \alpha P E (\alpha E^T P E + R)^{-1} E^T P] A \quad (\text{F.33})$$

et

$$\mathbf{v} = \alpha A^T (Id + \alpha P E R^{-1} E^T)^{-1} (\mathbf{v} - P F) \quad (\text{F.34})$$

Finalement, à partir des équations (F.26), (F.28) et (F.20) avec $P_{k+1} = P$ et $\mathbf{v}_{k+1} = \mathbf{v}$:

$$\mathbf{u}_k = -\alpha (R + \alpha E^T P E)^{-1} E^T P A \mathbf{x}_k - \alpha (R + \alpha E^T P E)^{-1} E^T [P F - \mathbf{v}] \quad (\text{F.35})$$

En définissant $\bar{A} = \sqrt{\alpha} A$ et $\bar{E} = \sqrt{\alpha} E$,

$$P = Q - \bar{A}^T P \bar{E} [\bar{E}^T P \bar{E} + R]^{-1} \bar{E}^T P \bar{A} + \bar{A}^T P \bar{A} \quad (\text{F.36})$$

L'équation (F.36) correspond à l'équation classique de Riccati appliquée à la paire (\bar{A}, \bar{E}) . ■

Exemple de POMDP à deux dimensions

Sommaire

G.1 Paramètre du problème	217
G.2 Cas avec l'espace de recherche simplifié	218
G.3 Cas avec l'espace de recherche complet	219

Cette annexe illustre la résolution d'un problème à deux dimensions dans un premier temps avec l'espace de recherche simplifié, dans un second temps avec l'espace de recherche complet. Cette illustration permet notamment de mettre en avant la complexité de calcul avec l'espace de recherche complet.

G.1 Paramètre du problème

Pour illustrer cette méthode, le système est simplifié à deux modes et à un horizon $H = 2$. En deux dimensions, les hyperplans α_k^a sont des droites ce qui est facile à illustrer. Supposons que l'on a :

- deux modes, un mode nominal m_1 et un mode dégradé m_2 où l'élevon droit est en panne ;
- l'état de croyance $\boldsymbol{\rho} = [\rho(m_1) \quad \rho(m_2)]^T$ avec $\rho(m_2) = 1 - \rho(m_1)$;
- deux actions, a_1 qui sélectionne le contrôleur nominal c_1 et a_2 qui sélectionne le contrôleur pour le mode dégradé c_2 ;
- quatre coût, $R_{k'}(m_1, a_1)$ associé à l'état $\{m_1, c_1\}$, $R_{k'}(m_1, a_2)$ associé à l'état $\{m_1, c_2\}$, $R_{k'}(m_2, a_1)$ associé à l'état $\{m_2, c_1\}$ et $R_{k'}(m_2, a_2)$ associé à l'état $\{m_2, c_2\}$. Les valeurs des récompenses utilisées sont celles du Tableau G.1.

Indice	k	$k + 1$	$k + 2$
$R_{k'}(m_1, a_1)$	0.15	0.1	0.05
$R_{k'}(m_1, a_2)$	0.25	0.2	0.15
$R_{k'}(m_2, a_1)$	1.1	1.05	1
$R_{k'}(m_2, a_2)$	0.3	0.25	0.2

Tableau G.1 – Valeurs des récompenses pour les indices allant de k à $k + 2$

La pénalité de changer de contrôleur vaut $P(c_1, a_2) = P(c_2, a_1) = 0.1$. Le facteur

de réduction vaut $\gamma = 0.75$. La matrice de transition entre les modes est la suivante :

$$T_m = \begin{bmatrix} 1 - e^{-8} & e^{-9} \\ e^{-8} & 1 - e^{-9} \end{bmatrix}$$

A l'instant k , le contrôleur courant est le contrôleur c_1 .

G.2 Cas avec l'espace de recherche simplifié

L'espace de recherche simplifié est associé à l'équation 7.39.

— Initialisation à $k + 2$: $V_{k+2}^a(\boldsymbol{\rho}) = \sum_m R_{k+2}(m, a)\rho(m)$

Les droites $\alpha_{k+2}^a(m)$ sont donc égales à $R_{k+2}(m, a)$. La Figure G.1 représente les deux fonctions de valeurs $V_{k+2}^{a_1}$ et $V_{k+2}^{a_2}$. Les deux droites se coupent en $\rho(m_1) = 0.8889$ avec une valeur de 0.1556.

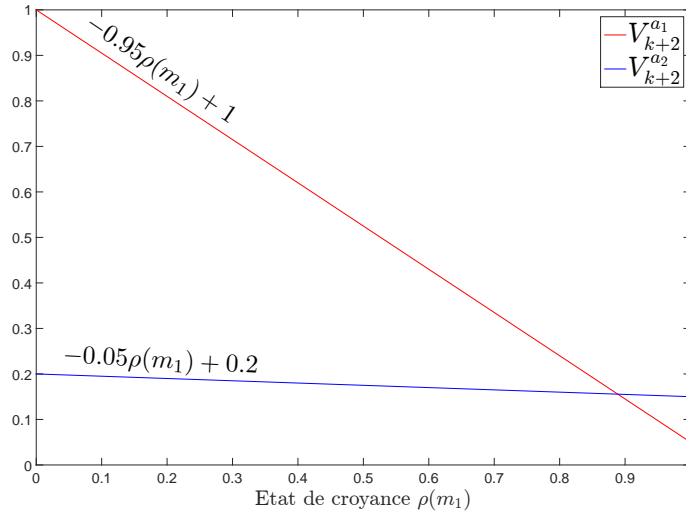


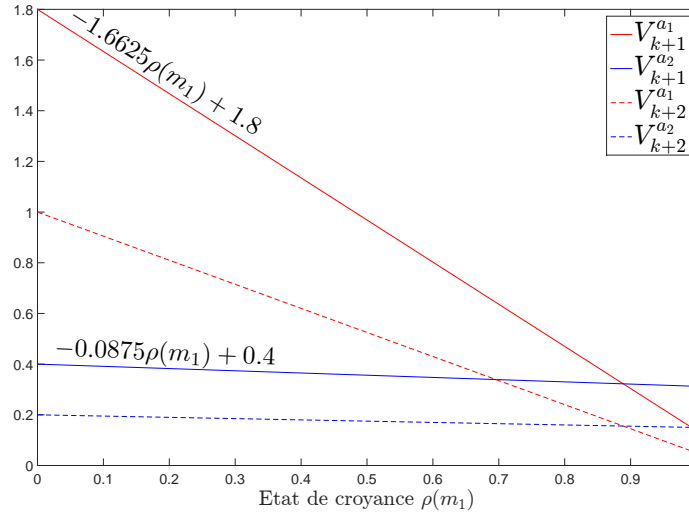
FIGURE G.1 – Fonctions de valeur $V_{k+2}^{a_1}$ et $V_{k+2}^{a_2}$

— Itération à $k + 1$: utiliser l'équation récursif 7.39 en remplaçant $\alpha_{k+2}^a(m)$ par son expression : $V_{k+1}^a(\boldsymbol{\rho}) = \sum_m R_{k+1}(m, a)\rho(m) + \gamma \sum_m p(m|\boldsymbol{\rho})R_{k+2}(m, a)$

En rassemblant les termes en $\rho(m_1)$ et les termes en $\rho(m_2)$, on trouve l'expression pour les droites $\alpha_{k+1}^a(m)$: $\alpha_{k+1}^a(m) = R_{k+1}(m, a) + \gamma \sum_{m'} R_{k+2}(m', a)p(m'|m)$. La Figure G.2 représente les deux fonctions de valeurs $V_{k+1}^{a_1}$ et $V_{k+1}^{a_2}$. Les deux droites se coupent en $\rho(m_1) = 0.8889$, comme pour $k + 2$, avec une valeur de 0.3222.

— Itération à k : utiliser l'équation récursif 7.39 en remplaçant $\alpha_{k+1}^a(m)$ par son expression :

$$\begin{aligned} V_k^a(\boldsymbol{\rho}) &= \sum_m [R_k(m, a) + P(c_1, a)]\rho(m) \\ &+ \gamma \sum_m p(m|\boldsymbol{\rho}) [R_{k+1}(m, a) + \gamma \sum_{m'} R_{k+2}(m', a)p(m'|m)] \end{aligned}$$

FIGURE G.2 – Fonctions de valeur $V_{k+1}^{a_1}$ et $V_{k+1}^{a_2}$

En rassemblant les termes en $\rho(m_1)$ et les termes en $\rho(m_2)$, on trouve les expressions suivantes pour les droites $\alpha_k^{a_1}(m)$ et $\alpha_k^{a_2}(m)$:

$$\begin{aligned}\alpha_k^{a_1}(m) &= R_k(m, a_1) + \gamma \sum_{m'} R_{k+1}(m', a_1) p(m'|m) \\ &\quad + \gamma^2 \sum_{m''} \sum_{m'} R_{k+2}(m', a_1) p(m'|m'') p(m''|m) \\ \alpha_k^{a_2}(m) &= R_k(m, a_1) + P(c_1, a_2) + \gamma \sum_{m'} R_{k+1}(m', a_1) p(m'|m) \\ &\quad + \gamma^2 \sum_{m''} \sum_{m'} R_{k+2}(m', a_1) p(m'|m'') p(m''|m)\end{aligned}$$

La Figure G.3 représente les deux fonctions de valeurs $V_k^{a_1}$ et $V_k^{a_2}$. Les deux droites se coupent en $\rho(m_1) = 0.8408$ avec une valeur de 0.6028.

