



HAL
open science

Sur la synthèse de lois de commande robuste et intelligente pour le contrôle de drones convertibles de type QTW

Khaled Benkhoud

► **To cite this version:**

Khaled Benkhoud. Sur la synthèse de lois de commande robuste et intelligente pour le contrôle de drones convertibles de type QTW. Automatique / Robotique. Normandie Université; Faculté des Sciences de Monastir (Tunisie), 2020. Français. NNT : 2020NORMR105 . tel-03330111

HAL Id: tel-03330111

<https://theses.hal.science/tel-03330111>

Submitted on 31 Aug 2021

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 CO - TUTELLE INTERNATIONALE

Pour obtenir le diplôme de doctorat

Spécialité AUTOMATIQUE, SIGNAL, PRODUCTIQUE ET ROBOTIQUE
et
Physique

Préparée au sein de l'Université de Rouen Normandie
et de la Faculté des Sciences de Monastir

Sur la synthèse de lois de commande robuste et intelligente pour
le contrôle de drones convertibles de type QTW

Présentée et soutenue par
Khaled BENKHOUD

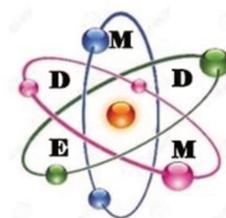
Thèse soutenue publiquement le 22-09-2020
devant le jury composé de

M. Mohsen MACHHOUT	Professeur, Faculté des Sciences de Monastir	Président
M. Patrick SIARRY	Professeur, Université Paris-Est Créteil	Rapporteur
M. Anis SAKLY	Professeur, ENI de Monastir, Université de Monastir	Rapporteur
Mme. Ionela PRODAN	Maître de conférences, Institut Polytechnique de Grenoble	Examinatrice
M. Nicolas LANGLOIS	DR (HDR), IRSEEM, ESIGELEC Rouen	Directeur de thèse
M. Mounir AYADI	Professeur, ENI de Tunis, Université de Tunis - El Manar	Codirecteur de thèse

Thèse dirigée par

M. **Nicolas LANGLOIS**, Pôle Automatique et Systèmes - Institut de Recherche en Systèmes Électroniques Embarqués IRSEEM- ESIGELEC

M. **Mounir AYADI**, Laboratoire de Recherche en Automatique LARA - Ecole Nationale d'Ingénieurs de Tunis ENIT



Sur la synthèse de lois de commande robuste et intelligente pour le contrôle de drones convertibles de type QTW

Khaled BENKHOUD

Résumé

Ces travaux de recherche portent sur la synthèse de lois de commande robuste et intelligente pour le contrôle d'un drone convertible à ailes/rotors basculants QTW. Une formulation multimodèle floue de type Takagi-Sugeno (T-S) a été adoptée pour la modélisation du système complexe étudié. Face à la complexité de ces systèmes volants, la proposition de méthodes de commande avancée, permettant l'élaboration de lois de commande robuste et performante tout en réduisant au mieux le temps de conception et les ressources de calcul des paramètres de réglage, constitue une préoccupation d'actualité dans ce cadre. Pour mieux gérer cette complexité, deux méthodes de réglage et d'optimisation des paramètres de commande sont proposées dans cette thèse.

Dans la première approche, le problème du choix de ces paramètres effectifs est contourné via la reformulation sous forme des problèmes d'optimisation statique sous contraintes des approches classiques de contrôle LQG, floue, prédictive MPC et non linéaire par Backstepping. Le principe consiste à favoriser des indices de performance tels que les critères ISE (Integral of Square Error) et MO (Maximum Overshoot) issus de la théorie de la commande optimale sous diverses contraintes opérationnelles. Une deuxième approche est proposée pour le réglage adaptatif en ligne des paramètres de la commande prédictive MPC basée sur un algorithme métaheuristique d'optimisation dynamique à comportement quantique (mQPSO). Cette dernière permet d'ajuster rapidement et avec une efficacité remarquable les paramètres effectifs de la commande MPC en assurant la stabilité optimale des sous modèles locaux issus d'une formulation multimodèle floue de type T-S (Takagi-Sugeno) du drone convertible QTW étudié.

Mots clés— drones convertibles QTW, modélisation floue Takagi-Sugeno, multimodèle, commande prédictive MPC, optimisation dynamique, métaheuristic globales.

On the synthesis of robust and intelligent control laws for the control of QTW convertible UAVs

Khaled BENKHOUD

Abstract

The research work presented in the present PhD thesis focuses on the design and tuning of robust and intelligent control laws for a convertible Quad Tilt-Wings (QTW) drone. A fuzzy Takagi-Sugeno (T-S) based formulation of the studied drone was adopted for the modeling. Faced with the complexity of these flying systems, advanced linear and nonlinear control methods, allowing less complex tuning algorithms and reduced design time and resources, has been proposed and successfully applied to stabilize the attitude and altitude dynamics of the drone. To better manage this complexity, two effective control parameters' tuning methods are proposed.

In the first approach, the parameters' selection problem is investigated via the reformulation in the form of static optimization problems under constraints of classical control approaches such as LQG, fuzzy, predictive MPC and nonlinear Backstepping. The principle consists in favoring performance indices such as ISE (Integral of Square Error) and MO (Maximum Overshoot) criteria derived from the theory of optimal control under various operational constraints. A second systematic and efficient tuning approach is proposed for the adaptive tuning of the predictive MPC parameters based on a dynamic metaheuristic algorithm such as a quantum particle swarm optimization (mQPSO). Such an online optimization algorithm makes an easy and systematic tuning of the main and effective parameters of the well-known MPC control approach. High performance specifications are satisfied and improved for the stabilization and tracking of the flight attitude and altitude dynamics.

Keywords— convertible drones QTW, fuzzy Takagi-Sugeno modeling, MPC predictive control, dynamic optimization, global metaheuristics.

*À mes très chers parents ;
en témoignage de gratitude d'estime et d'attachement.
Puisse Dieu vous accorder santé, longue vie et prospérité.*

Avant-propos

Les travaux présentés dans ce mémoire de thèse en co-tutelle internationale ont été effectués au sein du Laboratoire de Recherche en Automatique (L.A.R.A) de l'ENI de Tunis en Tunisie et du Pôle Automatique et Systèmes de l'Institut de Recherche en Systèmes Electroniques Embarqués (IRSEEM) de l'ESIGELEC Rouen, France.

Je suis très redevable envers mes directeurs de thèse, Messieurs Mounir AYADI, Professeur à l'ENI de Tunis, et Nicolas LANGLOIS, Enseignant Chercheur HDR, Directeur de la formation par la recherche et responsable du pôle A& S, pour l'accueil qu'ils m'ont réservé au sein de leurs équipes de recherche et qui m'ont offert la possibilité d'évoluer dans mes travaux de recherche. Qu'ils trouvent ici le témoignage de ma profonde gratitude pour l'intérêt qu'ils ont porté à mes travaux et les conseils éclairés qu'ils m'ont prodigués avec le sérieux, la compétence et la constante disponibilité qui les caractérisent.

Je tiens à remercier également mon co-encadreur de thèse Monsieur Mohamed Soufiene BOUALLÈGUE pour ses qualités pédagogiques et scientifiques, pour ses encouragements incessants ainsi que pour m'avoir accordé sa confiance durant ces années de thèse. Ses conseils et son aide ont été les plus utiles pour l'aboutissement de ce travail.

Je remercie Monsieur Mohsen MACHHOUT, Professeur à la Faculté des Sciences de Monastir, d'avoir accepté de présider le jury de cette thèse et de participer à l'examen de ce travail et en avoir assuré la critique.

J'adresse toute ma reconnaissance à Monsieur Patrick SIARRY, Professeur à l'Université Paris-Est Créteil (UPEC), et Monsieur Anis SAKLY, Professeur à l'Ecole Nationale d'Ingénieurs de Monastir (ENIM) d'avoir accepté d'être rapporteurs de mes travaux. Je les remercie vivement de leur lecture approfondie de mon mémoire et leurs remarques constructives et enrichissantes.

Je remercie également Madame Ionela PRODAN, Maître de conférences à l'Institut Polytechnique de Grenoble (INP-ESISAR), d'avoir accepté avec grand plaisir de participer à ce jury et d'examiner ce travail.

Je tiens à remercier l'école doctorale Matériaux, Dispositifs et Microsystèmes de Monastir (EDMDM) ainsi que le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique de Tunisie (MESRS) qui ont financés mes séjours scientifiques à Rouen pendant trois années. Je remercie également le Laboratoire de Recherche en Automatique (L.A.R.A) pour leur participation au bon déroulement de mes séjours scientifiques en France.

Je remercie finalement tous mes amis et collègues de l'IRSEEM et du Laboratoire LARA pour leur amicale présence et pour la sympathie qu'ils m'ont constamment témoignée. Je leur exprime ici toute ma gratitude.

Table des matières

Table des figures	ix
Liste des tableaux	xi
Glossaire	xii
Introduction générale	xiv
1 État de l’art sur les drones convertibles UAV	1
1.1 Introduction	2
1.2 Historique des drones	2
1.3 Classification des drones	4
1.3.1 Classification selon l’altitude et l’endurance	4
1.3.2 Classification selon le mode de propulsion	6
1.4 Les drones convertibles	7
1.4.1 Typologie	7
1.4.2 Les convertibles à basculement entier	9
1.4.3 Les convertibles à variation de structure	9
1.5 Drone convertible « Quad-Tilt Wing »	11
1.5.1 Structure mécanique et modes du vol	11
1.5.2 Modélisation dynamique	13
1.5.2.1 Equations cinématiques	13
1.5.2.2 Equations dynamiques	14
1.5.2.3 Forces extérieures	15
1.5.2.4 Moments extérieurs	17
1.5.2.5 Dynamiques des rotors	18
1.5.2.6 Modèle dynamique dans l’espace d’état	18
1.6 Modélisation T-S de systèmes non linéaires	20
1.6.1 Principe	20
1.6.2 Représentation d’état d’un modèle T-S	21
1.6.3 Approches d’obtention d’un modèle T-S	22
1.6.3.1 Multi-modèle par identification	22
1.6.3.2 Multi-modèle par linéarisation	23

1.6.3.3	Multi-modèle par secteurs non linéaires	23
1.6.4	Stabilité des systèmes T-S	24
1.6.5	Stabilisation des systèmes T-S	26
1.6.6	Modélisation T-S d'un drone convertible QTW	27
1.7	Critères de performance considérés	30
1.8	Conclusion	31
2	Commande de drones convertibles QTW	32
2.1	Introduction	33
2.2	Commande linéaire quadratique	33
2.2.1	Formulation du problème	33
2.2.2	Résultats de simulation et discussion	34
2.3	Commande par backstepping avec action intégrale	35
2.3.1	Contrôle de l'attitude	35
2.3.2	Contrôle de la position et de l'altitude	39
2.3.3	Résultats et discussion : cas de la stabilisation des dynamiques d'attitude et d'altitude	40
2.3.4	Résultats et discussion : cas de la poursuite de trajectoires en vol vertical	40
2.4	Commande prédictive à base de modèle	45
2.4.1	Principe	45
2.4.2	Choix des paramètres de réglage	46
2.4.3	Représentation d'état augmentée à temps discret	47
2.4.4	Formulation de la commande en absence de contraintes	48
2.4.5	Formulation de la commande en présence de contraintes	50
2.4.5.1	Contraintes sur les incréments de commande	51
2.4.5.2	Contraintes sur l'amplitude des entrées de commande	51
2.4.5.3	Contraintes sur les sorties	52
2.4.6	Application à la commande du drone QTW	54
2.4.7	Résultats de simulation et discussion	55
2.5	Conclusion	59
3	Synthèse de commandes intelligentes à base des métaheuristique	60
3.1	Introduction	61
3.2	État de l'art sur les algorithmes métaheuristiques d'optimisation statique	61
3.2.1	L'algorithme PSO standard	61
3.2.1.1	Principe et formulation mathématique	61
3.2.1.2	Pseudo-code	63
3.2.2	L'algorithme PSO à facteur d'inertie variable PSO-In	63
3.2.3	L'algorithme PSO à position gbest perturbée PSO-gbest	64
3.2.4	L'algorithme PSO à comportement Quantique QPSO	65
3.2.4.1	La variante QPSO-GM	67
3.2.4.2	La variante QPSO-HPD	67
3.3	Commande PID à supervision floue et réglage par PSO	68

3.3.1	Formulation du problème	68
3.3.2	Résultats de simulation et discussion	70
3.4	Commande LQG à réglage par métaheuristiques	80
3.4.1	Formulation du problème	80
3.4.2	Résultats de simulation et discussion	81
3.5	Commande prédictive MPC à réglage par métaheuristiques	86
3.5.1	Formulation du problème	86
3.5.2	Résultats de simulation et discussion	87
3.6	Conclusion	92
4	Réglage en ligne de la commande prédictive	94
4.1	Introduction	95
4.2	État de l'art sur l'optimisation continue dynamique	95
4.2.1	Principe	95
4.2.2	L'algorithme PSO multi-essaims à comportement quantique : mQPSO	98
4.2.2.1	Motivations et nouvelles équations de mouvement	98
4.2.2.2	Mécanisme d'exclusion	100
4.2.2.3	Test de changement de l'environnement	100
4.2.2.4	Pseudo-code de l'algorithme mQPSO	101
4.2.3	Principaux jeux de test	102
4.2.3.1	Moving Peaks Benchmark	102
4.2.3.2	Generalized Dynamic Benchmark Generator	103
4.3	Stratégie de réglage dynamique de la commande TSMPC	106
4.4	Résultats de simulation et discussion	107
4.4.1	Étude comparative	110
4.4.2	Analyse de robustesse de la méthode proposée	113
4.5	Conclusion	119
	Conclusion générale et perspectives	120
	Annexes	123
	Liste des publications	129
	Références bibliographiques	131

Table des figures

1.1	Premier drone Target conçu par Archibald Low en 1916 (Mills, 2019).	3
1.2	Prototype Hewitt-Sperry Automatic Airplane sur sa voie de lancement (Munson, 1988).	3
1.3	Prototype RQ-2 Pioneer en vol (Munson, 1995).	4
1.4	Premier drone algérien AMEL (menadefense.net, 2013).	4
1.5	Drones de type MALE (Nonami et al., 2010).	5
1.6	Drones de type HALE (Austin, 2010).	5
1.7	Drones tactiques TUAV (Safran, 2014).	6
1.8	Drones à voilure fixe.	6
1.9	Drones à voilure tournante.	8
1.10	Drones à voilure battante.	8
1.11	Drones convertibles.	9
1.12	Modes du vol de XFY-1 POGO (Koehl, 2012).	9
1.13	Modes du vol de V-22 Osprey(Boeing, 2016).	10
1.14	Drone convertible de type Quad tilt Wing.	10
1.15	Modes de vol du convertible QTW : horizontal, vertical et de transition.	12
1.16	Différents modes de vol du drone convertible QTW (BenKhoud et al., 2017).	12
1.17	Coordonnées et repères aérodynamiques du convertible QTW.	13
1.18	Forces extérieures agissant sur le corps du convertible QTW.	15
1.19	Angle d'attaque de profil d'aile.	16
1.20	Schéma blocs pour le modèle dynamique du drone QTW.	20
1.21	Réponses du système non linéaire et modèle TS : angle de roulis.	29
1.22	Erreur $\phi_{NL} - \phi_{TS}$.	29
1.23	Réponses du système non linéaire et modèle TS : angle de tangage.	29
1.24	Erreur $\theta_{NL} - \theta_{TS}$.	29
1.25	Réponses du système non linéaire et modèle TS : angle de lacet.	30
1.26	Erreur $\psi_{NL} - \psi_{TS}$.	30
2.1	Réponses indicielles en attitude pour la régulation LQ du drone QTW.	36
2.2	Réponses indicielles en position pour la régulation LQ du drone QTW.	36
2.3	Schéma de commande par backstepping à action intégrale du convertible QTW.	37

2.4	Dynamiques d'attitude et d'altitude stabilisées.	41
2.5	Signaux de commande backstepping pour la stabilisation.	41
2.6	Positions longitudinale et transversale contrôlées du drone QTW.	42
2.7	Variation des signaux de commande sous l'effet de perturbations.	42
2.8	Performances en poursuite de la commande backstepping intégrale.	43
2.9	Variation des signaux de commande en poursuite.	44
2.10	Poursuite de trajectoire hélicoïdale dans l'espace.	44
2.11	Illustration des signaux de commande et de sortie de la commande prédictive MPC.	46
2.12	Principe de la commande prédictive par interpolation (TSMPC) appliquée au drone QTW.	54
2.13	Effet de la variation de l'horizon de prédiction pour un horizon de contrôle fixé à $N_c = 15$	56
2.14	Effet de la variation de l'horizon de contrôle pour un horizon de prédiction fixé à $N_p = 65$	56
2.15	Réponses du système en fonctionnement nominal : angles d'attitude.	57
2.16	Réponses du système en fonctionnement nominal : vitesses angulaires.	57
2.17	Signaux de commande appliqués au système.	58
2.18	Evolution des fonctions d'activation μ_i	58
3.1	Evolution de la position d'une particule dans son espace de recherche.	62
3.2	Structure proposée de la supervision floue de correcteurs PID.	69
3.3	Surfaces de décision floue des gains des correcteurs GS-PID.	72
3.4	Historiques de convergence des algorithmes PSO pour le problème (3.32).	73
3.5	Stabilisation des dynamiques de roulis, tangage et lacet du véhicule QTW.	74
3.6	Performances comparées des contrôleurs flou, GS-PID flou et GS-PID flou optimisé : dynamique de roulis.	76
3.7	Performances comparées des contrôleurs flou, GS-PID flou et GS-PID flou optimisé : dynamique de tangage.	76
3.8	Performances comparées des contrôleurs flou, GS-PID flou et GS-PID flou optimisé : dynamique de lacet.	77
3.9	Robustesse vis-à-vis d'un défaut de 20% sur les signaux de commande.	78
3.10	Variation temporelle des gains du PID pour la dynamique de roulis.	78
3.11	Variation temporelle des gains du PID pour la dynamique de tangage.	79
3.12	Variation temporelle des gains du PID pour la dynamique de lacet.	79
3.13	Structure de commande LQG à réglage par métaheuristiques.	80
3.14	Historiques de convergence des algorithmes pour le problème (3.36).	83
3.15	Réponses indicielles comparées de la commande LQG avec réglage optimisé.	84
3.16	Performances temporelles comparées des commandes LQG-PSO, PID et LQR.	84
3.17	Réglage systématique des paramètres de la commande MPC pour le drone QTW.	87
3.18	Historiques de convergence des algorithmes pour le problème (3.37).	88

3.19 Performances comparées de la commande MPC avec réglage optimisé : altitude $z(t)$	90
3.20 Performances comparées de la commande MPC avec réglage optimisé : roulis $\phi(t)$	91
3.21 Performances comparées de la commande MPC avec réglage optimisé : tangage $\theta(t)$	91
3.22 Performances comparées de la commande MPC avec réglage optimisé : lacet $\psi(t)$	92
4.1 Illustration du paysage de la fonction dynamique Easm.	97
4.2 Illustration du mécanisme de partage des informations entre les essais du QPSO.	100
4.3 Illustration du paysage de la fonction de test MPB.	104
4.4 Structure de contrôle proposée à base de l'optimisation dynamique des paramètres TSMPC en ligne.	106
4.5 Évolution des dynamiques d'attitudes et d'altitude en fonction du temps.	109
4.6 Évolution des vitesses angulaires et linéaires en fonction du temps.	109
4.7 Évolution des signaux de commande en fonction du temps.	110
4.8 Exemple d'évolution des paramètres de la TSMPC au cours du temps.	111
4.9 Performances en poursuite de la commande TSMPC avec réglage par mQPSO.	112
4.10 Poursuite d'une trajectoire hélicoïdale dans l'espace.	112
4.11 Performances temporelles comparées des commandes PDC, ATSMPC et mQPSO-TSMPC : Signaux de sortie.	114
4.12 Performances temporelles comparées des commandes PDC, ATSMPC et mQPSO-TSMPC : Signaux de commande.	115
4.13 Performance nominale et robuste de la dynamique de roulis.	116
4.14 Performance nominale et robuste de la dynamique de tangage.	116
4.15 Performance nominale et robuste de la dynamique d'attitude.	117
4.16 Robustesse vis-à-vis d'un défaut de 20% sur la dynamique d'attitude.	117
4.17 Variation temporelle des paramètres de la technique MPC pour le sous-modèle T-S 1.	118
4.18 Variation temporelle des paramètres de la technique MPC pour le sous-modèle T-S 2.	118
4.19 Variation temporelle des paramètres de la technique MPC pour le sous-modèle T-S 3.	119
B.1 Stratégie de réglage dynamique de la commande TSMPC.	125
B.2 Modèle Dynamique de QTW en vol vertical.	126
C.1 Code source LabVIEW du modèle QTW.	127
C.2 Code source LabVIEW pour la co-simulation commande LQG.	128

Liste des tableaux

1.1	Quelques projets de recherche sur les drones convertibles.	11
2.1	Paramètres physiques du drone convertible QTW étudié (Cetinsoy et al., 2012).	35
3.1	Base des règles floues pour le gain K_p	69
3.2	Base des règles floues pour le gain K_i	70
3.3	Base des règles floues pour le gain K_d	70
3.4	Paramètres de contrôle des différents algorithmes PSO.	71
3.5	Résultats d'optimisation de 30 essais indépendants sur le problème (3.32).	71
3.6	Performances temporelles de la dynamique : roulis $\phi(t)$	73
3.7	Performances temporelles de la dynamique : tangage $\theta(t)$	75
3.8	Performances temporelles de la dynamique : lacet $\psi(t)$	75
3.9	Performances temporelles comparées de la dynamique de roulis $\phi(t)$	75
3.10	Performances temporelles comparées de la dynamique de tangage $\theta(t)$	77
3.11	Performances temporelles comparées de la dynamique de lacet $\psi(t)$	77
3.12	Résultats d'optimisation de vingt exécutions du problème (3.36).	82
3.13	Tailles de population et maximum d'itérations : problème (3.36).	82
3.14	Valeurs moyennes des coefficients optimaux de matrices de pondération LQG sur 20 exécutions du problème (3.36)	83
3.15	Performances temporelles de la dynamique de lacet de $\psi(t)$	85
3.16	Performances temporelles de la dynamique de altitude de $z(t)$	85
3.17	Performances temporelles de la dynamique de position de $x(t)$	85
3.18	Performances temporelles de la dynamique de position de $y(t)$	85
3.19	Résultats d'optimisation sur 30 essais du problème (3.37).	88
3.20	Valeurs moyennes des paramètres optimaux de commande MPC sur 20 exécutions du problème (3.37)	89
3.21	Performances temporelles comparées pour la commande MPC de la dynamique de lacet $\phi(t)$	89
3.22	Performances temporelles comparées de la commande MPC de la dynamique de tangage $\theta(t)$	89
3.23	Performances temporelles comparées de la commande MPC de la dynamique de roulis $\psi(t)$	90

3.24	Performances temporelles comparées de la commande MPC de la dynamique d'altitude $z(t)$	90
4.1	Fonctions statiques utilisées pour générer les fonctions de test du GDBG. .	105
4.2	Paramètres de contrôle d'algorithmes mQPSO.	108
4.3	Paramètres optimaux de la technique mQPSO-TSMPC obtenus en régime permanent.	108
4.4	Comparaison des performances temporelles de la dynamique de roulis $\phi(t)$	113
4.5	Comparaison des performances temporelles de la dynamique de tangage $\theta(t)$	113
4.6	Comparaison des performances temporelles de la dynamique de lacet $\psi(t)$	114

Glossaire

ABC	Ant Bee Colony
BMI	Inégalité(s) matricielle(s) bilinéaire(s)(Bilinear Matrix Inequality)
CSE	Énergie du signal de commande (Control Signal Energy)
CEE	Énergie de l'effort du signal de commande (Control Effort Energy)
FPSOMA	Fractional Particle Swarm Optimization Memetic Algorithm
GA	Genetic Algorithm
HALE	High-Altitude Long-Endurance
HSA	Harmony Search Algorithm
HW	Hardware
IAE	Integral of Absolute Error
ISE	Integral of Square Error
LQG	Linéaire Quadratique Gaussienne
LMI	Linear Matrix Inequalities
LTI	Linéaire invariant dans le temps (Linear Time Invariant)
MALE	Medium-Altitude Long-Endurance
MAV	Micro Aerial Vehicle
MIMO	Multi-entrées multi-sorties (Multi-Input Multi-Output)
MO	Maximum Overshoot
mQPSO	PSO multi-essais à comportement Quantique
MPC	Model Predictive Control
NI	National Instruments
OV	Dépassement (Overshoot)
PDC	Parallel Distributed Compensation
PFC	Predictive Functional Control
PID	Proportionnel Intégral Dérivé
PSO	Particle Swarm Optimization
pPSO	Perturbed Particle Swarm Optimization
QPSO	Quantum Behaved Particle Swarm Optimization
QTW	Quad Tilt Wing
RIO	Reconfigurable Input Output
RT	Temps de montée (Rise Time)
RPAS	Remotely Piloted Aircraft Systems
SDI	Index de degré de stabilité (Stability Degree Index)
ST	Temps de réponse (Settling Time)
SW	Software

TUAV	Tactical Unmanned Aerial Vehicle
TSF	Télégraphie Sans Fil
T-S	Takagi-Sugeno
UAV	Unmanned Aerial Vehicle
UPC	Unified Predictive Control
VAA	Vehicle Aerial Autonome
VARU	VARiance du signal de commande (VARiance of control signal)
VTOL	Vertical Take-Off and Landing
WCA	Water Cycle Algorithm

Introduction générale

Durant ces dernières décennies, les aéronefs Unmanned Aerial Vehicle (UAV) ont connu des utilisations de plus en plus remarquables et fréquentes dans les domaines de la reconnaissance militaire ainsi que dans des applications civiles variées relevant de la surveillance, la prévention et l'inspection, la collecte de données, etc. Ces drones, à voilure fixe ou tournante, sont des engins sans pilotes capables de mener à bien des missions à risque ou difficiles d'accès pour l'être humain. Ces missions peuvent être réalisées de façon semi autonome, autonome ou téléguidée à distance pendant une durée de vol pouvant varier en fonction de leurs capacités dans un environnement hostile et perturbé. Motivés par les récentes avancées technologiques dans les domaines de la miniaturisation des actionneurs et des capteurs, d'une part, et de l'électronique embarquée et de l'Automatique, d'autre part, les développements théoriques liés aux aéronefs UAV ont connu une activité très importante. La maîtrise de l'aérodynamique de ces engins, des lois de contrôle associées, de l'observation d'état et éventuellement des perturbations est toutefois indispensable pour la conception de systèmes de commande embarquée performants et robustes. Ceci explique tout l'intérêt porté par les communautés Roboticienne et Automaticienne à cet axe de recherche qui ne cesse de s'accroître et de se diversifier. Les drones convertibles QTW ayant des mécanismes de basculement au niveau des ailes et des rotors d'hélices ont aussi montré un intérêt grandissant par rapport aux architectures classiques. Ces aéronefs sont constitués d'un ensemble d'hélices/ailes mobiles permettant de décoller verticalement et d'effectuer aisément des vols stationnaires et d'avancement plus rapide pour bénéficier des effets de portance aérodynamique et de minimiser ainsi la consommation d'énergie en vol planaire.

Face à la complexité de ces systèmes, la proposition de méthodes de commande avancée, permettant l'élaboration de lois de commande robuste et performante tout en réduisant au mieux le temps de conception et les ressources de calcul, constitue une préoccupation d'actualité dans ce cadre. Toutefois, le choix d'utiliser une loi de commande plus qu'une autre dépend fortement des informations disponibles sur le système à contrôler, de la présence ou non de contraintes, de la complexité mathématique liée à l'intégration d'une loi de commande et bien évidemment à l'accessibilité des ressources nécessaires, qu'elles soient informatiques ou électroniques, pour l'implémentation de la commande.

Pour mieux gérer cette complexité, le recours à la théorie de l'optimisation présente une solution prometteuse. En effet, les problèmes de commande et d'observation peuvent être reformulés sous la forme de problèmes d'optimisation, mono- ou multi-objectifs pour lesquels une multitude d'algorithmes peut être adoptée. Les métaheuristiques d'optimisation difficile, considérées comme des algorithmes stochastiques présentant un haut niveau

d'abstraction vis-à-vis d'une large gamme de problèmes différents, constituent ici une solution intéressante, ayant montré une efficacité remarquable dans la résolution de problèmes variés d'ingénierie. La puissance de calcul et la disponibilité des logiciels de CAO ont largement contribué à la proposition, régulière, de nouvelles métaheuristiques plus puissantes et en constante amélioration. L'hybridation des approches classiques de commande, conventionnelles et non conventionnelles, avec des techniques relevant de cette famille d'algorithmes stochastiques et non reproductibles, présente ainsi l'une des principales contributions de mes travaux.

Plan du mémoire

Les résultats des travaux obtenus à l'issue de ces trois années de thèse seront présentés dans ce mémoire qui est, à son tour, composé de quatre chapitre résumés ci-dessous.

Chapitre 1 : État de l'art sur les drones convertibles UAV

Le premier chapitre aborde une étude bibliographique sur les différents types de drones en présentant l'évolution de leurs technologies et de leurs structures. Une classification des drones est ainsi établie selon l'altitude, l'endurance et leur mode de propulsion. Un intérêt particulier a été porté aux drones convertibles qui présentent une typologie particulière sous forme de voilure tournante ou fixe. Le drone convertible de type QTW-UAV a été retenu comme étant le système à commander par des approches classiques et intelligentes. Une modélisation dynamique de ce drone a été introduite en tenant compte des différents paramètres de vol et des forces permettant de représenter cette dynamique dans les différentes phases de vol. Après avoir défini le modèle dynamique dans l'espace d'état, une modélisation de type T-S du système non linéaire, a été proposée en vue de déterminer des lois de commande robuste permettant le contrôle du drone. La comparaison des résultats de simulation de ce modèle avec le modèle non linéaire, montre la fiabilité de l'approche adoptée.

Chapitre 2 : Commande de drones convertibles QTW

Dans le deuxième chapitre, et après avoir validé le modèle dynamique développé, différentes lois de commande ont été élaborées telles que la commande linéaire quadratique (LQ) qui a été appliquée autour d'un point de fonctionnement bien choisi. Par la suite, une commande par backstepping incorporant une action intégrale, a été conçue pour contrôler l'altitude et les positions du drone. Une fonction quadratique candidate de Lyapunov a été retenue pour calculer une commande stabilisante. Les résultats obtenus montrent les bonnes performances de la commande proposée même en présence de perturbations. La troisième commande élaborée est une commande prédictive à base de modèle (MPC) formulée dans un cadre de temps discret. La stratégie de commande TSMPC a été ainsi développée et appliquée au modèle non linéaire du drone. Les effets des paramètres de la commande prédictive ont été étudiés et présentés par simulations numériques.

Chapitre 3 : Synthèse de commande intelligentes à base des méta-heuristique

Le troisième chapitre est consacré à une autre approche permettant la synthèse de lois de commande en exploitant des techniques métaheuristiques. Une étude bibliographique a été ainsi effectuée sur les algorithmes des méthodes d'optimisation en se focalisant sur les différentes variantes de la méthode PSO allant de la méthode PSO classique aux variantes les plus récentes telles que QPSO-GM et QPSO-HPD. Ces algorithmes ont été exploités pour le réglage des paramètres de régulateurs LQG, PID à supervision floue et TSMPC. Plusieurs comparaisons entre ces algorithmes, ont été effectuées en utilisant plusieurs critères de performances.

Chapitre 4 : Réglage en ligne de la commande prédictive

Le quatrième chapitre de ce mémoire est dédié au réglage en ligne de la commande prédictive qui reste une problématique ouverte et d'actualité. Dans ces travaux, un algorithme PSO multi-essaims à comportement quadratique (mQPSO) pour la synthèse d'une commande MPC, est proposé et appliqué au drone objet de cette étude. Les résultats obtenus montrent les performances élevées de cette nouvelle approche de commande robuste. Une étude comparative entre cette méthode avec d'autres méthodes de commande, a confirmé les performances obtenues. Une étude de robustesse a permis de mettre en exergue l'intérêt qu'apporte cette approche originale pour le contrôle d'un système non linéaire complexe.

Chapitre 1

État de l'art sur les drones convertibles UAV

Sommaire

1.1	Introduction	2
1.2	Historique des drones	2
1.3	Classification des drones	4
1.3.1	Classification selon l'altitude et l'endurance	4
1.3.2	Classification selon le mode de propulsion	6
1.4	Les drones convertibles	7
1.4.1	Typologie	7
1.4.2	Les convertibles à basculement entier	9
1.4.3	Les convertibles à variation de structure	9
1.5	Drone convertible « Quad-Tilt Wing »	11
1.5.1	Structure mécanique et modes du vol	11
1.5.2	Modélisation dynamique	13
1.6	Modélisation T-S de systèmes non linéaires	20
1.6.1	Principe	20
1.6.2	Représentation d'état d'un modèle T-S	21
1.6.3	Approches d'obtention d'un modèle T-S	22
1.6.4	Stabilité des systèmes T-S	24
1.6.5	Stabilisation des systèmes T-S	26
1.6.6	Modélisation T-S d'un drone convertible QTW	27
1.7	Critères de performance considérés	30
1.8	Conclusion	31

1.1 Introduction

Ce premier chapitre a pour intérêt de mettre en relief un état de l'art sur les différents types de drones en présentant l'évolution de leurs technologies et de leurs structures. Une classification des drones sera ainsi établie selon l'altitude, l'endurance et leur mode de propulsion. Un intérêt particulier sera porté aux drones convertibles qui présentent une typologie particulière sous forme de voilure tournante ou fixe. Ensuite, Une modélisation dynamique, en vue de la commande, sera introduite du drone convertible Quad Tilt Wing (QTW) retenu dans nos travaux. Après avoir donné la description de ce drone, ses mouvements possibles et son principe du vol, un modèle mathématique est établi.

Dans un second temps, une structure de type Takagi-Sugeno (T-S) sera introduite en tant que technique de modélisation représentative et de contrôle. Cette technique permet l'écriture d'un système non linéaire sous forme quasi-LPV et peut représenter exactement le comportement de ce dernier en adaptant l'approche d'obtention d'un modèle T-S par secteurs non linéaires. La mise sous forme T-S de modèle dynamique d'un drone QTW sera justifiée en vue de déterminer des lois de commande robuste permettant le contrôle du drone.

1.2 Historique des drones

Il est intéressant de lire ce que retiennent les dictionnaires pour définir un drone. Selon l'encyclopédie LAROUSSE, le mot drone signifie en anglais "faux bourdon", c'est un petit avion télécommandé et utilisé pour des tâches diverses (missions de reconnaissance tactique à haute altitude, surveillance du champ de bataille et guerre électronique, etc.). Le sigle UAV, ou encore RPAS, désigne cette catégorie des aéronefs qui sont capables de voler et d'effectuer une mission sans présence humaine à bord (Mills, 2019; Nonami et al., 2010). Notons que le mot drone peut également désigner les engins terrestres ou sous-marins autonomes. Cependant, il est courant de réserver son usage aux Vehicle Aerial Autonome (VAA) (Castillo et al., 2004).

Le concept des drones était déjà connu au cours de la première guerre mondiale. En 1916, Archibald Low lance l'aéronef Target représenté dans la figure 1.1. C'est un projet d'avion-cible commandé à distance par les ondes TSF en Grande-Bretagne (Mills, 2019). Il s'agit de former les apprentis-pilotes au tir sans risque d'abattre l'avion remorquant la cible. Une idée similaire naît en 1917 aux Etats-Unis où les ingénieurs Elmer Ambrose Sperry, Lawrence Sperry et Peter Cooper Hewitt travaillent sur le projet d'avion radiocommandé Hewitt-Sperry Automatic Airplane. Ce prototype, figure 1.2, possède des gyroscopes pour la stabilisation et la direction du corps (Munson, 1988).

Le 2 juillet 1917, le pilote français Max Boucher a réussi à faire voler, sans pilote à bord, un avion sur 1 km (Munson, 1995). Il reprenait les travaux menés depuis 1894 par Octave Détable qui grâce à une voilure dotée de cônes divergents apporta une stabilité automatique à des planeurs.

Le grand essor des drones date de la guerre de Corée et celle du Vietnam. Pendant la guerre du Vietnam, les Américains ont utilisé des drones pour localiser les rampes de

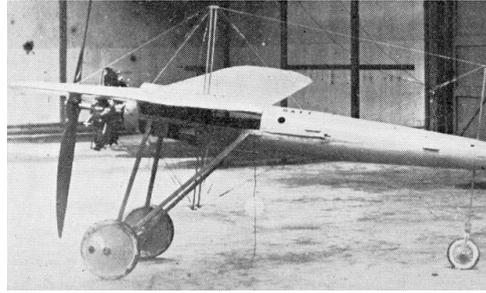


FIGURE 1.1 – Premier drone Target conçu par Archibald Low en 1916 ([Mills, 2019](#)).

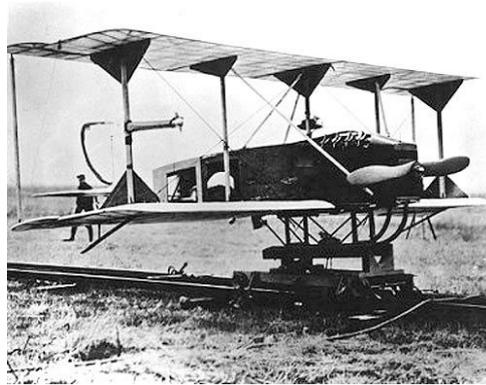


FIGURE 1.2 – Prototype Hewitt-Sperry Automatic Airplane sur sa voie de lancement ([Munson, 1988](#)).

lancement des missiles sol-air soviétiques "SAM-2". Plus tard, en 1991, lors de la guerre du Golfe, ils ont fait appel au drone RQ-2 Pioneer montré dans la figure 1.3 pour la surveillance jour/nuit, l'acquisition des objectifs et les réglages de l'artillerie. Le RQ-2 Pioneer est un drone développé en collaboration entre les États-Unis et Israël. C'est le premier drone tactique (TUAV) entré en service dans les forces armées des États-Unis ([Munson, 1995](#)).

De leur côté, les pays du monde arabe essaient de trouver un chemin dans ce domaine avec le drone AMEL, figure 1.4. C'est un prototype de conception algérienne opérationnel depuis octobre 2013 avec une autonomie de vol de 6 heures sur une distance de 200 km ([menadefense.net, 2013](#)). Ce drone peut accomplir ses missions à 3 500 mètres d'altitude de sa cible. Après que les États-Unis ont refusé de fournir des drones armés, un autre projet est lancé par l'Arabie Saoudite qui veut mener son propre programme de drone national avec l'aide de l'Afrique du Sud.

Les exemples d'utilisation civile se font plus rares. On peut toutefois citer la surveillance urbaine en Afrique du Sud et le traitement des cultures effectué au Japon par des drones hélicoptères télécommandés ([Nonami, 2007](#)). En général, le champ d'application des drones est très vaste et ceci a fait que plusieurs projets de recherche à travers le monde ont vu le jour. Des chercheurs de plusieurs communautés scientifiques, telles que la robotique, l'aéronautique, l'automatique, le traitement du signal et l'informatique, avec le support de l'industrie et des gouvernements, participent au développement de tels robots



FIGURE 1.3 – Prototype RQ-2 Pioneer en vol ([Munson, 1995](#)).



FIGURE 1.4 – Premier drone algérien AMEL ([menadefense.net, 2013](#)).

dans le but de réaliser des prototypes fortement manœuvrables et extrêmement stables ([Austin, 2010](#)).

1.3 Classification des drones

Il n'existe pas une façon unique pour classifier les drones. Ces derniers peuvent se regrouper selon divers paramètres comme l'altitude du vol, l'endurance (temps du vol de l'aéronef), les dimensions (capacité d'emport, ce qui est sensiblement équivalent) ou encore le type de voilure qui définit l'architecture aéromécanique.

1.3.1 Classification selon l'altitude et l'endurance

Les drones sont classiquement répartis selon leurs tailles. Dans ce cadre, les domaines opérationnels des drones peuvent se décomposer en trois catégories ([García Carrillo et al., 2013](#)) :

- **drones de Moyenne Altitude et Longue Endurance (MALE)** : ces véhicules volent en principe entre 10 000 et 30 000 pieds (de quelque 3 000 mètres à 9 100 mètres) d'altitude et pour une durée de 24 à 48 heures. La figure 1.5 présente les
-

drones MQ-9 Reaper et MQ1-Predator. Tous les deux sont développés par General Atomics Aeronautical (Nonami et al., 2010).



MQ-9 Reaper



MQ1-Predator

FIGURE 1.5 – Drones de type MALE (Nonami et al., 2010).

- **drones de Haute Altitude et Longue Endurance (HALE)** : ces drones volent plus longtemps et plus haut encore que les MALE. Ainsi, le plus connu étant le drone américain RQ-4 Global Hawk de la société Northrop Grumman. Cet aéronef peut voler jusqu'à 18 300 mètres d'altitude pendant 36 heures. La figure 1.6 présente les drones RQ-4B Global Hawk, construits par Northrop Grumman, et TAI Anka développés par Turkish Aerospace Industries.



RQ-4B Global Hawk



TAI Anka

FIGURE 1.6 – Drones de type HALE (Austin, 2010).

- **drones tactiques (TUAV)** : ils peuvent voler entre 100 et 300 km. Ces engins volants sont employés pour des missions précises ayant de dimensions bien plus petites que les drones MALE ou HALE. Cette catégorie se décompose elle-même en plusieurs sous-catégories où les micro-drones MAV constituent un exemple remarquable. Les MAV ont une autonomie de trentaines de minutes, une masse inférieure
-

à 500 gramme et pouvant être contenus dans une sphère de 30 centimètres. La figure 1.7 présente les drones WatchKeeper WK450, construits par la société Thales, et Sperwer développés par la compagnie Sagem Défense Sécurité.