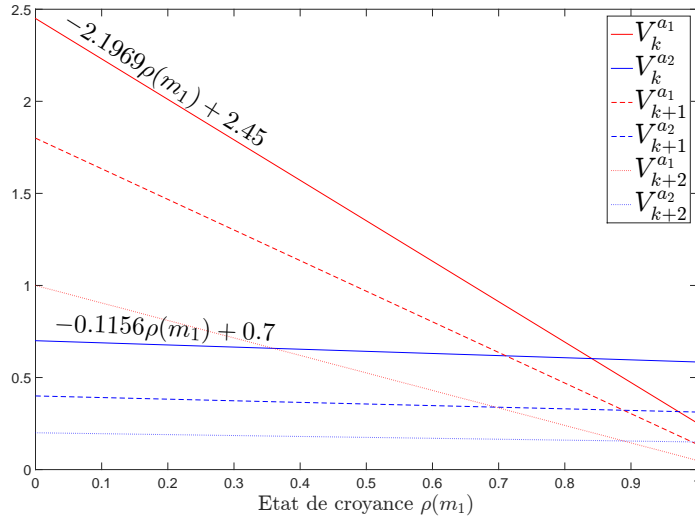
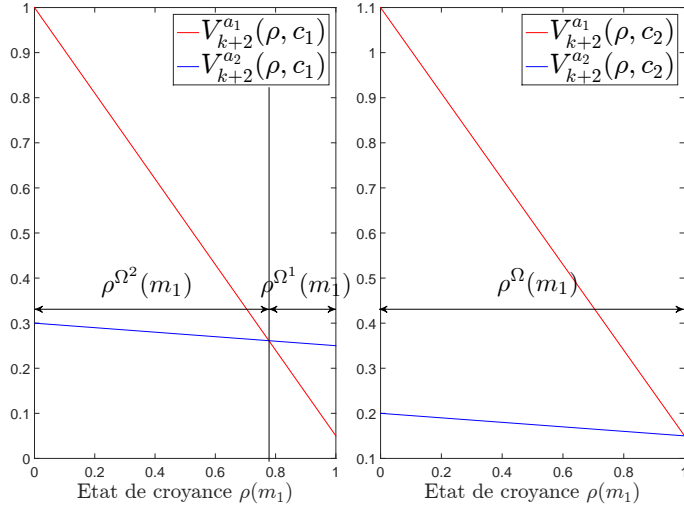
Ainsi, si $\rho(m_1)$ est supérieur à 0.8408 l'action a_1 sera choisie, sinon l'action a_2 est choisie.

G.3 Cas avec l'espace de recherche complet

L'espace de recherche complet est associé à l'équation 7.33.

— Initialisation à $k+2$: $V_{k+2}(\rho, c) = \min_{a \in A} \sum_m [R_{k+2}(m, a) + P(c, a)] \rho(m)$

Deux choix sont alors possibles : $c = c_1$ et $c = c_2$, ces deux choix sont montrés sur la Figure G.4 :

FIGURE G.3 – Fonctions de valeur V_k^{a1} et V_k^{a2} FIGURE G.4 – Fonctions de valeur $V_{k+2}^a(c)$

Les expressions de $V_{k+2}^a(\rho, c)$ sont les suivantes :

$$\begin{aligned}
 V_{k+2}^{a1}(\rho, c_1) &= \sum_m R_{k+2}(m, a_1)\rho(m) \\
 V_{k+2}^{a2}(\rho, c_1) &= \sum_m [R_{k+2}(m, a_2) + P(c_1, a_2)]\rho(m) \\
 V_{k+2}^{a1}(\rho, c_2) &= \sum_m [R_{k+2}(m, a_1) + P(c_2, a_1)]\rho(m) \\
 V_{k+2}^{a2}(\rho, c_2) &= \sum_m [R_{k+2}(m, a_2)]\rho(m)
 \end{aligned}$$

— Itération à $k + 1$: $V_{k+1}^a(\boldsymbol{\rho}, c) = r_{k+1}(\boldsymbol{\rho}, c, a) + \gamma \sum_{i=1}^{|\Omega|} p(\Omega^i | \boldsymbol{\rho}) V_{k+2}(\boldsymbol{\rho}^{\Omega^i}, a)$

Il est possible de partitionner l'espace d'observation de façon à rassembler les observations conduisant à la même action. Ici, il y a deux actions donc deux espaces d'observations Ω_1 et Ω_2 sont construits.

Pour l'action a_1 , il faut utiliser sur les fonctions de la Figure de gauche de G.4 associées au contrôleur c_1 , $V_{k+2}^{a_1}(\boldsymbol{\rho}, c_1)$ qui minimise le coût pour l'observation Ω^1 et $V_{k+2}^{a_2}(\boldsymbol{\rho}, c_1)$ qui minimise le coût pour l'observation Ω^2 et les remplacer par leur expression :

$$\begin{aligned} V_{k+1}^{a_1}(\boldsymbol{\rho}, c) &= r_{k+1}(\boldsymbol{\rho}, c, a_1) + \gamma [p(\Omega^1 | \boldsymbol{\rho}) \sum_m R_{k+2}(m, a_1) \rho^{\Omega^1}(m) \\ &\quad + p(\Omega^2 | \boldsymbol{\rho}) \sum_m [R_{k+2}(m, a_2) + P(a_2, c_1)] \rho^{\Omega^2}(m)] \end{aligned} \quad (\text{G.1})$$

Sachant que $\rho^{\Omega^i}(m) = \frac{p(\Omega^i | m) p(m | \boldsymbol{\rho})}{p(\Omega^i | \boldsymbol{\rho})}$,

$$\begin{aligned} V_{k+1}^{a_1}(\boldsymbol{\rho}, c) &= r_{k+1}(\boldsymbol{\rho}, c, a_1) + \gamma [\sum_m R_{k+2}(m, a_1) p(\Omega^1 | m) p(m | \boldsymbol{\rho}) \\ &\quad + \sum_m [R_{k+2}(m, a_2) + P(a_2, c_1)] p(\Omega^2 | m) p(m | \boldsymbol{\rho})] \end{aligned} \quad (\text{G.2})$$

Pour l'action a_2 , il faut utiliser sur la fonction de la Figure de droite de G.4 associées au contrôleur c_2 , $V_{k+2}^{a_2}(\boldsymbol{\rho}, c_2)$ qui minimise le coût pour les observations Ω^1 et Ω^2 et la remplacer par son expression :

$$\begin{aligned} V_{k+1}^{a_2}(\boldsymbol{\rho}, c) &= r_{k+1}(\boldsymbol{\rho}, c, a_1) + \gamma [p(\Omega^1 | \boldsymbol{\rho}) \sum_m [R_{k+2}(m, a_2)] \rho^{\Omega^1}(m) \\ &\quad + p(\Omega^2 | \boldsymbol{\rho}) \sum_m [R_{k+2}(m, a_2)] \rho^{\Omega^2}(m)] \end{aligned} \quad (\text{G.3})$$

Sachant que $\rho^{\Omega^i}(m) = \frac{p(\Omega^i | m) p(m | \boldsymbol{\rho})}{p(\Omega^i | \boldsymbol{\rho})}$ et $p(\Omega^1 | m) + p(\Omega^2 | m) = 1$,

$$V_{k+1}^{a_2}(\boldsymbol{\rho}, c) = r_{k+1}(\boldsymbol{\rho}, c, a_1) + \gamma [\sum_m [R_{k+2}(m, a_2)] p(m | \boldsymbol{\rho})] \quad (\text{G.4})$$

On se retrouve de nouveau dans un cas où l'on a deux choix, c_1 et c_2 . Il faut procéder de la même manière pour avoir $V_k^{a_1}(\boldsymbol{\rho}, c)$ $V_k^{a_2}(\boldsymbol{\rho}, c)$. A l'instant k , un seul choix est possible, le contrôleur courant qui est dans cette exemple le contrôleur c_1 .

Comparaison entre le filtre TSKF avec et sans contrainte

Sommaire

H.1 Pannes moteur	223
H.2 Pannes élevons	228

Les résultats montrés lors de cette annexe proviennent de simulations réalisées avec le filtre adaptatif TSKF prenant en compte les contraintes sur \mathbf{b}_1^1 et le filtre adaptatif TSKF sans prise en compte les contraintes. Pour plus de visibilité sur les courbes, la comparaison entre le filtre contraint et le filtre non contraint est réalisée pour 100 simulations pour chaque type de panne.

Le scénario suivant est utilisé : lors d'un atterrissage une panne permanente survient sur un actionneur à 10s ; la simulation se termine au bout de 100s. La panne peut être un biais, une perte d'efficacité ou un blocage. Les figures suivantes montrent l'erreur RMSE calculé après l'apparition de la faute, en régime établi, c'est-à-dire une seconde après l'apparition de la faute, de 11s à 100s.

H.1 Pannes moteur

La Figure H.1 montre les erreurs d'estimation des paramètres de fautes associés au moteur, c'est-à-dire b_0^1 et b_1^1 , en cas de perte d'efficacité au niveau du moteur.

Les erreurs d'estimation diminuent de façon exponentielle lorsque la perte d'efficacité augmente. Une moins bonne précision dans des petites pertes d'efficacité est néanmoins moins grave pour le comportement du drone. Alors qu'une bonne précision en cas de fortes pertes d'efficacité est importante pour un recouvrement total du comportement du drone. Pour des pertes d'efficacité supérieure à 30%, les erreurs sont très faibles ; inférieur à 0.01 pour b_0^1 et inférieur à 0.02 pour b_1^1 . Globalement, le filtre est plus précis avec les contraintes sur \mathbf{b}_1 , notamment pour des pertes d'efficacité inférieur à 30%.

Concernant la détection des pannes, il y a plusieurs cas. Pour les 12 simulations ayant une très faible perte d'efficacité, c'est-à-dire inférieure ou égale à 10.2%, on observe différents résultats. Pour le filtre avec contrainte, il y a :

- une *MD* (pour une perte d'efficacité de 0.51%) pour laquelle il n'y a pas eu de réouverture de la covariance ;

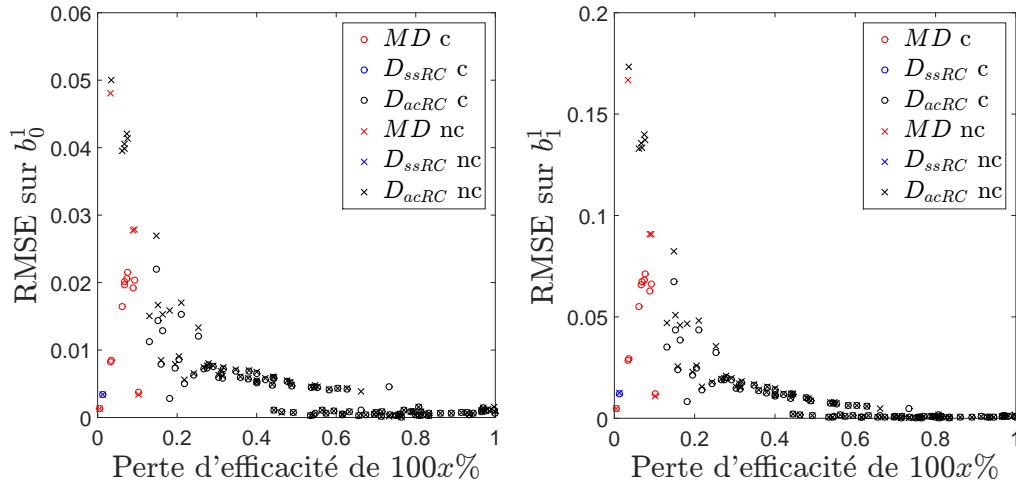


FIGURE H.1 – Erreur d'estimation en cas de perte d'efficacité moteur sur 100 simulations

- une détection sans réouverture de la covariance (pour une perte d'efficacité de 1.32%) ;
- dix *MD* (pour des pertes d'efficacité entre 3.3% et 10.2%) pour laquelle il y a eu une réouverture de la covariance.

Dans le premier cas, la *MD* est dû à une trop faible valeur de la perte d'efficacité. Dans le deuxième cas, la détection se fait grâce à une valeur légèrement plus élevée et une covariance faible. Dans le troisième cas, les valeurs des pertes d'efficacité restent faibles mais il y a une réouverture de la covariance, entraînant un manque de confiance en la valeur estimée et par conséquent des non-détections.

Pour le filtre non contraint, il y a :

- une *MD* (pour une perte d'efficacité de 0.51%) pour laquelle il n'y a pas eu de réouverture de la covariance ;
- une détection sans réouverture de la covariance (pour une perte d'efficacité de 1.32%) ;
- cinq *MD* (pour des pertes d'efficacité entre 3.3% et 10.2%) pour laquelle il y a eu une réouverture de la covariance.
- six détections tardives, $T_D = [73.56 \ 50.9 \ 41.4 \ 40.04 \ 24.32 \ 24.14]$, (pour des pertes d'efficacité entre 3.5% et 7.6%).

Contrairement au filtre contraint, le filtre non contraint réussit à détecter six pannes supplémentaires. Néanmoins, les détections tardives ne sont pas non plus satisfaisantes.

Les 88 autres simulations correspondent à des détections rapides grâce à la réouverture de la covariance des paramètres de fautes associés au moteur. Les temps de détection de ces simulations sont montrés sur la Figure H.2.

Le temps de détection est plus rapide pour des fortes pertes d'efficacité. Pour une perte d'efficacité supérieure à 30%, le temps de détection est inférieur à 0.5s

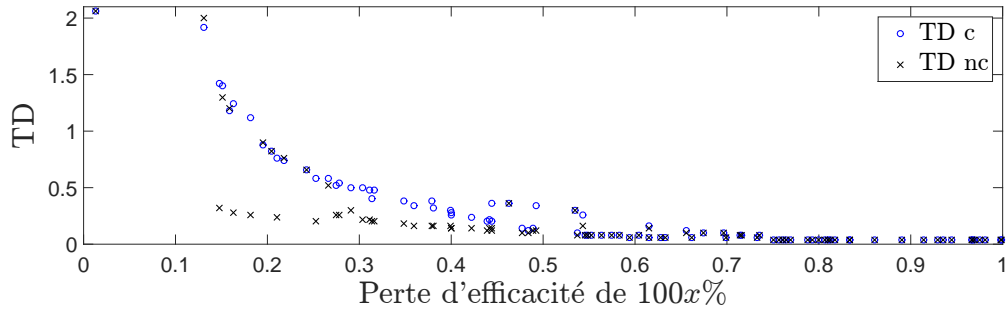


FIGURE H.2 – Temps de détection en cas de perte d’efficacité moteur sur 89 simulations (1 sans réouverture de la covariance et 88 avec réouverture de la covariance)

pour le filtre contraint et inférieur à 0.22s pour le filtre non contraint. Pour une perte d’efficacité supérieure à 70%, le temps de détection est inférieur à 0.1s pour le filtre contraint et le filtre non contraint.

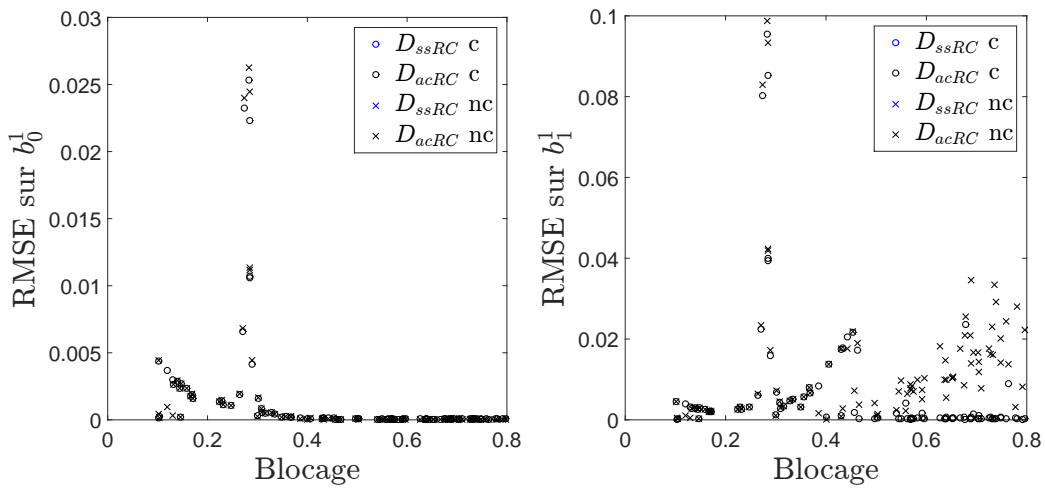


FIGURE H.3 – Erreur d’estimation en cas de blocage moteur sur 99 simulations

La Figure H.3 montre les erreurs d’estimation des paramètres de fautes associés au moteur en cas de blocage moteur pour uniquement 99 simulations. En effet, une simulation parmi les 100 a une valeur de blocage très proche de celle de l’équilibre, ainsi l’estimation est plus difficile pour ce cas. Pour des questions de visibilité, elle a été enlevée des Figures H.3 et H.4, néanmoins le déroulement de la simulation est expliquée. Lors de cette simulation, l’erreur au niveau des résidus n’est pas suffisamment importante et par conséquent la covariance n’est pas réouverte. Néanmoins, la valeur de l’estimation b_0^1 va légèrement dériver de zéro en gardant une covariance petite, d’où la détection de la panne à 7.44s. La panne est détectée mais les valeurs des paramètres de fautes ne sont pas les bonnes. L’erreur $RMSE$ sur b_0^1 est de 0.29 et l’erreur sur b_1^1 est de 0.995. Pour cette panne, le moteur est bloqué proche de sa

valeur d'équilibre, par conséquent cette panne n'a pas d'influence sur le comportement du drone, ce qui la rend très peu observable, d'où une mauvaise estimation pour les paramètres de fautes.