WatchKeeper WK450



Sperwer

FIGURE 1.7 – Drones tactiques TUAV (Safran, 2014).

1.3.2 Classification selon le mode de propulsion

Les drones peuvent être classés en trois familles en fonction de leur configuration aérodynamique (García Carrillo et al., 2013) :

- **drones à voilure fixe** : ce sont des drones utilisant des ailes fixes dans leur mode de propulsion. D'une manière générale, si un avion nécessite une piste d'atterrissage, il est considéré comme un avion à voilure fixe, Figure 1.8.



FIGURE 1.8 – Drones à voilure fixe.

- **drones à voilure tournante** : ce sont des drones à décollage et à atterrissage verticaux capables d'effectuer un vol stationnaire à basse vitesse et à faible altitude. Ils utilisent un ou plusieurs rotors pour se sustenter et se propulser. Nous établissons un classement des drones à voilures tournantes en 4 catégories (Bertrand, 2007) ;

- **Mono-rotor** : ils se composent d'un rotor principal au-dessus qui permet la sustentation et la propulsion et un autre secondaire à la queue pour la stabilité, Figure 1.9(a) ;
- **Birotors contrarotatifs** : ils se composent de deux rotors coaxiaux ou tandems tournant en sens opposés et à même vitesse de rotation pour annuler les effets de couple et ainsi, faire l'économie du rotor anti-couple et d'augmenter la charge utile, Figure 1.9(b) ;
- **Quadri-rotors** : ils se composent de quatre rotors qui sont généralement placés aux extrémités d'une croix et répartis en deux groupes en diagonale. Leurs moteurs tournent par paire dans le même sens. Les hélices fixées sur des bras opposés tournent dans le sens horaire et les deux autres dans le sens antihoraire. Cela permet d'annuler le couple induit par l'effort de chaque moteur pour faire tourner ses hélices. On notera que deux types de configuration de quads existent : en "x" et en "+", Figure 1.9(c) ;
- **Multi-rotors** : ce sont les drones avec six ou huit rotors. Ils sont agiles et volent même si un moteur tombe en panne car il y a une redondance en raison du grand nombre des rotors. Ces drones sont illustrés dans la figure 1.9(d).
- **drones à voilure battante** : ils sont inspirés des oiseaux et des insectes volants. Ces drones ont de petites ailes et une très faible charge et endurance. D'autre part, ils ont une faible consommation d'énergie et peuvent effectuer des décollages et atterrissages verticaux ([Rakotomamonjy, 2006](#)), Figure 1.10.

1.4 Les drones convertibles

1.4.1 Typologie

Deux grandes catégories de véhicules aériens se présentent : les systèmes à voilure tournante dont la sustentation est assurée par la rotation des pales d'hélices et les systèmes à voilure fixe dont la sustentation est assurée par le flux d'air sur les ailes induit par le mouvement propre du véhicule.

Les véhicules de la première catégorie ont pour intérêt de pouvoir décoller verticalement et d'effectuer un vol stationnaire ce qui est très utile pour de diverses applications (inspection, surveillance, décollage à partir d'une zone restreinte, etc.). Cependant, ils ne peuvent pas voler vers l'avant à grande vitesse transportant des grandes charges utiles. En contrepartie, leur efficacité énergétique est bien moindre que celle des véhicules à voilure fixe. Les capacités de décollage/atterrissage d'aéronefs à voilure tournante et l'efficacité du vol avant de ceux à voilure fixe peuvent être combinées dans un seul avion. Ce véhicule aérien est appelé aéronef convertible ([Snyder, 2000](#); [Escareno et al., 2006, 2007](#)).

La transition entre ces deux phases du vol peut se réaliser par basculement complet du véhicule lorsque celui-ci est équipé d'un rotor surmontant une voilure (Vertigo de Supaéro) ([Poinsot, 2008](#)), ou uniquement par basculement de ses rotors lorsque ceux-ci sont disposés sur des nacelles pivotantes tel est le cas du prototype (Eagle Eye de Bell) ([Helicopter, 2016](#)), voir Figure 1.11



(a) Hélicoptère mono-rotor.



(b) Hélicoptère Birotors contrarotatifs.



(c) Quadri-rotors.



(d) Multi-rotors.

FIGURE 1.9 – Drones à voilure tournante.

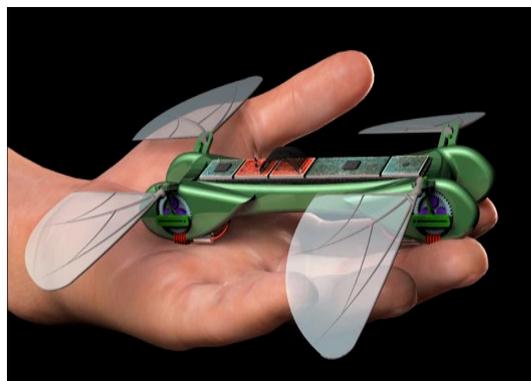
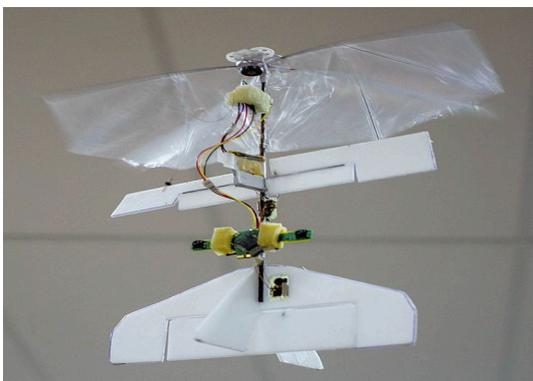
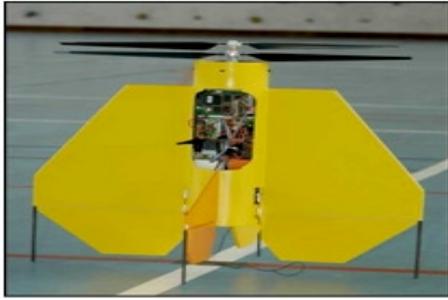


FIGURE 1.10 – Drones à voilure battante.



Vertigo de Supaéro (Poinsot, 2008)



Eagle Eye de Bell (Helicopter, 2016)

FIGURE 1.11 – Drones convertibles.

1.4.2 Les convertibles à basculement entier

C'est une famille des aéronefs convertibles qui ne possède pas de degré de liberté entre le bâti moteur et le reste de la structure. Les engins les plus connus de cette famille sont les aéronefs dénommés "tail-sitter" ou "tilt-body". En novembre 1954, le convair XFY-1 POGO a réussi pour la première fois de passer d'un vol vertical à un vol horizontal (Osborne, 2007). Comme montré dans la figure 1.12, le fuselage est posé verticalement sur le sol ayant la capacité de basculer le corps de l'appareil pour passer en mode du vol avion (Koehl, 2012).



FIGURE 1.12 – Modes du vol de XFY-1 POGO (Koehl, 2012).

1.4.3 Les convertibles à variation de structure

Ce sont les aéronefs convertibles qui possèdent des rotors ou des ailes basculants dénommés "tilt-rotor" ou "tilt-wing". Ces drones combinent ainsi le décollage vertical d'un hélicoptère à la vitesse de croisière et à la charge utile d'un avion.

Le Boeing-Bell-V22 Osprey, Figure 1.13, est un aéronef américain qui a été lancé en 1989 (Boeing, 2016). C'est le premier avion de type "tilt-rotor". Il s'agit du croisement entre un avion de transport militaire et un hélicoptère pouvant décoller et atterrir verticalement.



FIGURE 1.13 – Modes du vol de V-22 Osprey(Boeing, 2016).

Le type le plus célèbre du drone à rotors basculants est Eagle Eye fabriqué par la compagnie Bell (Helicopter, 2016). Eagle Eye possède des rotors avec un mécanisme d'inclinaison de son aile. Ceci est une caractéristique clé pour ce type de drone. Eagle Eye peut réaliser des décollages et atterrissages verticaux et le vol stationnaire avec une haute vitesse de croisière en changeant l'angle de la face du rotor par un mécanisme d'inclinaison. Cependant, l'inconvénient d'utiliser deux rotors, comme dans Eagle Eye, est qu'il est impossible de voler quand l'un de ces rotors est non fonctionnel. Par conséquent, il est souhaitable de disposer de quatre rotors. En outre, étant donné que seuls les rotors sont inclinés, l'efficacité de l'effet de souffle diminue parce que les ailes ne sont pas liées. Un nouveau design de drone convertibles, capable de résoudre les problèmes mentionnés ci-dessus, est développé. Ce type de drone, dit Quad Tilt-Wing (QTW), possède quatre rotors et des ailes avec un mécanisme d'inclinaison pour chacun, Figure 1.14. Plusieurs plate-formes expérimentales ont été réalisées selon cette structure de corps avec des ailes et de rotors basculants (Snyder, 2000; Papachristos et al., 2013; Oner et al., 2012; Nonami, 2007).



FIGURE 1.14 – Drone convertibles de type Quad tilt Wing.

Les recherches et les développements liés aux drones convertibles sont très actifs au cours des dernières années. Plusieurs centres de recherche et des groupes de spécialistes en aéronautique ont commencé à publier les premiers résultats liés à la modélisation et à la commande de ce type d'aéronefs tels que les travaux de l'université de Compiègne "BIROTAN" et "Quad-plan" (Kendoul et al., 2005; Fantoni, 2007; Gerardo and Lozano, 2013; Gerardo and Lozano, 2014) et ceux du TÜBITAK, le Conseil de la Recherche

Scientifique et Technologique en Turquie (Oner, 2009; Oner et al., 2008). Le tableau 1.1 dénombre des projets de recherche sur les drones convertibles.

Tableau 1.1 – Quelques projets de recherche sur les drones convertibles.

Projet	Université/Société	Configuration	Lancement
HARVee	 Université d'État de l'Arizona	Tilt-wing	2004
SUAVI	 Conseil de la Recherche Scientifique et Technologique en Turquie	Tilt-wing	2009
QTW UAS-FS4	 Université de Chiba, Japon	Tilt-wing	2006
Zero	 Agusta Westland	Tilt-rotor	2010
IAI Panther	 Israël Aerospace Industries	Tilt-rotor	2010

1.5 Drone convertible « Quad-Tilt Wing »

1.5.1 Structure mécanique et modes du vol

Le véhicule QTW étudié est équipé de quatre ailes montées à l'avant et à l'arrière du châssis mécanique, pouvant être tournées de la position verticale à l'horizontale et vice versa (Cetinsoy et al., 2012). La figure 1.15 montre l'architecture mécanique d'un tel prototype dans les différents modes du vol (horizontal, vertical et de transition). Avec

cette configuration, le fuselage du véhicule se transforme en un Quadrotor classique, si les ailes sont à la position verticale, et en une structure d'avion si les ailes sont à la position horizontale (Cetinsoy et al., 2012; BenKhoud et al., 2017; BenKhoud and Bouallègue, 2016b,a; Bouallègue and BenKhoud, 2016).

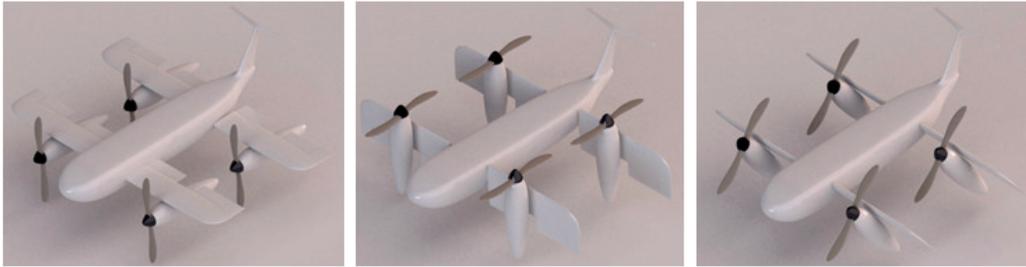


FIGURE 1.15 – Modes de vol du convertible QTW : horizontal, vertical et de transition.

En raison du profil de vol de cet aéronef convertible, nous pouvons distinguer les trois modes de fonctionnement, suivants (voir Figure 1.16) (Cetinsoy et al., 2012) :

- vol vertical : le vol ne repose que sur les rotors. Dans cette phase, le véhicule peut décoller et atterrir verticalement (VTOL). Les angles d'inclinaison des ailes et des rotors sont proches de 90° par rapport au plan horizontal ;
- transition : c'est le mode de fonctionnement intermédiaire qui relie le mode vertical au mode horizontal. C'est probablement la dynamique la plus complexe du drone ;
- vol horizontal : le véhicule se comporte comme un avion. Les angles d'inclinaison des ailes et des rotors sont presque nuls. Le drone gagne assez de vitesse pour générer des forces aérodynamiques pour soulever et contrôler son mouvement.

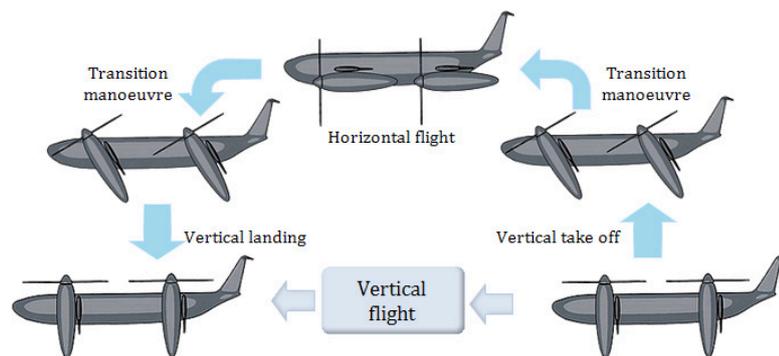


FIGURE 1.16 – Différents modes de vol du drone convertible QTW (BenKhoud et al., 2017).

1.5.2 Modélisation dynamique

1.5.2.1 Equations cinématiques

Le formalisme de Newton-Euler est utilisé pour établir un modèle dynamique non linéaire du drone étudié. Afin de simplifier cette étude, différentes hypothèses sont considérées (Cetinsoy et al., 2012; Abdessameud and Tayebi, 2013) :

- les flexibilités des ailes de l'aéronef et du fuselage sont négligées ;
- la force de trainée du fuselage est négligée ;
- le foyer aérodynamique et le centre de gravité coïncident ;
- la masse d'aéronef est considérée constante.

Afin d'établir les expressions mathématiques des forces et des moments aérodynamiques qui régissent le fonctionnement du drone QTW, nous retenons conformément à la figure 1.17 les deux repères suivants :

- un repère inertiel fixe, noté $\mathbf{R}_i (O_i, x_i, y_i, z_i)$, supposé galiléen avec z_i pointé vers le centre de la terre ;
- un repère lié au corps du véhicule, noté $\mathbf{R}_b (O_b, x_b, y_b, z_b)$, dont l'origine coïncide avec le centre de gravité, x_b définit l'axe longitudinal pointé vers l'avant, y_b définit l'axe transversal et z_b est l'axe vertical dirigé vers le haut.

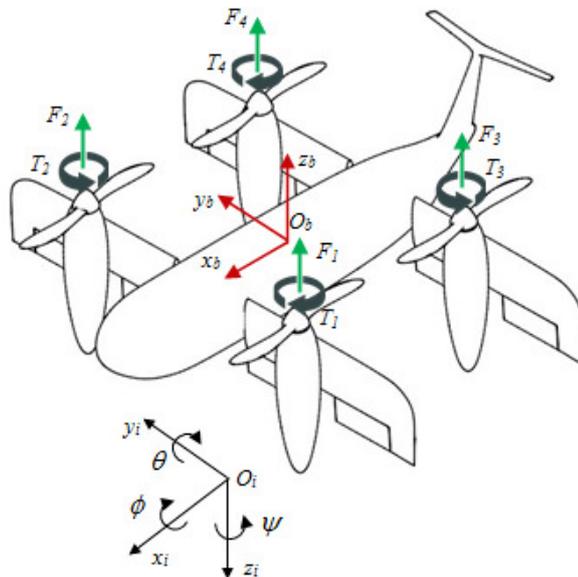


FIGURE 1.17 – Coordonnées et repères aérodynamiques du convertible QTW.

On désigne par $-\pi/2 \leq \phi \leq \pi/2$, $-\pi/2 \leq \theta \leq \pi/2$ et $-\pi \leq \psi \leq \pi$ les angles d'Euler relatifs respectivement aux mouvements de roulis, de tangage et de lacet. Ces angles définissent l'orientation d'un corps solide dans l'espace décrite par l'application

$\mathcal{R} : \mathbf{R}_i \rightarrow \mathbf{R}_b$ suivante :

$$\mathcal{R}(\phi, \theta, \psi) = \begin{pmatrix} c\phi c\theta & s\phi s\theta c\psi - s\phi c\theta & c\phi s\theta c\psi + s\psi s\phi \\ s\phi c\theta & s\phi s\theta s\psi + c\psi c\theta & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \quad (1.1)$$

avec $c = \cos(\cdot)$ et $s = \sin(\cdot)$.

L'attitude du drone par rapport au repère terrestre est définie par les angles repérant \mathbf{R}_b dans \mathbf{R}_i . Par convention, on reprend le système de Cardon (ϕ, θ, ψ) pour définir la matrice de rotation de l'équation (1.1). La position et la vitesse linéaire du centre de gravité G du drone dans le repère inertiel sont données respectivement par les équations (1.2) et (1.3) :

$$\boldsymbol{\xi} = (x, y, z)^T \in \mathbb{R}^3 \quad (1.2)$$

$$\dot{\boldsymbol{\xi}} = (\dot{x}, \dot{y}, \dot{z})^T \in \mathbb{R}^3 \quad (1.3)$$

De même, on définit respectivement l'attitude et la vitesse angulaire, dans le même référentiel \mathbf{R}_i , comme suit :

$$\boldsymbol{\eta} = (\phi, \theta, \psi)^T \in \mathbb{R}^3 \quad (1.4)$$

$$\dot{\boldsymbol{\eta}} = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T \in \mathbb{R}^3 \quad (1.5)$$

avec ϕ , θ et ψ sont les angles d'Euler qui définissent, respectivement, les mouvements de roulis, de tangage et de lacet.

Dans le repère \mathbf{R}_b , les vitesses angulaires de roulis, de tangage et de lacet sont définies par le vecteur $\boldsymbol{\vartheta} = (p, q, r)^T$. Ces vitesses sont liées à celles définies dans \mathbf{R}_i comme suit :

$$\boldsymbol{\vartheta} = \mathbf{T}(\phi, \theta) \boldsymbol{\eta} \quad (1.6)$$

où $\mathbf{T}(\phi, \theta)$ désigne la matrice de transformation de la vitesse (Voir Annexe A) définie comme suit :

$$\mathbf{T}(\phi, \theta) = \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{pmatrix} \quad (1.7)$$

Idem, pour les vitesses linéaires $\boldsymbol{\nu} = (v_x, v_y, v_z)^T$ dans le repère \mathbf{R}_b en retenant l'équation (1.8) suivante :

$$\boldsymbol{\nu} = \mathcal{R}^T(\phi, \theta, \psi) \dot{\boldsymbol{\xi}} \quad (1.8)$$

1.5.2.2 Equations dynamiques

Les dynamiques de position et d'attitude sont exprimées dans le repère du corps \mathbf{R}_b . Ainsi, en utilisant la formulation de Newton-Euler, la dynamique du véhicule est décrite comme suit :

$$\begin{pmatrix} m\mathbf{I} & 0 \\ 0 & \mathbf{J} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\nu}} \\ \dot{\boldsymbol{\vartheta}} \end{pmatrix} + \begin{pmatrix} 0 \\ \boldsymbol{\vartheta} \wedge \mathbf{J}\boldsymbol{\vartheta} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_t \\ \mathbf{M}_t \end{pmatrix} \quad (1.9)$$

\mathbf{I} est la matrice identité, m la masse du drone QTW et \mathbf{J} sa matrice d'inertie, \mathbf{F}_t et \mathbf{M}_t respectivement les forces extérieures et les moments agissant sur le drone dans le repère \mathbf{R}_b .

Les équations (1.9) décrivant le mouvement du corps rigide sont valables pour les vols horizontal, vertical et de transition. Cependant, les forces extérieures \mathbf{F}_t et les moments extérieurs \mathbf{M}_t sont définis en fonction des angles d'inclinaison des ailes $\gamma_{1,2,3,4}$, Figure 1.18.

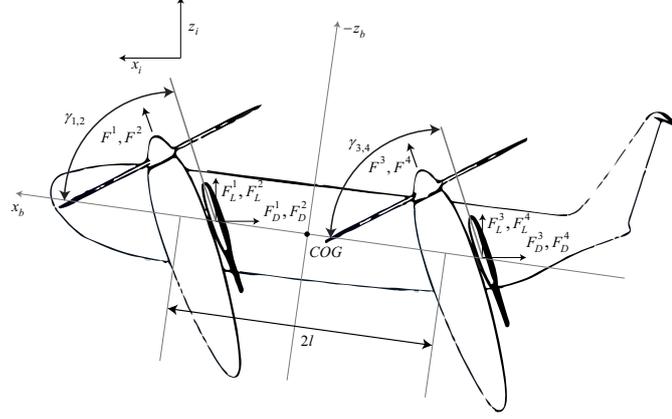


FIGURE 1.18 – Forces extérieures agissant sur le corps du convertible QTW.

1.5.2.3 Forces extérieures

La force totale extérieure \mathbf{F}_t agissant sur le drone se compose essentiellement des forces de poussée \mathbf{F} créées par les rotors, des forces aérodynamiques \mathbf{F}_a générées par les ailes, de la force de gravité \mathbf{F}_g et des perturbations externes \mathbf{F}_p comme les rafales de vent. Etant donné que ces forces sont exprimées dans le repère \mathbf{R}_b , elles doivent être transformées moyennant la matrice de rotation $\mathcal{R}(\phi, \theta, \psi)$ dans le repère inertiel \mathbf{R}_i pour avoir :

$$\mathbf{F}_t = \mathbf{F}_g + \mathbf{F} + \mathbf{F}_a + \mathbf{F}_p \quad (1.10)$$

- **Forces de poussée** : ce sont les forces générées par les systèmes de propulsion de l'avion. Pour un aéronef à hélices, ce type des forces s'exerce suivant l'axe du rotor. Ces forces de poussée \mathbf{F} sont proportionnelles aux carrés des vitesses de rotation des moteurs ω_i^2 ainsi que l'angle d'inclinaison γ_i . La force de poussée de l'i-ème moteur est définie comme suit :

$$F_i = k\omega_i^2 \quad (1.11)$$

avec $i \in \{1, 2, 3, 4\}$ et $k > 0$ est le coefficient de poussée pouvant être déterminé moyennant des essais expérimentaux.

En tenant compte des angles d'inclinaison γ_i , les forces de poussée pour tous les modes du vol sont exprimées comme suit :

$$\mathbf{F} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = \mathcal{R}(\phi, \theta, \psi) \begin{pmatrix} c\gamma_1 & c\gamma_2 & c\gamma_3 & c\gamma_4 \\ 0 & 0 & 0 & 0 \\ -s\gamma_1 & -s\gamma_2 & -s\gamma_3 & -s\gamma_4 \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} \quad (1.12)$$

- **Force de gravité** : c'est la force agissant sur le véhicule volant. Il s'agit d'un vecteur fixe tout au long de l'axe z_i du repère inertiel \mathbf{R}_i . Cette force est définie dans ce même repère comme suit :

$$\mathbf{F}_g = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \quad (1.13)$$

g étant la constante de gravité.

- **Forces aérodynamiques** : elles sont créées par l'ensemble des pressions locales dues à la vitesse de l'air autour de l'avion en déplacement. Afin d'intégrer le processus de transition, les forces de portance $F_L(\gamma_i, v_x, v_z)$ et de traînée $F_D(\gamma_i, v_x, v_z)$ sont modélisées non seulement en fonction de la vitesse longitudinale v_x et celle de décollage/atterrissage v_z du véhicule mais également en fonction des angles d'inclinaison γ_i et d'attaque α_i (voir Figure 1.19). Ces forces sont ainsi exprimées dans le repère du corps comme suit (Cetinsoy et al., 2012) :

$$\begin{pmatrix} F_D^i \\ 0 \\ F_L^i \end{pmatrix} = \mathbf{R}(\gamma_i - \alpha_i) \begin{pmatrix} -\frac{1}{2}C_D\rho v_{tot}^2 S \\ 0 \\ -\frac{1}{2}C_L\rho v_{tot}^2 S \end{pmatrix} \quad (1.14)$$

ρ est la densité de l'air, S est la surface de l'aile, C_D et C_L représentent respectivement les coefficients de traînée et de portance, $\mathbf{R}(\gamma_i - \alpha_i)$ dénote la matrice de rotation d'angle $(\gamma_i - \alpha_i)$ par rapport à l'axe y .

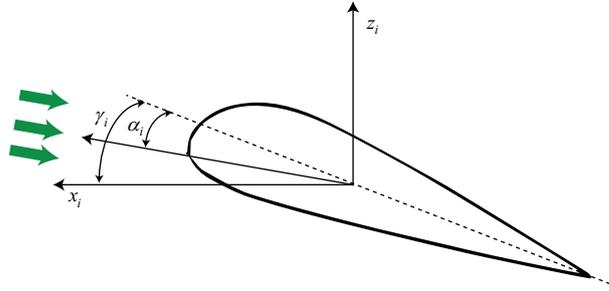


FIGURE 1.19 – Angle d'attaque de profil d'aile.

L'angle d'attaque α_i ainsi que la vitesse de flux v_{tot} d'air sont respectivement exprimés comme suit :

$$\alpha_i = \gamma_i + \arctan 2(v_x, v_z) \quad (1.15)$$

$$v_{tot} = \sqrt{v_x^2 + v_z^2} \quad (1.16)$$

v_x et v_z sont les composantes du vecteur de vitesse linéaire (1.8).

Ainsi, la somme des forces de portance et de traînée \mathbf{F}_a , exprimées dans le repère du corps de drone, est donnée par l'expression (1.17) suivante :

$$\mathbf{F}_a = \begin{pmatrix} F_{a,x} \\ F_{a,y} \\ F_{a,z} \end{pmatrix} = \mathbf{R}(\phi, \theta, \psi) \begin{pmatrix} F_D^1 + F_D^2 + F_D^3 + F_D^4 \\ 0 \\ F_L^1 + F_L^2 + F_L^3 + F_L^4 \end{pmatrix} \quad (1.17)$$

Afin de simplifier la modélisation de ce véhicule aérien, nous supposons que les ailes en avant et en arrière tournent ensemble avec un même angle d'inclinaison tel que $\gamma_{1,2} = \gamma_{3,4} = \gamma$.

1.5.2.4 Moments extérieurs

Le moment extérieur total \mathbf{M}_t est la somme du moment \mathbf{M} induit par les forces de poussée des rotors, celui généré par les forces aérodynamiques \mathbf{M}_a , des moments \mathbf{M}_{gyro}^h et \mathbf{M}_{gyro}^c générés par les effets gyroscopiques et des moments \mathbf{M}_p dûs aux perturbations externes, supposées nuls durant nos développements. Ce moment total est exprimé comme suit :

$$\mathbf{M}_t = \mathbf{M} + \mathbf{M}_a + \mathbf{M}_{gyro}^h + \mathbf{M}_{gyro}^c + \mathbf{M}_p \quad (1.18)$$

- **Moments de poussée** : ces moments sont constitués de deux composantes comme indiqué dans (1.19). La première est la somme des couples créés par les rotors qui sont proportionnels à la poussée. La deuxième composante est en raison du déséquilibre des forces de poussée agissant en différents angles d'inclinaison et différentes amplitudes.

$$\mathbf{M} = l \begin{pmatrix} s\gamma_1 & -s\gamma_2 & s\gamma_3 & -s\gamma_4 \\ s\gamma_1 & s\gamma_2 & -s\gamma_3 & -s\gamma_4 \\ c\gamma_1 & -c\gamma_2 & c\gamma_3 & -c\gamma_4 \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} + \begin{pmatrix} -c\gamma_1 & -c\gamma_2 & -c\gamma_3 & -c\gamma_4 \\ 0 & 0 & 0 & 0 \\ s\gamma_1 & s\gamma_2 & s\gamma_3 & s\gamma_4 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix} \quad (1.19)$$

l représente la distance entre les rotors et le centre de gravité du véhicule, $T_i = \tau_i F_i$ les couples créés par les rotors proportionnels aux forces de poussée et τ_i les coefficients couple/force qui dépendent de la géométrie des hélices. Pour le sens de rotation horaire des hélices, nous prenons $\tau_{2,3} = \tau$, et pour les hélices tournant dans le sens antihoraire, nous retenons $\tau_{1,4} = -\tau$.

- **Moments aérodynamiques** : ce sont les couples dus au déséquilibre des forces de portance et de traînée créées par les ailes. En supposant que ces forces agissent sur un seul point de la surface de l'aile, les couples aérodynamiques peuvent être exprimés comme suit :

$$\mathbf{M}_a = \begin{pmatrix} M_\phi \\ M_\theta \\ M_\psi \end{pmatrix} = l \begin{pmatrix} F_L^1 - F_L^2 \\ (F_L^1 + F_L^2) - (F_L^3 + F_L^4) \\ F_L^2 - F_L^1 \end{pmatrix} \quad (1.20)$$

- **Moments gyroscopiques** les effets gyroscopiques sont dûs aux changements du plan de rotation d'un objet tournant. Les couples gyroscopiques ont un effet dynamique plus significatif sur les objets qui tournent avec des vitesses très grandes. Pour cette raison, les effets gyroscopiques dûs aux rotations des hélices et aux mouvements du corps de drone doivent être inclus dans le modèle dynamique. Les moments gyroscopiques des hélices et du corps sont définis comme suit :

$$\mathbf{M}_{gyro}^h = \sum_{i=1}^4 J_r \left(\mu_i \boldsymbol{\vartheta} \wedge \begin{pmatrix} c\gamma_i \\ 0 \\ -s\gamma_i \end{pmatrix} \omega_i \right) \quad (1.21)$$

$$\mathbf{M}_{gyro}^c = \boldsymbol{\vartheta} \wedge \mathbf{J}\boldsymbol{\vartheta} \quad (1.22)$$

$\mu_{1,2,3,4} = (1, -1, -1, 1)$ et J_r est l'inertie des hélices par rapport à l'axe z .

1.5.2.5 Dynamiques des rotors

Les moteurs électriques les plus utilisés pour l'entraînement des hélices dans les drones à rotors en tandem multiples sont de type Brushless DC. En régime permanent, ces actionneurs électriques présentent une dynamique semblable à celle des moteurs à courant continu conventionnels (Bouallègue and Fessi, 2016).

Chaque moteur M_i est couplé à une hélice avec effet d'anti-battement. La tension de l'induit, comme entrée de commande de ces moteurs, est donnée comme suit :

$$v_i = \frac{R_{mot}}{k_{mot}} J_r \dot{\omega}_i + k_{mot} \omega_i + d R_{mot} \omega_i^2 \quad (1.23)$$

R_{mot} et k_{mot} sont respectivement la résistance d'induit du moteur et le coefficient du couple, d étant le coefficient de traînée.

Comme le coefficient de traînée est généralement très faible, nous pouvons approximer cette dynamique à celle d'un système du premier ordre dont la fonction de transfert est donnée par :

$$H_i(s) = \frac{W_i(s)}{V_i(s)} = \frac{K}{1 + \tau s} \quad (1.24)$$

$K = 1/k_{mot}$ étant le gain statique et $\tau = R_{mot} J_r / k_{mot}^2$ la constante de temps. Ces paramètres peuvent être déterminés expérimentalement, moyennant des méthodes classiques d'identification.

Ainsi, le drone QTW est piloté moyennant la variation indépendante des vitesses de rotation des différents moteurs Brushless DC et les entrées effectives de commande étant définies comme suit :

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = k \begin{pmatrix} 1 & 1 & 1 & 1 \\ l & -l & l & -l \\ l & l & -l & -l \\ \tau & -\tau & -\tau & \tau \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \quad (1.25)$$

On observe que l'entrée u_1 représente la force de poussée totale qui agit sur le corps du drone selon l'axe z , les entrées u_2 , u_3 et u_4 représentent respectivement les moments de roulis, de tangage et de lacet permettant les rotations du drone dans l'espace autour des axes x , y et z .

1.5.2.6 Modèle dynamique dans l'espace d'état

En exprimant toutes les forces extérieures, agissant sur le corps du véhicule dans le repère inertiel, et à partir des équations de mouvement selon le formalisme de Newton-Euler (Austin, 2010; Lozano, 2010), le modèle dynamique complet du convertible QTW est établi comme suit :

$$m\ddot{\boldsymbol{\xi}} = \mathbf{F}_t \quad (1.26)$$

$$\mathbf{J}\dot{\boldsymbol{\vartheta}} = -\mathbf{M}_t + \boldsymbol{\vartheta} \wedge \mathbf{J}\boldsymbol{\vartheta} \quad (1.27)$$

$\mathbf{J} = \text{diag}(J_{xx}, J_{yy}, J_{zz})$ est la matrice d'inertie du robot et J_{xx} , J_{yy} et J_{zz} sont les inerties du corps respectivement par rapport aux axes x , y et z .

En remplaçant dans les équations (1.26) et (1.27) chaque terme par son expression, précédemment établie, on obtient les équations différentielles suivantes qui définissent les mouvements de translation et de rotation du drone :

$$\left\{ \begin{array}{l} \ddot{x} = \frac{1}{m} ((c\psi c\theta c\gamma - (c\phi s\theta c\psi + s\phi s\psi) s\gamma) u_1 + F_{a,x}) \\ \ddot{y} = \frac{1}{m} ((s\psi c\theta c\gamma - (c\phi s\theta s\psi - s\phi c\psi) s\gamma) u_1 + F_{a,y}) \\ \ddot{z} = \frac{1}{m} ((-s\theta c\gamma - c\phi c\theta s\gamma) u_1 + F_{a,z}) + g \\ \dot{p} = \frac{u_2}{J_{xx}} + \frac{J_{yy} - J_{zz}}{J_{xx}} qr - \frac{J_r}{J_{xx}} q\omega_r s\gamma + M_\phi \\ \dot{q} = \frac{u_3}{J_{yy}} + \frac{J_{zz} - J_{xx}}{J_{yy}} pr + \frac{J_r}{J_{yy}} (ps\gamma + rc\gamma) \omega_r + M_\theta \\ \dot{r} = \frac{u_4}{J_{zz}} + \frac{J_{xx} - J_{yy}}{J_{zz}} pq - \frac{J_r}{J_{zz}} qc\gamma \omega_r + M_\psi \end{array} \right. \quad (1.28)$$

$\omega_r = \omega_1 - \omega_2 - \omega_3 + \omega_4$ représente la vitesse angulaire résiduelle des rotors et u_1 , u_2 , u_3 et u_4 sont les entrées de commande du véhicule.

Durant le vol vertical ($\gamma = \pi/2$), le convertible QTW se comporte comme un Quadrotor classique et ne repose que sur ses rotors pour la sustentation. Les forces et les moments aérodynamiques sont négligés. Dans ce cas, le mouvement du drone est accompli par la variation des vitesses de rotation des rotors provoquant ainsi la variation des forces de poussée produites. Le QTW s'incline ainsi vers la direction du rotor le plus lent et prend ensuite un mouvement de translation suivant cet axe de rotor. Les dynamiques de position et d'attitude sont alors régies par le modèle dynamique local suivant :

$$\left\{ \begin{array}{l} \ddot{x} = \frac{1}{m} (-c\phi s\theta c\psi - s\phi s\psi) u_1 \\ \ddot{y} = \frac{1}{m} (-c\phi s\theta s\psi + s\phi c\psi) u_1 \\ \ddot{z} = -\frac{1}{m} c\phi c\theta u_1 + g \\ \dot{p} = \frac{u_2}{J_{xx}} + \frac{J_{yy} - J_{zz}}{J_{xx}} qr - \frac{J_r}{J_{xx}} q\omega_r \\ \dot{q} = \frac{u_3}{J_{yy}} + \frac{J_{zz} - J_{xx}}{J_{yy}} pr + \frac{J_r}{J_{yy}} p\omega_r \\ \dot{r} = \frac{u_4}{J_{zz}} + \frac{J_{xx} - J_{yy}}{J_{zz}} pq \end{array} \right. \quad (1.29)$$

En retenant $X = (\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, z, \dot{z}, x, \dot{x}, y, \dot{y})^T \in \mathbb{R}^{12}$ comme vecteur état du modèle

dynamique, ce dernier peut se mettre sous la forme suivante :

$$\dot{X} = f(X, u) = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_4 \omega_r + b_1 u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_3 x_2 x_6 + a_4 x_2 \omega_r + b_2 u_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_5 x_2 x_4 + b_3 u_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = -\frac{cx_1 cx_3}{m} u_1 + g \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = -\frac{1}{m} (cx_1 sx_3 cx_4 + sx_1 sx_4) u_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = \frac{1}{m} (-cx_1 sx_3 sx_4 + sx_1 cx_4) u_1 \end{cases} \quad (1.30)$$

avec : $a_1 = (J_{yy} - J_{zz})/J_{xx}$, $a_2 = -J_r/J_{xx}$, $a_3 = (J_{zz} - J_{xx})/J_{yy}$, $a_4 = J_r/J_{yy}$, $a_5 = (J_{xx} - J_{yy})/J_{zz}$, $b_1 = 1/J_{xx}$, $b_2 = 1/J_{yy}$ et $b_3 = 1/J_{zz}$.

En retenant la même modélisation des rotors que pour le cas d'un drone Quadrotor, nous pouvons retenir finalement le schéma synoptique de la figure 1.20 qui présente la modélisation complète du véhicule étudié.

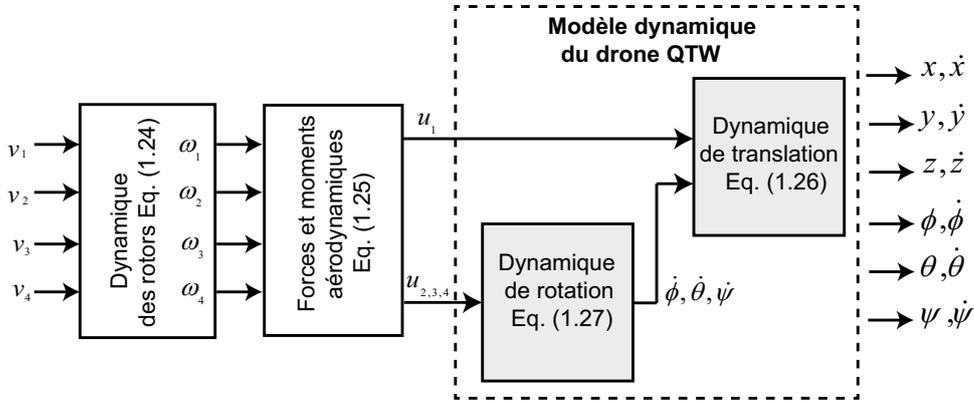


FIGURE 1.20 – Schéma blocs pour le modèle dynamique du drone QTW.

1.6 Modélisation T-S de systèmes non linéaires

1.6.1 Principe

Au cours de ces dernières décennies, les modèles T-S ont été largement utilisés pour la représentation de systèmes non linéaires. Ce type de modélisation permet de représenter un système non linéaire à l'aide de plusieurs sous-modèles linéaires pondérés par le biais de fonctions non linéaires satisfaisant la propriété de convexité (Tanaka et al., 1998a), (Sala,

2009) et (Allouche, 2016). Le passage du modèle non linéaire à sa représentation T-S est effectué soit en linéarisant autour de plusieurs points de fonctionnement (Johansen et al., 2000) soit en appliquant l'approche par secteurs non linéaires (Ohtake et al., 2003).

La modélisation T-S consiste en l'appréhension du comportement non linéaire dans le temps d'un système dynamique par un ensemble des modèles locaux le caractérisant dans différentes zones de fonctionnement. Par conséquent, les différents multi-modèles s'écrivent sous forme d'une interpolation entre des modèles linéaires invariants dans le temps (LTI) pondérés par le biais de « Fonctions d'appartenance » (Tanaka and Sugeno, 1992; Tanaka and Wang, 2001; Hamouda et al., 2013, 2014, 2016; Allouche, 2016; Youssef et al., 2017).

1.6.2 Représentation d'état d'un modèle T-S

Un modèle flou T-S peut représenter exactement le comportement d'un système non linéaire. Afin de mettre en place ce type de modélisation, un système non linéaire peut être écrit sous forme d'un modèle d'état continu avec des paramètres variables (modèle quasi-linéaire à paramètres variants) comme suit :

$$\begin{aligned}\dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) \\ \dot{y}(t) &= C(\theta)x(t)\end{aligned}\tag{1.31}$$

où A , B et C sont les matrices d'état du système, θ est le vecteur contenant les variables de prémisses qui dépendent à la fois de l'état du système x et des entrées de commande u (Huang and Jadbabaie, 1999; Leith and Leithead, 2002; Orjuela et al., 2008, 2009; Nagy et al., 2010; Chadli and Karimi, 2013). Une condition importante qui doit être respectée dans la constitution des modèles T-S est la bornitude du vecteur des variables de décision :

$$\theta^{min} \leq \theta \leq \theta^{max}\tag{1.32}$$

où θ^{min} et θ^{max} sont respectivement les valeurs minimale et maximale de θ . Le modèle flou est alors obtenu à partir d'un nombre fini de sous-modèles LTI bornés et interconnectés grâce à des fonctions satisfaisant la propriété de somme convexe comme suit :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{N_{lti}} \mu_i(\theta) (A_i(x(t))x(t) + B_i(x(t))u(t)) \\ y(t) = \sum_{i=1}^{N_{lti}} \mu_i(\theta) C_i(x(t))x(t) \end{cases}\tag{1.33}$$

avec N_{lti} le nombre des sous-modèles LTI, $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_u \times n_x}$ et $C_i \in \mathbb{R}^{n_y \times n_x}$ des matrices invariantes dans le temps décrivant la dynamique du système et $\mu_i(\theta)$ représente le degré d'activation du $i^{\text{ème}}$ sous-modèle associé.