Les 99 autres simulations correspondent à des détections avec réouverture de la covariance. Ainsi, il n'y a pas de MD que ce soit avec ou sans les contraintes sur \mathbf{b}_1 . Les erreurs d'estimation sont plus importantes autour de la valeur d'équilibre mais restent plutôt faibles. Lorsque l'on s'éloigne de la valeur d'équilibre, les erreurs deviennent très faibles, inférieures à 0.005 pour b_0^1 et inférieures à 0.04 pour b_1^1 . Globalement, le filtre est plus précis avec les contraintes sur \mathbf{b}_1 , notamment sur b_1^1 pour des blocages supérieurs à 0.6.

La Figure H.4 montre le temps de détection en cas de blocage moteur.

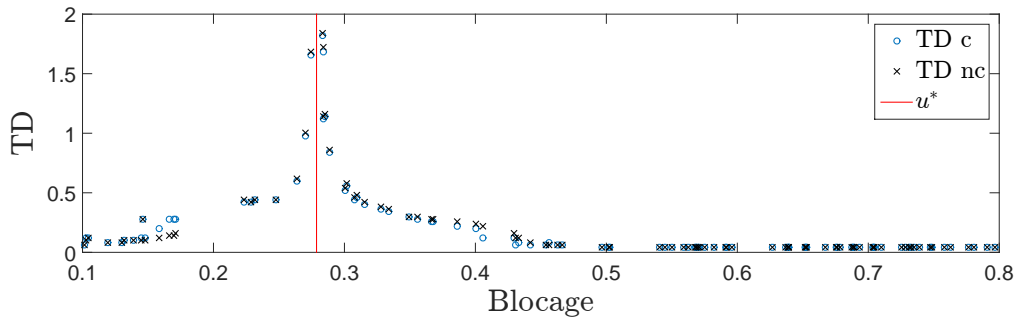


FIGURE H.4 – Temps de détection en cas de blocage moteur sur 99 simulations

Le temps de détection est plus important autour de la valeur d'équilibre mais reste inférieur à 2s. De plus, le temps de détection est similaire, que ce soit avec le filtre contraint ou non contraint.

La Figure H.5 montre les erreurs d'estimation des paramètres de fautes associés au moteur en cas de biais sur le moteur.

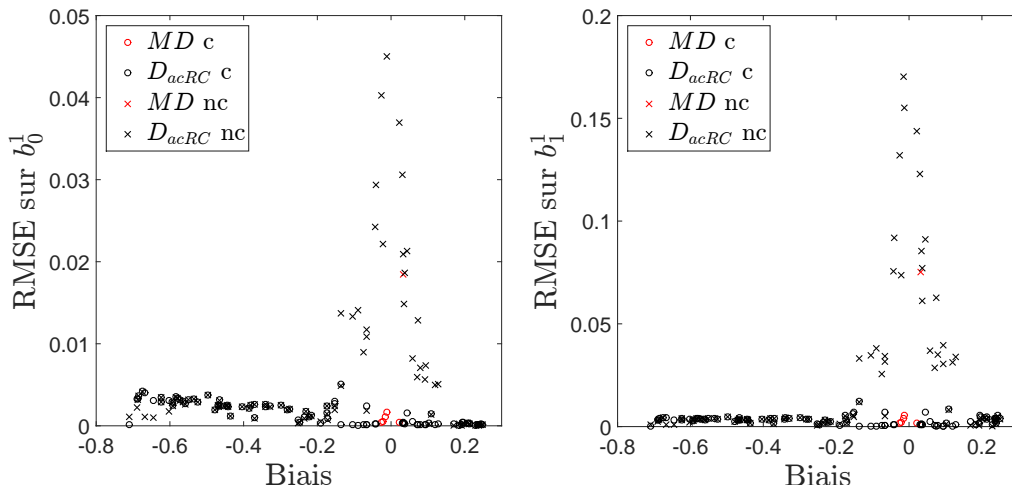


FIGURE H.5 – Erreur d'estimation en cas de biais moteur sur 100 simulations

Pour le filtre prenant en compte les contraintes, il y a :

- cinq MD (pour des biais compris entre -0.0265 et 0.0218) pour lesquelles il y a eu une réouverture de la covariance ;

Il n'y a pas de détection car les valeurs des biais sont trop faibles et la covariance est élevée.

Pour le filtre ne prenant pas en compte les contraintes, il y a :

- une MD (pour un biais de 0.0324) pour laquelle il y a eu une réouverture de la covariance ;
- quatre détections tardives $TD = [63.16 \quad 47.74 \quad 27.16 \quad 10.46]$, pour lesquelles il y a eu une réouverture de la covariance (pour des biais entre -0.012% et 0.033%).

Ces 5 simulations correspondent à des biais très faibles, et qui sont par conséquent moins dramatiques pour le comportement du drone.

Les 95 autres simulations correspondent à des détections avec réouverture de la covariance. Les erreurs d'estimation sont plus importantes lorsque le biais est proche de zéro. Pour un biais inférieur à -0.2 , les erreurs deviennent très faibles, inférieures à 0.005 pour b_0^1 et b_1^1 . Globalement, le filtre est plus précis avec les contraintes sur \mathbf{b}_1 , notamment pour des biais faibles. Avec le filtre contraint, les erreurs sont inférieures à 0.006 pour b_0^1 et inférieures à 0.013 pour b_1^1 .

La Figure H.6 montre le temps de détection en cas de biais moteur.

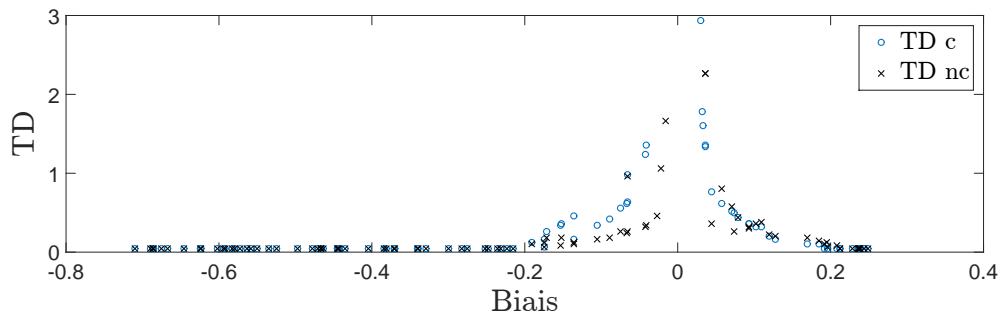


FIGURE H.6 – Temps de détection en cas de biais moteur pour 95 simulations

Le temps de détection est plus important autour de zéro. Le temps de détection du filtre sans contraint est légèrement plus rapide pour des biais compris entre -0.2 et 0 . Si le biais est supérieur à ± 0.2 , le temps de détection est inférieur à $0.1s$.

Remarque H.1

Les erreurs d'estimation des paramètres de fautes associés aux élevons, c'est-à-dire b_0^2 , b_0^3 , b_1^2 , b_1^3 , b_1^4 et b_1^5 , restent faibles en cas de panne moteur, de l'ordre de 10^{-5} , c'est pourquoi leurs erreurs ne sont pas tracées ici. Ces erreurs restent faibles car il n'y a pas de réouverture de la covariance de ces paramètres lors de panne moteur.

H.2 Pannes élevons

La Figure H.7 montre les erreurs d'estimation des paramètres de fautes associés au moteur en cas de perte d'efficacité de l'élevon 1.

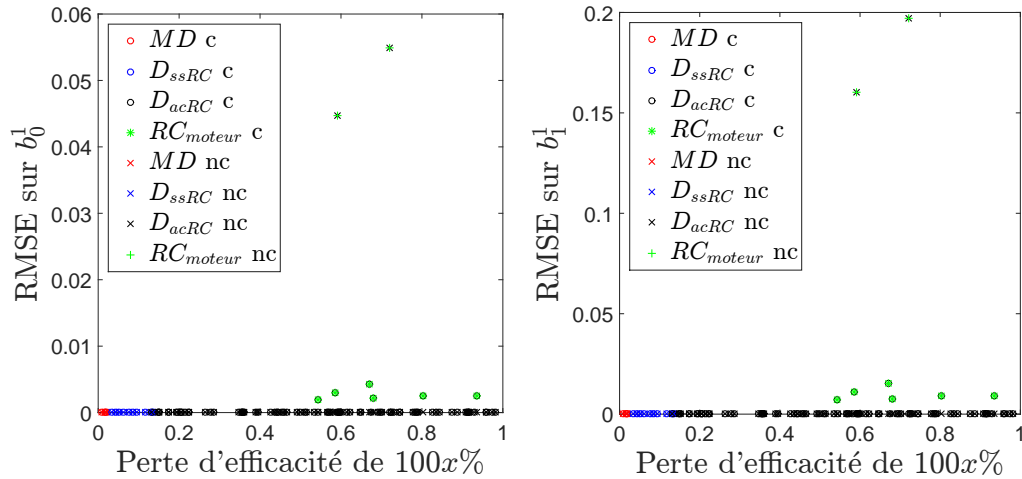


FIGURE H.7 – Erreur d'estimation en cas de perte d'efficacité élévon 1 sur 100 simulations

Avec le filtre contraint, il y a 6 réouvertures inappropriées. Les erreurs d'estimation des paramètres de fautes associés au moteur sont alors plus importantes mais elles restent assez faibles, inférieures à 0.005 pour b_0^1 et inférieures à 0.016 pour b_1^1 . Avec le filtre non contraint, il y a seulement 2 réouvertures inappropriées, cependant les erreurs d'estimation deviennent beaucoup plus importantes, inférieures à 0.06 pour b_0^1 et inférieures à 0.2 pour b_1^1 .

La Figure H.8 montre les erreurs d'estimation des paramètres de fautes associés au moteur en cas de biais sur l'élevon 1.

Avec le filtre contraint, il y a 36 réouvertures inappropriées. Les erreurs d'estimation des paramètres de fautes associés au moteur sont alors plus importantes mais elles restent assez faibles, inférieures à 0.004 pour b_0^1 et inférieures à 0.011 pour b_1^1 . Avec le filtre non contraint, il y a 33 réouvertures inappropriées et les erreurs d'estimation deviennent beaucoup plus importantes, inférieures à 0.07 pour b_0^1 et inférieures à 0.24 pour b_1^1 .

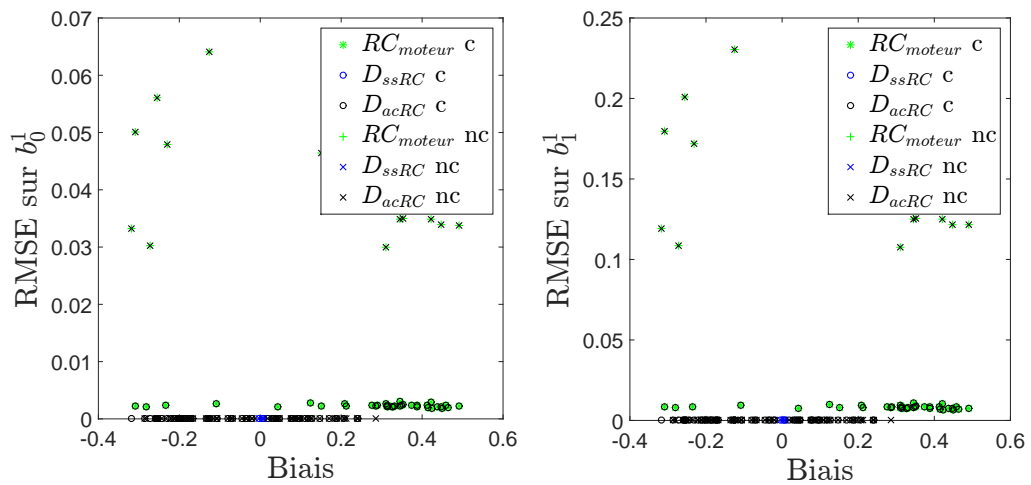


FIGURE H.8 – Erreur d'estimation en cas de biais élevon 1 sur 100 simulations

Publications

Article de revue scientifique

- Boche, A., Farges, J-L. et De Plinval, H. Reconfiguration control method for non-redundant actuator faults on UAV. Soumis à *Journal of Aerospace Engineering*, 2018.

Articles de conférences internationales

- Boche, A., Farges, J-L. et De Plinval, H. A continuous and discrete framework for reconfiguration of control of faulty systems. Dans *Computational Intelligence (SSCI)*, Athènes, Grèce, 2016.
- Boche, A., De Plinval, H. et Farges, J-L. Reconfiguration Control Method for Faulty Actuator on UAV. Dans *Advances in Aerospace Guidance, Navigation and Control*, pages 3-23, Varsovie, Pologne, 2018.
- Boche, A., Farges, J-L. et De Plinval, H. Reconfiguration control method for multiple actuator faults on UAV. Dans *IFAC-PapersOnLine*, Toulouse, France, 2017.
- Boche, A., Farges, J-L. et De Plinval, H. Reconfiguration control method for non-redundant actuator faults on UAV. Présenté à EUCASS 2017, Milan, Italie, 2017.

Bibliographie

- [Aldon 2001] Marie-José Aldon. *Capteurs et méthodes pour la localisation des robots mobiles*. Techniques de l'ingénieur. Informatique industrielle, vol. 6, no. S7852, pages 1–19, 2001. (Cité en page 21.)
- [Alouani 1993] AT Alouani, P Xia, TR Rice et WD Blair. *On the optimality of two-stage state estimation in the presence of random bias*. IEEE Transactions on Automatic Control, vol. 38, no. 8, pages 1279–1283, 1993. (Cité en page 50.)
- [Athans 1976] Michael Athans et Chaw-Bing Chang. *Adaptive estimation and parameter identification using multiple model estimation algorithm*. Rapport technique, Massachusetts inst of tech Lexington Lincoln Lab, 1976. (Cité en page 53.)
- [Basseville 1993] Michèle Basseville, Igor V Nikiforov *et al.* *Detection of abrupt changes : theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993. (Cité en pages 56 et 57.)
- [Bayouhdh 2008] Mehdi Bayouhdh, Louise Travé-Massuyes et Xavier Olive. *Hybrid systems diagnosis by coupling continuous and discrete event techniques*. IFAC Proceedings Volumes, vol. 41, no. 2, pages 7265–7270, 2008. (Cité en page 60.)
- [Bernardini 2009] Daniele Bernardini et Alberto Bemporad. *Scenario-based model predictive control of stochastic constrained linear systems*. In Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, pages 6333–6338. IEEE, 2009. (Cité en page 65.)
- [Bertsekas 1995] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas et Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995. (Cité en page 211.)
- [Blom 1988] Henk AP Blom et Yaakov Bar-Shalom. *The interacting multiple model algorithm for systems with Markovian switching coefficients*. IEEE transactions on Automatic Control, vol. 33, no. 8, pages 780–783, 1988. (Cité en page 60.)
- [Blom 2007] Henk AP Blom et Edwin A Bloem. *Exact Bayesian and particle filtering of stochastic hybrid systems*. IEEE Transactions on Aerospace and Electronic Systems, vol. 43, no. 1, 2007. (Cité en page 60.)
- [Boche 2016] Adèle Boche, Jean-Loup Farges et Henry De Plinval. *A continuous and discrete framework for reconfiguration of control of faulty systems*. In Computational Intelligence (SSCI), 2016 IEEE Symposium Series on, pages 1–8. IEEE, 2016. (Cité en page 86.)
- [Boche 2017a] A Boche, JL Farges et H De Plinval. *Reconfiguration control method for multiple actuator faults on UAV*. IFAC-PapersOnLine, vol. 50, no. 1, pages 12691–12697, 2017. (Cité en pages 144 et 183.)

- [Boche 2017b] A Boche, JL Farges et H De Plinval. *Reconfiguration control method for non redundant actuator faults on UAV*. EUCASS, 2017. (Cit  en pages 144 et 183.)
- [Boche 2018] Ad le Boche, Henry De Plinval et Jean-Loup Farges. *Reconfiguration Control Method for Faulty Actuator on UAV*. In Advances in Aerospace Guidance, Navigation and Control, pages 3–23. Springer, 2018. (Cit  en pages 144 et 183.)
- [Boers 2003] Y Boers et JN Driessen. *Interacting multiple model particle filter*. IEE Proceedings-Radar, Sonar and Navigation, vol. 150, no. 5, pages 344–349, 2003. (Cit  en page 60.)
- [Boskovic 2001] JD Boskovic et RK Mehra. *A hybrid fault-tolerant scheme for flight control applications*. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, August AIAA, pages 2001–4400, 2001. (Cit  en pages 65 et 66.)
- [Branicky 1998] Michael S Branicky. *Multiple Lyapunov functions and other analysis tools for switched and hybrid systems*. IEEE Transactions on automatic control, vol. 43, no. 4, pages 475–482, 1998. (Cit  en pages 42 et 187.)
- [Caliskan 2016] Fikret Caliskan et Chingiz Hajiyev. *Active fault-tolerant control of UAV dynamics against sensor-actuator failures*. Journal of Aerospace Engineering, vol. 29, no. 4, page 04016012, 2016. (Cit  en pages 40, 52 et 61.)
- [Chanthery 2013] Elodie Chanthery et Pauline Ribot. *An integrated framework for diagnosis and prognosis of hybrid systems*. arXiv preprint arXiv :1308.5332, 2013. (Cit  en pages 42 et 60.)
- [Chen 1995] Chi-Tsong Chen. Linear system theory and design. Oxford University Press, Inc., 1995. (Cit  en page 195.)
- [Chowdhary 2010] Girish Chowdhary et Ravindra Jategaonkar. *Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter*. Aerospace science and technology, vol. 14, no. 2, pages 106–117, 2010. (Cit  en page 208.)
- [Cocquempot 2004] Vincent Cocquempot, Touria El Meznyani et Marcel Staroswiecki. *Fault detection and isolation for hybrid systems using structured parity residuals*. In Control conference, 2004. 5th asian, volume 2, pages 1204–1212. IEEE, 2004. (Cit  en page 60.)
- [Cordier 2004] M-O Cordier, Philippe Dague, Fran ois L vy, Jacky Montmain, Marcel Staroswiecki et Louise Trav -Massuy s. *Conflicts versus analytical redundancy relations : a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 5, pages 2163–2177, 2004. (Cit  en pages 43 et 46.)