Selon la zone où évolue le système, la fonction $\mu_i(\theta)$ indique la contribution plus au moins importante du modèle local vis-à-vis du modèle global. Ces fonctions d'activation sont non linéaires, normalisées et peuvent être de formes sinusoïdale, triangulaire ou gaussienne vérifiant la propriété de convexité suivante :

$$\sum_{i=1}^{N_{lti}} \mu_i(\theta) = 1\tag{1.34}$$

avec

$$0 \leq \mu_i(\theta) \forall i = 1, 2, \dots, N_{lti}$$

Un modèle T-S est décrit alors par des règles floues de la forme « **si-alors** » comme suit :

$$R_i : \mathbf{Si} \theta_1(t) \text{ est } F_1^i(\theta_1(t)) \mathbf{Et} \theta_2(t) \text{ est } F_2^i(\theta_2(t)) \mathbf{Et} \dots \theta_p(t) \text{ est } F_p^i(\theta_p(t))$$

Alors

$$\begin{cases} \dot{x}(t) = A_i(x(t))x(t) + B_i(x(t))u(t) \\ y(t) = C_i(x(t))x(t) \end{cases} \quad (1.35)$$

où R_i représente la $i^{\text{ème}}$ règle floue, F_j^i pour $j \in \{1, 2, \dots, p\}$ les fonctions d'appartenance des ensembles flous, $(\theta_1(t), \theta_2(t), \dots, \theta_p(t))$ constitue le vecteur des variables de prémisses θ pouvant être définie en fonction des variables d'état, des entrées, ou d'une combinaison des deux et des variables mesurées ou non.

Les fonctions d'activation μ_i dépendent à la fois du vecteur des variables de prémisses $\theta(t) = [\theta_1(t), \theta_2(t), \dots, \theta_p(t)]$ et du choix de l'opérateur logique. Dans la modélisation T-S, l'opérateur "**Et**" est souvent choisi comme étant le produit, par conséquent :

$$\mu_i(\theta(t)) = \prod_{j=1}^p F_j^i(\theta_j(t)) \quad (1.36)$$

avec $\mu_i(\theta(t)) > 0, \forall t \geq 0$.

Finalement, le modèle T-S global du système (1.6.2) est obtenu en utilisant la somme pondérée suivante (Ichalal, 2009; Hamouda et al., 2016) :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{N_{lti}} \mu_i(\theta(t)) \left(A_i(x(t))x(t) + B_i(x(t))u(t) \right) \\ \dot{y}(t) = \sum_{i=1}^{N_{lti}} \mu_i(\theta(t)) \left(C_i(x(t))x(t) \right), \quad i \in 1, \dots, N_{lti} \end{cases} \quad (1.37)$$

1.6.3 Approches d'obtention d'un modèle T-S

Dans la littérature, trois approches différentes peuvent être employées pour obtenir une représentation multi-modèle sous forme T-S d'un modèle non linéaire (Oudghiri, 2008; Ichlal, 2009; Hamouda et al., 2016; Allouche, 2016) :

1.6.3.1 Multi-modèle par identification

Cette technique est appropriée dans le cas des systèmes sans modèles de connaissance où les modèles considérés sont de type boîte noire (Tanaka and Sugeno, 1992). Cette approche est développée moyennant à des techniques d'identification des sous-systèmes correspondant aux modèles locaux définis aux différents points de fonctionnement. La construction d'un multi-modèle par identification à partir des données disponibles d'entrées/sorties passe généralement par les quatre étapes suivantes (Gasso, 2000) :

- définition de la structure ;
- définition des fonctions d'appartenance ;
- estimation des modèles locaux et des paramètres des fonctions d'appartenance (les méthodes d'optimisation numériques sont employées dans cette étape) ;
- validation du modèle final (évaluation de la représentativité des multi-modèles).

1.6.3.2 Multi-modèle par linéarisation

Cette approche est basée sur la linéarisation d'un système non linéaire à partir de son modèle mathématique autour d'un nombre donné de points de fonctionnement méticuleusement choisis. Le modèle non linéaire doit être représentatif du système étudié vis-à-vis de sa dynamique et de son évolution physique au cours du temps. Ensuite, une interpolation entre les différents modèles locaux obtenus par le biais des fonctions non linéaires doit être établie afin d'obtenir la structure T-S globale (Akhenak, 2004; Nagy et al., 2010).

1.6.3.3 Multi-modèle par secteurs non linéaires

Cette approche a été initialement proposée dans (Wang et al., 1996; Tanaka et al., 1996). Son avantage majeur est de réduire au mieux le nombre de modèles locaux et d'assurer une erreur d'approximation minimale (Youssef et al., 2017). Contrairement aux approches citées précédemment, cette méthode permet effectivement d'obtenir une structure T-S décrivant exactement la dynamique du système étudié. L'approche T-S par secteurs non linéaires repose sur les bornes inférieures et supérieures des termes non linéaires, comme décrit dans le Lemme 1.6.1. Son principe de construction est basé sur une Transformation Polytopique Convexe (TPC) des termes non linéaires du modèle (Tanaka and Wang, 2001; Morère, 2001; Ohtake et al., 2003; Allouche, 2016).

Lemme 1.6.1. *Soit $\theta_j(x, u), j \in \{1, 2, \dots, p\}$ l'ensemble des fonctions non linéaires et bornées sur un domaine compact $D \subset \mathbb{R}^{n_1} \times \mathbb{R}^{m_1}$ à valeurs dans \mathbb{R} avec $x \in \mathbb{R}^{n_1}$ et $u \in \mathbb{R}^{m_1}$. Il existe deux fonctions ($v = 1, 2$) telles que :*

$$\mu_v : D \mapsto [0, 1] \quad (1.38)$$

$$(x, u) \mapsto \mu_v(x, u) \quad (1.39)$$

Les fonctions $\theta_j(x, u)$ peuvent être formulées sous forme convexe comme suit :

$$\theta(x, u) = \mu_1(x, u)\theta^{max} + \mu_2(x, u)\theta^{min} \quad (1.40)$$

avec :

$$\begin{cases} \theta^{max} = \max_{(x,u) \in D} \{\theta(x, u)\} \\ \theta^{min} = \min_{(x,u) \in D} \{\theta(x, u)\} \end{cases} \quad (1.41)$$

sachant que :

$$\mu_1(x, u) + \mu_2(x, u) = 1 \quad (1.42)$$

Les fonctions de pondération μ_1 et μ_2 sont déterminées à partir de la relation suivante :

$$\mu_1 = \frac{\theta(x, u) - \theta^{min}}{\theta^{max} - \theta^{min}}; \quad \mu_2 = 1 - \mu_1 = \frac{\theta^{max} - \theta(x, u)}{\theta^{max} - \theta^{min}} \quad (1.43)$$

□

La structure obtenue du multi-modèle T-S selon l'approche des secteurs non linéaires n'est pas unique (Allouche, 2016). En effet, en changeant le choix du vecteur des variables de prémisses, les propriétés de ces dernières changent et une nouvelle représentation T-S peut être formulée. Toujours selon la même approche de construction des modèles T-S, le nombre fini des modèles locaux linéaires obtenus est $N_{lti} = 2^r$ où r est un entier naturel désignant le nombre de variables de prémisses. Par conséquent, le nombre de règles floues et le nombre de termes non linéaires sont liés d'une manière exponentielle. Cette liaison peut être néfaste surtout lors de la manipulation de systèmes non linéaires complexes. Un nombre de règles élevé peut conduire à des problèmes de calcul importants (Tanaka and Wang, 2001; Morère, 2001; Ohtake et al., 2003).

Dans la suite de ce manuscrit, l'intérêt est porté à la modélisation via la troisième approche d'obtention des modèles T-S. Ce choix est justifié par ses avantages majeurs du point de vue précision et connaissance des fonctions d'activation qui assurent l'interconnexion entre les différents modèles locaux LTI.

1.6.4 Stabilité des systèmes T-S

La stabilité des systèmes non linéaires écrits sous forme d'une représentation d'état T-S a été largement étudiée dans la littérature (Ichalal et al., 2009; Djeghaba, 2012; Bezzaoucha, 2013; Chadli and Karimi, 2013; Hamouda, 2015; Allouche, 2016). Grâce à l'écriture particulière d'un modèle T-S, l'extension des techniques d'étude de stabilité des systèmes linéaires au cas des systèmes non linéaires est rendue possible. Une technique souvent utilisée pour cette étude est basée sur la théorie de Lyapunov. La formulation des Inégalités Matricielles Linéaires (LMI) est employée afin de garantir des conditions suffisantes assurant la stabilité des systèmes T-S comme décrit dans le théorème 1.6.1 :

Théorème 1.6.1. (Tanaka and Sugeno, 1992; Tanaka and Wang, 2001)

Soit un modèle T-S autonome ($u(t) = 0$) donné par :

$$\dot{x}(t) = \sum_{i=1}^{N_{lti}} \mu_i(\theta) (A_i x(t)) \quad (1.44)$$

Le système (1.44) est dit asymptotiquement stable s'il existe une matrice P tel que $P = P^T > 0$ vérifiant les LMIs suivantes pour $i \in \{1, 2, \dots, N_{lti}\}$:

$$A_i^T P + P A_i < 0 \quad (1.45)$$

□

Soit la fonction candidate de Lyapunov suivante :

$$V(x(t)) = x(t)^T P x(t) \quad (1.46)$$

avec $P = P^T > 0$.

Un recours à la propriété de la somme convexe des fonctions de Lyapunov, formulées en termes LMIs, permet d'obtenir les N_{lti} conditions suffisantes pour la stabilité du système (1.44) pouvant être résolues simultanément.

Le système (1.33) autonome est stable si :

$$\dot{V}(x(t)) = \dot{x}^T(t) P x(t) + x^T(t) P \dot{x}(t) < 0 \quad (1.47)$$

ce qui est équivalent à :

$$\dot{V}(x(t)) = \dot{x}^T(t) \left(\sum_{i=1}^{N_{lti}} \mu_i(\theta(t)) (A_i^T P + P A_i) \right) x(t) < 0 \quad (1.48)$$

D'après (Tanaka et al., 1998a), l'interconnexion des sous-modèles stables n'est pas forcément stable. En effet, il a été vérifié, dans le cas où le nombre de sous-modèles locaux N_{lti} est élevé, que la détermination d'une matrice commune P satisfaisant en même temps tous les N_{lti} LMIs devient un problème difficile à résoudre. D'autres travaux publiés ultérieurement ont porté sur la réduction du conservatisme des conditions imposées par le Théorème 1.6.1. En effet, l'utilisation des fonctions de Lyapunov polyquadratiques de la forme (1.49) permet tout de même d'assurer la stabilité des systèmes non linéaires autonomes (1.33) mais avec moins de restrictions (Johansson, 1999; Chadli, 2002; Tanaka et al., 2003) :

$$\dot{V}(x(t)) = x(t)^T \sum_{i=1}^{N_{lti}} \mu_i(\theta(t)) (P_i x(t)) \quad (1.49)$$

L'avantage de l'utilisation des fonctions de Lyapunov polyquadratiques est de calculer plusieurs matrices P_i au lieu d'une seule matrice commune P . Par conséquent, les contraintes de stabilité du Théorème 1.6.1 sont relaxées. Partant de l'idée des fonctions continues par morceaux, une fonction candidate de Lyapunov non quadratique a été introduite et définie de la manière suivante :

$$V(x(t)) = \max\{V_1(x(t)), V_2(x(t)), \dots, V_{N_{lti}}(x(t))\} \quad (1.50)$$

avec :

$$V_i(x(t)) = x(t)^T P_i x(t), \quad P_i = P_i^T > 0, \text{ pour } i = \{1, 2, \dots, N_{lti}\} \quad (1.51)$$

Ce type des fonctions a été exploité lors de l'étude de la stabilité des systèmes non linéaires à paramètres variant dans le temps (Boyd et al., 1994). Elles ont aussi fait l'objet d'application dans le cas des systèmes flous (Cao et al., 1999; Johansson, 1999; Zhang et al., 2001) et pour les systèmes non linéaires décrits en T-S (Johansson, 1999; Chadli, 2002). Le Théorème 1.6.2 suivant donne ainsi les conditions de stabilité de l'utilisation de ces fonctions de Lyapunov particulière :

Théorème 1.6.2. (*Johansson, 1999*)

Le modèle T-S autonome (1.44) est stable, s'il existe des matrices symétriques et définies positives P_j et des scalaires strictement positifs τ_{ijk} tels que :

$$A_i^T P_j + P_j A_i + \sum_{k=1}^{N_{li}} \tau_{ijk} (P_j - P_k) < 0 \quad (1.52)$$

$$\text{avec } \begin{cases} P_j > 0 \\ \tau_{ijk} > 0 \end{cases} \text{ et } i, j = \{1, 2, \dots, N_{li}\}.$$

□

Ces conditions de stabilité sont moins restrictives que celles issues de la stabilité en utilisant les fonctions quadratiques. Par contre, le problème de résolution devient plus compliqué. En effet, l'introduction des fonctions quadratiques conduit à l'apparition d'inégalités matricielles bilinéaires (*Tanaka and Wang, 2001; Chadli, 2002; Tanaka et al., 2003; Guerra et al., 2006; Yoneyama, 2008*)

1.6.5 Stabilisation des systèmes T-S

Le problème de stabilisation des systèmes non linéaires décrits sous forme T-S consiste à trouver une loi de commande $u(t)$ afin que le système (1.33) satisfasse les performances souhaitées en boucle fermée. De nombreux travaux dans la littérature s'intéressent à l'étude de la stabilisation des systèmes non linéaires décrits en multi-modèles T-S (*Takagi and Sugeno, 1985; Tanaka et al., 1998b; Guerra and Vermeiren, 2001; Kim and Lee, 2000; Chadli et al., 2001*). Vu que l'étude de la stabilité des systèmes T-S est souvent basée sur la technique de Lyapunov quadratique, qui, à son tour, exige des hypothèses restrictives et conduit à des conditions de stabilité suffisantes (*Tanaka and Sugeno, 1992*), l'emploi de la formulation convexe sous forme LMI facilite la résolution de tels problèmes (*Boyd et al., 1994*). Cependant, l'intégration d'un nombre élevé de contraintes et le souhait de trouver en même temps une matrice définie positive limite le nombre de solutions. Par conséquent, l'étude de la stabilité par la technique de Lyapunov n'est pas toujours possible.

Dans la suite, le système multi-modèle (1.33) est supposé localement observable et commandable. Plusieurs techniques ont été proposées afin d'assurer la stabilisation des systèmes T-S. Nous citons, dans la suite, quelques-unes :

1. **Stabilisation par la commande PDC** : la technique de commande PDC (en anglais Parallel Distributed Compensation) est utilisée pour déterminer une loi de commande $u(t)$ pour les systèmes T-S lorsque les variables d'état sont toutes disponibles (*Wang et al., 1996; Tanaka et al., 1996, 1998a; Wudhichai et al., 2004*). Le principe est de calculer un régulateur par retour d'état relatif à chaque modèle LTI local. La loi de commande globale est obtenue alors par agrégation des lois de commande locales linéaires (*Oudghiri, 2008*).
2. **Stabilisation par retour de sortie** : dans le cas où l'état du système T-S n'est pas disponible à tout instant, une approche de synthèse par retour de sortie (statique ou

dynamique) est proposée pour la stabilisation (Cao et al., 1998; Kar, 1999; Chadli et al., 2002; Lo and Lin, 2003; Kau et al., 2007; Huang and Nguang, 2007; Wu, 2008).

3. **Stabilisation par contrôleur H_∞** : cette technique à base d'observateurs rend possible le calcul des gains à la fois du contrôleur et de l'observateur d'état. Le principe de cette approche est de concevoir une commande PDC basée sur un observateur qui assure la stabilité en boucle fermée du système T-S sous contraintes H_∞ (Xiaodong and Qingling, 2003; Lin et al., 2005; Oudghiri et al., 2007).
4. **Stabilisation par poursuite de trajectoires selon la norme L_2** : cette technique utilise l'écriture de la dynamique d'erreur d'estimation d'état sous la forme d'un système incertain. Les incertitudes sont supposées bornées ou encore constantes. Cette technique fait partie des approches de stabilisation des systèmes T-S à base d'un observateur. Le gain de cet observateur est déterminé de telle manière qu'il assure la stabilité du système sous des contraintes L_2 (Bara, 2001; Weinmann, 2012; Hamouda, 2015).

1.6.6 Modélisation T-S d'un drone convertible QTW

Considérons le modèle dynamique (1.53) qui définit les mouvements de rotation du drone de type QTW en mode de vol vertical :

$$\begin{cases} \ddot{\phi} = \frac{\tau_\phi}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \\ \ddot{\theta} = \frac{\tau_\theta}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\phi} \dot{\psi} \\ \ddot{\psi} = \frac{\tau_\psi}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \end{cases} \quad (1.53)$$

où ϕ , θ et ψ représentent respectivement les angles d'attitude du drone (roulis, tangage et lacet), l la distance entre les rotors et le centre de gravité, I_{xx} , I_{yy} et I_{zz} sont les moments d'inertie du système respectivement sur les axes x , y et z , les entrées de commande $u_{2;3;4}$ sont explicitement décrites comme suit :

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} kl & -kl & kl & -kl \\ kl & kl & -kl & -kl \\ k\lambda & -k\lambda & -k\lambda & k\lambda \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (1.54)$$

avec k le coefficient de portance, λ le rapport couple/force et ω_i la vitesse angulaire du i -ème moteur.

Pour un vecteur d'état $x = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]^T$ et des entrées de commande $\tau_{\phi;\theta;\psi}$, le système est décrit par une représentation d'état de la forme :

$$\begin{cases} \dot{x} = A(x)x + Bu \\ y = Cx \end{cases} \quad (1.55)$$

avec :

$$A(x) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\dot{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\dot{\phi} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\dot{\theta} & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 \end{bmatrix}$$

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}}; a_2 = \frac{I_{zz} - I_{xx}}{I_{yy}}; a_3 = \frac{I_{xx} - I_{yy}}{I_{zz}}$$

A partir de $A(x)$, deux fonctions non linéaires sont distinguées $(\dot{\phi}, \dot{\psi})$. Donc le vecteur des prémisses choisi $z(x(t)) = [\dot{\phi}, \dot{\psi}]$ et dimension $p = 2$.

La seconde étape consiste à appliquer la transformation polytopique convexe, présentée précédemment, à chaque variable de prémisses $z_i(x(t))$. D'après le lemme 1.6.1, nous réalisons ensuite pour chacune des variables de prémisses, une partition en deux zones pour obtenir :

$$z_1(x(t)) = F_0^1(z_1(x)) \underline{z}_1 + F_1^1(z_1(x)) \bar{z}_1 \quad (1.56)$$

$$z_2(x(t)) = F_0^2(z_2(x)) \underline{z}_2 + F_1^2(z_2(x)) \bar{z}_2 \quad (1.57)$$

où :

$$\begin{cases} F_0^1 = \frac{\bar{z}_1 - z_1(x(t))}{\bar{z}_1 - \underline{z}_1} \\ F_1^1 = \frac{z_1(x(t)) - \underline{z}_1}{\bar{z}_1 - \underline{z}_1} \\ F_0^2 = \frac{\bar{z}_2 - z_2(x(t))}{\bar{z}_2 - \underline{z}_2} \\ F_1^2 = \frac{z_2(x(t)) - \underline{z}_2}{\bar{z}_2 - \underline{z}_2} \end{cases} \quad (1.58)$$

telles que :

$$\bar{z}_p = \max \{z_p(x(t))\} \quad \text{et} \quad \underline{z}_p = \min \{z_p(x(t))\}, \quad \forall p = 1, 2 \quad (1.59)$$

En faisant toutes les combinaisons possibles, $\mu_1 = F_0^1 F_0^2$, $\mu_2 = F_0^1 F_1^2$, $\mu_3 = F_1^1 F_0^2$ et $\mu_4 = F_1^1 F_1^2$, le modèle non linéaire est écrit sous la forme de quatre sous-modèles linéaires ($r = 2^p = 2^2$) inter-connectés avec les fonctions d'appartenance μ_i , $i \in \{1, 2, 3, 4\}$, où les matrices constantes sont données par :

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1 \underline{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2 \underline{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3 \underline{z}_2 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1 \bar{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2 \bar{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3 \bar{z}_2 & 0 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1 \bar{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2 \bar{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3 \bar{z}_2 & 0 & 0 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1 \bar{z}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2 \bar{z}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3 \bar{z}_2 & 0 & 0 \end{bmatrix}$$

Dans le but de valider le modèle TS développé, des signaux aléatoires de type bruit blanc de moyenne nulle sont utilisés comme des entrées, pour le système non linéaire et le modèle T-S du QTW. Les figures 1.21-1.26 montrent la conformité des réponses en attitude obtenues avec le modèle TS développé avec celles obtenues avec le système non linéaire du drone. Sur la base de ces résultats, nous pouvons retenir le modèle T-S développé est représentatif du modèle non linéaire.

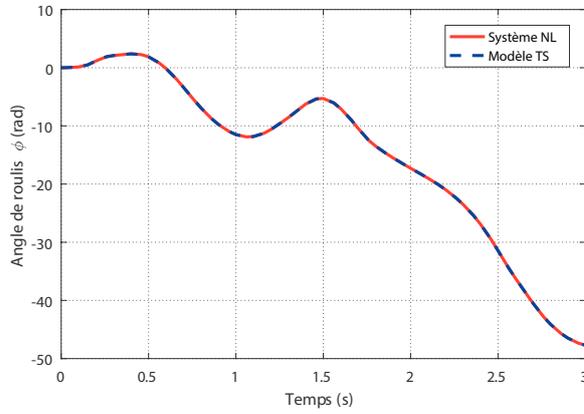


FIGURE 1.21 – Réponses du système non linéaire et modèle TS : angle de roulis.

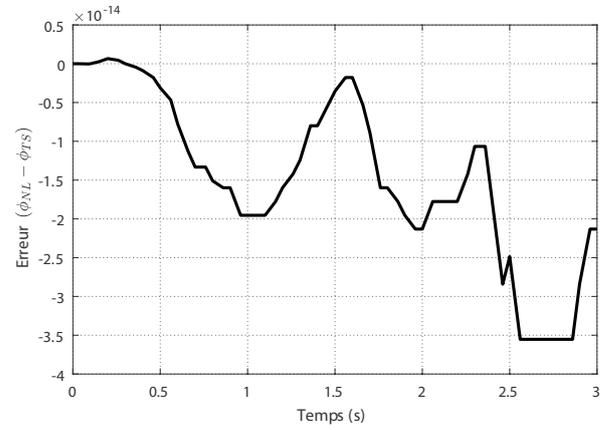


FIGURE 1.22 – Erreur $\phi_{NL} - \phi_{TS}$.

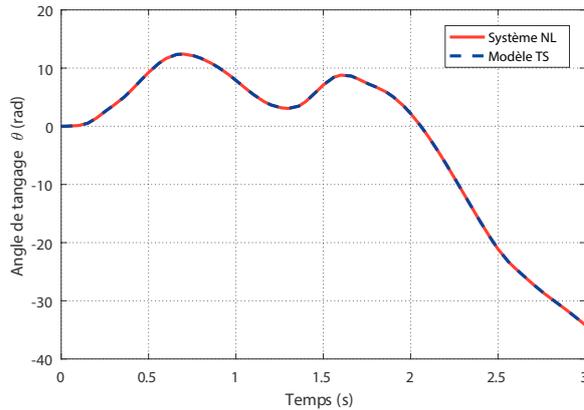


FIGURE 1.23 – Réponses du système non linéaire et modèle TS : angle de tangage.

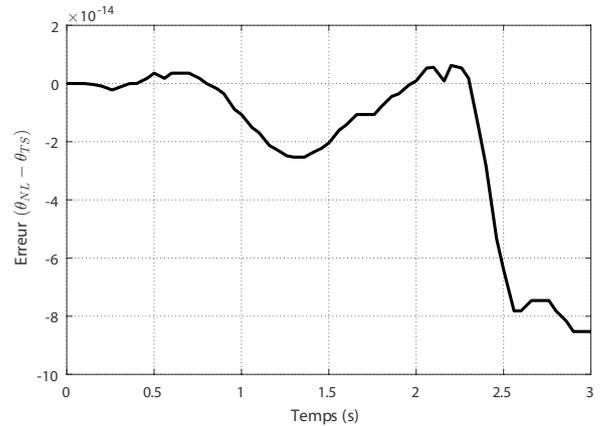


FIGURE 1.24 – Erreur $\theta_{NL} - \theta_{TS}$.

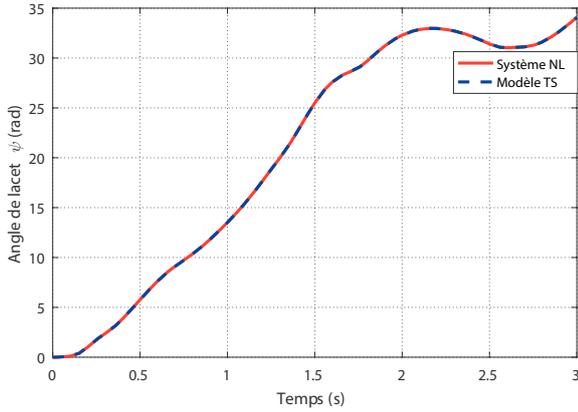


FIGURE 1.25 – Réponses du système non linéaire et modèle TS : angle de lacet.

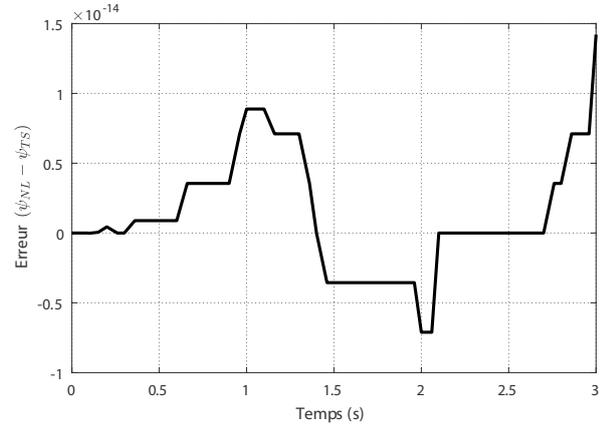


FIGURE 1.26 – Erreur $\psi_{NL} - \psi_{TS}$.

1.7 Critères de performance considérés

Dans l'objectif d'établir une étude comparative entre quelques méthodes de réglage des paramètres des lois de commande publiées dans la littérature et celles que nous proposons en réponse à une consigne de type échelon, nous avons considéré les critères de performances suivants :

1. **Le degré de stabilité (Stability Degree Index, SDI)** : Ce critère est utilisé pour évaluer la stabilité du système en boucle fermée défini à partir de la différence entre le rayon du cercle unité et le module du pôle le plus éloigné du centre du cercle unité comme suit (Mamoundou and Langlois, 2011) :

$$SDI(k) = 1 - \max(|p_1|, |p_2|, \dots, |p_{A_N}|), \quad (1.60)$$

où $\{p_1, p_2, \dots, p_{A_N}\}$ sont les valeurs propres de la matrice d'état du système en boucle fermée A_{cl} définie par :

$$A_{cl} = A - BK_{mpc} \quad (1.61)$$

Par conséquent, le système en boucle fermée est stable si son SDI appartient à l'intervalle ouvert $]0, 1[$. C'est-à-dire que le système sera considéré en limite de stabilité si le SDI prend une valeur nulle. De même, il sera considéré instable si le SDI devient de valeur strictement négative.

2. **La VARiance du signal de commande (VARiance of control signal, VARU)** : Ce critère permet d'observer la valeur moyenne des écarts quadratiques de $u(k)$ de sa moyenne comme suit (Abdenmour et al., 2001; Mamoundou and Langlois, 2011) :

$$VARU(k) = \bar{u}(k)^2 - [\bar{u}(k)]^2 \quad (1.62)$$

où \bar{u} est la valeur moyenne de u .

3. **Le temps de montée (Rise Time, RT)** : entre 10% à 90%.

4. **Le temps de réponse (Settling Time, ST)** : choisi égal à 5%.
5. **Le temps de pic (Peak Time, PT)** : le temps mis par le système pour atteindre le premier pic du dépassement.
6. **Le dépassement (Overshoot, OV)** : Ce critère présente le dépassement maximal sur les signaux de sortie.
7. **La moyenne de l'erreur de poursuite (MTE)** : Ce critère est utilisé pour évaluer la précision du système à chaque pas d'échantillonnage.
8. **L'énergie du signal de commande (Control Signal Energy, CSE)** ([Bagheri and Khaki-Sedigh, 2010](#)) :

$$CSE = \sum_{k=0}^{ST} u^2(k) \quad (1.63)$$

9. **L'énergie de l'effort du signal de commande (en anglais Control Effort Energy ou CEE)** ([Bagheri and Khaki-Sedigh, 2010](#)) :

$$CEE = \sum_{k=0}^{ST} \Delta u^2(k) \quad (1.64)$$

1.8 Conclusion

Ce chapitre a présenté un aperçu général sur les drones et leur classification selon l'endurance et le type de voilure. Le drone convertible de type QTW a été retenu comme étant le système à commander par des approches classiques et intelligentes. La structure mécanique ainsi que les mouvements possibles et les cycles du vol de cet aéronef sont présentés. Un modèle dynamique du véhicule a été développé en détaillant les trois modes de vol possibles : vertical, horizontal et de transition. Le modèle dynamique non linéaire ainsi obtenu a été mis sous forme d'une représentation d'état.

Une modélisation dynamique de ce drone a été introduite en tenant compte des différents paramètres de vol et des forces permettant de représenter cette dynamique dans les différentes phases de vol. Après avoir défini le modèle dynamique dans l'espace d'état, une modélisation de type T-S du système non linéaire a été proposée. La comparaison des résultats de simulation de ce modèle avec le modèle non linéaire, montre la fiabilité de l'approche adoptée.

Le chapitre suivant sera consacré à la commande drones convertibles QTW dans le but de proposer des lois de commande robuste permettant le contrôle et la stabilisation du drone.

Commande de drones convertibles QTW

Sommaire

2.1	Introduction	33
2.2	Commande linéaire quadratique	33
2.2.1	Formulation du problème	33
2.2.2	Résultats de simulation et discussion	34
2.3	Commande par backstepping avec action intégrale	35
2.3.1	Contrôle de l'attitude	35
2.3.2	Contrôle de la position et de l'altitude	39
2.3.3	Résultats et discussion : cas de la stabilisation des dynamiques d'attitude et d'altitude	40
2.3.4	Résultats et discussion : cas de la poursuite de trajectoires en vol vertical	40
2.4	Commande prédictive à base de modèle	45
2.4.1	Principe	45
2.4.2	Choix des paramètres de réglage	46
2.4.3	Représentation d'état augmentée à temps discret	47
2.4.4	Formulation de la commande en absence de contraintes	48
2.4.5	Formulation de la commande en présence de contraintes	50
2.4.6	Application à la commande du drone QTW	54
2.4.7	Résultats de simulation et discussion	55
2.5	Conclusion	59

2.1 Introduction

Les drones convertibles de type QTW ont montré un intérêt grandissant dans les recherches en robotique aérienne. Comme montré dans le premier chapitre, ce type de drone présente une bonne solution aux problèmes de vol mentionnés pour les véhicules à voilure fixe et ceux à voilure tournante.

Ce deuxième chapitre est consacré à la commande du drone convertible QTW retenu dans nos travaux. Diverses lois de commande relevant des approches de commande linéaire quadratique (LQ), la commande par backstepping incorporant une action intégrale ainsi qu'une commande prédictive à base de modèle (MPC) sont synthétisées pour cet aéronef afin de stabiliser son vol vertical.

2.2 Commande linéaire quadratique

2.2.1 Formulation du problème

La commande linéaire quadratique LQ est une stratégie de contrôle optimale et robuste (Zhou et al., 1996). L'idée de cette approche de commande est de déterminer un retour d'état \mathbf{K} qui minimise le critère de performance J suivant :

$$J(x, u) = \int_0^{\infty} x^T(t) \mathbf{Q} x(t) + u^T(t) \mathbf{R} u(t) dt \quad (2.1)$$

avec $\mathbf{Q} = \mathbf{Q}^T > 0$ et $\mathbf{R} = \mathbf{R}^T > 0$ les matrices de pondération sur l'état et la commande du système qui définissent la dynamique du correcteur LQR.

Une loi de commande linéaire quadratique LQ minimisant le critère (2.1) est alors donnée par :

$$u(t) = -\mathbf{K}x(t) \quad (2.2)$$

avec \mathbf{K} le gain du correcteur donné par :

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (2.3)$$

La matrice \mathbf{P} définie positive dans (2.3) est la solution de l'équation algébrique de RICCATI donnée par :

$$\mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P} + \mathbf{Q} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} = 0 \quad (2.4)$$

Le choix des matrices de pondération \mathbf{Q} et \mathbf{R} dans ((2.1)) se fait selon le comportement souhaité du système en boucle fermée. Plus la matrice \mathbf{Q} est grande, plus l'énergie de l'état x sera minimisée au détriment de la commande u qui est maximisée. La dynamique de la boucle fermée est rapide dans ce cas. Au contraire, plus la pondération \mathbf{R} est grande, plus l'énergie de la commande u sera minimisée, et plus la dynamique de la boucle fermée sera lente.

Bien que le vecteur d'état du drone QTW contient 12 variables en mode VTOL, ce véhicule aérien est un système sous-actionné avec 4 degrés de liberté. Par conséquent, les sorties à commander du système sont la position $\boldsymbol{\xi} = (x, y, z)$ et l'angle de lacet ψ .

Pour la conception du contrôleur LQ, les équations du modèle de véhicule sont linéarisées autour du point de fonctionnement (x^*, u^*) de l'équation (2.5).

$$(u^*, x^*) : \begin{cases} u_1^* = -g \\ u_{2,3,4}^* = 0 \\ x_{1,2,3,4,5,6,7,9,11}^* = 0 \\ x_{8,10,12}^* = \text{constante} \end{cases} \quad (2.5)$$

Les matrices du système linéarisé A , B et C sont alors calculées comme suit :

$$A = \begin{bmatrix} \frac{\partial f_1(X,u)}{\partial X_1|_{x=x^*}} & \cdots & \frac{\partial f_1(X,u)}{\partial X_{12}|_{x=x^*}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{12}(X,u)}{\partial X_1|_{x=x^*}} & \cdots & \frac{\partial f_{12}(X,u)}{\partial X_{12}|_{x=x^*}} \end{bmatrix} \quad (2.6)$$

$$B = \begin{bmatrix} \frac{\partial f_1(X,u)}{\partial u_1|_{u=u^*}} & \cdots & \frac{\partial f_1(X,u)}{\partial u_4|_{u=u^*}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{12}(X,u)}{\partial u_1|_{u=u^*}} & \cdots & \frac{\partial f_{12}(X,u)}{\partial u_4|_{u=u^*}} \end{bmatrix} \quad (2.7)$$

$$C = I_{12} \quad (2.8)$$

2.2.2 Résultats de simulation et discussion

A partir de la position $\xi_0 = (0, 0, 0)^T$ et l'attitude $\eta_0 = (0, 0, 0)^T$ initiales du véhicule, les résultats de simulation sous l'environnement MATLAB sont donnés dans les figures 2.1 et 2.2. Ces résultats montrent la réponse indicielle en attitude et en position du drone avec des entrées de référence $\xi_{ref} = (3, 3, 5)^T$ et $\eta_{ref} = (0, 0, \frac{\pi}{2})^T$.

Moyennant une procédure d'essais-erreurs, nous choisissons les matrices de pondération Q et R de la commande LQ comme suit :

$$Q = 10^{-1} \times I_{12} \quad (2.9)$$

$$R = \begin{bmatrix} 10^{-2} & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad (2.10)$$

Avec ces matrices de pondération, nous obtenons le gain K de retour d'état de l'équation (2.11) :

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -3.16 & -5.94 & 0 & 0 & 0 & 0 \\ 1.72 & 0.88 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.31 & -0.73 \\ 0 & 0 & 1.54 & 0.71 & 0 & 0 & 0 & 0 & 0.31 & 0.70 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.31 & 0.43 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.11)$$

Dans ce qui suit, nous retenons, pour les simulations, les paramètres physiques du tableau 2.1 relatifs au prototype réel utilisé dans (Cetinsoy et al., 2012).

Tableau 2.1 – Paramètres physiques du drone convertible QTW étudié (Cetinsoy et al., 2012).

Paramètres	Description	Valeur/unité
m	Masse du drone	1.250 kg
l	Distance entre l'hélice et le COG	0.20 m
J_{xx}	Inertie du drone selon l'axe x	1.353e-03 kg.m ²
J_{yy}	Inertie du drone selon l'axe y	1.353e-03 kg.m ²
J_{zz}	Inertie du drone selon l'axe z	1.706e-03 kg.m ²
J_r	Inertie d'hélices selon l'axe z	2.838e-05 kg.m ²
τ	Coefficient couple/force	0.01 Nm/N
k	Coefficient de portance du corps	2.92e-03 N/s ²
g	Constante de gravité	9.81 m/s ²

Sur la base de ces résultats, les performances de l'algorithme de commande LQ sont satisfaisantes. La loi de commande calculée permet de stabiliser le drone autour du point de fonctionnement désiré avec une dynamique de poursuite relativement lente mais précise. L'avantage de cette approche de commande réside dans sa simplicité de mise en œuvre. Cependant, cette technique présente un inconvénient majeur qui réside dans le nombre important des capteurs devraient être installés pour la commande. En effet, toutes les variables d'état du système devant être disponibles, ce qui n'est pas toujours le cas pour des systèmes complexes comme les drones QTW. Aussi, le problème de choix des pondération Q et R présente un inconvénient, qui peut rendre la synthèse de commande LQ difficile, non systématique et gourmande en temps de calcul.

2.3 Commande par backstepping avec action intégrale

Le modèle du véhicule QTW de l'équation (1.30) est sous-actionné, fortement couplé et non linéaire. La structure de commande par backstepping proposée est divisée en deux boucles en cascade comme montrées dans la figure 2.3. Une boucle interne pour la stabilisation de l'attitude et une autre externe pour les dynamiques de position x et y , sont implémentées (Bouallègue and BenKhoud, 2016; Bouabdallah and Siegwart, 2007; Zhou and Wen, 2008). Les entrées de référence définies pour la dynamique d'attitude sont générées à travers les trajectoires de roulis et de tangage, notées respectivement par ϕ_d et θ_d .

2.3.1 Contrôle de l'attitude

Considérons le sous-système dynamique suivant relatif à l'angle de roulis ϕ depuis le modèle d'état complet de l'équation (1.30) :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_4 \omega_r + b_1 u_2 \end{cases} \quad (2.12)$$

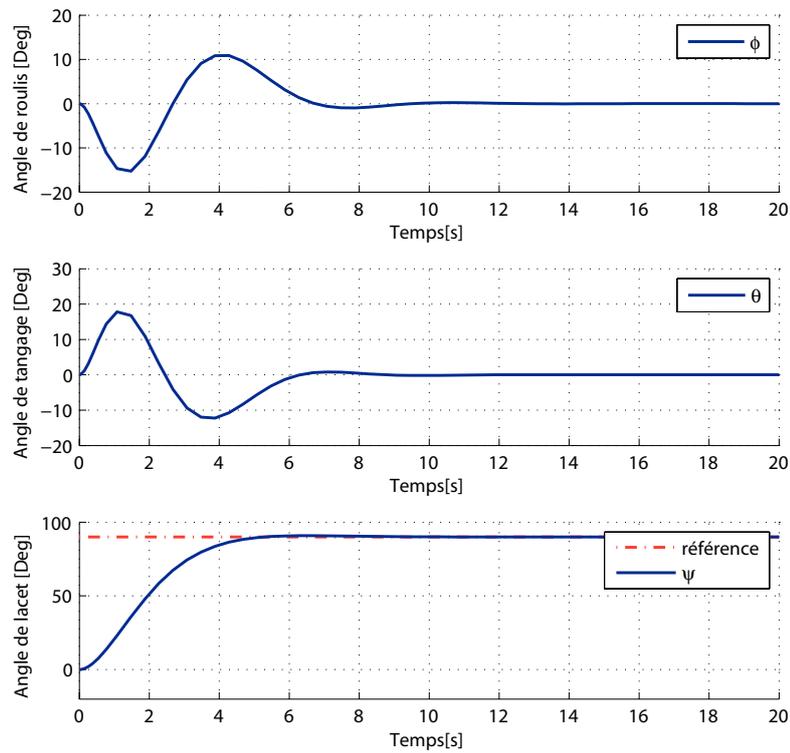


FIGURE 2.1 – Réponses indicielles en attitude pour la régulation LQ du drone QTW.

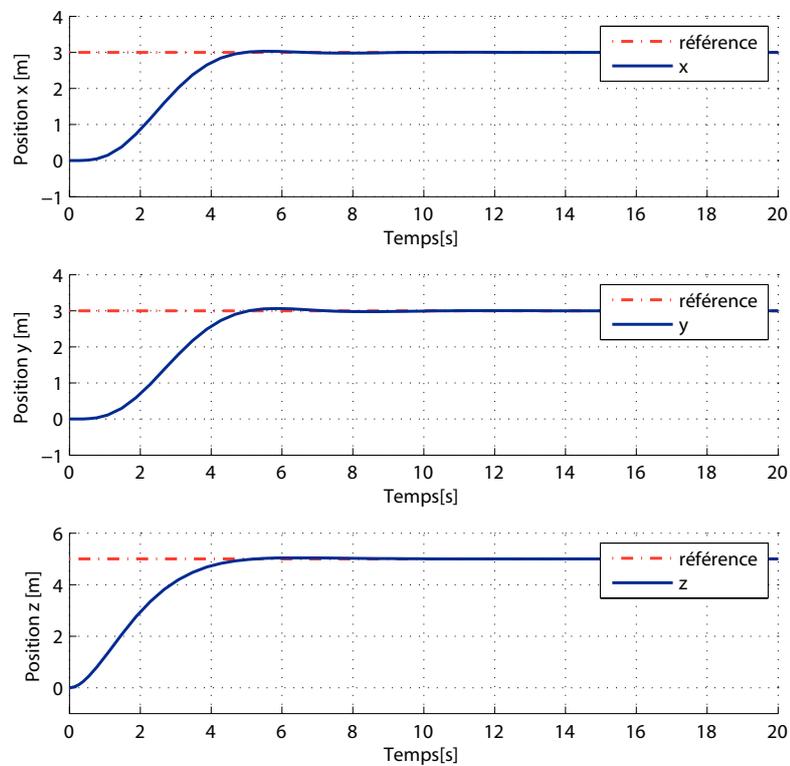


FIGURE 2.2 – Réponses indicielles en position pour la régulation LQ du drone QTW.