- [Doucet 2001] Arnaud Doucet, Neil J Gordon et Vikram Krishnamurthy. *Particle filters for state estimation of jump Markov linear systems*. IEEE Transactions on signal processing, vol. 49, no. 3, pages 613–624, 2001. (Cité en page 60.)
- [Ducard 2008] Guillaume Ducard et Hans P Geering. *Efficient nonlinear actuator fault detection and isolation system for unmanned aerial vehicles*. Journal of Guidance Control and Dynamics, vol. 31, no. 1, pages 225–237, 2008. (Cité en pages 53 et 62.)
- [Ducard 2009] Guillaume JJ Ducard. *Fault-tolerant flight control and guidance systems : Practical methods for small unmanned aerial vehicles*. Springer Science & Business Media, 2009. (Cité en pages 15, 16, 17, 18, 36 et 62.)
- [Freeman 2014] Paul Michael Freeman. *Reliability assessment for low-cost unmanned aerial vehicles*. PhD thesis, University of Minnesota, 2014. (Cité en pages 16 et 18.)
- [Friedland 1969] Bernard Friedland. *Treatment of bias in recursive filtering*. IEEE Transactions on Automatic Control, vol. 14, no. 4, pages 359–367, 1969. (Cité en page 50.)
- [Gao 1991] Zhiqiang Gao et Panos J Antsaklis. *Stability of the pseudo-inverse method for reconfigurable control systems*. International Journal of Control, vol. 53, no. 3, pages 717–729, 1991. (Cité en page 64.)
- [Gao 2015a] Zhiwei Gao, Carlo Cecati et Steven X Ding. *A survey of fault diagnosis and fault-tolerant techniques-Part I : Fault diagnosis with model-based and signal-based approaches*. IEEE Transactions on Industrial Electronics, vol. 62, no. 6, pages 3757–3767, 2015. (Cité en page 44.)
- [Gao 2015b] Zhiwei Gao, Carlo Cecati et Steven X Ding. *A survey of fault diagnosis and fault-tolerant techniques-Part II : Fault Diagnosis With Knowledge-Based and Hybrid Active Approaches*. IEEE Transactions on Industrial Electronics, vol. 62, no. 6, pages 3768–3774, 2015. (Cité en page 44.)
- [Geng 2008] Yanrui Geng et Jinling Wang. *Adaptive estimation of multiple fading factors in Kalman filter for navigation applications*. GPS Solutions, vol. 12, no. 4, pages 273–279, 2008. (Cité en page 54.)
- [Gillijns 2007] Steven Gillijns et Bart De Moor. *Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough*. Automatica, vol. 43, no. 5, pages 934–937, 2007. (Cité en page 54.)
- [Gordon 1993] Neil J Gordon, David J Salmond et Adrian FM Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*. In IEE Proceedings F (Radar and Signal Processing), volume 140, pages 107–113. IET, 1993. (Cité en page 47.)
- [Goshen-Meskin 1992] DRORA Goshen-Meskin et IY Bar-Itzhack. *Observability analysis of piece-wise constant systems. I. Theory*. IEEE Transactions on Aerospace and Electronic systems, vol. 28, no. 4, pages 1056–1067, 1992. (Cité en page 199.)

- [Goupil 2010] Philippe Goupil. *Oscillatory failure case detection in the A380 electrical flight control system by analytical redundancy*. Control Engineering Practice, vol. 18, no. 9, pages 1110–1119, 2010. (Cité en pages 16 et 28.)
- [Goupil 2011] Philippe Goupil. *AIRBUS state of the art and practices on FDI and FTC in flight control system*. Control Engineering Practice, vol. 19, no. 6, pages 524–539, 2011. (Cité en page 19.)
- [Gustafsson 2000] Fredrik Gustafsson. Adaptive filtering and change detection, volume 1. Wiley New York, 2000. (Cité en pages 56 et 57.)
- [Hajiyev 2007] Ch Hajiyev. *Adaptive filtration algorithm with the filter gain correction applied to integrated INS/radar altimeter*. Proceedings of the Institution of Mechanical Engineers, Part G : Journal of Aerospace Engineering, vol. 221, no. 5, pages 847–855, 2007. (Cité en page 54.)
- [Hajiyev 2013] Chingiz Hajiyev et Halil Ersin Soken. *Robust adaptive Kalman filter for estimation of UAV dynamics in the presence of sensor/actuator faults*. Aerospace Science and Technology, vol. 28, no. 1, pages 376–383, 2013. (Cité en page 54.)
- [Hayhurst 2007] Kelly J Hayhurst, Jeffrey M Maddalon, Paul S Miner, George N Szatkowski, Michael L Ulrey, Michael P DeWalt et Cary R Spitzer. *Preliminary considerations for classifying hazards of unmanned aircraft systems*. 2007. (Cité en pages 11, 12, 13 et 14.)
- [Henzinger 2000] Thomas A Henzinger. *The theory of hybrid automata*. In Verification of Digital and Hybrid Systems, pages 265–292. Springer, 2000. (Cité en page 42.)
- [Heredia 2005] Guillermo Heredia, Aníbal Ollero, Rajesh Mahtani, Manuel Béjar, Volker Remuß et Marek Musial. *Detection of sensor faults in autonomous helicopters*. In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pages 2229–2234. IEEE, 2005. (Cité en page 17.)
- [Hermann 1977] Robert Hermann et Arthur Krener. *Nonlinear controllability and observability*. IEEE Transactions on automatic control, vol. 22, no. 5, pages 728–740, 1977. (Cité en page 195.)
- [Hoey 2005] Jesse Hoey et Pascal Poupart. *Solving POMDPs with continuous or large discrete observation spaces*. In IJCAI, pages 1332–1338, 2005. (Cité en page 41.)
- [Hsieh 1995] Chien-Shu Hsieh et Fu-Chuang Chen. *Optimal solution of the two-stage Kalman estimator*. In Decision and Control, 1995., Proceedings of the 34th IEEE Conference on, volume 2, pages 1532–1537. IEEE, 1995. (Cité en page 50.)
- [Hu 2003] Congwei Hu, Wu Chen, Yongqi Chen, Dajie Liuet al. *Adaptive Kalman filtering for vehicle navigation*. Journal of Global Positioning Systems, vol. 2, no. 1, pages 42–47, 2003. (Cité en pages 53 et 54.)

- [Hwang 2010] Inseok Hwang, Sungwan Kim, Youdan Kim et Chze Eng Seah. *A survey of fault detection, isolation, and reconfiguration methods*. IEEE Transactions on Control Systems Technology, vol. 18, no. 3, pages 636–653, 2010. (Cité en pages 39, 40, 44 et 55.)
- [Ignagni 1990] Mario B Ignagni. *Separate bias Kalman estimator with bias state noise*. IEEE Transactions on Automatic Control, vol. 35, no. 3, pages 338–341, 1990. (Cité en page 50.)
- [Ingimundarson 2008] Ari Ingimundarson et Ricardo S Sánchez Pena. *Using the unfalsified control concept to achieve fault tolerance*. IFAC Proceedings Volumes, vol. 41, no. 2, pages 1236–1242, 2008. (Cité en page 37.)
- [Jazwinski 2007] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007. (Cité en page 207.)
- [Jiang 2012] Jin Jiang et Xiang Yu. *Fault-tolerant control systems : A comparative study between active and passive approaches*. Annual Reviews in control, vol. 36, no. 1, pages 60–72, 2012. (Cité en pages 33 et 34.)
- [Julier 2004] Simon J Julier et Jeffrey K Uhlmann. *Unscented filtering and nonlinear estimation*. Proceedings of the IEEE, vol. 92, no. 3, pages 401–422, 2004. (Cité en page 208.)
- [Kalman 1960] Rudolph Emil Kalman *et al.* *A new approach to linear filtering and prediction problems*. Journal of basic Engineering, vol. 82, no. 1, pages 35–45, 1960. (Cité en page 48.)
- [Kalman 1961] Rudolph E Kalman et Richard S Bucy. *New results in linear filtering and prediction theory*. Journal of basic engineering, vol. 83, no. 1, pages 95–108, 1961. (Cité en page 48.)
- [Keller 1997] Jean-Yves Keller et Mohamed Darouach. *Optimal two-stage Kalman filter in the presence of random bias*. Automatica, vol. 33, no. 9, pages 1745–1748, 1997. (Cité en pages 48, 50 et 51.)
- [Kim 2003] Ki-Seok Kim, Keum-Jin Lee et Youdan Kim. *Reconfigurable flight control system design using direct adaptive method*. Journal of Guidance Control and Dynamics, vol. 26, no. 4, pages 543–550, 2003. (Cité en pages 36 et 39.)
- [Kim 2009] Kwang Hoon Kim, Jang Gyu Lee et Chan Gook Park. *Adaptive two-stage extended Kalman filter for a fault-tolerant INS-GPS loosely coupled system*. IEEE Transactions on Aerospace and Electronic Systems, vol. 45, no. 1, pages 125–137, 2009. (Cité en pages 51 et 52.)
- [Lesprier 2015a] J Lesprier. *Synthèse et validation de lois de commande de vol robustes en présence d'incertitudes paramétriques et de non-linéarités*. PhD thesis, Université de Toulouse, 2015. (Cité en page 19.)
- [Lesprier 2015b] Jeremy Lesprier, Jean-Marc Biannic et Clément Roos. *Modeling and robust nonlinear control of a fixed-wing UAV*. In Control Applications (CCA), 2015 IEEE Conference on, pages 1334–1339. IEEE, 2015. (Cité en pages 25 et 189.)

- [Lu 2015] P Lu, L Van Eykeren, E Van Kampen, CC De Visser et QP Chu. *Adaptive three-step Kalman filter for air data sensor fault detection and diagnosis*. Journal of Guidance, Control, and Dynamics, no. null, pages 590–604, 2015. (Cité en page 54.)
- [Lundin 2017] Gustav Öman Lundin, Philippe Mouyon et Augustin Manecy. *A GLR algorithm for multiple consecutive measurement bias estimation*. In Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2017 Workshop on, pages 252–257. IEEE, 2017. (Cité en page 187.)
- [Lunze 2008] Jan Lunze et Jan H Richter. *Reconfigurable fault-tolerant control : a tutorial introduction*. European journal of control, vol. 14, no. 5, pages 359–386, 2008. (Cité en page 60.)
- [Mahmoud 2003] Mufeed Mahmoud, Jin Jiang et Youmin Zhang. *Active fault tolerant control systems : stochastic analysis and synthesis*, volume 287. Springer Science & Business Media, 2003. (Cité en page 39.)
- [Maquin 1997] Didier Maquin, Vincent Cocquempot, Jean-Philippe Cassar, Marcel Staroswiecki et José Ragot. *Generation of analytical redundancy relations for FDI purposes*. In IFAC Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, SDEMPED'97, pages 86–93, 1997. (Cité en page 46.)
- [Maybeck 1991] Peter S Maybeck et Richard D Stevens. *Reconfigurable flight control via multiple model adaptive control methods*. IEEE Transactions on Aerospace and Electronic systems, vol. 27, no. 3, pages 470–480, 1991. (Cité en pages 65 et 66.)
- [Maybeck 1999] Peter S Maybeck. *Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems*. International Journal of Robust and Nonlinear Control, vol. 9, no. 14, pages 1051–1070, 1999. (Cité en page 53.)
- [Menke 1995] Timothy E Menke et Peter S Maybeck. *Sensor/actuator failure detection in the Vista F-16 by multiple model adaptive estimation*. IEEE Transactions on aerospace and electronic systems, vol. 31, no. 4, pages 1218–1229, 1995. (Cité en page 53.)
- [Merheb 2013] Abdel-Razzak Merheb, Hassan Noura et Francois Bateman. *Passive fault tolerant control of quadrotor uav using regular and cascaded sliding mode control*. In Control and Fault-Tolerant Systems (SysTol), 2013 Conference on, pages 330–335. IEEE, 2013. (Cité en page 33.)
- [Mignone 2002] Domenico Mignone. *Control and estimation of hybrid systems with mathematical optimization*. PhD thesis, 2002. (Cité en pages 43 et 64.)
- [Page 1954] Ewan S Page. *Continuous inspection schemes*. Biometrika, vol. 41, no. 1/2, pages 100–115, 1954. (Cité en page 56.)
- [Patton 1991] RJ Patton et J Chen. *A review of parity space approaches to fault diagnosis*. IFAC Proceedings Volumes, vol. 24, no. 6, pages 65–81, 1991. (Cité en page 46.)

- [Patton 1997] Ron J Patton. *Fault-tolerant control : the 1997 situation*. IFAC Proceedings Volumes, vol. 30, no. 18, pages 1029–1051, 1997. (Cité en pages 35 et 60.)
- [Puterman 2014] Martin L Puterman. *Markov decision processes : discrete stochastic dynamic programming*. John Wiley & Sons, 2014. (Cité en page 202.)
- [Qi 2013] Xin Qi, Didier Theilliol, Juntong Qi, Youmin Zhang et Jianda Han. *A literature review on fault diagnosis methods for manned and unmanned helicopters*. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 1114–1118. IEEE, 2013. (Cité en page 17.)
- [Raol 2004] Jitendra R Raol, Gopalrathnam Girija et Jatinder Singh. *Modelling and parameter estimation of dynamic systems*, volume 65. Iet, 2004. (Cité en page 55.)
- [Rawlings 2006] James B Rawlings et Bhavik R Bakshi. *Particle filtering and moving horizon estimation*. *Computers & chemical engineering*, vol. 30, no. 10, pages 1529–1541, 2006. (Cité en page 48.)
- [Rhudy 2013] M Rhudy et Y Gu. *Understanding Nonlinear Kalman Filters Part I : Selection of EKF or UKF*. *Interactive Robotics Letters*, 2013. (Cité en page 208.)
- [Robertson 1996] Douglas G Robertson, Jay H Lee et James B Rawlings. *A moving horizon-based approach for least-squares estimation*. *AIChE Journal*, vol. 42, no. 8, pages 2209–2224, 1996. (Cité en page 48.)
- [Rotondo 2014] Damiano Rotondo, Fatiha Nejjari et Vicenç Puig. *A virtual actuator and sensor approach for fault tolerant control of LPV systems*. *Journal of Process Control*, vol. 24, no. 3, pages 203–222, 2014. (Cité en page 62.)
- [Rotondo 2015] Damiano Rotondo, Jean-Christophe Ponsart, Didier Theilliol, Fatiha Nejjari et Vicenç Puig. *A virtual actuator approach for the fault tolerant control of unstable linear systems subject to actuator saturation and fault isolation delay*. *Annual Reviews in Control*, vol. 39, pages 68–80, 2015. (Cité en page 62.)
- [Sadeghzadeh 2011] Iman Sadeghzadeh et Youmin Zhang. *A review on fault-tolerant control for unmanned aerial vehicles (UAVs)*. Infotech@ Aerospace, St. Louis, MO, 2011. (Cité en page 33.)
- [Saied 2017] Majd Saied, Benjamin Lussier, Isabelle Fantoni, Hassan Shraim et Clovis Francis. *Fault Diagnosis and Fault-Tolerant Control of an Octorotor UAV using motors speeds measurements*. *IFAC-PapersOnLine*, vol. 50, no. 1, pages 5263–5268, 2017. (Cité en page 64.)
- [Sampath 1995] Meera Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinamohideen et Demosthenis Teneketzis. *Diagnosability of discrete-event systems*. *IEEE Transactions on automatic control*, vol. 40, no. 9, pages 1555–1575, 1995. (Cité en pages 57 et 201.)

- [Sampath 1996] Meera Sampath, Raja Sengupta, Stephane Lafortune, Kasim Sinnamohideen et Demosthenis C Teneketzi. *Failure diagnosis using discrete-event models*. IEEE transactions on control systems technology, vol. 4, no. 2, pages 105–124, 1996. (Cité en page 40.)
- [Sircoulomb 2008] Vincent Sircoulomb, Ghaleb Hoblos, Houcine Chafouk et José Ragot. *Analyse et comparaison de techniques d'estimation d'état sous contraintes inégalité non linéaires*. In 5ème Conférence Internationale Francophone d'Automatique, CIFA'2008, page CDROM, 2008. (Cité en page 54.)
- [Tan 2002] Chee Pin Tan et Christopher Edwards. *Sliding mode observers for detection and reconstruction of sensor faults*. Automatica, vol. 38, no. 10, pages 1815–1821, 2002. (Cité en page 47.)
- [Theilliol 2008] Didier Theilliol, Cédric Join et Youmin Zhang. *Actuator fault tolerant control design based on a reconfigurable reference input*. International Journal of Applied Mathematics and Computer Science, vol. 18, no. 4, pages 553–560, 2008. (Cité en page 64.)
- [Tsuda 2001] Kazuro Tsuda, Domenico Mignone, Giancarlo Ferrari-Trecate et Manfred Morari. *Reconfiguration strategies for hybrid systems*. In American Control Conference, 2001. Proceedings of the 2001, volume 2, pages 868–873. IEEE, 2001. (Cité en page 65.)
- [Vachhani 2005] Pramod Vachhani, Raghunathan Rengaswamy, Vikrant Gangwal et Shankar Narasimhan. *Recursive estimation in constrained nonlinear dynamical systems*. AIChE Journal, vol. 51, no. 3, pages 946–959, 2005. (Cité en page 48.)
- [Valavanis 2014] Kimon P Valavanis et George J Vachtsevanos. *Handbook of unmanned aerial vehicles*. Springer Publishing Company, Incorporated, 2014. (Cité en pages 2, 11 et 12.)
- [Veillette 1995] Robert J Veillette. *Reliable linear-quadratic state-feedback control*. Automatica, vol. 31, no. 1, pages 137–143, 1995. (Cité en pages 33, 34 et 64.)
- [Wald 1945] Abraham Wald. *Sequential tests of statistical hypotheses*. The Annals of Mathematical Statistics, vol. 16, no. 2, pages 117–186, 1945. (Cité en page 56.)
- [Wang 1996] H Wang et S Daley. *Actuator fault diagnosis : An adaptive observer-based technique*. IEEE transactions on Automatic Control, vol. 41, no. 7, pages 1073–1078, 1996. (Cité en page 47.)
- [Wang 2017] Ban Wang, Youmin Zhang, Jean-Christophe Ponsart et Didier Theilliol. *Fault-tolerant adaptive control allocation for unmanned multirotor helicopter*. IFAC-PapersOnLine, vol. 50, no. 1, pages 5269–5274, 2017. (Cité en page 62.)
- [Willsky 1976] Alan Willsky et H Jones. *A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems*. IEEE Transactions on Automatic control, vol. 21, no. 1, pages 108–112, 1976. (Cité en page 57.)