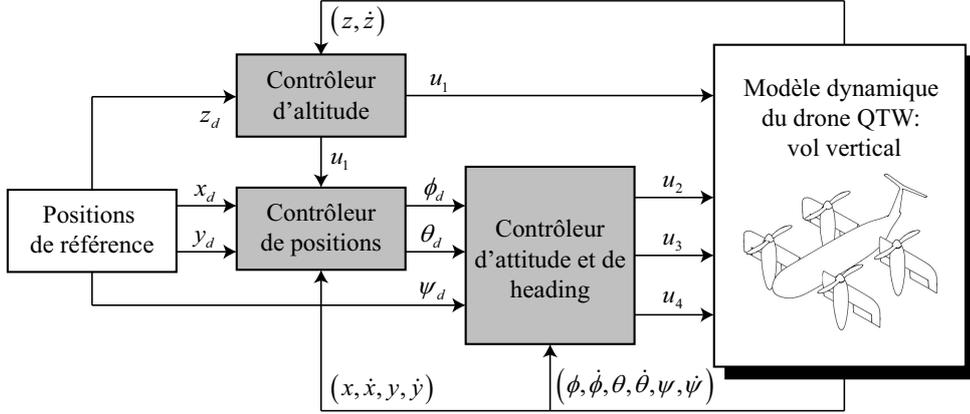


FIGURE 2.3 – Schéma de commande par backstepping à action intégrale du convertibile QTW.

Les étapes de synthèse d'une commande par Backstepping pour la dynamique de roulis sont présentées dans ce qui suit (Bouallègue and BenKhoud, 2016) :

- **Etape 1** : L'erreur de poursuite pour l'angle de roulis et sa dérivée temporelle peuvent être écrites respectivement comme suit :

$$e_1 = x_1^d - x_1 \quad (2.13)$$

$$\dot{e}_1 = \dot{x}_1^d - \dot{x}_1 = \dot{x}_1^d - x_2 \quad (2.14)$$

où x_1^d est la trajectoire désirée de la variable d'état x_1 , ici l'angle de roulis $\phi(t)$.

Une fonction quadratique candidate de Lyapunov V_1 , définie positive et à dérivée temporelle \dot{V}_1 semi-définie négative, est choisie comme suit (Krstic et al., 1995) :

$$V_1 = \frac{1}{2} (e_1^2 + \lambda_1 h_1^2) \quad (2.15)$$

$$\dot{V}_1 = e_1 \dot{e}_1 + \lambda_1 h_1 \dot{e}_1 = e_1 (\dot{x}_1^d - x_2 + \lambda_1 h_1) \quad (2.16)$$

où $\lambda_1 > 0$ est une constante à déterminer et $h_1 = \int_0^t e_1(v) dv$ représente l'intégrale de l'erreur de poursuite sur l'angle de roulis.

Selon le théorème de Krasovskii-La Salle (Kanellakopoulos et al., 1991; Krstic et al., 1995) et pour satisfaire $\dot{V}_1 < 0$, on introduit une entrée virtuelle de commande x_2^d qui, si elle est suivie par la variable réelle x_2 , permet d'avoir un système du premier ordre stabilisable ($e_1 \rightarrow 0$). Une telle entrée de commande est définie comme suit :

$$x_2^d = \dot{x}_1^d + \beta_1 e_1 + \lambda_1 h_1 \quad (2.17)$$

où β_1 est une constante positive qui détermine la vitesse de convergence de l'erreur de poursuite.

Remplaçons la commande virtuelle désirée x_2^d par son expression, l'équation (2.16) devient :

$$\dot{V}_1 = -\beta_1 e_1^2 < 0 \quad (2.18)$$

avec $\beta_1 > 0$ et $e_1 \neq 0$.

- **Etape 2** : Retournons à l'équation (2.17), il est clair que le terme x_2 provient de la dynamique de roulis de l'équation (2.12). Par conséquent, pour garantir que la variable d'état converge vers sa valeur désirée x_2^d , une deuxième boucle de commande doit être placée en définissant une nouvelle variable d'erreur e_2 comme suit :

$$e_2 = x_2 - x_2^d = x_2 - (\dot{x}_1^d + \beta_1 e_1 + \lambda_1 h_1) \quad (2.19)$$

Sa dérivée temporelle est donnée par :

$$\dot{e}_2 = \dot{x}_2 - \ddot{x}_1^d - \beta_1 \dot{e}_1 - \lambda_1 e_1 \quad (2.20)$$

Une autre fonction candidate de Lyapunov, notée V_2 , est choisie pour la dynamique de roulis du second ordre comme suit :

$$V_2 = V_1 + \frac{1}{2} (e_2^2 + \lambda_2 h_2^2) \quad (2.21)$$

$$\dot{V}_2 = \dot{V}_1 + e_2 \dot{e}_2 + \lambda_2 h_2 \dot{e}_2 \quad (2.22)$$

où $\lambda_2 > 0$ est une constante à déterminer et $h_2 = \int_0^t e_2(v) dv$ représente l'intégrale de l'erreur de poursuite sur la vitesse de roulis.

En réécrivant les équations (2.14) et (2.20) donnant les termes d'erreur de poursuite \dot{e}_1 et \dot{e}_2 comme suit :

$$\dot{e}_1 = \dot{x}_1^d - x_2 = -e_2 - \beta_1 e_1 - \lambda_1 h_1 \quad (2.23)$$

$$\dot{e}_2 = (a_1 x_4 x_6 + a_2 x_4 \omega_r + b_1 u_2) - \ddot{x}_1^d + \alpha_1 (e_2 + \beta_1 e_1 + \lambda_1 h_1) - \lambda_1 e_1 \quad (2.24)$$

nous obtenons la nouvelle expression de l'erreur de poursuite sur la vitesse de roulis :

$$\dot{V}_2 = e_1 (-e_2 - \beta_1 e_1) + e_2 \left(\begin{array}{l} (a_1 x_4 x_6 + a_2 x_4 \omega_r + b_1 u_2) - \ddot{x}_1^d + \beta_1 (e_2 + \beta_1 e_1 + \lambda_1 h_1) \\ -\lambda_1 e_1 + \lambda_2 h_2 \end{array} \right) \quad (2.25)$$

Enfin, nous pouvons tirer l'expression de la loi de commande u_2 pour la dynamique du roulis en considérant $\dot{V}_2 < 0$, comme suit :

$$u_2 = \frac{1}{b_1} (e_1 + \ddot{x}_1^d - a_1 x_4 x_6 - a_2 x_4 \omega_r - \beta_1 (e_2 + \beta_1 e_1 + \lambda_1 h_1) + \lambda_1 e_1 - \lambda_2 h_2 - \beta_2 e_2) \quad (2.26)$$

β_2 étant une constante positive.

Ainsi, la dérivée temporelle de cette fonction candidate est donnée par :

$$\dot{V}_2 = -\beta_1 e_1^2 - \beta_2 e_2^2 \quad (2.27)$$

qui est négative pour $(\beta_1, \beta_2) > 0$ et $(e_1, e_2) \neq 0$.

En utilisant la même procédure de synthèse backstepping avec action intégrale, nous donnons respectivement les lois de commande pour la stabilisation des dynamiques de tangage et de lacet du convertible QTW suivantes :

$$u_3 = \frac{1}{b_2} (e_3 + \ddot{x}_3^d - a_3 x_2 x_6 - a_4 x_2 \omega_r - \beta_3 (e_4 + \beta_3 e_3 + \lambda_3 h_3) + \lambda_3 e_3 - \lambda_4 h_4 - \beta_4 e_4) \quad (2.28)$$

$$u_4 = \frac{1}{b_3} (e_5 + \ddot{x}_5^d - a_5 x_2 x_4 - \beta_5 (e_6 + \beta_5 e_5 + \lambda_5 h_5) + \lambda_5 e_5 - \lambda_6 h_6 - \beta_6 e_6) \quad (2.29)$$

où $(\beta_3, \beta_4, \lambda_3, \lambda_4)$ et $(\beta_5, \beta_6, \lambda_5, \lambda_6)$ sont des constantes positives respectivement pour le contrôle backstepping des dynamiques de tangage et de lacet.

2.3.2 Contrôle de la position et de l'altitude

Pour le sous-système d'altitude, la loi de commande est calculée de la même manière que pour les sous-systèmes d'attitude (Bouallègue and BenKhoud, 2016). Ainsi, nous pouvons définir l'erreur de poursuite relative $e_7 = x_7^d - x_7$, la commande virtuelle $x_8^d = \dot{x}_7^d + \beta_7 e_7 + \lambda_7 h_7$ et sa variable d'erreur associée $e_8 = x_8 - (x_7^d + \beta_7 e_7 + \lambda_7 h_7)$ pour obtenir la loi de commande suivante :

$$u_1 = \frac{1}{\cos(x_1) \cos(x_3)} (e_7 + \ddot{x}_7^d - \beta_7 (e_8 + \beta_7 e_7 + \lambda_7 h_7) + \lambda_7 e_7 - \lambda_8 h_8 - \beta_8 e_8 + g) \quad (2.30)$$

où $(\beta_7, \beta_8, \lambda_7, \lambda_8)$ sont des constantes positives, paramètres de réglage de la commande backstepping pour la dynamique d'altitude, et $\cos(x_1) \cos(x_3) \neq 0$.

En revenant à la dynamique du convertible QTW, donnée par l'équation (1.30), on note le couplage entre les mouvements de rotation et de translation. Ainsi, afin de commander indépendamment les dynamiques de position longitudinale x et transversale y , un contrôleur de position est mis en cascade avec celui d'attitude, Figure 2.3. L'objectif principal étant de régler les angles désirés de roulis et de tangage (ϕ_d, θ_d) . Ces angles de référence peuvent être calculés par les lois de commande virtuelles (Bouallègue and BenKhoud, 2016) données comme suit :

$$u_x = \cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi) \quad (2.31)$$

$$u_y = \cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi) \quad (2.32)$$

Ainsi, en résolvant les équations (2.31) et (2.32) pour un angle ψ donné, on obtient les angles désirés de roulis ϕ_d et de tangage θ_d , donnés respectivement par :

$$\phi_d = x_1^d = \arctan \left(\frac{u_x \sin(x_5) - u_y \cos(x_5)}{\sqrt{(1 - u_x^2 \sin^2(x_5) + 2u_x u_y \cos(x_5) \sin(x_5) + u_y^2 \sin^2(x_5) - u_y^2)}} \right) \quad (2.33)$$

$$\theta_d = x_3^d = \arcsin \left(\frac{u_x \cos(x_5) + u_y \sin(x_5)}{\sqrt{(1 - u_x^2 \sin^2(x_5) + 2u_x u_y \cos(x_5) \sin(x_5) + u_y^2 \sin^2(x_5) - u_y^2)}} \right) \quad (2.34)$$

Les lois de commande u_x et u_y , sont calculées en utilisant la même procédure de synthèse précédemment utilisée pour les dynamiques d'attitude et d'altitude. De cette façon, les erreurs de poursuite pour les positions x et y sont respectivement définies comme suit :

$$e_9 = x_9^d - x_9 \quad (2.35)$$

$$e_{11} = x_{11}^d - x_{11} \quad (2.36)$$

Les variables d'erreurs associées à ces entrées virtuelles sont respectivement données par :

$$e_{10} = x_{10} - (\dot{x}_9^d + \beta_9 e_9 + \lambda_9 h_9) \quad (2.37)$$

$$e_{12} = x_{12} - (\dot{x}_{11}^d + \beta_{11} e_{11} + \lambda_{11} h_{11}) \quad (2.38)$$

Ainsi, les lois de commande backstepping pour les positions x et y du convertible QTW sont respectivement données par les expressions suivantes :

$$u_x = \frac{m}{u_1} (e_9 + \ddot{x}_9^d - \beta_9 (e_{10} + \beta_9 e_9 + \lambda_9 h_9) + \lambda_9 e_9 - \lambda_{10} h_{10} - \beta_{10} e_{10}) \quad (2.39)$$

$$u_y = \frac{m}{u_1} (e_{11} + \ddot{x}_{11}^d - \beta_{11} (e_{12} + \beta_{11} e_{11} + \lambda_{11} h_{11}) + \lambda_{11} e_{11} - \lambda_{12} h_{12} - \beta_{12} e_{12}) \quad (2.40)$$

où $(\beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \lambda_9, \lambda_{10}, \lambda_{11}, \lambda_{12})$ sont des constantes positives à ajuster en tant que paramètres de réglage de la commande backstepping.

2.3.3 Résultats et discussion : cas de la stabilisation des dynamiques d'attitude et d'altitude

En partant des conditions initiales $\boldsymbol{\eta}_i = (0.5, 0.3, -0.6)^T$ et $z_i = 0$ et compte tenu des consignes désirées $\boldsymbol{\eta}_d = (0, 0, 0)^T$ et $z_d = 5$, les figures 2.4 et 2.5 montrent les résultats de simulation obtenus. Les dynamiques contrôlées sont stabilisées et des performances satisfaisantes en régime transitoire sont garanties.

Considérons à présent la simulation de la stabilisation des positions x et y . La performance de la stratégie de commande backstepping avec action intégrale proposée est comparée aux approches PID et backstepping classique. Le convertible QTW est soumis à des perturbations extérieures selon les axes latéral et longitudinal, i.e. rafales de vent, définies par $\Delta_x(t) = 10 \sin(3t)$ et $\Delta_y(t) = 10 \cos(3t)$. Les figures 2.6 et 2.7 donnent les résultats de simulation.

D'après la figure 2.6, les performances de la boucle fermée de tous les contrôleurs proposés sont proches dans le cas sans perturbations. Cependant, en présence de telles entrées externes, il est clair que le contrôleur backstepping intégral est capable de rejeter les perturbations appliquées et stabiliser ainsi le système autour de la référence souhaitée. Ceci est dû à l'action intégrale qui est introduite pour diminuer, voire annuler, les erreurs en régime permanent. Le régulateur PID stabilise les dynamiques de position mais avec dégradations en régime transitoire par rapport au backstepping avec action intégrale. Les dépassements sont bien amortis et la précision est nettement meilleure. D'autre part, les signaux de commande de la figure 2.7 sont d'amplitudes plus modérées pour la variante backstepping proposée.

2.3.4 Résultats et discussion : cas de la poursuite de trajectoires en vol vertical

Considérons à présent le problème du suivi de trajectoires du véhicule convertible QTW. Dans cette étude, les résultats de mise en œuvre par co-simulation matérielle HW sont

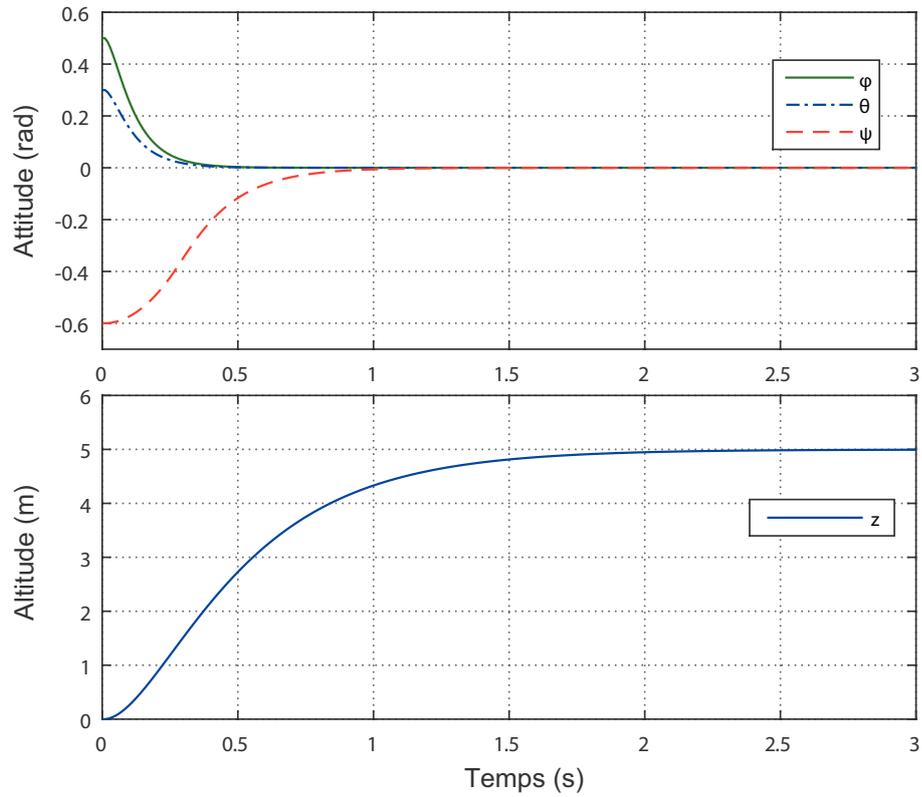


FIGURE 2.4 – Dynamiques d'attitude et d'altitude stabilisées.

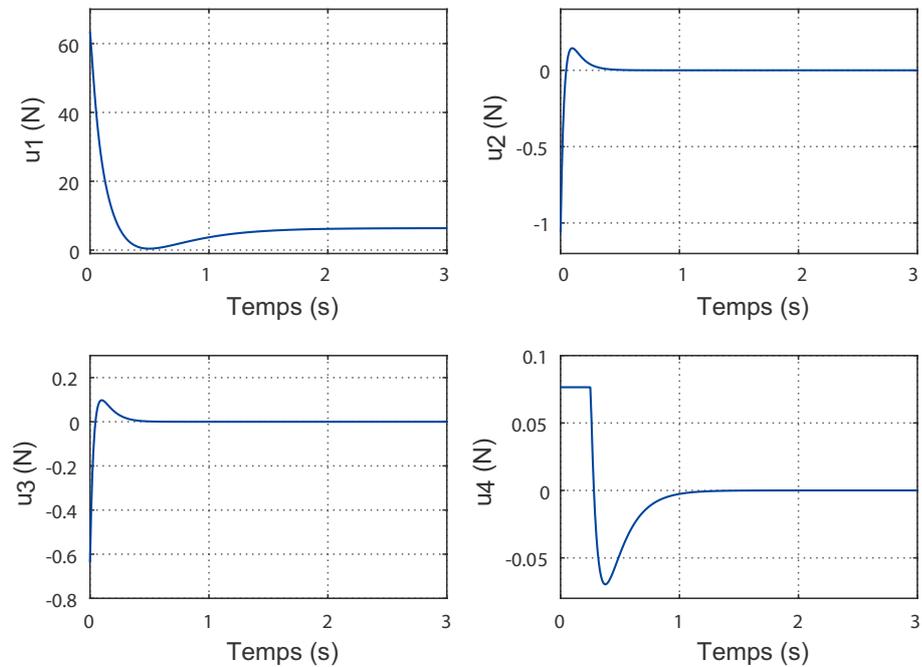


FIGURE 2.5 – Signaux de commande backstepping pour la stabilisation.

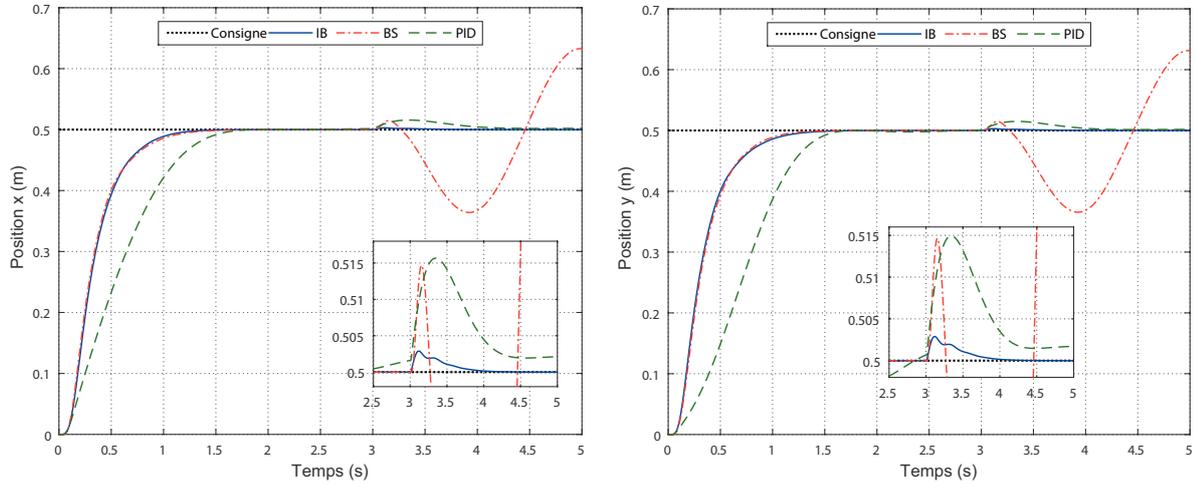


FIGURE 2.6 – Positions longitudinale et transversale contrôlées du drone QTW.

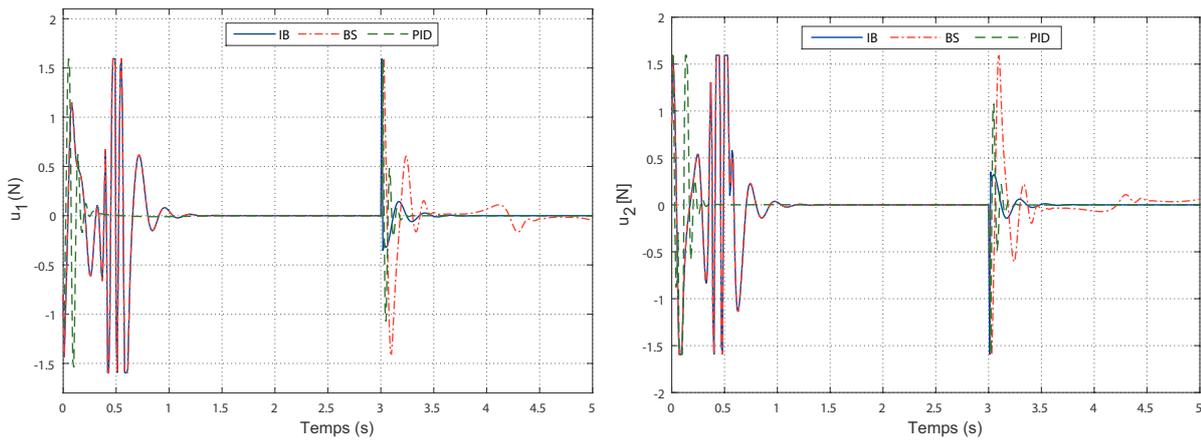


FIGURE 2.7 – Variation des signaux de commande sous l'effet de perturbations.

montrés dans les figures 2.8, 2.9 et 2.10. Pour cela, nous retenons la plate-forme de co-design proposée à base de la carte NI myRIO-1900 et l'environnement LabVIEW (Voir Annexe C) (BenKhoud et al., 2017; Instruments, 2009, 2013).

Les trajectoires circulaires de référence retenues sont définies par les équations (2.41), (2.42) et (2.43) suivantes :

$$x_d(t) = 0.5 \sin(0.25t) \quad (2.41)$$

$$y_d(t) = 0.5 \cos(0.25t) \quad (2.42)$$

$$z_d(t) = 5t \quad (2.43)$$

Dans ce scénario de co-simulation, la poursuite d'une trajectoire hélicoïdale tridimensionnelle est envisagée. Le résultat de mise en œuvre est représenté dans la figure 2.10. Une telle réponse témoigne de la haute performance de la stratégie de commande backstepping avec action intégrale proposée en termes de précision et de rapidité. Les signaux de commande, montrés dans la figure 2.9, sont plus modérés et présentent moins d'oscillations. Nous remarquons également que les résultats de co-simulation matérielle HW et ceux de la simulation logicielle SW sont bien proches, ce qui justifie davantage la validité de l'approche d'émulation proposée à base de la plate-forme LabVIEW/NI myRIO-1900.

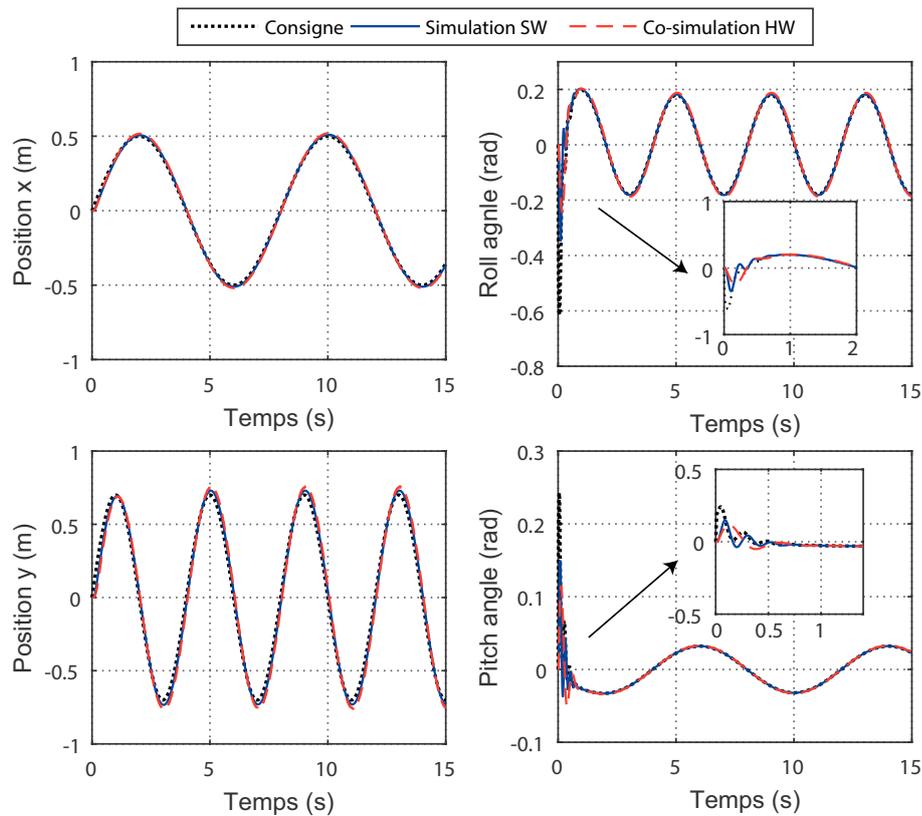


FIGURE 2.8 – Performances en poursuite de la commande backstepping intégrale.

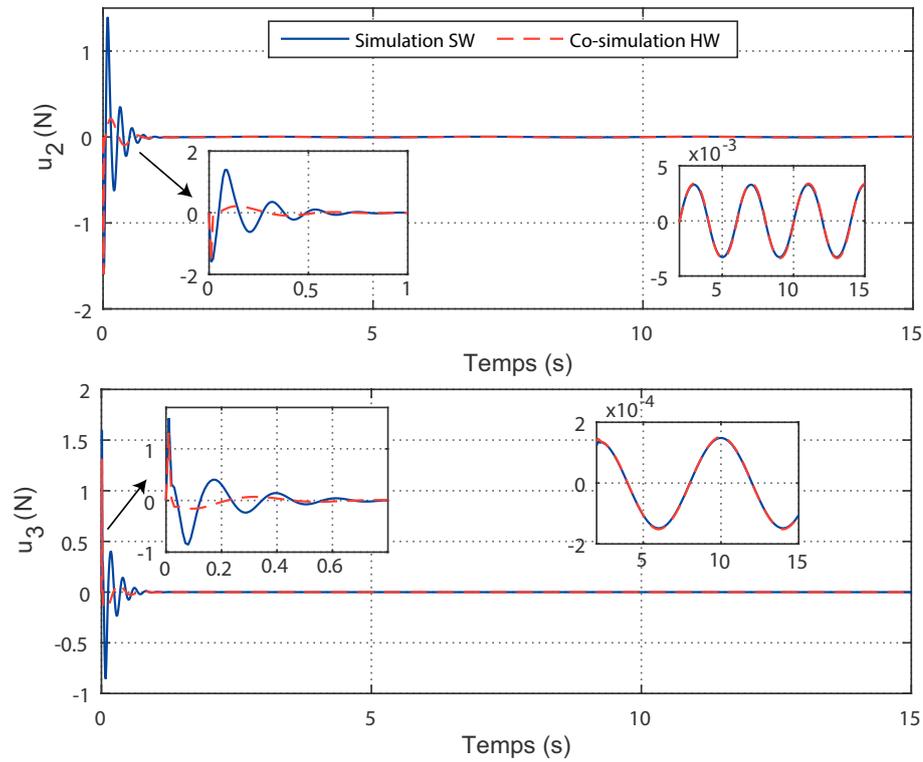


FIGURE 2.9 – Variation des signaux de commande en poursuite.

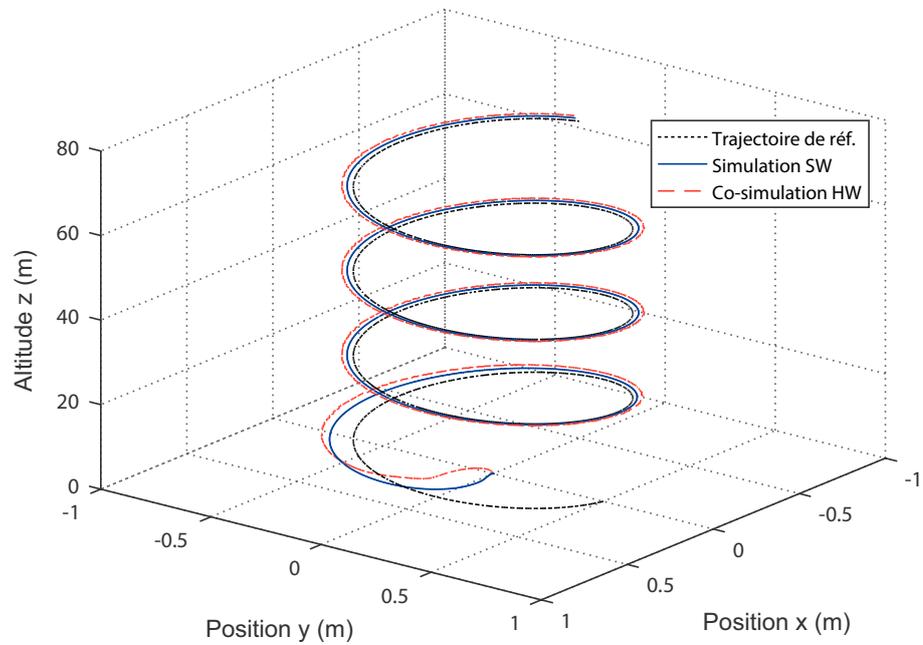


FIGURE 2.10 – Poursuite de trajectoire hélicoïdale dans l'espace.

2.4 Commande prédictive à base de modèle

2.4.1 Principe

La commande prédictive à base de modèle (MPC) a été appliquée dans plusieurs domaines d'activité. Depuis les années 70, les industriels de la pétrochimie et métallurgie ont commencé à manifester de l'intérêt pour la MPC (Richalet et al., 1978). Le point commun de ces deux domaines est qu'ils imposent de nombreuses variables et contraintes en entrée/sortie (Balchen et al., 1989; Wills and Heath, 2005). Son aptitude à être appliquée aux systèmes avec un grand nombre de variables rend la commande intéressante dans le domaine de robotique et surtout pour des systèmes de plus en plus complexes. Nous citons à titre d'exemple l'utilisation de la commande prédictive pour le guidage d'un robot humanoïde (Dimitrov et al., 2011) et le suivi de chemin d'un robot de type HRP-2 (Koenemann et al., 2015). Dans le domaine aérospatial, l'approche MPC a été largement appliquée aux vannes électromécaniques (Banerjee et al., 2010), dans la commande des engins spatiaux (Ikeda et al., 2012; Pirouzmand and Ghahramani, 2013; Kalabic et al., 2014; Gupta et al., 2015) et des satellites (Pirouzmand and Ghahramani, 2013). Sans oublier son intégration dans le contrôle de systèmes énergétiques comme les centrales hydro-électriques (Dinh et al., 2011), les éoliennes (Mirzaei et al., 2012), les échangeurs thermiques (Xiaohong et al., 2014), les réseaux de distribution d'eau (Piotr et al., 2010) et les centrales de traitement des eaux (Honggui and Junfei, 2014).

Plusieurs méthodologies ont ainsi fait leurs apparitions, s'attachant toujours aux mêmes idées de base. Par exemple, nous citons la commande MUSMUR proposée par (Greco et al., 1984), la commande PFC suggérée par (Richalet et al., 1987), ou encore UPC (en anglais Unified Predictive Control) élaborée par (Soeterboek, 1992). La liste des commandes présentées ici n'est pas exhaustive. Dans (Granado, 2004), (Marcelin, 2010) et (Larminat, 2013) plus de détails sont donnés. Le principe de la commande prédictive consiste à minimiser une fonction coût qui conduit à la détermination d'une séquence de commande sur un horizon défini. Ce dernier doit être convenablement choisi en tenant compte de plusieurs types de contraintes : les contraintes sur les entrées, sur les états ou bien sur les sorties du système.

La commande prédictive fait appel au principe de l'horizon fuyant. En effet, à chaque instant, une séquence de commande optimale en boucle ouverte, minimisant la fonction coût sur un horizon de prédiction N_p , est calculée. Cependant, seul le premier élément est appliqué au système. Cette procédure est reprise à la période d'échantillonnage suivante, mettant à jour les paramètres courants du système. La Figure 2.11 illustre le principe de la commande prédictive : le calcul de la séquence de commande optimale u à appliquer au système se fait à chaque instant k jusqu'à l'instant $k + N_c$ où N_c est l'horizon de commande pour la MPC. Pour cela, un modèle du système à contrôler permettant de prédire la sortie y jusqu'à l'instant $k + N_p$ par rapport à une trajectoire de référence r .

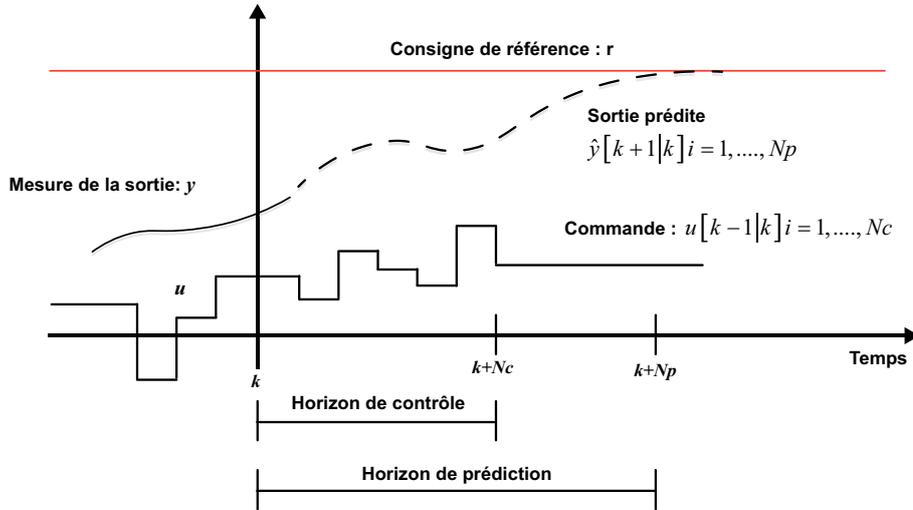


FIGURE 2.11 – Illustration des signaux de commande et de sortie de la commande prédictive MPC.

2.4.2 Choix des paramètres de réglage

Dans la littérature, la synthèse d'une loi de commande prédictive requiert la détermination de trois paramètres suivants :

- l'horizon de commande $N_c \in \mathbb{N}$;
- l'horizon de prédiction $N_p \in \mathbb{N}$;
- le facteur de pondération $\lambda \in \mathbb{R}$.

Ces paramètres influencent d'une manière significative le comportement du système en boucle fermée en termes de stabilité, rapidité et précision. Le choix de ces paramètres doit être fait avec grande attention et leur réglage dépend de plusieurs critères tels que la période d'échantillonnage, la dynamique du système, le retard, etc. Dans la littérature, plusieurs travaux ont été menés sur la problématique du réglage de ces paramètres (Nagoudi-Layerle, 2011; Cazy, 2016; Turki et al., 2018b; Mamboundou and Langlois, 2012).

Le choix de N_c égal à l'unité est reconnu comme étant suffisant dans la plupart des cas (Hamouda, 2015). Cependant, une valeur plus élevée permet d'atteindre des objectifs plus complexes lorsque les systèmes à contrôler présentent un nombre élevé de degrés de liberté (Clarke et al., 1987). Or, plus la valeur de l'horizon de commande est élevée, plus le temps de calcul nécessaire pour synthétiser la loi de commande est important. Cela s'explique par le fait que N_c fixe la dimension des matrices à inverser dans le calcul de la séquence de commande optimale. Un compromis doit être alors trouvé (Nagoudi-Layerle, 2011).

Comme mentionné précédemment, la commande prédictive fait appel au principe de l'horizon fuyant. À chaque itération, l'horizon de prédiction N_p permet d'avoir la visibilité sur les futurs états du système jusqu'à un temps fini, ce qui donne à l'algorithme de

résolution une meilleure qualité des informations relatives à la dynamique d'évolution du système. Cependant, plus N_p est grand, plus la qualité de la prédiction est faible. Dans la littérature, il est difficile de trouver une méthode générique permettant de déterminer la valeur de l'horizon de prédiction de manière exacte. Mais, un compromis peut toutefois être trouvé entre une petite valeur, qui offre peu d'informations sur l'évolution du système mais assurant une meilleure qualité de prédiction, et une grande valeur, qui fournit une plus large quantité d'informations mais des incertitudes plus importantes par rapport à l'état futur réel du système (Nagoudi-Layerle, 2011; Cazy, 2016).

Tout comme l'indique son nom, le facteur de pondération λ , joue un rôle primordial de dosage. Il garantit un certain équilibre dans la fonction coût entre l'erreur de poursuite et l'impact de la commande. Une augmentation excessive de λ implique une demande exigeante de la commande et, par conséquent, la dynamique du système est ralentie. Par contre, une diminution de λ entraîne une commande énergique qui rend la réponse du système de plus en plus rapide. Il faut noter que le réglage de ce paramètre est assez délicat. En effet, tout comme l'horizon de commande, il influence la stabilité du système en boucle fermée (Nagoudi-Layerle, 2011; Hamouda, 2015).

2.4.3 Représentation d'état augmentée à temps discret

Soit un système linéaire multivariable MIMO représenté par le modèle d'état à temps discret suivant :

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) \\ y(k) &= C_m x_m(k) \end{aligned} \quad (2.44)$$

avec $y \in \mathbb{R}^{n_{out}}$ est la sortie du système, $\mathbb{R}^{n_{in}}$ est le vecteur des commandes et x est le vecteur d'état de dimensions $(n_1 \times 1)$.

Dans (2.44), A_m , B_m et C_m sont respectivement les matrices d'état, de commande et de sortie de dimensions successives $(n_1 \times n_1)$, $(n_1 \times n_{in})$ et $(n_{out} \times n_1)$. Dans le but d'obtenir le modèle augmenté à base d'intégrateurs, nous introduisons les différences du vecteur d'état et de la variable de commande telles que :

$$\Delta x_m(k) = x_m(k) - x_m(k-1) \quad (2.45)$$

$$\Delta u(k) = u(k) - u(k-1) \quad (2.46)$$

En intégrant les incréments des variables $x(k)$ et $u(k)$, la différence (2.46) devient alors :

$$\begin{aligned} \Delta x_m(k+1) &= A_m(x_m(k) - x_m(k-1)) + B_m(u(k) - u(k-1)) \\ &= A_m \Delta x_m(k) + B_m \Delta u(k) \end{aligned} \quad (2.47)$$

Soit ainsi le nouveau vecteur d'état $x(k)$ défini par :

$$x(k) = \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (2.48)$$

En procédant par récurrence, la sortie $y(k)$ devient :

$$\begin{aligned} y(k+1) - y(k) &= C_m(x_m(k+1) - x_m(k)) \\ &= C_m\Delta x_m(k+1) \\ &= C_mA_mA_m\Delta x_m(k) + C_mB_m\Delta u(k) \end{aligned} \quad (2.49)$$

La combinaison des équations (2.47) et (2.49) donne la représentation d'état augmentée d'un modèle multivariable à temps discret :

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ y(k) &= Cx(k) \end{aligned} \quad (2.50)$$

avec :

$$x(k+1) = \begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} \quad (2.51)$$

$$x(k) = \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (2.52)$$

$$A = \begin{bmatrix} A_m & 0_m^T \\ C_mA_m & I_{(n_{out} \times n_{out})} \end{bmatrix} \quad (2.53)$$

$$B = \begin{bmatrix} B_m \\ C_mB_m \end{bmatrix} \quad (2.54)$$

$$C = [0_m \quad I_{(n_{out} \times n_{out})}] \quad (2.55)$$

L'indice T représente la transposée d'une matrice et I est la matrice identité. Dans (2.4.3), 0_m est une matrice nulle de dimension $(n_{out} \times n_1)$. Admettons que $n_{etat} = n_1 + n_{out}$, le triplet (A, B, C) de dimension respectivement $(n_{etat} \times n_{etat})$, $(n_{etat} \times n_{in})$ et $(n_{out} \times n_{etat})$ constitue le modèle augmenté discret invariant dans le temps du système MIMO considéré.

2.4.4 Formulation de la commande en absence de contraintes

Admettons que le système (2.44) soit observable et commandable. Le vecteur de l'incrément de la commande ΔU de dimensions $(1 \times n_{in}N_c)$ est défini par :

$$\Delta U = [\Delta u(k)^T \quad \Delta u(k+1)^T \quad \dots \quad \Delta u(k+N_c-1)^T]^T \quad (2.56)$$

Soient les vecteurs de la sortie désirée Y_{des} et de la sortie prédite \hat{Y} définis respectivement comme suit :

$$Y_{des} = [y_{des}(k+1)^T \quad y_{des}(k+2)^T \quad \dots \quad y_{des}(k+N_p)^T]^T$$

$$\hat{Y} = [\hat{y}(k+1|k)^T \quad \hat{y}(k+2|k)^T \quad \dots \quad \hat{y}(k+N_p|k)^T]^T$$

Dans l'objectif de synthétiser une loi de commande optimale MPC la fonction coût J à minimiser est définie par l'expression suivante :

$$J = (Y_{des} - \hat{Y})^T (Y_{des} - \hat{Y}) + \Delta U^T \bar{R} \Delta U \quad (2.57)$$

avec \bar{R} une matrice carrée de dimensions $(n_{in}N_c \times n_{in}N_c)$ définie par :

$$\bar{R} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \vdots & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{n_{in}N_c} \end{bmatrix} \quad (2.58)$$

En se basant sur le modèle d'état augmenté (2.50), la séquence future du vecteur d'état est calculée par récurrence comme suit (Wang, 2009) :

$$\begin{aligned} x(k+1 | k) &= Ax(k) + B\Delta u(k) \\ x(k+2 | k) &= Ax(k+1 | k) + B\Delta u(k+1) \\ &= A^2x(k) + AB\Delta u(k) + B\Delta u(k+1) \\ &\vdots \\ x(k+N_p | k) &= A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) \\ &\quad + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1) \end{aligned}$$

À partir du vecteur d'état prédit et en procédant par récurrence, la sortie estimée est définie par :

$$\begin{aligned} y(k+1 | k) &= CAx(k) + CB\Delta u(k) \\ y(k+2 | k) &= CA^2x(k) + CAB\Delta u(k) + CB\Delta u(k+1) \\ y(k+3 | k) &= CA^3x(k) + CA^2B\Delta u(k) + CAB\Delta u(k+1) \\ &\quad + CB\Delta u(k+2) \\ &\vdots \\ y(k+N_p | k) &= CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1) \\ &\quad + \cdots + CA^{N_p-N_c}B\Delta u(k+N_c-1) \end{aligned}$$

La sortie prédite peut être exprimée alors sous la forme matricielle suivante :

$$\hat{Y} = Fx(k) + \Phi\Delta U \quad (2.59)$$

où F et Φ sont deux matrices de dimensions respectives $(n_{out}N_p \times n_{etat})$ et $(n_{out}N_p \times n_{in}N_c)$ définies par :

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix} \quad (2.60)$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix} \quad (2.61)$$

En associant les équations (2.57) et (2.59), la dérivée partielle de J par rapport à ΔU peut être écrite comme suit :

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T(Y_{des} - Fx(k)) + 2(\Phi^T\Phi + \bar{R})\Delta U \quad (2.62)$$

À ce stade et afin d'obtenir la loi de commande optimale à appliquer au système, la condition d'optimalité suivante doit être respectée :

$$\frac{\partial J}{\partial \Delta U} = 0 \quad (2.63)$$

Dans le cas où la commande prédictive ne présente pas de contraintes, la séquence de commande optimale est alors exprimée comme suit :

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(Y_{des} - Fx(k)) \quad (2.64)$$

Selon le principe de l'horizon fuyant, seuls les premiers n_{in} éléments de la loi de commande optimale sont appliqués au système comme suit :

$$u(k) = u(k-1) + \Delta u(k)$$

$$u(k) = u(k-1) + \overbrace{[I_m \quad 0_m \quad \dots \quad 0_m]}^{n_{in}N_c} (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(Y_{des} - Fx(k))$$

Ainsi, le vecteur d'état $x(k)$ est le seul élément requis par le contrôleur prédictif pour construire la sortie prédite. Lorsque ce vecteur n'est pas accessible, un observateur d'état doit être utilisé.

2.4.5 Formulation de la commande en présence de contraintes

En réalité, des contraintes sont souvent présentes dans la plupart des systèmes dynamiques. Ces contraintes rendent la synthèse des lois de commande plus délicate, surtout face à un procédé industriel complexe. Du point de vue conceptuel, les contraintes peuvent être de différentes natures. Nous distinguons alors deux types de contraintes : celles qui sont imposées par les limitations physiques et celles qui sont présentes afin de garantir des performances désirées (Nguyen, 2012). Un avantage majeur de la commande prédictive est sa capacité de prendre en compte ces contraintes.

2.4.5.1 Contraintes sur les incréments de commande

Dans ce cadre, la classe des contraintes sur l'incrément des variables de commande est introduite afin d'éviter la saturation. Ces contraintes sont de dimensions égales à celles du vecteur de commande et considérées comme "dures". Ainsi, nous supposons que la variable de commande incrémentale ΔU est soumise à des contraintes d'inégalité linéaires comme suit :

$$\Delta U^{min} \leq \Delta U \leq \Delta U^{max} \quad (2.65)$$

avec $\Delta U^{min} = [\Delta u_1^{min} \quad \Delta u_2^{min} \quad \dots \quad \Delta u_{n_{in}}^{min}]$ et $\Delta U^{max} = [\Delta u_1^{max} \quad \Delta u_2^{max} \quad \dots \quad \Delta u_{n_{in}}^{max}]$ sont les valeurs minimale et maximale de ΔU .

Les incréments des variables de commande sont donnés comme suit :

$$\left\{ \begin{array}{l} \Delta u_1^{min} \leq \Delta u_1 \leq \Delta u_1^{max} \\ \Delta u_2^{min} \leq \Delta u_2 \leq \Delta u_2^{max} \\ \vdots \\ \Delta u_{n_{in}}^{min} \leq \Delta u_{n_{in}} \leq \Delta u_{n_{in}}^{max} \end{array} \right. \quad (2.66)$$

L'intégration de ces contraintes dans la formulation du problème de synthèse de la commande prédictive à base d'état se fait à travers la transformation en un problème d'optimisation contraint avec :

$$-\Delta U \leq -\Delta U^{min} \quad (2.67)$$

$$\Delta U \leq \Delta U^{max} \quad (2.68)$$

Ce qui est équivalent une écriture matricielle aux équations suivantes :

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix} \quad (2.69)$$

2.4.5.2 Contraintes sur l'amplitude des entrées de commande

Une autre classe de contraintes aussi considérée étant les contraintes sur l'amplitude du vecteur de commande U fréquemment rencontrées en pratique et définie comme suit (Wang, 2009) :

$$U^{min} \leq U \leq U^{max} \quad (2.70)$$

où $U^{min} = [u_1^{min} \quad u_2^{min} \quad \dots \quad u_{n_{in}}^{min}]$ et $U^{max} = [u_1^{max} \quad u_2^{max} \quad \dots \quad u_{n_{in}}^{max}]$ sont les valeurs minimale et maximale de U .

tout en vérifiant :

$$\left\{ \begin{array}{l} u_1^{min} \leq u_1 \leq u_1^{max} \\ u_2^{min} \leq u_2 \leq u_2^{max} \\ \vdots \\ u_{n_{in}}^{min} \leq u_{n_{in}} \leq u_{n_{in}}^{max} \end{array} \right. \quad (2.71)$$

D'une manière générale, les contraintes sont appliquées durant les instants futurs. Elles peuvent être exprimées en fonction de ΔU comme suit :

$$\overbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_c-1) \end{bmatrix}}^U = \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \end{bmatrix} u(k-1) + \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ I & I & 0 & \cdots & 0 \\ I & I & I & \cdots & 0 \\ \vdots & & & & \\ I & I & \cdots & I & I \end{bmatrix} \overbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}}^{\Delta U} \quad (2.72)$$

2.4.5.3 Contraintes sur les sorties

Il s'agit de contraintes opérationnelles sur les sorties du système qui sont souvent considérées comme des contraintes "soft" et définies comme suit :

$$Y^{min} \leq Y \leq Y^{max} \quad (2.73)$$

où $Y^{min} = [y_1^{min} \ y_2^{min} \ \cdots \ y_{n_{out}}^{min}]$ et $Y^{max} = [y_1^{max} \ y_2^{max} \ \cdots \ y_{n_{out}}^{max}]$ sont les valeurs minimale et maximale de Y de dimensions chacune $(1 \times n_{out})$.

telles que :

$$\begin{cases} y_1^{min} \leq y_1 \leq y_1^{max} \\ y_2^{min} \leq y_2 \leq y_2^{max} \\ \vdots \\ y_{n_{out}}^{min} \leq y_{n_{out}} \leq y_{n_{out}}^{max} \end{cases} \quad (2.74)$$

À l'instar des contraintes sur l'incrément et l'amplitude de commande, les contraintes sur les sorties peuvent être écrites en fonction de ΔU et seront abordées par la suite.

L'équation (2.73) peut se réécrire comme suit (Wang, 2009) :

$$Y^{min} \leq Fx(k) + \Phi \Delta U \leq Y^{max} \quad (2.75)$$

ce qui est équivalent à :

$$-\Phi \Delta U \leq -Y^{min} + Fx(k) \quad (2.76)$$

$$\Phi \Delta U \leq Y^{max} - Fx(k) \quad (2.77)$$

L'écriture matricielle des équations (2.76) et (2.77) conduit à :

$$\begin{bmatrix} -\Phi \\ \Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} -Y^{min} + Fx(k) \\ Y^{max} - Fx(k) \end{bmatrix} \quad (2.78)$$

Les résultats des équations (2.69) et (2.78) permettent d'établir l'inégalité suivante :

$$M_{cons} \Delta U \leq \gamma_{cons} \quad (2.79)$$

où M_{cons} et γ_{cons} sont des matrices qui reflètent la présence des contraintes sur les sorties dans le système et qui sont définies par :

$$M_{cons} = \begin{bmatrix} -C_2 \\ C_2 \\ -I \\ I \\ -\Phi \\ \Phi \end{bmatrix}, \gamma_{cons} = \begin{bmatrix} -U^{min} + C_1 u(k-1) \\ U^{max} - C_1 u(k-1) \\ -\Delta U^{min} \\ \Delta U^{max} \\ -Y^{min} + Fx(k) \\ Y^{max} - Fx(k) \end{bmatrix} \quad (2.80)$$

Dans le cas où toutes les contraintes sont présentes, le nombre de lignes de la matrice M_{cons} sera égal à $4 \times n_{in} \times N_c + 2 \times n_{out} \times N_p$. A ce stade, le problème de prise en compte des contraintes est devenu un problème d'optimisation quadratique (Wang, 2009). Pour résoudre ce dernier, l'algorithme multiplicateur de Lagrange peut être utilisé.

L'intégration de la sortie prédite \hat{Y} présentée dans l'équation (2.57) conduit à la forme développée de la fonction coût J suivante :

$$J = (Y_{des} - Fx(k))^T (Y_{des} - Fx(k)) - 2\Delta U^T \Phi^T (Y_{des} - Fx(k)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (2.81)$$

Selon (Kerrigan and Maciejowski, 2000), le critère J peut être exprimé comme suit :

$$J = \frac{1}{2} \Delta U^T E_J \Delta U + \Delta U^T F_J + \Psi^T \Psi \quad (2.82)$$

Sous les contraintes inégalité (2.79) et avec :

$$\begin{cases} E_J = 2(\Phi^T \Phi + \bar{R}) \\ F_J = -2\Phi^T \Psi \\ \Psi = Y_{des} - Fx(k) \end{cases}$$

Dans ce mémoire, la matrice E_J est supposée symétrique et définie positive. Puisque le dernier élément de la fonction coût ($\Psi^T \Psi$) ne tient pas compte de ΔU , il peut être alors simplifié. L'intégration du vecteur multiplicateur de Lagrange η_{cons} dans l'expression de J conduit à :

$$J = \frac{1}{2} \Delta U^T E_J \Delta U + \Delta U^T F_J + \eta_{cons}^T (M_{cons} \Delta U - \gamma_{cons}) \quad (2.83)$$

La synthèse d'une loi de commande prédictive sous contraintes est ramenée alors à résoudre un problème d'optimisation quadratique (Fletcher, 1981; Luenberger and Yinyu, 1984; Boyd and Vandenberghe, 2004). Dans ce problème, l'incrément de commande ΔU et le vecteur multiplicateur de Lagrange η_{cons} représentent les variables de décision (Wang, 2009).

Notons par S_{act} l'ensemble des contraintes actives. Par conséquent, les conditions d'optimalité au sens de Kuhn-Trucker sont définis comme (Wang, 2009) :

$$E_J \Delta u(k) + F_J + \sum_{i \in S_{act}} M_{i_{cons}}^T \eta_{i_{cons}} = 0 \quad (2.84)$$

avec :

$$\begin{cases} M_{icons}\Delta U - \gamma_{icons} = 0, & i \in S_{act} \\ M_{icons}\Delta U - \gamma_{icons} < 0, & i \notin S_{act} \\ \eta_{icons} \geq 0, & i \in S_{act} \\ \eta_{icons} = 0, & i \notin S_{act} \end{cases} \quad (2.85)$$

où M_{icons} désigne la i ème ligne de la matrice M_{cons} .

D'après les travaux de (Kerrigan and Maciejowski, 2000), quand les contraintes sont actives (c'est-à-dire i appartient à S_{act}), les vecteurs optimaux η_{cons} et ΔU de l'équation (2.84) sont respectivement donnés par :

$$\eta_{cons} = -(M_{cons}E_J^{-1}M_{cons}^T)^{-1}(\gamma_{cons} + M_{cons}E_J^{-1}F_J) \quad (2.86)$$

$$\Delta U = -E_J^{-1}(F_J + M_{cons}^T\eta_{cons}) \quad (2.87)$$

2.4.6 Application à la commande du drone QTW

La stratégie de commande TSMPC proposée est appliquée pour le sous-système non linéaire d'attitude du drone convertible QTW décrit par des multi-modèles T-S présenté dans la section 1.6.6. Les sous-modèles d'attitude sont pondérés par des fonctions d'activation dépendant des paramètres du système. Comme le système considéré présente deux variables non linéaires $\dot{\phi}$ et $\dot{\psi}$, alors quatre sous-modèles LTI ($r = 2^p = 2^2$) interconnectés moyennant les fonctions d'appartenance μ_i , $i \in \{1, 2, 3, 4\}$ sont établis. La loi de commande à appliquer au système non linéaire devient alors :

$$u(k) = \sum_{j=1}^4 \mu_j(\theta(t))u_j(k) \quad (2.88)$$

Ici, la commande prédictive développée est appliquée aux quatre sous-systèmes linéaires présentant chacun une zone de fonctionnement bien déterminée. En effectuant une interpolation entre les différentes lois de commande $\{u_1, u_2, u_3, u_4\}$, la stratégie proposée peut contrôler le système non linéaire initial conformément à la figure 2.12.

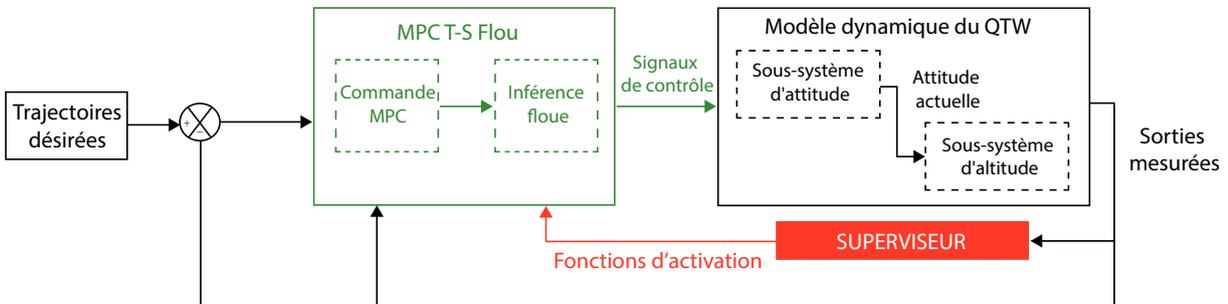


FIGURE 2.12 – Principe de la commande prédictive par interpolation (TSMPC) appliquée au drone QTW.

Pour la synthèse, le pas d'échantillonnage est fixé à $T_s = 0.1$ s. D'après la dynamique de vol, les trois angles d'Euler de roulis, de tangage et de lacet sont contraints comme suit :

$$-\frac{\pi}{2} < \phi, \theta < \frac{\pi}{2} \quad (2.89)$$

$$-\pi < \psi < \pi \quad (2.90)$$

En pratique, les entrées du système sont soumises à des contraintes opérationnelles souvent issues des limitations sur les vitesses angulaires des actionneurs et les facteurs de poussée et de traînée en vol stationnaire. Cela peut dégrader les performances du système en boucle fermée et conduire ainsi à l'instabilité du vol. En supposant que les caractéristiques d'un moteur à courant continu soient conformes à celles décrites dans (Alexis et al., 2010), les contraintes sur les vitesses angulaires et les entrées peuvent être définies comme suit :

$$-0,75 \leq \dot{\phi}, \dot{\theta}, \dot{\psi} \leq 0,75 \quad (2.91)$$

$$|\tau_\phi, \tau_\theta| \leq 5,61 \quad (2.92)$$

$$|\tau_\psi| \leq 0,16 \quad (2.93)$$

2.4.7 Résultats de simulation et discussion

Nous avons choisi, parmi les paramètres du contrôleur MPC, d'explorer les effets de la variation des l'horizon de prédiction N_p et de contrôle N_c pour la dynamique de lacet. À partir de diverses simulations et avec $N_c = 15$, nous pouvons voir sur la figure 2.13 les effets d'évolution de l'horizon de prédiction. Pour le choix $N_p = 15$, l'horizon de prédiction est trop court pour contrôler le processus réel et aboutir à une boucle de contrôle moins amortie. Cependant, pour les choix $N_p = 35, 45$ et 65 , le suivi de référence est garanti avec des dégradations mineures entre les différents scénarios. Le choix $N_p = 65$ assure une stabilité meilleure du système avec un dépassement minimal, voir nul.

En ce qui concerne le choix de l'horizon de contrôle, la figure 2.14 montre qu'après avoir fixé $N_p = 65$, une valeur diminuée de N_c permet de suivre les références avec un temps de réaction trop lent, de sorte que le bon choix d'horizon de contrôle améliore dans une certaine mesure la réponse du système. Ainsi, le choix $N_c = 15$ conduit à obtenir une réponse du système satisfaisante.

La réponse temporelle du système en boucle fermée par rapport à la trajectoire de référence est donnée dans la figure 2.15 pour les attitudes, et dans la figure 2.16 pour les vitesses angulaires. Les entrées de commande du système sont représentées dans la figure 2.18.

Sur la base de ces résultats, les performances de l'approche prédictive MPC proposée sont satisfaisantes du point de vue stabilisation des dynamique de vol. Néanmoins, la différence notoire avec la technique non linéaire backstepping de la section précédente vient de l'amplitude beaucoup plus faible des signaux relatifs aux angles d'attitude. Un tel comportement, propice à la non saturation des actionneurs, induit cependant une convergence relativement lente des sorties vers ses références.

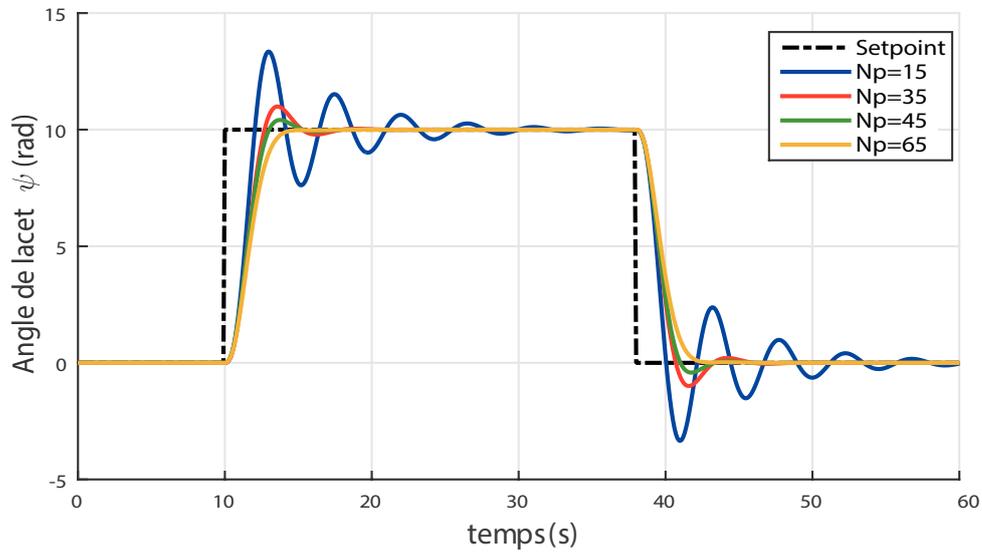


FIGURE 2.13 – Effet de la variation de l’horizon de prédiction pour un horizon de contrôle fixé à $N_c = 15$.

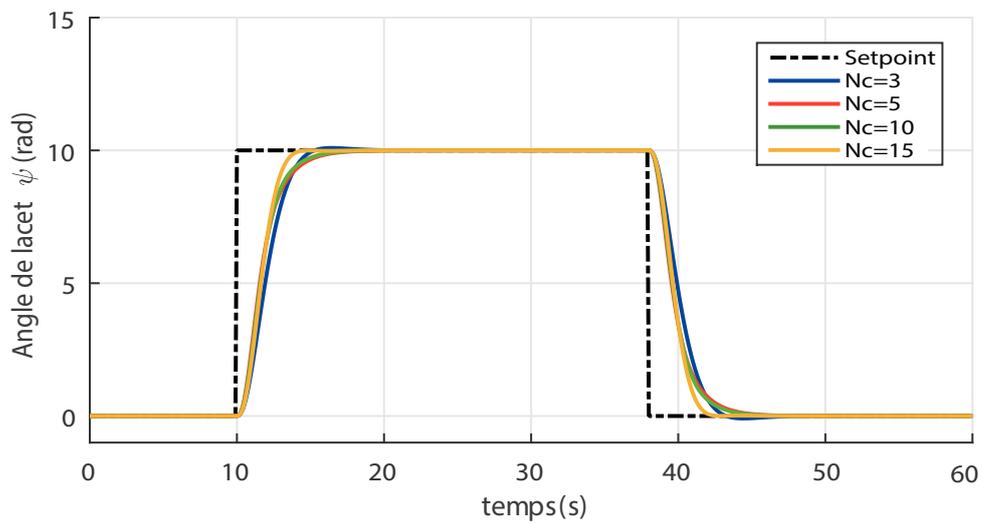


FIGURE 2.14 – Effet de la variation de l’horizon de contrôle pour un horizon de prédiction fixé à $N_p = 65$.

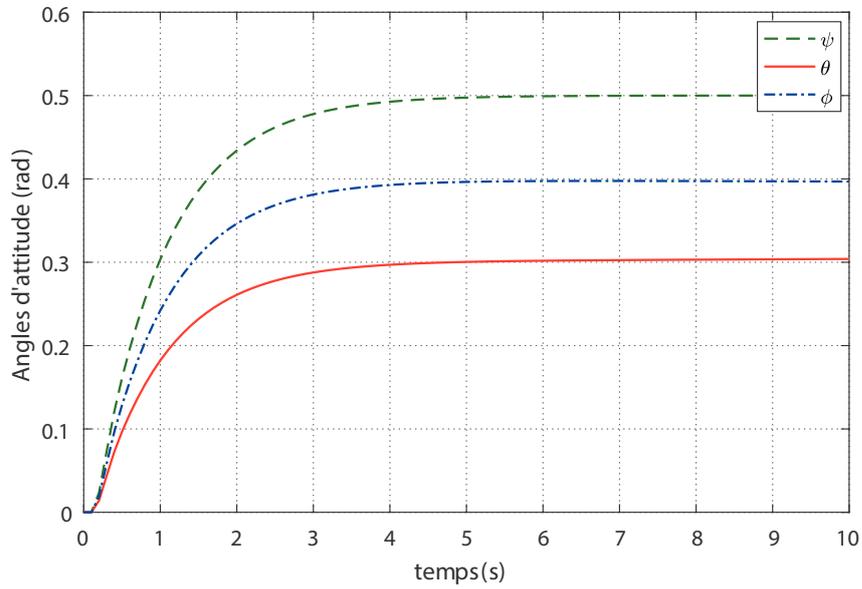


FIGURE 2.15 – Réponses du système en fonctionnement nominal : angles d'attitude.

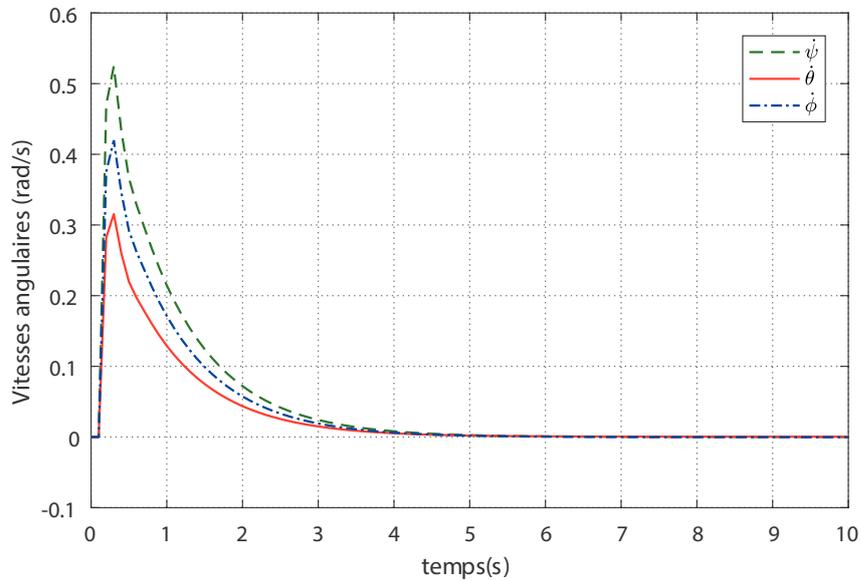


FIGURE 2.16 – Réponses du système en fonctionnement nominal : vitesses angulaires.

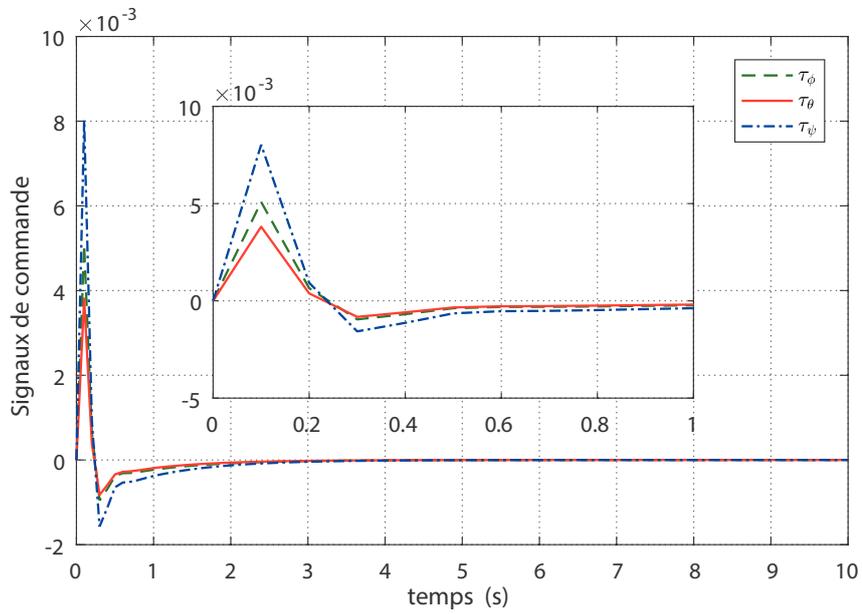
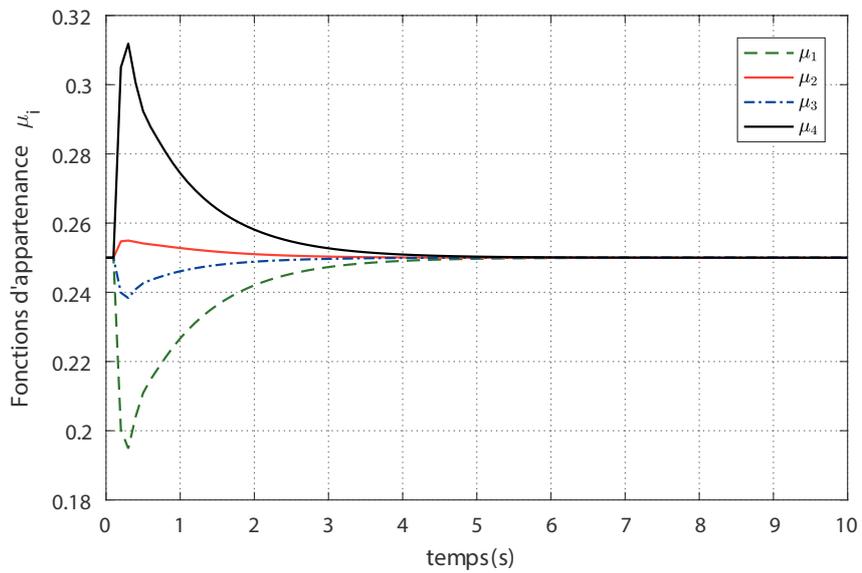


FIGURE 2.17 – Signaux de commande appliqués au système.

FIGURE 2.18 – Evolution des fonctions d'activation μ_i .

2.5 Conclusion

Dans ce chapitre, différentes lois de commande ont été élaborées telles que la commande linéaire quadratique (LQ) qui a été appliquée autour d'un point de fonctionnement bien choisi. Par la suite, une commande par backstepping incorporant une action intégrale, a été conçue pour contrôler l'altitude et les positions du drone. Une fonction quadratique candidate de Lyapunov a été retenue pour calculer une commande stabilisante. Les résultats obtenus montrent les bonnes performances de la commande proposée. La troisième commande élaborée est une commande prédictive à base de modèle (MPC) formulée dans un cadre de temps discret et s'appuyant sur le modèle T-S du drone. Les effets des paramètres de la commande prédictive ont été étudiés et présentés par simulations numériques. Ces paramètres influencent d'une manière significative le comportement du système en boucle fermée, notamment, en termes de stabilité, rapidité et précision entre la sortie prédite et la référence. Leurs valeurs doivent être donc méticuleusement choisies.

Le réglage de ces paramètres n'est pas assez trivial même pour des experts automaticiens. Généralement, les relations liant les paramètres de réglage aux phénomènes physiques restent encore non établies. Par conséquent, ce type de réglage présente, jusqu'à aujourd'hui, une difficulté pour les applications industrielles. La proposition des méthodes systématique pour le réglage de paramètres de lois de commande faisait l'objet de nos travaux de recherche présentés dans les prochaines chapitres.

Synthèse de commandes intelligentes à base des métaheuristique

Sommaire

3.1	Introduction	61
3.2	État de l'art sur les algorithmes métaheuristiques d'optimisation statique	61
3.2.1	L'algorithme PSO standard	61
3.2.2	L'algorithme PSO à facteur d'inertie variable PSO-In	63
3.2.3	L'algorithme PSO à position gbest perturbée PSO-gbest	64
3.2.4	L'algorithme PSO à comportement Quantique QPSO	65
3.3	Commande PID à supervision floue et réglage par PSO	68
3.3.1	Formulation du problème	68
3.3.2	Résultats de simulation et discussion	70
3.4	Commande LQG à réglage par métaheuristiques	80
3.4.1	Formulation du problème	80
3.4.2	Résultats de simulation et discussion	81
3.5	Commande prédictive MPC à réglage par métaheuristiques	86
3.5.1	Formulation du problème	86
3.5.2	Résultats de simulation et discussion	87
3.6	Conclusion	92

3.1 Introduction

Comme nous l'avons vu au chapitre précédent, la recherche d'une méthode systématique et à complexité réduite pour l'élaboration de lois de commande robuste, en vue d'une implémentation pratique aisée, constitue l'une des principales préoccupations d'actualité dans la recherche en théorie de la commande robuste.

Ce chapitre est consacré à la présentation d'une méthode hors ligne de réglage et d'optimisation, en vue d'une réduction de la complexité et d'une systématisation de la synthèse, de structures de commande robuste par des métaheuristiques de recherche globale pour le drone QTW étudié. Nous nous intéressons, plus particulièrement, aux problèmes liés à la synthèse de la commande optimale quadratique gaussienne LQG, la commande PID à supervision floue et la commande prédictive MPC.

Nous commençons par la présentation des algorithmes métaheuristiques proposés pour la résolution des problèmes de commande reformulés comme des problèmes d'optimisation non linéaires et contraints. La validation des algorithmes proposés est achevée moyennant un banc des fonctions test issues de la littérature. Nous nous intéressons dans ce chapitre aux techniques d'optimisation par PSO et ses variantes. Divers algorithmes métaheuristiques comme la colonie d'abeilles artificielles (ABC), l'algorithme du cycle d'eaux (WCA) et l'algorithme de recherche d'harmonie (HSA) sont considérés comme supports de comparaison pour les lois de commande développées.

3.2 État de l'art sur les algorithmes métaheuristiques d'optimisation statique

3.2.1 L'algorithme PSO standard

3.2.1.1 Principe et formulation mathématique

L'algorithme d'optimisation par essaims particulaires, ou Particle Swarm Optimization (PSO), est une métaheuristique à population de recherche globale proposée à l'origine par R. Eberhart et J. Kennedy ([Clerc and Kennedy, 2002](#); [Eberhart and Kennedy, 1995](#); [Eberhart and Shi, 2001](#); [Kennedy and Eberhart, 1995](#); [Shi and Eberhart, 1999](#)).

L'algorithme PSO est inspiré de l'éthologie, plus précisément du comportement social des animaux évoluant en essaim tels que les bancs de poissons et les groupes d'oiseaux. La coopération et l'échange d'informations entre individus permettent, bien que chacun des individus ait des capacités limitées, de résoudre des problèmes d'optimisation complexes ([Dréo et al., 2006](#); [Gendreau and Potvin, 2010](#); [Siarry and Michalewicz, 2008](#); [Sun et al., 2012](#); [Yang, 2010](#)). Cette technique a suscité récemment un intérêt remarquable dans divers domaines de l'ingénierie et de recherche, notamment celui de la commande de processus industriels ([Bouallègue et al., 2012, 2015](#); [Sandou, 2009](#); [Magni et al., 2009](#)).

Dans un espace de recherche de dimension D , la i -ème particule de l'essaim, de taille N_{pop} , est modélisée par son vecteur de position $\mathbf{x}_n^i = (x_n^{i,1}, x_n^{i,2}, \dots, x_n^{i,D})$ et par son vecteur de vitesse $\mathbf{v}_n^i = (v_n^{i,1}, v_n^{i,2}, \dots, v_n^{i,D})$. La qualité de sa position est déterminée au sens de la fonction objectif évaluée en ce point.

Au départ de l'algorithme, les particules de l'essaim sont initialisées de manière aléatoire dans l'espace de recherche. Ensuite, à chaque itération n , le vecteur de position et le vecteur de vitesse sont mis à jour respectivement à partir des équations (3.1) et (3.2), dites de mouvement (Bouallègue et al., 2012, 2015) :

$$\mathbf{x}_{n+1}^i = \mathbf{x}_n^i + \mathbf{v}_{n+1}^i \quad (3.1)$$

$$\mathbf{v}_{n+1}^i = w\mathbf{v}_n^i + c_1 r_{1,n}^i (\mathbf{p}_n^i - \mathbf{x}_n^i) + c_2 r_{2,n}^i (\mathbf{p}_n^g - \mathbf{x}_n^i) \quad (3.2)$$

avec w le coefficient d'inertie, c_1 et c_2 les coefficients d'accélération cognitif et social, $r_{1,n}^i$ et $r_{2,n}^i$ deux nombres aléatoires tirés uniformément de l'intervalle $[0, 1]$, à chaque itération n et pour chaque dimension j , \mathbf{p}_n^i est la meilleure position obtenue de la i -ème particule et \mathbf{p}_n^g dénote la meilleure position trouvée par l'essaim dans son ensemble.

Une fois le déplacement effectué, les nouvelles positions des particules sont évaluées conformément aux équations (3.1) et (3.2). Cette procédure est graphiquement illustrée dans la figure 3.1 pour le mouvement d'une particule dans un espace de recherche de dimension 2.

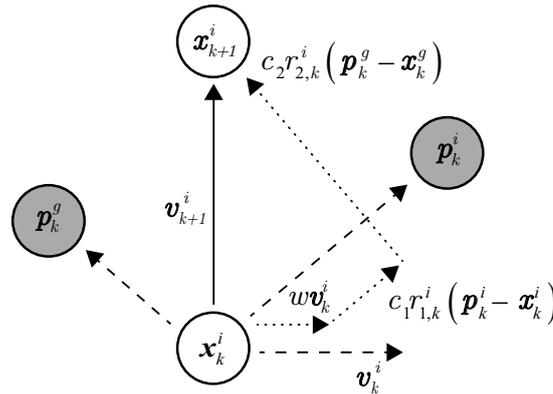


FIGURE 3.1 – Evolution de la position d'une particule dans son espace de recherche.

Dans le formalisme PSO, une particule peut sortir, lors de l'évolution de l'essaim, de l'espace de recherche initialement défini. Il convient alors de définir des contraintes de confinement sur les variables de décision du problème comme suit :

$$x_{\min}^j \leq x_n^{i,j} \leq x_{\max}^j \quad (3.3)$$

où $i \in [1, N_{pop}]$, $j \in [1, D]$ et $n \in [1, N_{iter}]$ représentent respectivement les indices de la particule dans l'essaim, des composantes du vecteur des variables de décision et des itérations de l'algorithme.

A ce propos, la technique la plus simple consiste à ne pas évaluer la nouvelle position de la particule. Néanmoins, il est souvent conseillé d'introduire dans l'algorithme PSO standard des mécanismes, dits de confinement d'intervalles, comme définis dans (Clerc and Kennedy, 2002; Siarry and Michalewicz, 2008). Le principe consiste à dire que, si une particule tend à sortir de l'espace de recherche, alors il faut la ramener au point le plus proche qui soit dans cet espace et modifier sa vitesse en conséquence.

Analytiquement, lorsqu'une coordonnée $x_n^{i,j}$ du vecteur de position, calculée selon les équations de mouvement (3.1) et (3.2), sort de l'intervalle $[x_{\min}^j, x_{\max}^j]$, le mécanisme de confinement introduit consiste à lui attribuer la valeur du point de frontière le plus proche. Il convient de modifier les équations de mouvement précédentes en remplaçant, pour chaque particule, l'équation (3.1) par l'équation (3.4) suivante :

$$x_n^{i,j} = \min \{ \max (x_n^{i,j} + v_n^{i,j}, x_{\min}^j), x_{\max}^j \} \quad (3.4)$$

La vitesse d'une telle particule $v_n^{i,j}$ est, par conséquent, annulée ou, plus généralement, peut être remplacée par son opposée, pondérée par un poids μ , $\tilde{v}_n^{i,j}$, comme suit :

$$\tilde{v}_n^{i,j} = -\mu v_n^{i,j}, \quad 0 \leq \mu < 1 \quad (3.5)$$

À l'origine, les algorithmes PSO ont été formulés pour des problèmes d'optimisation sans contraintes. Le besoin d'un algorithme PSO pour les problèmes contraints, très fréquents en pratique, s'avère important. Les approches basées sur les techniques de pénalisation sont les plus adoptées (Bouallègue et al., 2012). Dans nos travaux, nous nous proposons d'utiliser la technique de pénalité statique extérieure définie comme suit :

$$\varphi(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{n_{con}} \Lambda_j \max [0, g_j(\mathbf{x})]^2 \quad (3.6)$$

où Λ_j et n_{con} représentent respectivement le coefficient de pénalité correspondant à la j -ème contrainte de type inégalité et le nombre total de ces contraintes, $f(\cdot)$ la fonction coût du problème contraint et $g_j(\cdot)$ ses contraintes de type inégalité.

3.2.1.2 Pseudo-code

Pour pouvoir implémenter l'algorithme PSO en un langage évolué (C, C++, MATLAB, etc.) et résoudre les problèmes, reformulés plus tard, nous retenons le pseudo-code de l'algorithme 1.

Le développement de la théorie d'optimisation par PSO a marqué une multitude des variantes proposées dans la littérature (Clerc and Kennedy, 2002; Shi and Eberhart, 1999; Sun et al., 2012; Xinchao, 2010). Ces variantes ont été toujours des améliorations de l'algorithme standard du PSO et ont montré des supériorités remarquables en pratique. Dans ce qui suit, nous passons en revue les principales de ces méthodes à savoir les algorithmes PSO avec facteur d'inertie variable (PSO-In), PSO à position gbest perturbée (PSO-gbest ou pPSO) et PSO à comportement quantique (QPSO).