- [Wu 1998] N Eva Wu, Youmin Zhang et Kemin Zhou. *Control effectiveness estimation using an adaptive Kalman estimator*. In Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings, pages 181–186. IEEE, 1998. (Cit  en pages 40 et 52.)
- [Wu 2000] N Eva Wu, Youmin Zhang et Kemin Zhou. *Detection, estimation, and accommodation of loss of control effectiveness*. Int. J. Adapt. Control Signal Process, vol. 14, no. 775, page 795, 2000. (Cit  en pages 40, 52 et 64.)
- [Wu 2006] N Eva Wu, Sudha Thavamani, Youmin Zhang et Mogens Blanke. *Sensor fault masking of a ship propulsion system*. Control Engineering Practice, vol. 14, no. 11, pages 1337–1345, 2006. (Cit  en pages 52 et 62.)
- [Xiao-Zheng 2009] Jin Xiao-Zheng et Yang Guang-Hong. *Robust adaptive fault-tolerant compensation control with actuator failures and bounded disturbances*. Acta Automatica Sinica, vol. 35, no. 3, pages 305–309, 2009. (Cit  en page 36.)
- [Yang 2000] Y Yang, G-H Yang et YC Soh. *Reliable control of discrete-time systems with actuator failure*. IEE Proceedings-Control Theory and Applications, vol. 147, no. 4, pages 428–432, 2000. (Cit  en pages 33, 34 et 64.)
- [Yang 2001] Guang-Hong Yang, Jian Liang Wang et Yeng Chai Soh. *Reliable H_∞ controller design for linear systems*. Automatica, vol. 37, no. 5, pages 717–725, 2001. (Cit  en pages 33 et 34.)
- [Ye 2006] Dan Ye et G-H Yang. *Adaptive fault-tolerant tracking control against actuator faults with application to flight control*. IEEE Transactions on control systems technology, vol. 14, no. 6, pages 1088–1096, 2006. (Cit  en page 36.)
- [Zaytoon 2013] Janan Zaytoon et St ephane Lafortune. *Overview of fault diagnosis methods for discrete event systems*. Annual Reviews in Control, vol. 37, no. 2, pages 308–320, 2013. (Cit  en pages 57 et 58.)
- [Zhang 1998] Youmin Zhang et X Rong Li. *Detection and diagnosis of sensor and actuator failures using IMM estimator*. IEEE Transactions on aerospace and electronic systems, vol. 34, no. 4, pages 1293–1313, 1998. (Cit  en page 59.)
- [Zhang 1999a] Youmin Zhang et Jin Jiang. *An interacting multiple-model based fault detection, diagnosis and fault-tolerant control approach*. In Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, volume 4, pages 3593–3598. IEEE, 1999. (Cit  en page 59.)
- [Zhang 1999b] Youmin Zhang et N Eva Wu. *Fault diagnosis for a ship propulsion benchmark : part I*. IFAC Proceedings Volumes, vol. 32, no. 2, pages 7670–7675, 1999. (Cit  en page 52.)
- [Zhang 2001] Youmin Zhang et Jin Jiang. *Integrated active fault-tolerant control using IMM approach*. IEEE Transactions on Aerospace and Electronic systems, vol. 37, no. 4, pages 1221–1235, 2001. (Cit  en pages 59, 64, 65 et 66.)

-
- [Zhang 2002] YM Zhang et Jin Jiang. *Active fault-tolerant control system against partial actuator failures*. IEE proceedings-Control Theory and applications, vol. 149, no. 1, pages 95–104, 2002. (Cité en pages 40, 52 et 64.)
- [Zhang 2008] Youmin Zhang et Jin Jiang. *Bibliographical review on reconfigurable fault-tolerant control systems*. Annual reviews in control, vol. 32, no. 2, pages 229–252, 2008. (Cité en pages 33, 38, 45 et 61.)
- [Zhao 2005] Feng Zhao, Xenofon Koutsoukos, Horst Haussecker, James Reich et Patrick Cheung. *Monitoring and fault diagnosis of hybrid systems*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 35, no. 6, pages 1225–1240, 2005. (Cité en page 60.)

Méthodes indirectes d'adaptation et de décision pour la sécurisation du vol des drones à voilure fixe

Résumé De par l'augmentation de leur utilisation, la sécurisation du vol des drones devient de plus en plus importante. La commande tolérante aux fautes peut alors contribuer à l'obtention d'un niveau de sécurité acceptable. Le but de cette thèse est de développer et d'appliquer au vol des drones à voilure fixe une méthode de commande tolérante aux fautes basée d'une part sur une représentation de systèmes à l'aide de modèles décrivant des évolutions continues et d'autre part sur la représentation de systèmes à l'aide de modèles discrets ou logiques. Ainsi la première contribution de cette recherche est le développement d'une méthode générique de commande tolérante aux fautes utilisant les cadres de modélisation discret et continu. L'idée consiste à combiner une modélisation continue permettant d'estimer l'état et les paramètres de fautes et une modélisation discrète permettant de prendre une décision en ligne quant au contrôleur à utiliser. L'estimation continue permet d'avoir plus d'informations sur la faute qu'avec une modélisation discrète, alors que celle-ci prend en compte des probabilités de panne et des techniques d'optimisation qui sont plus adaptées à la tâche de décision. La seconde contribution concerne le développement et la validation d'une méthode active permettant de détecter et de diagnostiquer la faute. Un filtre de Kalman sensible aux sauts de panne pour l'estimation de l'état et des paramètres de fautes est développé. Pour la détection et le diagnostic de la panne, l'idée a été d'utiliser les données de l'estimation de façon probabiliste. Une fois la faute détectée et identifiée, le système de commande doit réagir pour pouvoir compenser cette faute. Le troisième contribution porte donc sur l'amélioration du suivi de la trajectoire par reconfiguration du système de commande. Les méthodes de commutation et d'adaptation sont combinées, afin de limiter le nombre de contrôleurs en utilisant des contrôleurs adaptatifs pour les modes dégradés, tout en ayant des contrôleurs faciles à concevoir. Des techniques d'optimisation sont alors utilisées de façon à prendre une décision en ligne quant au choix du contrôleur. Finalement, la méthode développée doit être vérifiée avant de pouvoir être implémentée sur un drone. La dernière contribution est l'évaluation de la capacité de la méthode à faire suivre une trajectoire d'atterrissage en cas de pannes capteurs ou actionneurs grâce à un modèle de drone.

Mots-clés : Drone, Adaptation, Reconfiguration, Commande tolérante aux fautes, Détection et Diagnostic de fautes, Estimation

Indirect adaptive and decisional methods to secure the flight of fixed-wing UAVs

Abstract Major security risks appear with the increase of the number of UAV in the air space. Thus, UAV security is more and more important and Fault Tolerant Control (FTC) methods could support the achievement of acceptable security level. The aims of this research is to develop and apply to a fixed wing UAV a FTC method which combines a continuous representation of the system and discrete or logical models. Thus, the first contribution of this thesis is the development of a generic fault tolerant control method which uses discrete and continuous frameworks. The idea was to combine a continuous framework to estimate the state and fault parameters and a discrete framework to take on line a decision about the controller. The continuous estimation provides more knowledge on the fault whereas a discrete model allows the use of different optimization tools which are more suited for the decision task. The second contribution is the development and the validation of an active method for fault detection and diagnosis. For its potential, a Kalman filter is adapted in order to be sensitive to abrupt faults and used for state and fault parameters estimation. These estimates are then used in a probabilistic way to detect and identify the fault. Once the fault is detected, the control system should react to compensate the fault. Thus, the third contribution of this thesis is the improvement of the trajectory tracking by reconfiguration of the control system. The aim is to combine switching and adaptive methods in order to limit the number of controllers by using adaptive controllers for degraded modes while having convenient controllers. Optimization tools are then used to take the decision about the controller to use. Finally, the method has to be validated before being implemented on line. The last contribution is the evaluation of the ability of the method to control the UAV trajectory despite the apparition of actuator or sensor faults during a landing approach.

Keywords : UAV, Adaptation, Reconfiguration, Fault Tolerant Control, Fault Detection and Diagnosis, Estimation