3.2.2 L'algorithme PSO à facteur d'inertie variable PSO-In

La variante PSO avec facteur d'inertie variable, notée PSO-In, a été proposée par Shi et Eberhart afin d'améliorer les capacités d'exploitation de l'algorithme (Shi and Eberhart, 1999). Cette variante consiste à introduire une loi de décroissance linéaire du facteur d'inertie w en fonction des itérations conformément à l'expression suivante :

$$w_{n+1} = w_{\max} - (w_{\max} - w_{\min}) n / N_{iter} \quad (3.7)$$

Algorithme 1 : PSO standard

1. Définir les paramètres de contrôle de l'algorithme : N_{pop} , c_1 , c_2 , w et N_{iter} .
2. Initialiser aléatoirement les N_{pop} positions \mathbf{x}_0^i et vitesses \mathbf{v}_0^i des particules dans l'essaim. Evaluer cette population initiale et déterminer \mathbf{p}_0^i et \mathbf{p}_0^g .
3. Incrémenter l'indice des itérations n et appliquer, pour chaque particule, les équations de mouvement (3.1) et (3.2), et évidemment (3.4) et (3.5) pour le confinement.
4. Evaluer la fonction coût à minimiser $\varphi_n^i = \varphi(\mathbf{x}_n^i)$:
 - (i) si $\varphi_n^i \leq lbest_n^i$, alors $lbest_n^i = \varphi_n^i$ et $\mathbf{p}_n^i = \mathbf{x}_n^i$,
 - (ii) si $\varphi_n^i \leq gbest_n$, alors $gbest_n = \varphi_n^i$ et $\mathbf{p}_n^g = \mathbf{x}_n^i$. $lbest_n^i = \varphi(\mathbf{p}_n^i)$ et $gbest_n = \varphi(\mathbf{p}_n^g)$ représentant respectivement les meilleures performances de la i -ème particule et de l'ensemble de l'essaim.
5. Si le critère d'arrêt est satisfait (nombre max d'itérations ou autre), l'algorithme s'arrête avec la solution $\mathbf{x}^* = \arg \min_{\mathbf{x}_n^i} \{\varphi(\mathbf{x}_n^i), \forall i, n\}$. Sinon, retour à l'étape 3.

où n est l'itération courante et N_{iter} est le nombre maximal d'itérations, w_{\max} et w_{\min} désignent respectivement les valeurs maximale et minimale du coefficient w .

En effet, le facteur d'inertie w , introduit par Shi et Eberhart, contrôle l'influence de la direction de la particule sur le déplacement futur. Les valeurs minimale et maximale du facteur d'inertie sont fixées dans (Shi and Eberhart, 1999) à $w_{\min} = 0.4$ et $w_{\max} = 0.9$ et la nouvelle équation de mouvement liée à la vitesse devient :

$$\mathbf{v}_{n+1}^i = w_n \mathbf{v}_n^i + c_1 r_{1,n}^i (\mathbf{p}_n^i - \mathbf{x}_n^i) + c_2 r_{2,n}^i (\mathbf{p}_n^g - \mathbf{x}_n^i) \quad (3.8)$$

Un pseudo-code de l'algorithme PSO-In peut être directement dérivé de celui d'Algorithme 1 en tenant compte des nouvelles équations de mouvement (3.1), (3.7) et (3.8) ainsi que des nouveaux paramètres de contrôle w_{\min} et w_{\max} .

3.2.3 L'algorithme PSO à position gbest perturbée PSO-gbest

La technique PSO à solution gbest perturbée, dite PSO-gbest ou pPSO, est une variante de l'algorithme PSO proposée en 2010 par Z. Xinchao (Xinchao, 2010). Au contraire de l'algorithme PSO standard, la technique PSO-gbest consiste à retenir, à chaque itération, une distribution normale de la position globale gbest de l'essaim \mathbf{p}_n^g , en se basant sur les concepts de la théorie des possibilités (Xinchao, 2010). Cette nouvelle formulation de l'algorithme est décrite comme suit :

$$\mathbf{v}_{n+1}^i = w \mathbf{v}_n^i + c_1 r_{1,n}^i (\mathbf{p}_n^i - \mathbf{x}_n^i) + c_2 r_{2,n}^i (\hat{\mathbf{p}}_n^g - \mathbf{x}_n^i) \quad (3.9)$$

où $\hat{\mathbf{p}}_n^g \sim \mathcal{N}(\mathbf{p}_n^g, \sigma)$ représente une distribution normale d'espérance \mathbf{p}_n^g et d'écart type σ pour la variable aléatoire gbest. Cette nouvelle position est appelée p-gbest pour "*possibly at gbest*" selon la théorie des possibilités.

Dans (Xinchao, 2010), la variance $\sigma = \sigma_n$ représente le degré d'incertitude sur l'optimalité de la position gbest \mathbf{p}_n^g de l'algorithme PSO standard. Pour la mise à jour des positions des particules dans l'algorithme PSO-gbest, la variance σ_n est modélisée comme une fonction non-croissante des itérations de l'algorithme. Dans la littérature, différents modèles mathématiques ont été proposés pour cette stratégie. On note principalement les modèles min-max (PSO-gbest 1) de l'équation (3.10), linéaire (PSO-gbest 2) de l'équation (3.11) et aléatoire (PSO-gbest 3) de l'équation (3.12) :

$$\sigma_n = \begin{cases} \sigma_{\max}, & n < \lambda N_{iter} \\ \sigma_{\min}, & \text{sinon} \end{cases} \quad (3.10)$$

$$\sigma_n = \sigma_{\max} - (n - 1)(\sigma_{\max} - \sigma_{\min})/N_{iter} \quad (3.11)$$

$$\sigma_n = \sigma_{\min} + \mathcal{U}(0, 1)(\sigma_{\max} - \sigma_{\min}) \quad (3.12)$$

avec σ_{\max} , σ_{\min} et λ des paramètres de réglage positifs à fixer et $\mathcal{U}(0, 1)$ un nombre aléatoire uniformément distribué dans l'intervalle $[0, 1]$.

Dans (Derouiche et al., 2016; Xinchao, 2010), les valeurs $\sigma_{\max} = 0.15$, $\sigma_{\min} = 0.001$ et $\lambda = 0.5$ sont recommandées comme paramètres de contrôle, d'un tel algorithme.

Un pseudo-code peut être directement dérivé de celui de l'Algorithme 1 en tenant compte des nouvelles équations de mouvement (3.1) et (3.9) ainsi que des modèles (3.10), (3.11) et (3.12) relativement à chaque variante PSO-gbest.

3.2.4 L'algorithme PSO à comportement Quantique QPSO

L'algorithme QPSO, variante relativement récente de J. Sun et al. (Sun et al., 2012), est basé sur des concepts de la mécanique quantique ainsi que des travaux d'analyse dynamique des trajectoires des particules, menés par M. Clerc et J. Kennedy (Clerc and Kennedy, 2002).

Il s'agit d'un algorithme probabiliste dont les équations récurrentes de mouvement sont différentes de celles du PSO standard. Le principe étant d'améliorer l'algorithme vis-à-vis des problèmes de convergence prématurée, de piégeage dans des optima locaux et de confinement des particules. Le vecteur de vitesse n'est plus nécessaire dans le calcul de la nouvelle position d'une particule qui se voit évoluer directement. Un nombre très réduit des paramètres de contrôle caractérise l'algorithme QPSO. Les expériences numériques et les analyses menées montrent que cet algorithme peut offrir des meilleures performances lors de la résolution d'une large gamme de problèmes d'optimisation continue.

Les analyses du comportement dynamique d'une particule de l'essaim, présentées dans (Clerc and Kennedy, 2002), montrent une propriété importante de la métaheuristique PSO. La convergence de l'algorithme est assurée lorsque chaque i -ème particule \mathbf{x}_n^i converge vers son attracteur local $\mathbf{a}_n^i = (a_n^{i,1}, a_n^{i,2}, \dots, a_n^{i,D})^T$ dont la j -ème composante est définie par :

$$a_n^{i,j} = \phi_n^{i,j} p_n^{i,j} + (1 - \phi_n^{i,j}) p_n^{g,j}, \quad \phi_n^{i,j} \sim \mathcal{U}(0, 1) \quad (3.13)$$

où $\phi_n^{i,j} = c_1 r_{1,n}^{i,j} / (c_1 r_{1,n}^{i,j} + c_2 r_{2,n}^{i,j})$ est un nombre aléatoire uniforme dans $[0, 1]$.

D'après les développements menés de l'approche (Sun et al., 2012), l'équation fondamentale de l'algorithme QPSO pour la mesure de la nouvelle position d'une particule est définie comme suit :

$$\mathbf{x}_{n+1}^i = \mathbf{a}_n^i \pm \frac{\mathbf{L}_n^i}{2} \ln \left(\frac{1}{\mathbf{u}_{n+1}^i} \right) \quad (3.14)$$

avec \mathbf{a}_n^i l'attracteur local de la i -ème particule, $\mathbf{u}_{n+1}^i \sim \mathcal{U}(0, 1)$ est une séquence de nombres aléatoires uniformément distribués dans l'intervalle $[0, 1]$ et $\mathbf{L}_n^i = (L_n^{i,1}, L_n^{i,2}, \dots, L_n^{i,D})^T$ représente l'écart type de la distribution du potentiel quantique, issue de la résolution de l'équation de Schrödinger relative permettant de déterminer la zone de recherche de chaque particule.

En pratique, plusieurs méthodes ont été proposées pour le calcul des termes \mathbf{L}_n^i de l'équation (3.14). Une technique simple consiste à fixer les valeurs des composantes $L_n^{i,j}$ durant le processus de recherche. Cette méthode permet de garantir le confinement des particules mais au détriment d'une convergence lente pendant les dernières phases de recherche. Une autre approche consiste à adopter une décroissance, le plus souvent linéaire, vers zéro tout en commençant par une valeur égale à celle de la taille de l'espace de recherche (Clerc and Kennedy, 2002).

Dans (Sun et al., 2012), deux stratégies efficaces sont proposées pour ce calcul. Dans la première, appelée algorithme QDPSO ou QPSO-Type 1, nous retenons, pour chaque composante j du vecteur de position, la formule suivante :

$$L_n^{i,j} = 2\kappa |x_n^{i,j} - a_n^{i,j}| \quad (3.15)$$

avec κ un nombre réel positif, appelé coefficient de Contraction-Elargissement, noté CE.

Ceci permet d'avoir une équation fondamentale de mouvement de la forme :

$$\mathbf{x}_{n+1}^i = \mathbf{a}_n^i \pm \kappa |\mathbf{x}_n^i - \mathbf{a}_n^i| \ln \left(\frac{1}{\mathbf{u}_{n+1}^i} \right) \quad (3.16)$$

La deuxième stratégie, dite QPSO-Type 2, propose une amélioration de la précédente variante et utilise l'expression suivante :

$$L_n^{i,j} = 2\kappa |x_n^{i,j} - \mathcal{C}_n^j| \quad (3.17)$$

où $\mathcal{C}_n = (\mathcal{C}_n^1, \mathcal{C}_n^2, \dots, \mathcal{C}_n^D)^T$ représente la moyenne des meilleures positions de toutes les particules, calculée comme suit :

$$\mathcal{C}_n^j = \frac{1}{N_{pop}} \sum_{i=1}^{N_{pop}} p_n^{i,j} \quad (3.18)$$

Nous obtenons ainsi l'équation fondamentale de mouvement de l'algorithme QPSO-Type 2 suivante :

$$\mathbf{x}_{n+1}^i = \mathbf{a}_n^i \pm \kappa |\mathbf{x}_n^i - \mathcal{C}_n| \ln \left(\frac{1}{\mathbf{u}_{n+1}^i} \right) \quad (3.19)$$

Finalement, un pseudo-code de l'algorithme QPSO peut être retenu de l'Algorithme 2. Nous utilisons les équations fondamentales de mouvement (3.16) ou (3.19) selon qu'il s'agit de la variante QPSO-Type 1 ou QPSO-Type 2.

Algorithme 2 : QPSO

1. Définir les paramètres de contrôle de l'algorithme : N_{pop} , N_{iter} et κ .
2. Initialiser aléatoirement les N_{pop} positions \mathbf{x}_0^i et vitesses \mathbf{v}_0^i des particules dans l'essaim. Evaluer cette population initiale et déterminer \mathbf{p}_0^i et \mathbf{p}_0^g .
3. Incrémenter l'indice des itérations n et calculer \mathbf{C}_n selon (3.18) pour l'algorithme QPSO-Type 2 ou l'attracteur local \mathbf{a}_n^i de l'équation (3.13) pour la variante QPSO-Type 1.
4. Mettre à jour la nouvelle position \mathbf{x}_n^i selon les équations de mouvement (3.16) ou (3.19).
5. Evaluer la fonction coût à minimiser $\varphi_n^i = \varphi(\mathbf{x}_n^i)$:
 - (i) si $\varphi_n^i \leq lbest_n^i$, alors $lbest_n^i = \varphi_n^i$ et $\mathbf{p}_n^i = \mathbf{x}_n^i$,
 - (ii) si $\varphi_n^i \leq gbest_n$, alors $gbest_n = \varphi_n^i$ et $\mathbf{p}_n^g = \mathbf{x}_n^i$. $lbest_n^i = \varphi(\mathbf{p}_n^i)$ et $gbest_n = \varphi(\mathbf{p}_n^g)$ représentant respectivement les meilleures performances de la i -ème particule et l'ensemble de l'essaim.
6. Si le critère d'arrêt est satisfait (nombre max d'itérations ou autre), l'algorithme s'arrête avec la solution $\mathbf{x}^* = \arg \min_{\mathbf{x}_n^i} \{\varphi(\mathbf{x}_n^i), \forall i, n\}$. Sinon, retour à l'étape 3.

3.2.4.1 La variante QPSO-GM

Cette variante de l'algorithme QPSO, proposée à l'origine par Liu en 2006, est basée sur l'introduction d'un mécanisme de mutation pour la meilleure position globale et la moyenne des meilleures positions (Sun et al., 2012). La nouvelle position résultante \mathbf{x}_n^i de la i -ème particule est obtenue par l'opérateur mutation $m(\mathbf{x}_n^i) = \mathbf{x}_n^i + \mathcal{G}(0, \delta)$.

En pratique, cet opérateur de mutation est appliqué soit sur la meilleure position locale \mathbf{p}_n^i soit sur la moyenne des meilleures positions \mathbf{C}_n comme suit :

$$m(\mathbf{p}_n^i) = \mathbf{p}_n^i + \mathcal{G}(0, \delta) \quad (3.20)$$

$$m(\mathbf{C}_n) = \mathbf{C}_n + \mathcal{G}(0, \delta) \quad (3.21)$$

avec $\mathcal{G}(0, \delta)$ est la distribution gaussienne de moyenne 0 et d'écart type δ .

3.2.4.2 La variante QPSO-HPD

Contrairement à la distribution exponentielle qui caractérise la position d'une particule dans l'essaim, une distribution hybride peut améliorer davantage les performances du QPSO (Sun et al., 2012). Dans cette variante à distribution hybride de probabilité, l'équation de mouvement (3.16) de la i -ème particule peut être réécrite comme suit :

$$\mathbf{x}_{n+1}^i = \mathbf{a}_n^i + \mathbf{\Lambda}_{n+1}^i \quad (3.22)$$

où $\mathbf{\Lambda}_n^i = (\Lambda_n^{i,1}, \Lambda_n^{i,2}, \dots, \Lambda_n^{i,D})^T$ est une séquence aléatoire qui converge vers zéro pour permettre à \mathbf{x}_n^i d'atteindre \mathbf{a}_n^i .

Dans le QPSO standard, cette séquence aléatoire suit une loi de distribution exponentielle et peut être décomposée comme suit :

$$\Lambda_{n+1}^i = \tau_{n+1}^i + \vartheta_{n+1}^i \quad (3.23)$$

avec τ_n^i et ϑ_n^i sont deux séquences aléatoires générées par deux distributions de probabilité différentes.

Deux exemples de choix des distributions τ_n^i et ϑ_n^i , proposés dans (Sun et al., 2012), sont les distributions exponentielle et normale définies, respectivement par :

$$\tau_{n+1}^i = \pm \lambda_1 |\mathbf{x}_n^i - \mathbf{a}_n^i| \ln \left(\frac{1}{\mathbf{u}_{n+1}^i} \right) \quad (3.24)$$

$$\vartheta_{n+1}^i = \lambda_2 |\mathbf{x}_n^i - \mathbf{C}_n| \zeta_{n+1}^i \quad (3.25)$$

avec \mathbf{u}_n^i est une séquence des nombres aléatoires uniformes dans $[0, 1]$, $\zeta_n^i = (\zeta_n^{i,j})_{j=1,2,\dots,D}$ est une séquence des nombres aléatoires à distribution normale, \mathbf{C}_n est la moyenne des meilleures positions et λ_1 et λ_2 sont des coefficients CE.

Ainsi, nous retenons la nouvelle équation fondamentale suivante de mouvement d'une particule dans l'algorithme QPSO-HPD :

$$\mathbf{x}_{n+1}^i = \mathbf{a}_n^i \pm \lambda_1 |\mathbf{x}_n^i - \mathbf{a}_n^i| \ln \left(\frac{1}{\mathbf{u}_{n+1}^i} \right) + \lambda_2 |\mathbf{x}_n^i - \mathbf{C}_n| \zeta_{n+1}^i \quad (3.26)$$

3.3 Commande PID à supervision floue et réglage par PSO

3.3.1 Formulation du problème

La commande proportionnelle-intégrale-dérivée PID a toujours été longtemps utilisée pour la conduite de processus dynamiques (Aström and Hägglund, 1995). Une structure de supervision par logique floue pour le réglage des paramètres d'un contrôleur PID (GS-PID) est proposée dans cette section pour la stabilisation des dynamiques du véhicule QTW. Dans le temps discret, nous retenons pour le superviseur par logique floue les variables d'entrée linguistiques e_k et Δe_k définies comme suit :

$$e(kT_s) = e_k = y_k^d - y_k, \quad k \in \mathbb{N} \quad (3.27)$$

$$\Delta e(kT_s) = \Delta e_k = (e_k - e_{k-1})/T_s \quad (3.28)$$

où T_s est la période d'échantillonnage du superviseur flou, y_k est la sortie échantillonnée du système et y_k^d la grandeur de consigne.

Les variables linguistiques de sortie sont les gains K_p , K_i et K_d du correcteur PID conçu. Afin de normaliser les univers de discours du système d'inférence, nous introduisons les facteurs d'échelle κ_e et $\kappa_{\Delta e}$ et retenons les nouvelles expressions des entrées comme suit :

$$\hat{e}_k = \kappa_e e_k \quad (3.29)$$

$$\Delta \hat{e}_k = \kappa_{\Delta e} \Delta e_k \quad (3.30)$$

Un ensemble des règles floues de la forme (3.31) est utilisé dans le système d'inférence du superviseur pour le calcul et le réglage des gains K_p , K_i et K_d (Sinthipsomboon et al., 2011; Zhao et al., 1993) :

$$\text{Si } \hat{e}_k \text{ est } \mathbf{A}_i \text{ ET } \Delta \hat{e}_k \text{ est } \mathbf{B}_i \text{ Alors } K_p \text{ est } \mathbf{C}_i, K_i \text{ est } \mathbf{D}_i \text{ ET } K_d \text{ est } \mathbf{E}_i \quad (3.31)$$

où \mathbf{A}_i , \mathbf{B}_i , \mathbf{C}_i , \mathbf{D}_i et \mathbf{E}_i sont respectivement les sous-ensembles flous des variables linguistiques \hat{e}_k , $\Delta \hat{e}_k$, K_p , K_i et K_d .

Dans cette approche de supervision intelligente, dont le schéma de principe est donné dans la figure 3.2, trois bases de 49 règles floues sont utilisées, Tableaux 3.1, 3.2 et 3.3. Les significations des labels ; désignant les normes des valeurs linguistique, sont N (Négatif), Z (Zéro), P (Positive), NG (Négative Grande) et PG (Positive Grande). Les fonctions

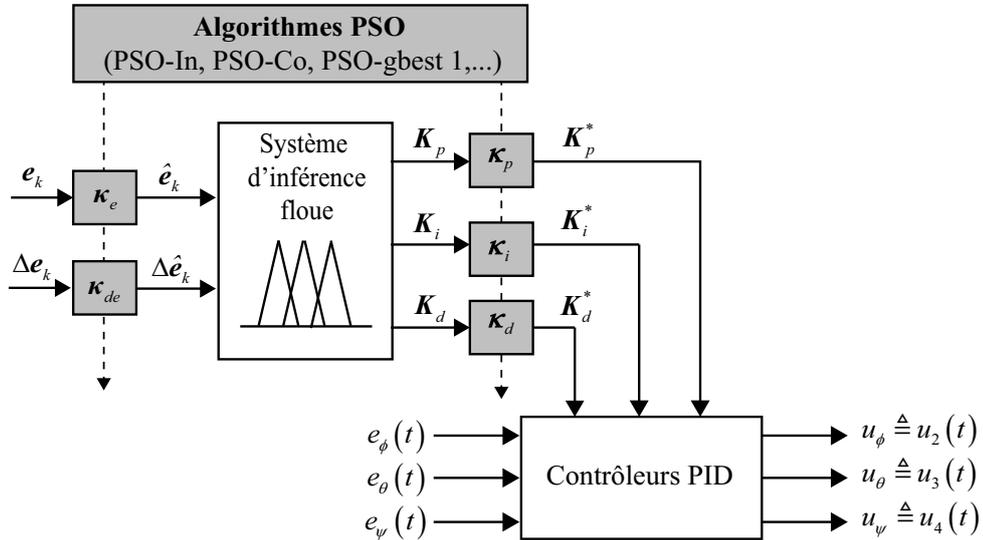


FIGURE 3.2 – Structure proposée de la supervision floue de correcteurs PID.

d'appartenance des différentes variables linguistiques sont choisies triangulaires, symétriques et équidistantes. Nous adoptons la méthode d'inférence Som-Prod ainsi qu'une défuzzification par le centre de gravité.

Tableau 3.1 – Base des règles floues pour le gain K_p .

$\Delta \hat{e}_k / \hat{e}_k$	NG	N	Z	P	PG
NG	NG	NG	NG	N	Z
N	NG	N	N	N	Z
Z	NG	N	Z	P	PG
P	Z	P	P	P	PG
PG	Z	P	PG	PG	PG

Tableau 3.2 – Base des règles floues pour le gain K_i .

$\Delta \hat{e}_k / \hat{e}_k$	NG	N	Z	P	PG
NG	PG	NG	PG	N	NG
N	PG	P	P	Z	NG
Z	P	P	Z	N	NG
P	Z	P	N	N	NG
PG	Z	N	NG	NG	NG

Tableau 3.3 – Base des règles floues pour le gain K_d .

$\Delta \hat{e}_k / \hat{e}_k$	NG	N	Z	P	PG
NG	NG	NG	NG	P	PG
N	N	N	N	Z	PG
Z	Z	N	Z	P	PG
P	P	N	P	P	PG
PG	PG	P	PG	PG	PG

Parmi les paramètres de réglage de la structure de supervision proposée, dans la figure 3.2 figurent les facteurs d'échelle κ_e , $\kappa_{\Delta e}$, κ_p , κ_i et κ_d . Nous nous proposons ainsi de les déterminer moyennant les algorithmes métaheuristiques PSO précédemment présentés. Pour cela, nous retenons la formulation (3.32) du problème de réglage des gains supervisés de correcteurs PID pour la stabilisation d'attitude du drone QTW :

$$\left\{ \begin{array}{l} \text{minimiser } f_i(\mathbf{x}) \\ \mathbf{x} = (\kappa_e, \kappa_{\Delta e}, \kappa_p, \kappa_i, \kappa_d)^T \in \mathcal{S} \subseteq \mathbb{R}_+^{15} \\ \text{sous contraintes :} \\ g_1(\mathbf{x}) = D_\phi - D_\phi^{\max} \leq 0 \\ g_2(\mathbf{x}) = D_\theta - D_\theta^{\max} \leq 0 \\ g_3(\mathbf{x}) = D_\psi - D_\psi^{\max} \leq 0 \end{array} \right. \quad (3.32)$$

avec $f_i : \mathbb{R}^{15} \rightarrow \mathbb{R}$ les fonctions objectifs ($i \in \{\phi, \theta, \psi\}$), $g_j : \mathbb{R}^{15} \rightarrow \mathbb{R}$ représentent les contraintes du problème, $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}_+^{15}, \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}\}$ est l'espace de recherche borné, et D_ϕ , D_θ et D_ψ sont respectivement les dépassements des réponses indicielles en roulis, tangage et lacet, $\kappa_p = (\kappa_p^\phi, \kappa_p^\theta, \kappa_p^\psi)$, $\kappa_i = (\kappa_i^\phi, \kappa_i^\theta, \kappa_i^\psi)$, $\kappa_d = (\kappa_d^\phi, \kappa_d^\theta, \kappa_d^\psi)$, $\kappa_e = (\kappa_e^\phi, \kappa_e^\theta, \kappa_e^\psi)$ et $\kappa_\Delta = (\kappa_\Delta^\phi, \kappa_\Delta^\theta, \kappa_\Delta^\psi)$.

3.3.2 Résultats de simulation et discussion

Les objectifs de la commande PID supervisée pour le drone QTW sont la stabilisation de ses dynamiques de roulis, de tangage et de lacet en vol stationnaire ainsi que la garantie de robustesse vis-à-vis des défauts au niveau des actionneurs électriques qui entraînent les hélices. Pour la synthèse des correcteurs, nous considérons le modèle non linéaire (1.30) du

convertible QTW. Les surfaces de décision floue des différents gains des correcteurs PID sont présentées dans la figure 3.3. Ces surfaces floues sont ensuite calibrées, relativement à chaque dynamique contrôlée, grâce aux valeurs optimales $(\kappa_p^*, \kappa_i^*, \kappa_d^*) \in \mathbb{R}_+^3 \times \mathbb{R}_+^3 \times \mathbb{R}_+^3$ des facteurs d'échelle du problème 3.32.

Pour la résolution numérique, tous les algorithmes PSO proposés sont exécutés sur un ordinateur PC de processeur CORE-i3 avec 2,4 GHz de vitesse et ressources en mémoire RAM de 4 Go. Les contraintes non analytiques du problème sont définies par les mesures des dépassements $D_\phi^{\max} = D_\theta^{\max} = D_\psi^{\max} = 5\%$. Pour toutes les expériences numériques menées, nous retenons les mêmes valeurs des paramètres communs des algorithmes PSO (taille de population $N_{pop} = 30$ et maximum d'itérations $N_{iter} = 100$). Les paramètres spécifiques sont donnés dans le tableau 3.4.

Tableau 3.4 – Paramètres de contrôle des différents algorithmes PSO.

Paramètres	Valeurs
Nombre des variables de décision	$D = 15$
Coefficient d'accélération cognitif	$c_1 = 0.5$
Coefficient d'accélération social	$c_2 = 0.3$
Facteur d'inertie	$w_{\max} = 0.9$ et $w_{\min} = 0.9$
Coefficients de gbest perturbée	$\sigma_{\max} = 0.15, \sigma_{\min} = 0.001, \lambda = 0.5$

Les résultats d'optimisation obtenus suite à 30 essais indépendants sur le problème (3.32) sont regroupés dans le tableau 3.5. Le temps de calcul, relatif à chaque essai, est également mentionné pour chaque algorithme.

Tableau 3.5 – Résultats d'optimisation de 30 essais indépendants sur le problème (3.32).

Algorithmes	Meilleur cas	Moyen cas	Pire cas	Ecart-type	Temps (s)
PSO	0,01404	0,01436	0,01490	3,540e-04	6766,741
PSO-In	0,01373	0,01418	0,01472	3,631e-04	6664,372
PSO-gbest 1	0,01441	0,01473	0,01497	1,816e-04	11484,21
PSO-gbest 2	0,01477	0,01542	0,01616	5,551e-04	6286,384
PSO-gbest 3	0,01535	0,01576	0,01606	3,140 e-04	11031,76

Pour ce problème d'optimisation non linéaire et de grande dimension, i.e. \mathbb{R}_+^{15} , des solutions faisables sont obtenues à 85% des essais et pendant un temps de calcul raisonnable, Tableau 3.5. Les historiques de convergence des différentes variantes PSO sont représentés dans la figure 3.4. Ce résultat montre la supériorité, en termes des capacités exploration/exploitation, de qualité des solutions obtenues et de la dynamique de convergence non-prématurée. La figure 3.5 montre les réponses temporelles des dynamiques contrôlées $\phi(t)$, $\theta(t)$ et $\psi(t)$ pour les différents algorithmes PSO (BenKhoud et al., 2018).

Les performances temporelles de la commande, telles que le temps de montée (RT), le temps d'établissement (ST), le dépassement (OV) et l'erreur statique en régime permanent (Ess) sont données dans les tableaux 3.6, 3.7 et 3.8. Ces résultats nous permettent de

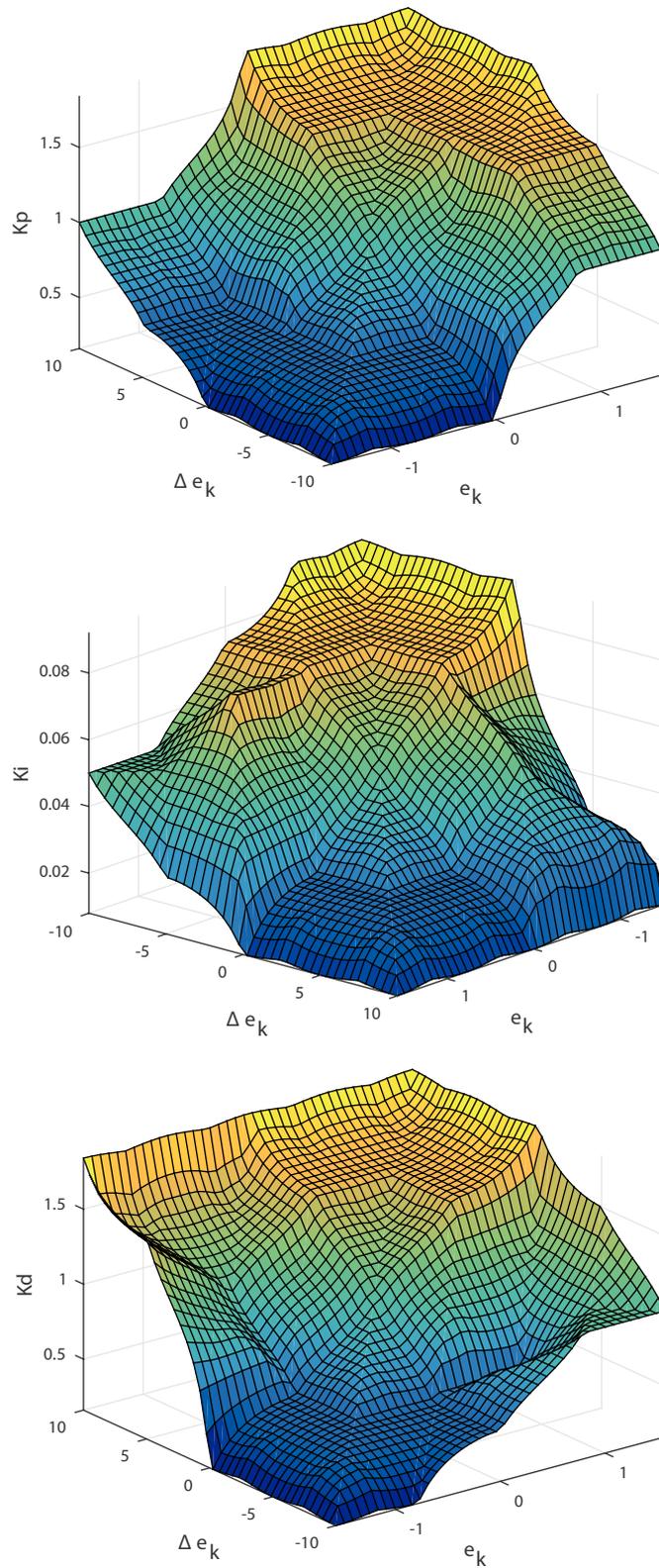


FIGURE 3.3 – Surfaces de décision floue des gains des correcteurs GS-PID.

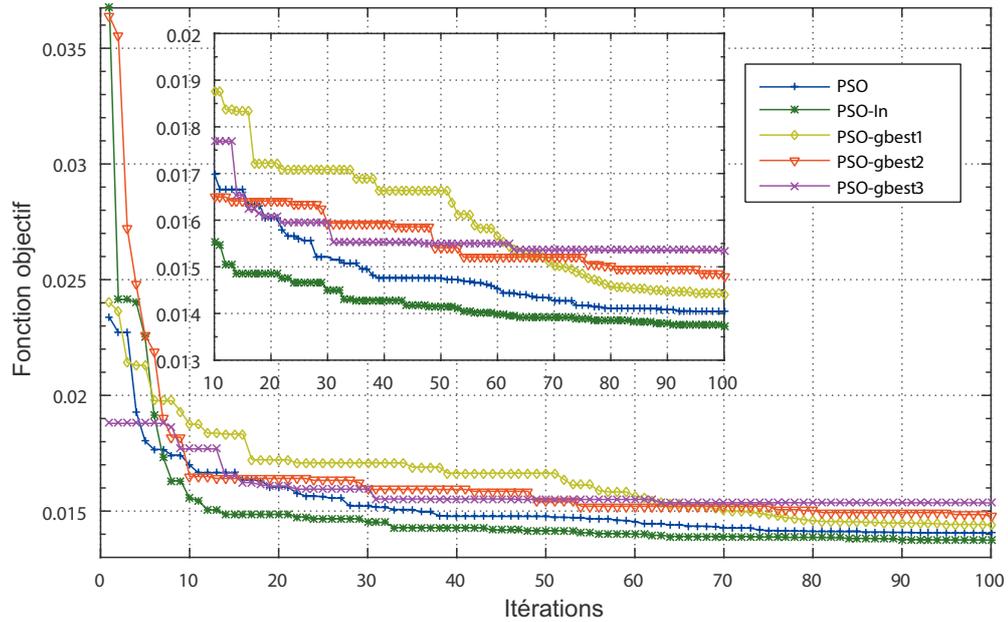


FIGURE 3.4 – Historiques de convergence des algorithmes PSO pour le problème (3.32).

constater que la stabilisation du véhicule QTW est garantie avec des performances comparables. Les résultats de simulation numérique indiquent clairement que l'algorithme PSO-In dépasse significativement tous les autres algorithmes pour la résolution du le problème de réglage des contrôleurs GS-PID flous.

Tableau 3.6 – Performances temporelles de la dynamique : roulis $\phi(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO	0,0569	0,0343	0,3561	3,627e-05
PSO-In	0,0661	0,0408	0,3991	4,866e-05
PSO-gbest 1	0,0573	0,0347	0,4382	1,473e-04
PSO-gbest 2	0,0749	0,0458	0,3550	1,487e-04
PSO-gbest 3	0,0568	0,0342	0,3946	1,563e-04

D'autre part, une étude comparative des performances de la boucle fermée pour les régulateurs GS-PID flous optimisés, des contrôleurs flous standard (Erginer and Altug, 2012) et des GS-PID flous non optimisés (Hasanjani et al., 2015), est menée à travers les figures 3.6, 3.7 et 3.8. Les tableaux 3.9, 3.10 et 3.11 regroupent les performance dans le domaine temporel des dynamiques contrôlées de roulis, tangage et lacet du drone. Il est clair que tous les contrôleurs stabilisent le vol du véhicule QTW mais une supériorité remarquable en rapidité et amortissement de la réponse des contrôleurs flous avec réglage PSO est mentionnée. La précision de la réponse en régime permanent est améliorée davantage avec les approches par réglage PSO. En outre, les efforts des signaux de commande, comme le montrent les figures 3.6, 3.7 et 3.8, sont nettement moins importants pour la

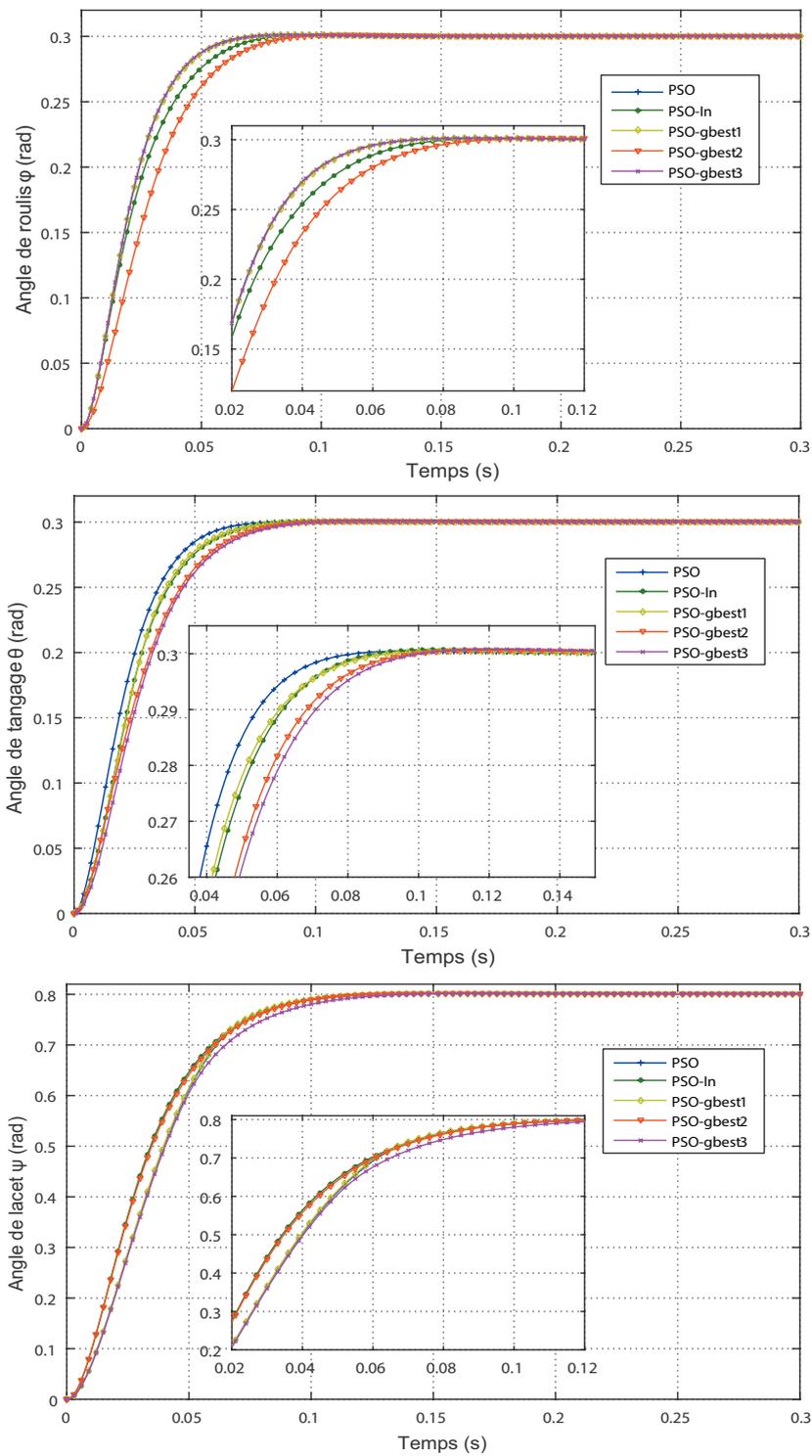


FIGURE 3.5 – Stabilisation des dynamiques de roulis, tangage et lacet du véhicule QTW.

Tableau 3.7 – Performances temporelles de la dynamique : tangage $\theta(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO	0,0597	0,0358	0,1573	8,568e-05
PSO-In	0,0673	0,0392	0,2609	2,740e-04
PSO-gbest 1	0,0669	0,0379	0,1562	8,200e-05
PSO-gbest 2	0,0746	0,0451	0,2049	2,441e-04
PSO-gbest 3	0,0774	0,0455	0,2712	3,332e-04

Tableau 3.8 – Performances temporelles de la dynamique : lacet $\psi(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO	0,0934	0,0540	0,1409	1,452e-04
PSO-In	0,0935	0,0551	0,1292	1,299e-04
PSO-gbest 1	0,0926	0,0535	0,1617	1,442e-04
PSO-gbest 2	0,0946	0,0561	0,2106	8,178e-04
PSO-gbest 3	0,1038	0,0590	0,1844	7,559e-04

stratégie de contrôle à base des métaheuristique PSO proposée. Ce résultat justifie encore l'efficacité de l'approche de contrôle robuste proposée pour une future mise en œuvre pratique dans le monde réel (BenKhoud et al., 2018).

Tableau 3.9 – Performances temporelles comparées de la dynamique de roulis $\phi(t)$.

Contrôleurs	PT (s)	ST (s)	RT (s)	OV(%)	Ess
Flou	0,0310	0,0256	0,0173	1,6542	0,0029
GS-PID flou	0,2090	0,2594	0,0985	2,7866	0,0053
GS-PID flou optimisé	0,1940	0,1369	0,0933	0,4361	7,376e-04

La robustesse des correcteurs PID est testée vis-à-vis des défauts d'actionneurs. En effet, nous considérons une perte d'efficacité de 20% sur les signaux de commande des quatre moteurs des hélices. La figure 3.9 montre la robustesse des régulateurs PID supervisés à la base de l'algorithme PSO-In. Ces contrôleurs, ayant désormais un comportement adaptatif, réduisent l'effet de ces défauts en ajustant, au cours de temps, les gains supervisés comme représenté dans les figures 3.10, 3.11 et 3.12. En effet, le système contrôlé retourne suite à l'apparition des défauts à l'attitude désirée avec une dynamique amortie et relativement rapide, i.e. un temps de réponse de l'ordre de 0,1 s pour les variables de tangage et de lacet et de 0.05 s pour l'état de roulis.

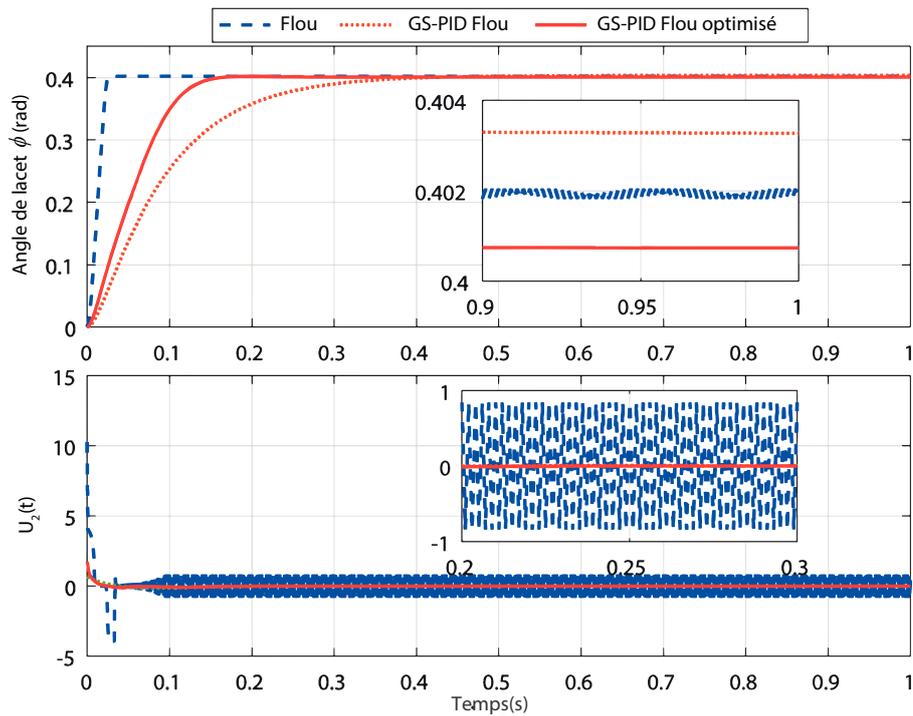


FIGURE 3.6 – Performances comparées des contrôleurs flou, GS-PID flou et GS-PID flou optimisé : dynamique de roulis.

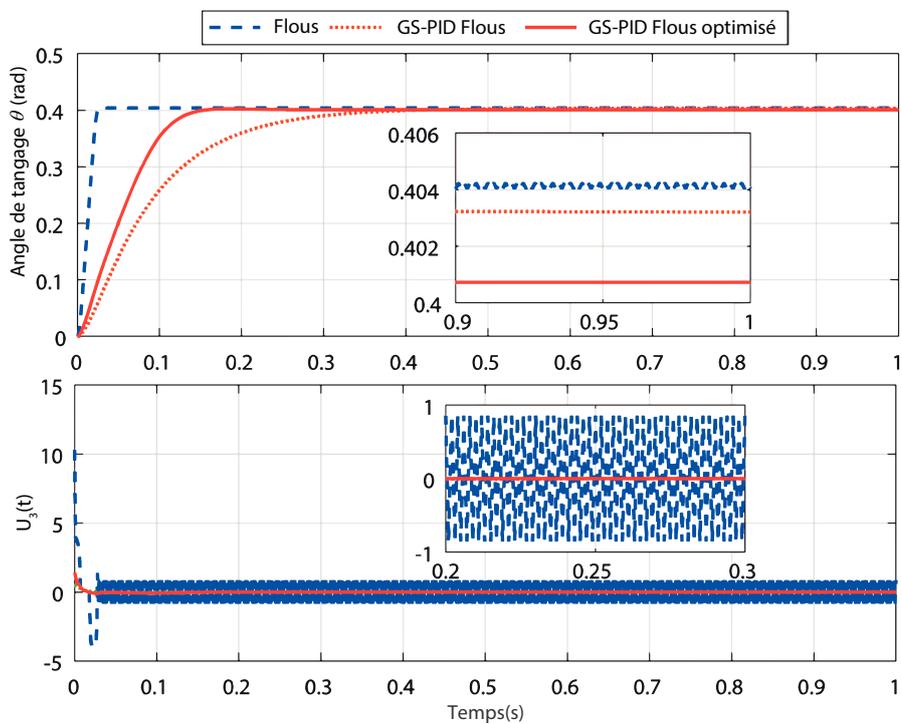


FIGURE 3.7 – Performances comparées des contrôleurs flou, GS-PID flou et GS-PID flou optimisé : dynamique de tangage.

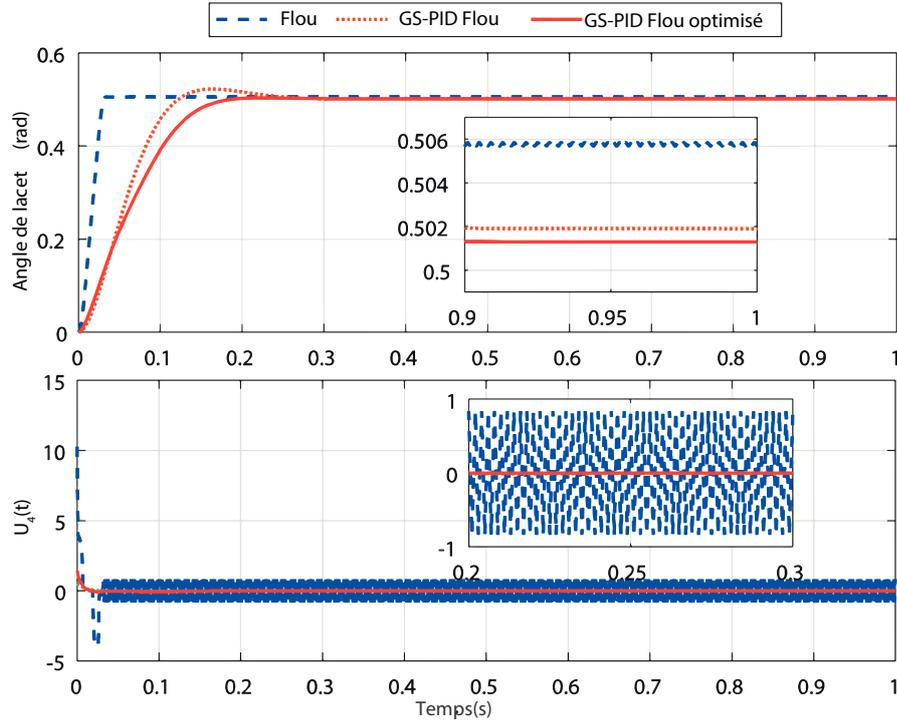


FIGURE 3.8 – Performances comparées des contrôleurs flou, GS-PID flou et GS-PID flou optimisé : dynamique de lacet.

Tableau 3.10 – Performances temporelles comparées de la dynamique de tangage $\theta(t)$.

Contrôleurs	PT (s)	ST (s)	RT (s)	OV(%)	Ess
Flou	0,0270	0,0211	0,0159	2,7548	0,0179
GS-PID flou	0,1920	0,2595	0,0908	3,6390	0,0027
GS-PID flous optimisé	0,1840	0,1325	0,0912	0,5496	7,234e-04

Tableau 3.11 – Performances temporelles comparées de la dynamique de lacet $\psi(t)$.

Contrôleurs	PT (s)	ST (s)	RT (s)	OV(%)	Ess
Flou	0,0330	0,0297	0,0206	1,1808	0,0057
GS-PID flou	0,1580	0,2326	0,0757	6,4252	0,0019
GS-PID flou optimisé	0,2230	0,1605	0,1103	0,6818	0,0013

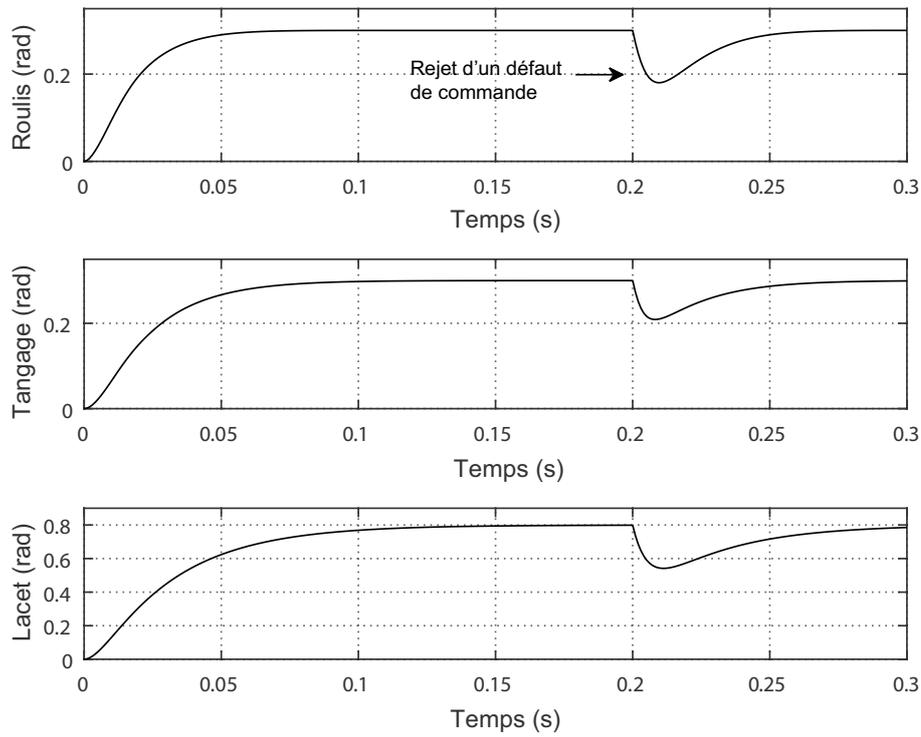


FIGURE 3.9 – Robustesse vis-à-vis d'un défaut de 20% sur les signaux de commande.

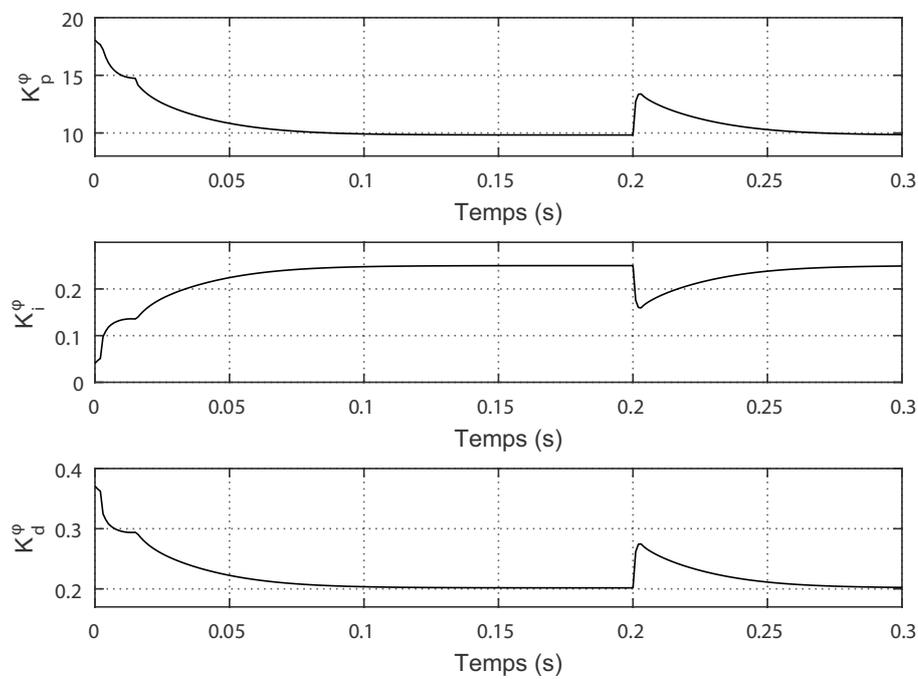


FIGURE 3.10 – Variation temporelle des gains du PID pour la dynamique de roulis.

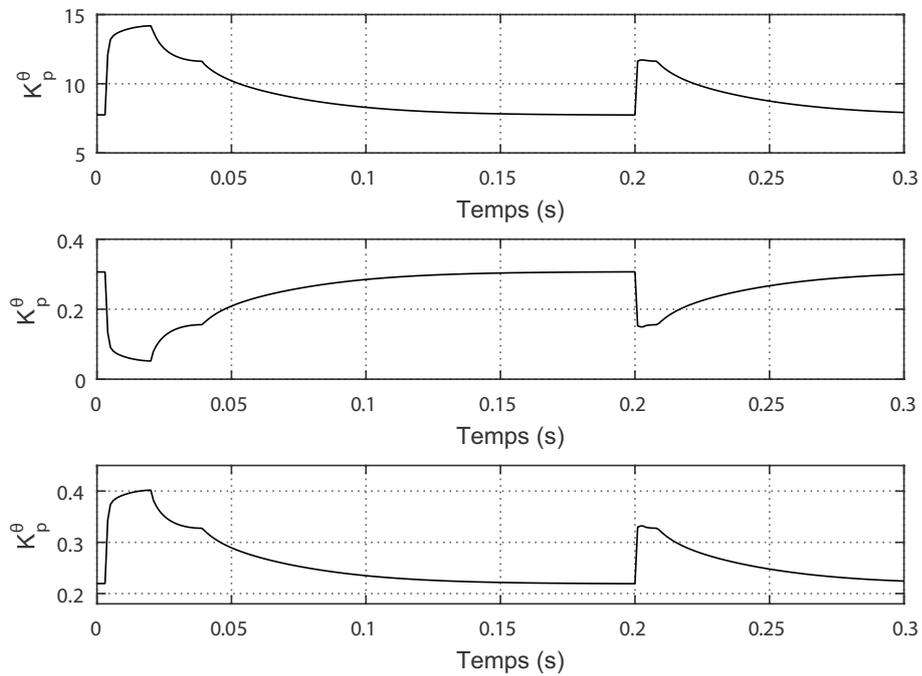


FIGURE 3.11 – Variation temporelle des gains du PID pour la dynamique de tangage.

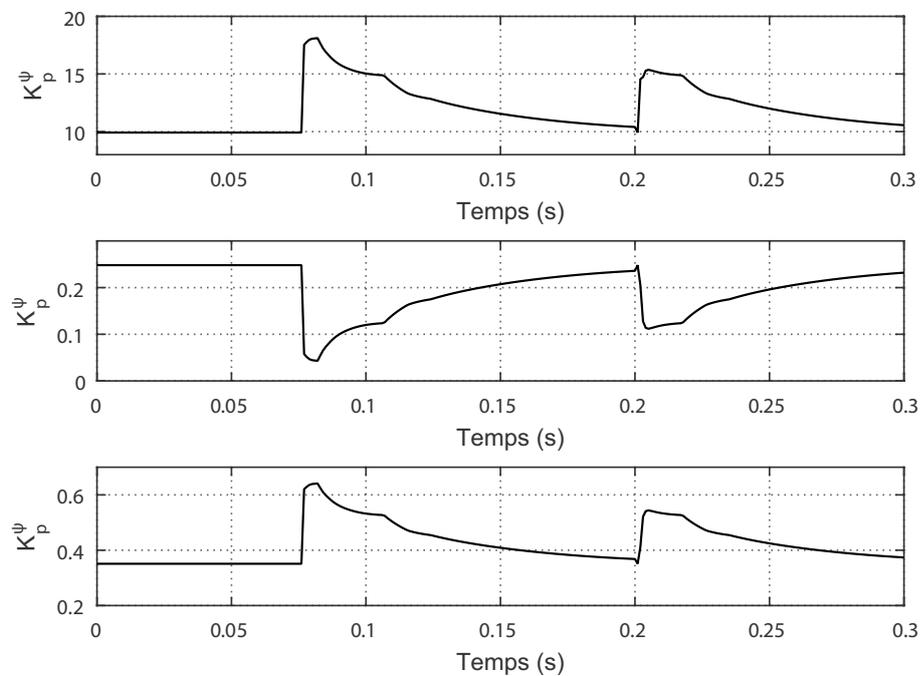


FIGURE 3.12 – Variation temporelle des gains du PID pour la dynamique de lacet.

3.4 Commande LQG à réglage par métaheuristiques

La synthèse d'une loi de commande optimale quadratique de type LQR et/ou quadratique gaussienne LQG pour le drone QTW dépend fortement du choix des filtres de pondération définis sur les états et les signaux de commande (BenKhoud and Bouallègue, 2016b; Fessi and Bouallègue, 2016; BenKhoud and Bouallègue, 2018; Argentim et al., 2013; Khatoun et al., 2014).

3.4.1 Formulation du problème

La difficulté du choix des tels paramètres de réglage a toujours présenté un handicap dans l'utilisation de cette stratégie de commande pour les véhicules aériens. Ainsi, les résultats présentés dans cette section portent sur la proposition d'une technique de synthèse systématique de ces filtres basée sur l'optimisation par métaheuristiques comme montré dans la figure 3.13.

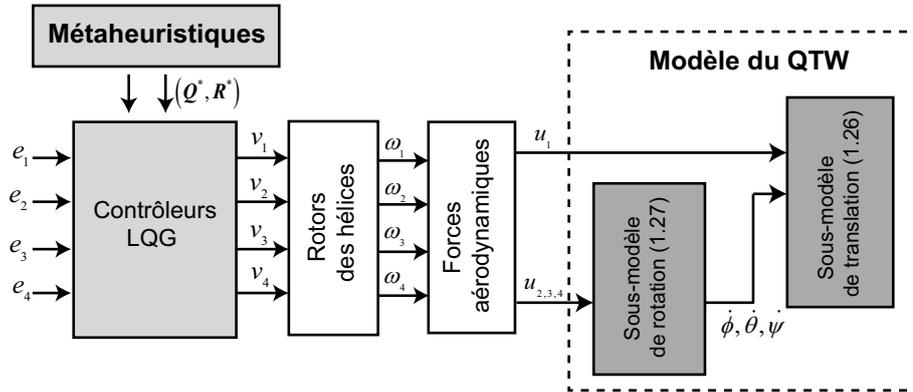


FIGURE 3.13 – Structure de commande LQG à réglage par métaheuristiques.

Les variables de décision du problème de commande reformulé sont les éléments des matrices de pondération $\mathbf{Q} = \mathbf{Q}^T \geq 0$ et $\mathbf{R} = \mathbf{R}^T > 0$, choisies de forme diagonale comme suit :

$$\mathbf{Q} = \begin{bmatrix} q_1 & 0 & \dots & 0 \\ 0 & q_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & q_{n_{\mathbf{Q}}} \end{bmatrix} = \text{diag}(q_1, q_2, \dots, q_{n_{\mathbf{Q}}}) \geq 0 \quad (3.33)$$

$$\mathbf{R} = \begin{bmatrix} r_1 & 0 & \dots & 0 \\ 0 & r_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & r_{n_{\mathbf{R}}} \end{bmatrix} = \text{diag}(r_1, r_2, \dots, r_{n_{\mathbf{R}}}) > 0 \quad (3.34)$$

Ces paramètres de commande et de réglage sont tous regroupés dans le vecteur des variables de décision $\mathbf{x} \in \mathbb{R}_+^{n_{\mathbf{Q}}+n_{\mathbf{R}}}$ suivant :

$$\mathbf{x} = [q_1, q_2, \dots, q_{n_{\mathbf{Q}}}, r_1, r_2, \dots, r_{n_{\mathbf{R}}}]^T \in \mathbb{R}_+^{n_{\mathbf{Q}}+n_{\mathbf{R}}} \quad (3.35)$$

où $\mathbb{R}_+^{n_Q+n_R}$ présente l'ensemble des réels positifs de dimension $n_Q + n_R$.

Dans cette formulation, la fonction coût à minimiser est la somme pondérée des critères de performance temporelle ITSE (Integral Time of Square Error). Les contraintes opérationnelles du problème portent sur les temps de montée t_m^x , t_m^y , t_m^z et t_m^ψ des réponses indicielles :

$$\left\{ \begin{array}{l} \underset{\mathbf{x} \in \mathbb{R}_+^{16}}{\text{minimiser}} f(\mathbf{x}) = \sum_{i \in \{x,y,z,\psi\}} \xi_i f_{ITSE,i}(\mathbf{x}) \\ \text{sous contraintes :} \\ g_1(\mathbf{x}) = t_m^x - t_m^{x,\max} \leq 0 \\ g_2(\mathbf{x}) = t_m^y - t_m^{y,\max} \leq 0 \\ g_3(\mathbf{x}) = t_m^z - t_m^{z,\max} \leq 0 \\ g_4(\mathbf{x}) = t_m^\psi - t_m^{\psi,\max} \leq 0 \end{array} \right. \quad (3.36)$$

où $\mathbf{x} = [q_1, q_2, \dots, q_{12}, r_1, r_2, r_3, r_4] \in \mathbb{R}_+^{16} - \{0\}$ est le vecteur des variables de décision, et $t_m^{x,\max} = t_m^{y,\max} = t_m^{z,\max} = t_m^{\psi,\max} = 2s$ sont les valeurs maximales des contraintes sur les temps de montée des dynamiques x , y , z et ψ et où les ξ_i représentent des coefficients de pondération sur les critères d'optimisation.

3.4.2 Résultats de simulation et discussion

Nous utilisons, pour la résolution du problème (3.36), la variante PSO-In de l'algorithme PSO comparé aux métaheuristiques ABC, HSA, WCA et FPSOMA. Les paramètres de contrôle des différents algorithmes sont choisis comme suit (BenKhoud and Bouallège, 2018) :

- PSO-In (Shi and Eberhart, 1999) : $c_1 = c_2 = 2$, $w_{\max} = 0.9$ et $w_{\min} = 0.4$;
- FPSOMA (Mousavi and Alfi, 2015) : $c_1 = c_2 = 0.9$, $w = 0.7298$ et $a = 1.4962$;
- ABC (Bouallège et al., 2015) : limite des abondances $L = 60$;
- HSA (Wang et al., 2015) : taux de mémoire $HMCR = 0.9$ et taux d'ajustement $PAR = 0.3$;
- WCA (Eskandar et al., 2012) : nombre de rivières $N_{sr} = 8$ et distance maximale $d_{\max} = 1e - 03$.

Les résultats d'optimisation relatifs à vingt essais indépendants sont récapitulés dans le tableau 3.12. Pour toutes les expériences, les valeurs des paramètres communs, telles que la taille de population $N_{pop} = 50$ et le nombre maximum d'itérations $N_{iter} = 200$, sont gardés les mêmes pour les différents algorithmes. En effet, ce choix est retenu après plusieurs séries d'exécution de tels algorithmes avec des jeux différents de paramètres. Nous avons procédé à augmenter le nombre d'itérations pour différentes valeurs de la taille de la population. Le tableau 3.13 montre les résultats d'une telle expérience numérique d'optimisation et conclut sur le choix des valeurs appropriées.

Les résultats obtenus avec les différents algorithmes sont très proches de point de vue qualité. Ceci augmente l'optimalité des solutions ainsi trouvées. En effet, on trouve à chaque fois presque le même résultat avec des solveurs différents. Pour les performances liées à la convergence des algorithmes proposés, nous présentons dans la figure 3.14 les historiques de recherche pour chaque méthode. Un examen de ces courbes montre les

Tableau 3.12 – Résultats d’optimisation de vingt exécutions du problème (3.36).

Algorithmes	Meilleur cas	Moyen cas	Pire cas	Ecart-type	Temps (s)
PSO-In	1,3873	1,9469	2,3005	0,5902	500,380
ABC	2,4951	2,8444	2,9440	0,1780	819,987
HSA	1,1431	1,2007	1,2210	0,0814	1099,55
WCA	0.7458	1,0054	1,2372	0,4456	1055,92
FPSOMA	0,8886	1,4040	1,6898	0,2288	1046,60

Tableau 3.13 – Tailles de population et maximum d’itérations : problème (3.36).

Max. d’itérations	Population	Algorithmes				
		PSO-In	ABC	HSA	WCA	FPSOMA
50	20	8,8560	12,6842	6,2918	4,1609	5,4037
	30	7,2450	10,1552	4,4805	3,9616	3,4405
	50	6.6364	8,2559	3,5979	3,9474	3,3829
100	20	5,4312	5,0647	2,2219	3,1527	2,3439
	30	4,7251	3,4215	2,2903	2,5005	1,3175
	50	2,4553	3,9726	2,1425	1,0177	1.3052
200	20	1,9633	3,7996	1,3547	1,9064	1,1019
	30	1,3474	2,7062	1,1467	0,8213	0,9547
	50	1.7910	2,4949	1,0082	1.7458	1,1426

bonnes capacités d’intensification et de diversification de différents algorithmes proposés. Une supériorité remarquable de la méthode PSO-In est notée pour une recherche globale et non prématurée, tandis qu’en recherche locale, c’est la méthode WCA qui est la plus significative. La méthode PSO-In est la plus rapide en terme de temps de calcul par rapport aux autres algorithmes. Afin d’évaluer la capacité des métaheuristiques proposées pour la recherche de solution optimale, les coefficients obtenus des matrices de pondération LQG sont présentés dans le tableau 3.14

Les résultats de simulation relatifs aux algorithmes proposés sont présentés dans la stabilisation des figure 3.15 pour les dynamiques x , y , z et ψ . Les dynamiques du vol considérées sont stabilisées autour des consignes spécifiées pour le drone. De légères différences portant sur les performances du domaine temporel des différents algorithmes sont toutefois notées sur les courbes de réponses indicielles.

Nous menons à présent des comparaisons des algorithmes proposés vis-à-vis des indices de performance du système corrigé dans le domaine temporel. Les tableaux 3.15 à 3.18 présentent les mesures d’erreurs statiques, de dépassements et de temps de montée et d’établissement des dynamiques corrigées. Nous effectuons également des comparaisons avec d’autres types de correcteurs, comme la commande PID et la commande quadratique LGR, afin de montrer les hautes performances et la supériorité de la stratégie de commande LQG-PSO proposée. La figure 3.16 présente les comparaisons menées.

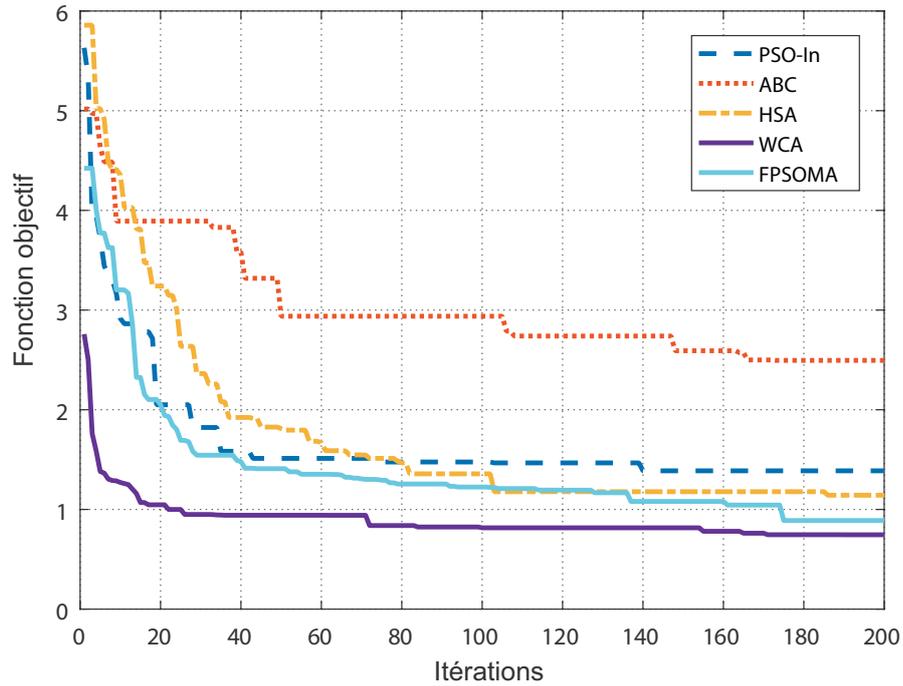


FIGURE 3.14 – Historiques de convergence des algorithmes pour le problème (3.36).

Tableau 3.14 – Valeurs moyennes des coefficients optimaux de matrices de pondération LQG sur 20 exécutions du problème (3.36)

Coefficients	PSO-In	ABC	HSA	WCA	FPSOMA
q_1	10,846	84,670	20,693	36,496	5,2643
q_2	0,2368	0,0659	0,0010	0,0351	0,1403
q_3	19,875	2,6239	15,227	20,616	1,5395
q_4	7,6416	0,4190	0,1712	0,8461	1,2165
q_5	30,790	7,3786	28,002	49,893	12,000
q_6	12,446	0,9119	6,7017	1,0200	4,6652
q_7	24,047	42,765	43,150	8,1457	10,968
q_8	0,8205	0,6453	0,0010	0,0215	0,8056
q_9	32,709	32,106	46,502	29,750	8,6108
q_{10}	29,948	15,400	20,763	7,9948	10,719
q_{11}	22,800	44,525	37,502	42,949	12,451
q_{12}	7,6821	0,2088	8,0742	12,103	8,6125
r_1	0,6580	0,0001	0,0010	2,2021	0,0019
r_2	0,3684	0,2254	0,0265	0,0357	0,0118
r_3	0,0357	0,0137	0,3056	0,0124	0,0644
r_4	15,560	0,0823	0,0235	2,5670	5,3935

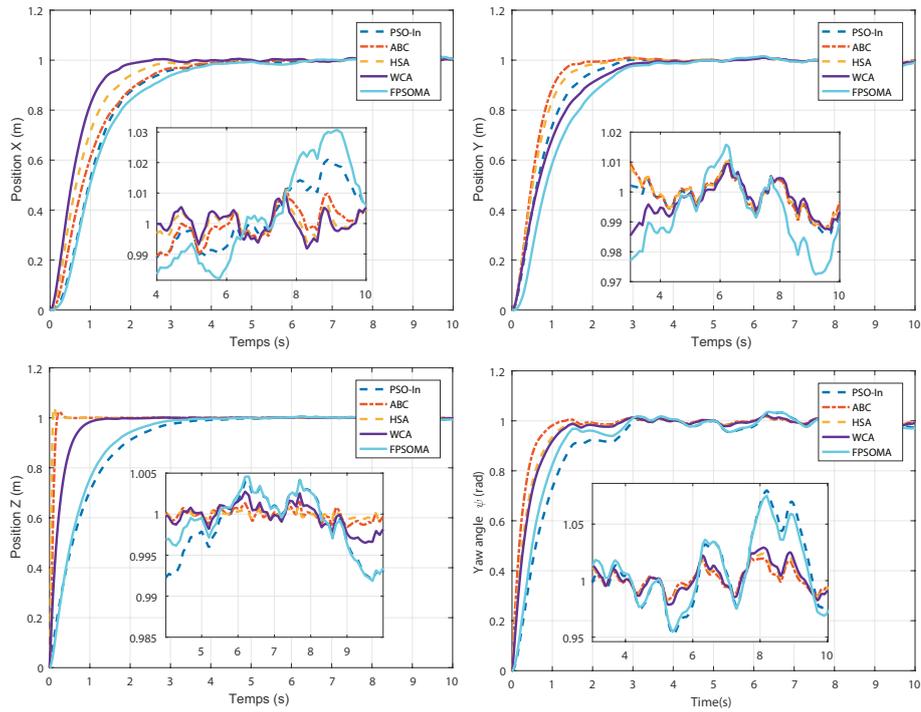


FIGURE 3.15 – Réponses indicielles comparées de la commande LQG avec réglage optimisé.

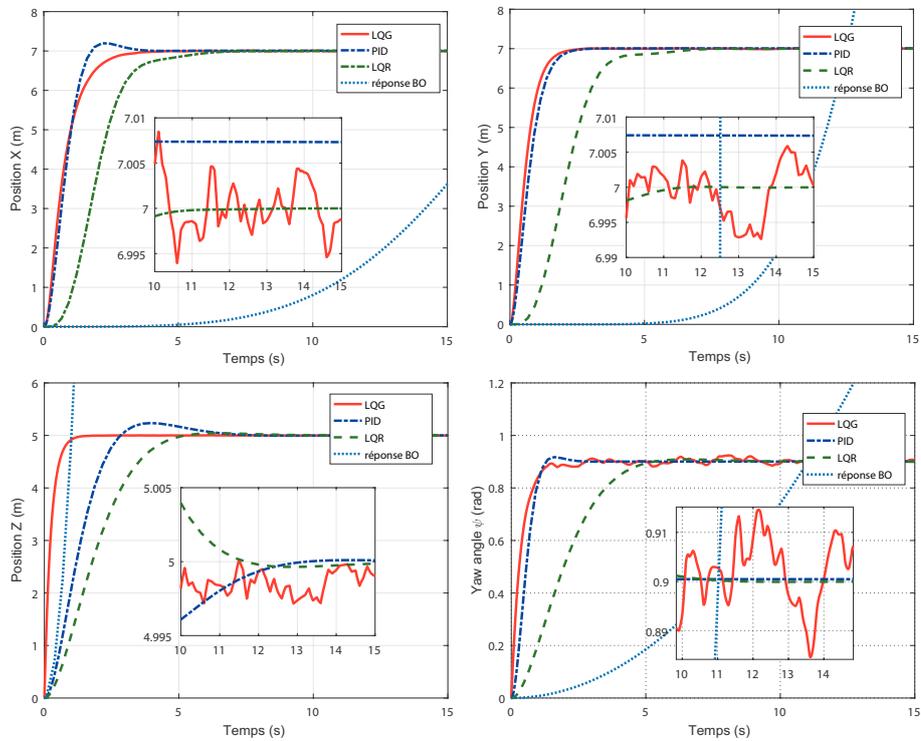


FIGURE 3.16 – Performances temporelles comparées des commandes LQG-PSO, PID et LQR.

Tableau 3.15 – Performances temporelles de la dynamique de lacet de $\psi(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO-In	3,5421	1,3076	7,9802	0,0250
ABC	0,9863	0,5448	1,9573	0,0044
HSA	2,9144	0,8157	1,0240	0,0068
WCA	3,9842	0,8659	2,9227	0,0087
FPSOMA	1,0751	1,0516	7,5058	0,0232

Tableau 3.16 – Performances temporelles de la dynamique de altitude de $z(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO-In	3,2161	1,8031	0,4181	0,0250
ABC	0,2772	0,1170	2,4165	0,0044
HSA	0,9583	0,5334	0,2743	0,0068
WCA	0,1495	0,0537	4,3461	0,0087
FPSOMA	2,6959	1,4846	0,4584	0,0232

Tableau 3.17 – Performances temporelles de la dynamique de position de $x(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO-In	8,9852	1,8145	2,1003	0,0250
ABC	3,5571	1,8470	1,1612	0,0044
HSA	2,5862	1,4961	1,0245	0,0068
WCA	0,9031	1,0630	0,9166	0,0087
FPSOMA	9,5501	2,1204	3,0657	0,0232

Tableau 3.18 – Performances temporelles de la dynamique de position de $y(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO-In	2,5068	1,3735	1,0704	0,0250
ABC	1,4569	0,8669	1,0247	0,0044
HSA	1,8898	1,0265	1,0120	0,0068
WCA	2,8672	1,7228	0,9371	0,0087
FPSOMA	9,7404	1,9019	1,5810	0,0232

3.5 Commande prédictive MPC à réglage par métaheuristiques

3.5.1 Formulation du problème

Le réglage des paramètres d'une loi de commande prédictive MPC pour les véhicules aériens n'est pas assez trivial même pour des experts automaticiens. Généralement, les relations liant les paramètres de réglage aux phénomènes physiques restent encore non établies. Par conséquent, ce type de réglage présente, jusqu'à aujourd'hui, une difficulté pour les applications industrielles de l'approche (Mamboundou and Langlois, 2012; Turki et al., 2018b; Derouiche et al., 2016; Tran et al., 2014; Garriga and Soroush, 2010).

La proposition des techniques de réglage de paramètres de la commande prédictive faisait l'objet de plusieurs travaux de recherche. Dans (Lino et al., 1993), les auteurs proposent une méthode de réglage basée sur l'analyse de la réponse fréquentielle du procédé, afin de garantir une robustesse de la stabilité. Une méthode de réglage par \mathcal{H}_∞ loop-shaping a été présentée dans (Rowe and Maciejowski, 2000). Une heuristique à base des règles d'expert établies est utilisée dans (Drogies and Geest, 1999). Les méthodes proposées par C. Rowe et Y. Lino nécessitent des procédures de calcul assez lourd, complexe et gourmand en temps de conception. La méthode présentée par S. Drogies, quant à elle, devrait être modifiée et adaptée particulièrement pour chaque problème traité. Ainsi, aucune des précédentes méthodes n'a abouti à une procédure systématique de réglage pouvant être adoptée en cas de manque d'expertise dans le domaine. Le problème de réglage des paramètres de la commande MPC, i.e. les horizons de prédiction $\mathbf{N}_p \in \mathbb{N}_+$ et de contrôle $\mathbf{N}_c \in \mathbb{N}_+$, et le facteur de pondération $\boldsymbol{\lambda} \in \mathbb{R}_+$ reste encore ouvert.

Une formulation de ce réglage sous la forme d'un problème d'optimisation, sous contraintes opérationnelles, est proposée pour la stabilisation du véhicule QTW comme suit :

$$\left\{ \begin{array}{l} \text{minimiser } f(\mathbf{x}, t) \\ \mathbf{x} = (\mathbf{N}_p, \mathbf{N}_c, \boldsymbol{\lambda}) \\ \text{sous contraintes :} \\ g_1(\mathbf{x}) = \mathbf{N}_c - \mathbf{N}_p \leq 0 \\ g_2(\mathbf{x}) = \boldsymbol{\lambda} \geq 0 \end{array} \right. \quad (3.37)$$

où $f(\cdot)$ est la fonction objectif du problème, $g_j(\cdot)$ les contraintes de type inégalité, \mathbf{N}_p et \mathbf{N}_c sont respectivement les horizons de prédiction et de contrôle, $\boldsymbol{\lambda}$ sont les facteurs de pondération et $\mathbf{x} = (\mathbf{N}_p, \mathbf{N}_c, \boldsymbol{\lambda}) \in \mathbb{R}_+^m$ est le vecteur de décision.

La figure 3.17 présente la nouvelle structure de commande MPC proposée pour le drone QTW avec un étage de réglage à base d'algorithmes métaheuristiques PSO. Ces derniers, présentant les outils de résolution d'un tel problème d'optimisation difficile sont déjà présentés au début de ce chapitre.

La fonction coût du problème (3.37) est choisie, pour chaque sous dynamique contrôlée comme le critère de performance "*Integral Time Absolute Error (ITAE)*" modifiée comme

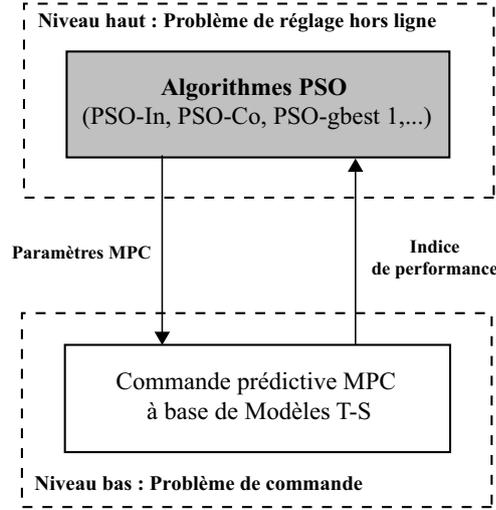


FIGURE 3.17 – Réglage systématique des paramètres de la commande MPC pour le drone QTW.

suit (Bouallègue et al., 2012, 2011) :

$$f_i(\mathbf{x}, t) = \int_0^{+\infty} t |e_i(\mathbf{x}, t)| + u_i^2(\mathbf{x}, t) dt \quad i \in \{z, \phi, \theta, \psi\} \quad (3.38)$$

où $u_i(t)$ est le signal de commande MPC et $e_i(\mathbf{x}, t)$ représente l'écart consigne-sortie du système à commander.

Tel qu'il est défini dans (3.37), le problème de réglage MPC est un problème d'optimisation multiobjectif. Une fonction d'agrégation est utilisée pour le traitement des objectifs multiples comme suit (BenKhoud and Bouallègue, 2018) :

$$f(\mathbf{x}, t) = w_z f_z(\mathbf{x}, t) + w_\phi f_\phi(\mathbf{x}, t) + w_\theta f_\theta(\mathbf{x}, t) + w_\psi f_\psi(\mathbf{x}, t) \quad (3.39)$$

où $w_i, i \in \{z, \phi, \theta, \psi\}$ sont des coefficients de pondération positifs sur les objectifs, vérifiant $w_z + w_\phi + w_\theta + w_\psi = 1$.

3.5.2 Résultats de simulation et discussion

Pour le réglage de cette approche de commande, nous utilisons les métaheuristiques PSO standard et quantique QPSO, QPSO-GM et QPSO-HPD. En adoptant une taille de population $N_{pop} = 40$ et un maximum d'itérations $N_{iter} = 100$, ces algorithmes aboutissent aux résultats d'optimisation des tableaux 3.19, 3.21, 3.22, 3.23 et 3.24 relatifs aux 30 essais indépendants sur le problème (3.37).

Des solutions réalisables du problème ont été trouvées à 75% des essais et au bout d'un temps de calcul raisonnable vis-à-vis de la complexité du système commandé et du problème d'optimisation de grand taille formulé. De plus, le fait que les algorithmes convergent le plus souvent vers des régions proches et similaires de l'espace de recherche, indique qu'une région intéressante à explorer et susceptible de contenir l'optimum du problème, Figure 3.18.

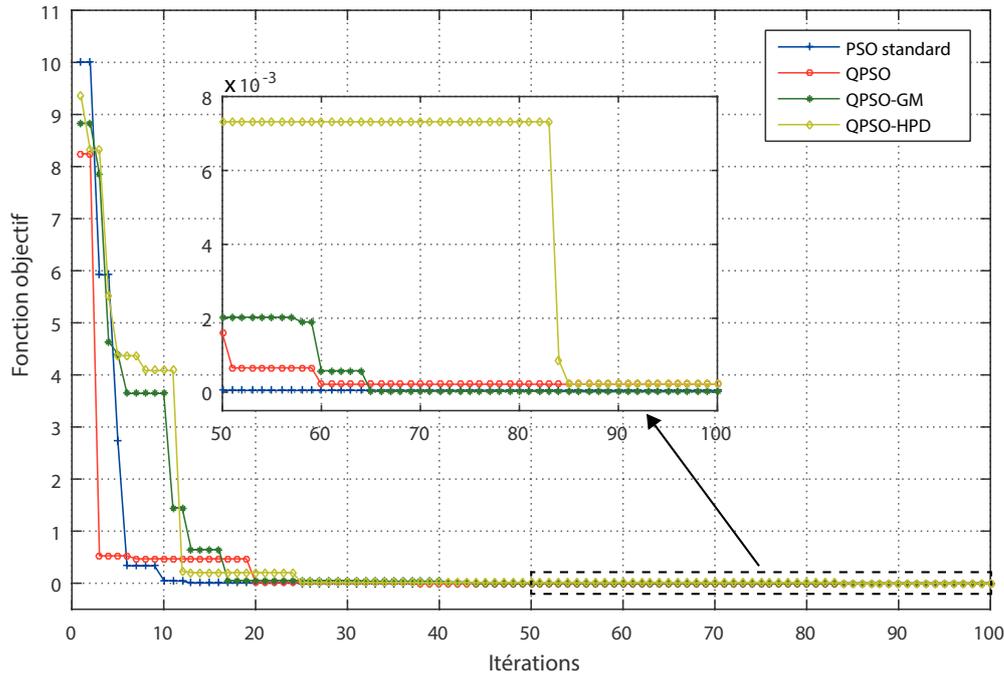


FIGURE 3.18 – Historiques de convergence des algorithmes pour le problème (3.37).

Tableau 3.19 – Résultats d'optimisation sur 30 essais du problème (3.37).

Algorithmes	Meilleur cas	Moyen cas	Pire cas	Ecart-type	Temps
PSO	4,396e-4	7,425e-2	3,703e-1	1,655e-1	1681
QPSO	2,141e-4	1,223e-2	4,330e-2	1,747e-1	847
QPSO-GM	9,807e-6	4,888e-2	1,488e-2	6,847e-3	1163
QPSO-HPD	2,146e-5	3,463e-3	1,138e-2	3,477e-3	858

Les paramètres de commande optimaux obtenus par chaque algorithme sont donnés dans le tableau 3.20. D'après ces expériences numériques, nous pouvons remarquer que tous les algorithmes métaheuristique donnent des résultats similaires et relativement proches. Les contraintes $g_1(\cdot)$ et $g_2(\cdot)$ du problème (3.37) sont vérifiées et respectées par tous les algorithmes.

Tous les algorithmes QPSO proposés produisent la commande MPC permettant de stabiliser la dynamique d'attitude et d'altitude du véhicule QTW, comme illustré aux figures 3.19 à 3.22. D'après les résultats statistiques ci-dessus, il est clair que les meilleurs résultats d'optimisation sont obtenus avec l'algorithme QPSO-GM. Souvent, les paramètres obtenus avec cet algorithme rendent le système plus efficace et plus rapide par rapport à ceux obtenus par d'autres algorithmes, notamment en dynamique d'altitude et de lacet.

Tableau 3.20 – Valeurs moyennes des paramètres optimaux de commande MPC sur 20 exécutions du problème (3.37)

Paramètres	PSO	QPSO	QPSO-GM	QPSO-HPD
N_p^1	25	35	20	36
N_p^2	21	33	10	12
N_p^3	22	10	15	20
N_p^4	22	26	17	18
N_c^1	15	22	3	33
N_c^2	20	10	6	10
N_c^3	10	6	3	8
N_c^4	16	10	10	13
λ^1	3,709	2,106	46,502	19,750
λ^2	9,948	15,400	20,763	7,9948
λ^3	2,800	4,525	37,502	2,949
λ^4	7,6821	0,2088	8,0742	12,103

Tableau 3.21 – Performances temporelles comparées pour la commande MPC de la dynamique de lacet $\phi(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO	0,2758	0,1826	0,3344	3,3309e-16
QPSO	0,2418	0,1538	0,3254	5,6647e-11
QPSO-GM	0,3456	0,2022	0,8361	5,6884e-07
QPSO-HPD	0,2116	0,1279	0,7354	1,0659e-08

Tableau 3.22 – Performances temporelles comparées de la commande MPC de la dynamique de tangage $\theta(t)$.

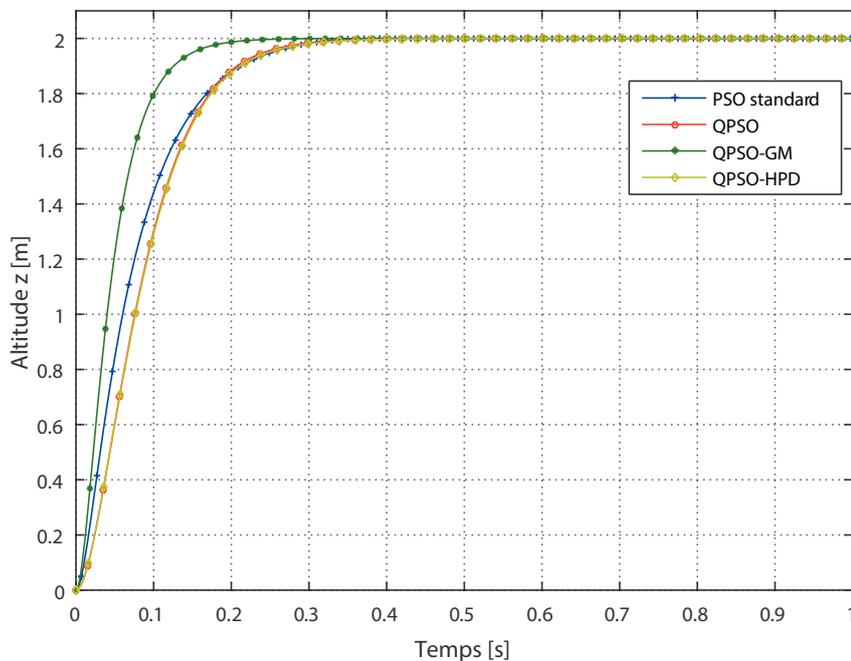
Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO	0,3993	0,1470	0,2506	4,9965e-16
QPSO	0,1573	0,2208	0,1480	3,8855e-16
QPSO-GM	0,2833	0,1211	0,2506	4,9960e-16
QPSO-HPD	0,4720	0,3053	0,2489	6,8348e-06

Tableau 3.23 – Performances temporelles comparées de la commande MPC de la dynamique de roulis $\psi(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO	0,4097	0,1426	0,1001	5,5517e-16
QPSO	0,7450	0,4281	0,0851	1,1108e-15
QPSO-GM	0,2012	0,1293	0,1883	1,9723e-12
QPSO-HPD	0,5385	0,2069	0,0805	0,0037e-05

Tableau 3.24 – Performances temporelles comparées de la commande MPC de la dynamique d'altitude $z(t)$.

Algorithmes	ST (s)	RT (s)	OV(%)	Ess
PSO	0,2625	0,1882	0,3527	4,2126e-12
QPSO	0,2528	0,1478	0,2320	1,6890e-08
QPSO-GM	0,1532	0,0889	0,0500	1,1102e-15
QPSO-HPD	0,2600	0,1496	0,6528	4,7791e-08

FIGURE 3.19 – Performances comparées de la commande MPC avec réglage optimisé : altitude $z(t)$.

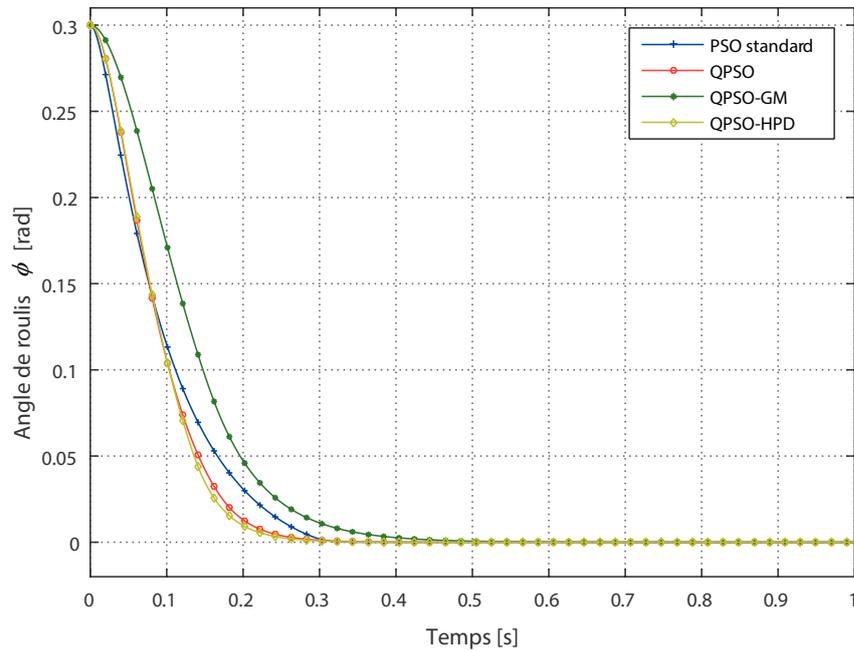


FIGURE 3.20 – Performances comparées de la commande MPC avec réglage optimisé : roulis $\phi(t)$.

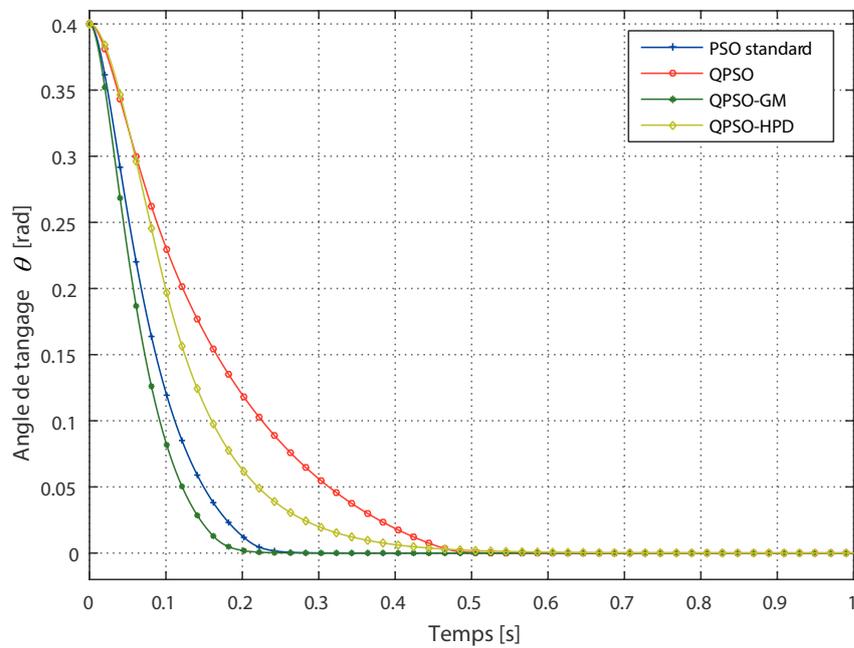


FIGURE 3.21 – Performances comparées de la commande MPC avec réglage optimisé : tangage $\theta(t)$.

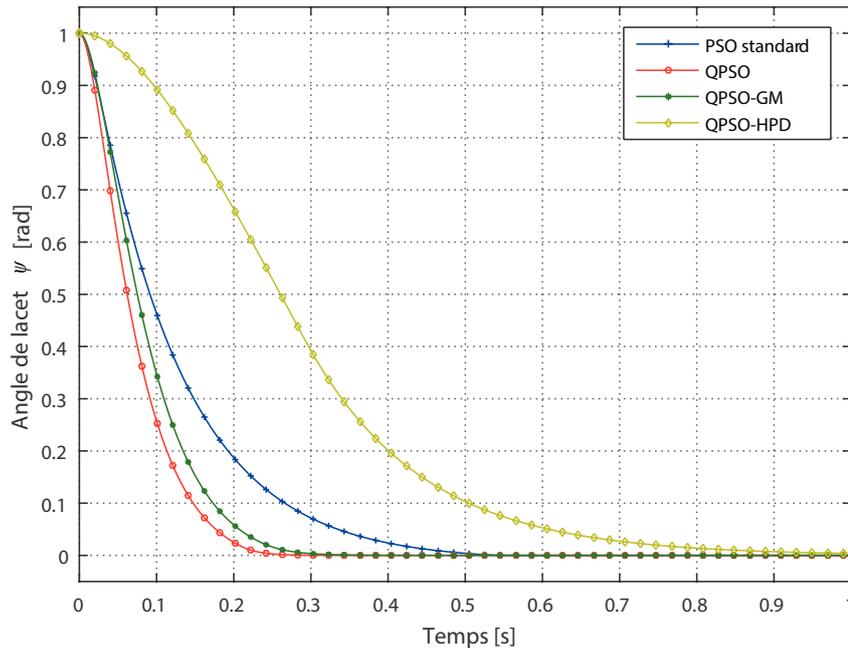


FIGURE 3.22 – Performances comparées de la commande MPC avec réglage optimisé : lacet $\psi(t)$.

3.6 Conclusion

Dans ce troisième chapitre, nous avons proposé une approche hors ligne d'optimisation et de réglage de structures de commande robuste à base d'algorithmes métaheuristiques, tout en améliorant la stabilité numérique du processus. Nous nous sommes plus particulièrement intéressés aux problèmes de réglage des commandes optimale quadratique gaussienne LQG, PID à supervision floue et prédictive MPC. Après avoir présenté les principaux algorithmes métaheuristiques proposés pour la résolution des problèmes de synthèse reformulés, nous avons proposé une approche de commande à base de la technique LQG, avec des matrices de pondération déterminées et optimisées par métaheuristiques PSO. Le problème de synthèse LQG hybride a été reformulé sous forme de problème d'optimisation sous contraintes liées aux performances temporelles de réglage. Pour le deuxième cas, en l'occurrence la commande PID à supervision floue, nous proposons une nouvelle méthode à base de la métaheuristique d'optimisation PSO pour son réglage systématique. Les gains du PID, devenus variables dans le temps, contribuent davantage à améliorer les performances de la structure de commande adaptative ainsi élaborée en termes de robustesse.

La dernière partie de ce chapitre a été consacrée à la méthodologie élaborée de réglage par métaheuristiques des paramètres de contrôle de la commande prédictive MPC, i.e. facteur de pondération et horizons de contrôle et de prédiction. Une application de cette stratégie a été envisagée, pour la stabilisation d'altitude et d'attitude de drone QTW étudié. Les résultats de mise en oeuvre ont montré des hautes performances, en régulation comme en poursuite, avec une systématisation remarquable de la procédure de synthèse.

Comme limitations de cette approche, le temps de calcul nécessaire afin de choisir un bon réglage de la MPC est important. En plus, la méthode proposée reste incapable de maintenir les performances désirées en cas des variations paramétriques ou d'éventuels défauts instantanés pouvant affecter le système à contrôler.

Comme solution à ces problèmes, nous avons proposé, dans le prochain chapitre, une nouvelle approche en ligne de synthèse à base d'algorithmes métaheuristiques issus de l'optimisation dynamique. ces derniers sont capables de détecter un changement éventuel dans la fonction objectif des problèmes d'optimisation reformulés.

Chapitre **4**

Réglage en ligne de la commande prédictive

Sommaire

4.1	Introduction	95
4.2	État de l'art sur l'optimisation continue dynamique	95
4.2.1	Principe	95
4.2.2	L'algorithme PSO multi-essaims à comportement quantique : mQPSO	98
4.2.3	Principaux jeux de test	102
4.3	Stratégie de réglage dynamique de la commande TSMPC	106
4.4	Résultats de simulation et discussion	107
4.4.1	Étude comparative	110
4.4.2	Analyse de robustesse de la méthode proposée	113
4.5	Conclusion	119

4.1 Introduction

Dans le chapitre précédent, une approche de réglage a été détaillée dans l'objectif d'améliorer la stabilité optimale du système en boucle fermée à base de métaheuristiques statiques. Ce chapitre est consacré à la présentation des résultats des travaux portant sur l'optimisation dynamique, en vue d'une réduction de la complexité et d'une systématisation de la synthèse de structures de commande MPC, par des métaheuristiques de recherche globale. Nous nous intéressons, plus particulièrement, au réglage des paramètres de la commande prédictive MPC. Par conséquent, nous allons proposer une approche en ligne de réglage adaptative des paramètres de la commande prédictive TSMPC pour les problèmes de stabilisation et de suivi des trajectoires pour le drone convertible QTW décrit sous forme d'une représentation T-S.

Dans un premier temps, un état de l'art sur les algorithmes métaheuristiques désignées aux problèmes dynamiques est présenté. Ensuite, nous proposons un algorithme PSO multi-essais à comportement Quantique (mQPSO) pour le réglage en ligne de la commande TSMPC reformulé comme un problème d'optimisation non linéaire et contraint. Finalement, la validation de l'approche adaptative de réglage des paramètres effectives de la technique TSMPC est présentée moyennant des simulations numériques. Diverses lois de commande MPC présentées dans la littérature sont considérées comme supports de comparaison pour les problèmes de commande envisagés.

4.2 État de l'art sur l'optimisation continue dynamique

4.2.1 Principe

Au cours des deux dernières décennies, un intérêt croissant a été porté aux problèmes d'optimisation dynamique (DOP) en raison de leur importance dans des applications réelles d'ingénierie et de recherche (S.Yang et al., 2007; Morrison, 2004; Yang and Yao, 2013). Une variété d'approches de calcul évolutif a été développée pour traiter les DOPs. Des nouvelles techniques et des jeux de test ont été construits et comparés sous différentes mesures de performance pour représenter un large éventail de problèmes réels. Un problème d'optimisation dynamique est caractérisé par une fonction objectif variable en fonction du temps. C'est le cas, par exemple, des problèmes classiques de tournées de véhicules et de routage dans les réseaux de télécommunication, etc (Branke, 2002; Alba et al., 2013; Nguyen et al., 2012).

Mathématiquement, un problème d'optimisation dynamique est régi par l'équation (4.1) où $f(\cdot)$ est la fonction objectif du problème à minimiser, $g_j(\cdot)$ et $h_k(\cdot)$ sont les contraintes non linéaires, voire non analytiques en les paramètres, respectivement de type inégalité et égalité, \mathbf{x} est le vecteur des variables de décision, inconnu du problème :

$$\begin{aligned} \text{Minimiser} \quad & f(\mathbf{x}, t) \\ \text{s.c.} \quad & h_j(\mathbf{x}, t) = 0 \quad \text{pour } j = 1, 2, \dots, u \\ & g_k(\mathbf{x}, t) \leq 0 \quad \text{pour } k = 1, 2, \dots, v \end{aligned} \tag{4.1}$$

La fonction f est déterministe, mais dépend du temps t , ainsi la position de l'optimum change au cours de temps. Dans la littérature, l'approche la plus simple pour résoudre un problème dynamique et suivre en permanence son optimum est de redémarrer un algorithme d'optimisation statique à chaque fois qu'un changement se produit dans la fonction objectif et de formuler ainsi un nouveau problème d'optimisation (Branke, 2002; Alba et al., 2013). Néanmoins, cette approche nécessite l'identification de l'instant du changement d'environnement et un temps suffisant entre chaque changement pour trouver une solution acceptable, ce qui n'est pas toujours le cas.

La figure 4.1 présente un exemple simple de la fonction dynamique *Easom* à trois différents instants où la position de l'optimum change au cours du temps. Le changement dynamique pourrait affecter ainsi d'autres paramètres comme les amplitudes des pics, la phase, etc.

Des algorithmes spécialement conçus pour la résolution des DOPs ont alors été proposés dans la littérature (Blackwell and Branke, 2004; Alba et al., 2013; Nguyen et al., 2012). Ils devraient pouvoir :

- trouver l'optimum au plus rapidement possible ;
- détecter un éventuel changement dans la fonction objectif ;
- suivre en permanence l'optimum ;
- prendre en compte, lors du processus d'optimisation, des informations d'environnements précédents pour accélérer l'optimisation après un changement.

Récemment, la problématique de l'optimisation des fonctions dynamiques pose de nouveaux défis. Par conséquent, résoudre ce type de problèmes est une tâche importante. Les algorithmes métaheuristiques constituent un bon choix pour la résolution des DOPs vu que ces algorithmes s'inspirent de nature des processus d'évolution et d'intelligence soumis aux changements d'environnements. Par contre, des nouvelles exigences rendent difficile, voire impossible, aux métaheuristiques traditionnelles d'optimisation statique de résoudre de manière satisfaisante les problèmes dynamique DOP, principalement pour les deux raisons suivantes (Branke, 2002; Alba et al., 2013) :

- mémoire périmée : après un changement dans l'environnement, les informations stockées dans la mémoire de chaque individu au sens de la meilleure solution locale et de la position correspondante, peuvent ne plus être valides ou bien « périmées » et donc orienter la recherche de façon erronée ;
- perte de la diversité : après un changement dans l'environnement, l'optimum global change probablement sa position dans le domaine de recherche. Ainsi, la réoptimisation dans ce cas semble être efficace. Cependant, si le décalage de la nouvelle position d'optimum est significativement éloigné de la précédente, l'algorithme doit assurer une diversification suffisante pour pouvoir le trouver. Par contre, les faibles vitesses des particules empêchent le suivi et ça va être difficile pour un algorithme statique d'écarter rapidement et explorer des nouvelles zones dans le domaine de recherche.

Dans (Branke, 2002; Jin and Branke, 2005), J. Branke a classifié comme suit les techniques dynamiques issues de la littérature pour la résolution des problèmes DOPs :

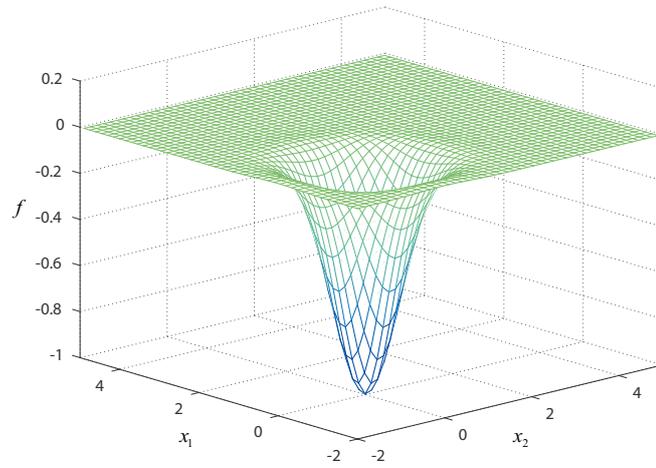
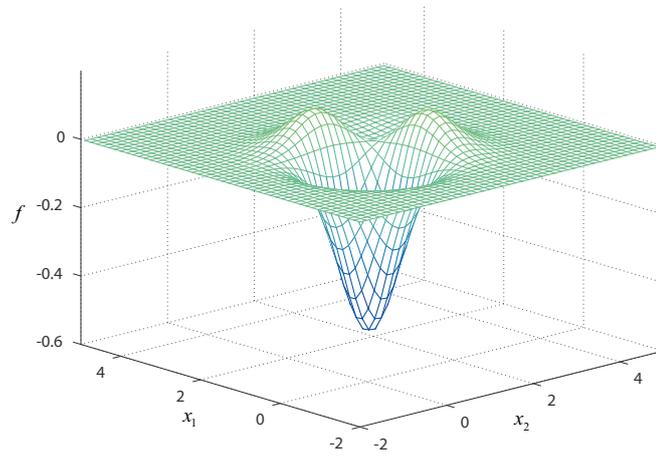
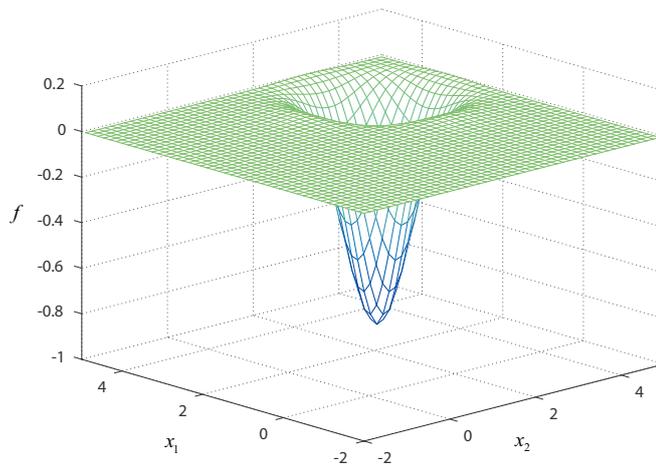
(a) temps t_1 .(b) temps $t_2 > t_1$.(c) temps $t_3 > t_2$.

FIGURE 4.1 – Illustration du paysage de la fonction dynamique Easom.

- **les méthodes réactives** : dès qu'un changement dans l'environnement est détecté, des traitements explicites sont pris pour augmenter la diversité des solutions et par conséquent faciliter la recherche de nouvel optimum. Les représentants typiques de cette approche sont l'hypermutation et la recherche locale ;
- **les méthodes de conservation de la diversification** : la convergence est maintenue au cours du temps, et l'on espère qu'une population largement répartie dans l'espace de recherche peut s'adapter plus efficacement aux changements ;
- **les méthodes avec mémoire** : l'algorithme dispose d'une mémoire qui permet de rappeler les informations utiles sur le passé des générations et de la fonction objectif, ce qui semble particulièrement utile lorsque l'optimum revient une autre fois aux emplacements précédents. Ces méthodes sont efficaces seulement dans le cas périodique (Luo et al., 2016) ;
- **les méthodes visant à prédire les changements** : récemment, certaines études ont été réalisées pour exploiter la prévisibilité des environnements dynamiques et essayer d'apprendre les changements survenant sur la fonction objectif et de les prédire à l'avenir en se basant sur ces schémas et à partir de l'expérience de recherche précédente ;
- **les méthodes à plusieurs populations** : la population est divisée en plusieurs sous-populations réparties sur différents optima locaux. Lorsqu'un changement se produit, l'une de ces solutions pourrait devenir l'optimum global, ou au moins de s'y rapprocher par la suite. La probabilité de retrouver et de suivre un tel optimum global est renforcée (Blackwell and Branke, 2006; Parrott and Li, 2004).

4.2.2 L'algorithme PSO multi-essaims à comportement quantique : mQPSO

4.2.2.1 Motivations et nouvelles équations de mouvement

Comme vu au chapitre précédent, le PSO est un algorithme à base de population qui a été appliqué avec succès pour résoudre une grande variété de problèmes statiques (Alba et al., 2013). Son utilisation pour les problèmes d'optimisation dynamique nécessite de modifier sa formulation canonique afin d'obtenir des bonnes performances. Dans (Janson and Middendorf, 2004), Janson et Middendorf ont proposé une organisation hiérarchique des essaims (PH-PSO) où les particules sont disposées dans une hiérarchie dynamique. Cette dernière est divisée en plusieurs sous-essaims pendant un nombre limité de générations après qu'un changement soit produit. Parrot et Li (Parrott and Li, 2004) ont adapté les méthodes de spéciation introduites dans les algorithmes génétiques pour le cas du PSO pour encourager le suivi simultané de plusieurs pics en empêchant le surpeuplement aux pics. Une multitude des variantes proposées dans la littérature est basée généralement sur des caractéristiques bien définies telles que les multi-essaims, la diversité ou/et l'exclusion intra-essaim (Hu et al., 2011; Ma et al., 2019; D.Yazdani et al., 2013; Kordestani et al., 2019; Ozsoydan and Baykasoglu, 2019). Ces variantes ont été toujours des améliorations de l'algorithme initialement conçu par Blackwell et Branke (Branke et al., 2000; Blackwell and Branke, 2006, 2004).

Plusieurs des variantes précédentes ont été testées avec le Scénario 2 du banc de test Moving Peaks Benchmark (MPB) (Branke, 2002; Alba et al., 2013). La variante la plus performante pour ce jeux de test fut une approche multi-essaims à comportement quantique (mQPSO) basée sur des concepts de la mécanique quantique ainsi que des travaux d'analyse dynamique des trajectoires des particules dans l'essaim (Branke et al., 2000; Blackwell and Branke, 2006). Les expériences numériques et les analyses menées ont montré que l'algorithme mQPSO offre des meilleures performances lors de la résolution d'une large gamme des problèmes d'optimisation dynamique. La figure 4.2 explique clairement le concept proposé de l'algorithme QPSO à essaims multiples. Chaque essaim n dans le mQPSO est formé de deux types des particules :

- **particules de trajectoire** : aussi connues comme classiques ou neutres, ce sont les particules utilisées par l'algorithme canonique PSO dont les positions sont mises à jour, à chaque itération k , en suivant les équations de mouvement suivantes :

$$\mathbf{v}_i^n(k+1) = \chi [\mathbf{v}_i^n(k) + c_1 r_1 (\mathbf{p}_g^n(k) - \mathbf{x}_i^n(k)) + c_2 r_2 (\mathbf{p}_i^n(k) - \mathbf{x}_i^n(k))] \quad (4.2)$$

$$\mathbf{x}_i^n(k+1) = \mathbf{x}_i^n(k) + \mathbf{v}_i^n(k+1) \quad (4.3)$$

où \mathbf{x} et \mathbf{v} sont respectivement les vecteurs de position et de vitesse, \mathbf{p}_i^n est la meilleure position obtenue de la i -ème particule et \mathbf{p}_g^n représente la meilleure position trouvée par l'ensemble des essaims à chaque itération, c_1 et c_2 sont des paramètres positifs qui caractérisent la dynamique des particules, r_1 et r_2 sont deux nombres aléatoires tirés uniformément de l'intervalle $[0, 1]$ et χ est le facteur de construction ;

- **particules chargées (électrons)** : ces particules ont été nouvellement introduites dans l'algorithme mQPSO. Elles ne suivent pas une trajectoire classique du PSO, mais plutôt elles sont réinitialisées aléatoirement dans une boule (sphère pleine) centrée sur la meilleure position trouvée par l'essaim. Ces particules chargées visent à atteindre un niveau plus élevé de diversité en se déplaçant de façon aléatoire dans une hypersphère de rayon r centrée sur \mathbf{p}_g^n . Ce mouvement aléatoire est effectué selon une distribution de probabilités sur l'hypersphère, en l'occurrence une distribution uniforme. D'après les développements menés de l'approche, l'équation fondamentale de l'algorithme QPSO pour la mesure de la nouvelle position de la i -ème particule quantique est définie comme suit (Blackwell and Branke, 2006) :

$$\mathbf{x}_i^n(k+1) = \text{rand}_{\text{hypersphere}}(\mathbf{p}_g^n, r) \quad (4.4)$$

L'idée fondamentale de l'approche proposée est de diviser l'essaim en un certain nombre de sous-essaims indépendants dans le but de positionner chacun d'eux sur des sommets différents (pics) prometteurs du paysage. Cependant, il est peu probable que cette méthode ne soit pas efficace puisque les essaims n'interagissent pas entre eux. Ainsi, un mécanisme d'*exclusion* a été appliqué pour garantir l'interaction locale et l'échange d'informations entre les essaims en collision de telle sorte que ces derniers soient conscients de l'état de leur convergence.

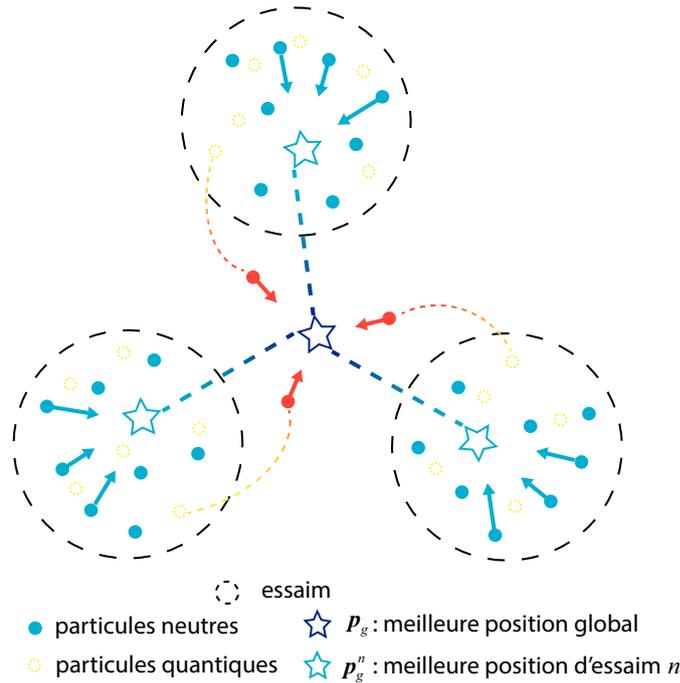


FIGURE 4.2 – Illustration du mécanisme de partage des informations entre les essaims du QPSO.

4.2.2.2 Mécanisme d'exclusion

Toutefois, il est possible que deux pics se rapprochent l'un de l'autre et que l'essaim quitte son pic couvert et se déplace vers un autre. Cela signifie également qu'un pic sera couvert par deux essaims au même temps. Ce scénario n'est pas souhaitable puisque la motivation derrière une approche multi-essaims est de positionner des différents essaims sur différents sommets (zone de recherche). Pour surmonter ce problème, la distance euclidienne d_{eucl} entre les meilleurs positions P_g^n pour $n = 1, 2, \dots, N$ de chaque sous-essaims doit être calculée à chaque itération. Par conséquent, deux essaims avec une distance inférieure à r_{excl} sont placés sur le même pic. Ce dernier est défini, en supposant que tous les pics sont répartis uniformément dans l'espace de recherche. Dans (Blackwell and Branke, 2006), une technique est proposée pour calculer la valeur optimale de la métrique r_{excl} moyennant la formule suivante :

$$r_{excl} = 0.5 (X/p^{1/d}) \quad (4.5)$$

avec p le nombre des pics, d la dimension du problème à optimiser et X la longueur de l'espace de recherche pour chaque dimension.

Un pseudo-code pour le mécanisme d'exclusion de l'algorithme QPSO peut être retenu depuis l'algorithme 3 suivant :

4.2.2.3 Test de changement de l'environnement

Comme mentionné dans la section (4.2.1), après un changement d'environnement, les algorithmes d'optimisation rencontrent deux problèmes majeurs de mémoire invalide et

Algorithme 3 : Exclusion()

```

1 pour Chaque paire d'essaims  $n$  et  $m$  faire
2   si la distance euclidienne  $d_{eucl}$  entre  $\mathbf{p}_g^n$  et  $\mathbf{p}_g^m$  est inférieure à  $r_{excl}$  alors
3     si  $f(\mathbf{p}_i^n) \leq f(\mathbf{p}_i^m)$  alors
4       Ré-initialisation de l'essaim  $n$  : les positions et les vitesses  $\mathbf{v}_i^n, \mathbf{p}_i^n = \mathbf{x}_i^n$ ;
5       Evaluation de la fonction objectif  $f(\mathbf{p}_i^n)$ ;
6        $\mathbf{p}_g^n := \arg \max \{\mathbf{p}_i^n\}$ ;
7     sinon
8       Ré-initialisation de l'essaim  $m$  : les positions et les vitesses
9        $\mathbf{v}_i^m, \mathbf{p}_i^m = \mathbf{x}_i^m$ ;
9       Evaluation de la fonction objectif  $f(\mathbf{p}_i^m)$ ;
10       $\mathbf{p}_g^m := \arg \max \{\mathbf{p}_i^m\}$ ;

```

de réduction remarquable de la diversité des essaims. Comme solution, un mécanisme spécifique a été utilisé dans (Blackwell and Branke, 2006).

A chaque itération k et pour chaque essaim n , la valeur de fitness de la meilleure position globale $f(\mathbf{p}_g^n)$ est évaluée et comparée à sa valeur précédente. Si les résultats de deux exécutions successives sont identiques, cela signifie que l'environnement n'a pas changé. D'autre part, leur différence illustre un changement dans l'environnement et les vitesses et les positions de toutes les particules doivent être évaluées conformément aux équations de mouvement (4.2)-(4.4).

Finalement, un pseudo-code de l'algorithme QPSO pour l'algorithme des DOPs peut être retenu de l'algorithme 4.

Algorithme 4 : Test de changement de l'environnement()

```

1 Evaluation de la fonction objectif  $f(\mathbf{p}_g^n)$ ;
2 si la nouvelle valeur est différente de celle de la dernière itération alors
3   pour Chaque particule  $i$  d'essaim  $n$  faire
4     Mettre à jour  $\mathbf{x}_i^n$  en utilisant les équations (4.2) et (4.4) selon le type de la
4     particule;
5     Mettre à jour  $\mathbf{v}_i^n$  en utilisant l'équation (4.3);
6      $\mathbf{p}_i^n = \mathbf{x}_i^n$ ;
7     Evaluation de la fonction objectif  $f(\mathbf{p}_i^n)$ ;
8    $\mathbf{p}_g^n := \arg \max \{\mathbf{p}_i^n\}$ ;

```

4.2.2.4 Pseudo-code de l'algorithme mQPSO

Pour pouvoir implémenter l'algorithme mQPSO en un langage évolué (C, C++, MATLAB, etc.) et résoudre les problèmes d'optimisation dynamique, reformulés plus tard, nous retenons le pseudo-code de l'algorithme 5.

Algorithme 5 : mQPSO

```

1 Définir les paramètres de contrôle de l'algorithme :  $N_{pop}$ ,  $N_{swarm}$  et  $FES$ ;
2 pour Chaque particule  $i$  d'essaim  $n$  faire
3   Initialisation aléatoire des positions et des vitesses  $\mathbf{v}_i^n, \mathbf{p}_i^n = \mathbf{x}_i^n$ ;
4   Evaluation de la fonction objectif  $f(\mathbf{p}_i^n)$ ;
5 pour Chaque essaim  $n$  faire
6    $\mathbf{p}_g^n := \arg \max \{\mathbf{p}_i^n\}$ ;
7 répéter
8    $Exclusion()$ ;
9   pour Chaque essaim  $n$  faire
10     $Test\ de\ changement\ de\ l'environnement()$ ;
11    pour Chaque particule  $i$  d'essaim  $n$  faire
12     Mise à jour des vitesses et des positions conformément aux équations
13     de mouvement (1.1) et (1.2);
14     Evaluation de la fonction objectif  $f(\mathbf{x}_i^n)$ ;
15     si  $f(\mathbf{x}_i^n) > f(\mathbf{p}_i^n)$  alors
16     |  $\mathbf{p}_i^n = \mathbf{x}_i^n$ 
17     si  $f(\mathbf{x}_i^n) > f(\mathbf{p}_g^n)$  alors
18     |  $\mathbf{p}_g^n = \mathbf{x}_i^n$ 
19  $\mathbf{p}_g := \arg \max \{\mathbf{p}_g^n\}$ ;
jusqu'à nombre d'évaluations de fonctions effectuées  $> FES$ ;

```

4.2.3 Principaux jeux de test

4.2.3.1 Moving Peaks Benchmark

Le *Moving Peaks Benchmark (MPB)* est un jeu de test de fonction d'optimisation dynamique proposé par Branke (Branke, 2002). Il est largement utilisé dans la littérature pour évaluer les divers algorithmes d'optimisation dans un environnement dynamique. Le problème d'optimisation MPB est composé d'un nombre fini des pics dans un espace multi-dimensionnel où la hauteur, la largeur et la position de chaque pic sont légèrement modifiées chaque fois qu'un changement se produit dans l'environnement au cours du temps, figure 4.3.

Le MPB se compose de m pics dans un espace de n dimensions, il s'agit d'un problème de maximisation d'une fonction objective f qui peut être formulée comme suit :

$$f(\mathbf{x}, t) = \max_{i=1, \dots, m} (h_i(t) - w_i(t) \|\mathbf{x} - \mathbf{p}_i(t)\|) \quad (4.6)$$

où f est la fonction définissant la forme des pics, $h_i(t)$, $w_i(t)$ et $\mathbf{p}_i(t)$ sont respectivement la hauteur, la largeur et la position du $i^{\text{ème}}$ pic à l'instant t .

Les changements dans la fonction objectif ont lieu toutes les Δe évaluations où la hauteur et la largeur de chaque pic sont modifiées en ajoutant une variable aléatoire gaussienne (Branke, 2002). Les nouveaux déplacements de chaque pic sont évalués et mis à

jour conformément aux équations (4.7), (4.8) et (4.9) où \mathbf{v}_i est un vecteur de vitesse avec une longueur s fixe dans une direction λ décorrélée (pour $\lambda = 0$) ou dans une direction corrélée avec la direction précédente (pour $\lambda > 0$). Les modifications apportées à chaque pic sont décrites par les équations suivantes :

$$h_i(t+1) = h_i(t) + h_{severity} \cdot N(0,1) \quad (4.7)$$

$$w_i(t+1) = w_i(t) + w_{severity} \cdot N(0,1) \quad (4.8)$$

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t+1) \quad (4.9)$$

Les modifications apportées aux paramètres de largeur et de hauteur dépendent des gravités données respectivement par $w_{severity}$ et $h_{severity}$ avec $N(0,1)$ est la fonction de densité de la loi normale.

La vitesse de déplacement d'un pic est une combinaison linéaire d'un vecteur aléatoire \mathbf{r} et de sa vitesse à l'instant précédent $\mathbf{v}_i(t)$. Elle est définie comme suit :

$$\mathbf{v}_i(t+1) = \frac{s}{|\mathbf{r} + \mathbf{v}_i(t)|} ((1-\lambda)\mathbf{r} + \lambda\mathbf{v}_i(t)) \quad (4.10)$$

où le vecteur aléatoire \mathbf{r} est créé en générant une séquence de nombres aléatoires uniformément distribués pour chaque dimension et le paramètre s indique la distance parcourue par un seul pic lorsqu'un changement d'environnement se produit.

4.2.3.2 Generalized Dynamic Benchmark Generator

Le *Generalized Dynamic Benchmark Generator (GDBG)* est un jeu de test des fonctions d'optimisation dynamique introduit dans (Changhe and Shengxiang, 2008; Changhe et al., 2011). Ce benchmark a été proposé lors de la compétition du congrès *IEEE CEC'09*, afin de tester et évaluer les performances des algorithmes métaheuristiques dédiés à l'optimisation dynamique.

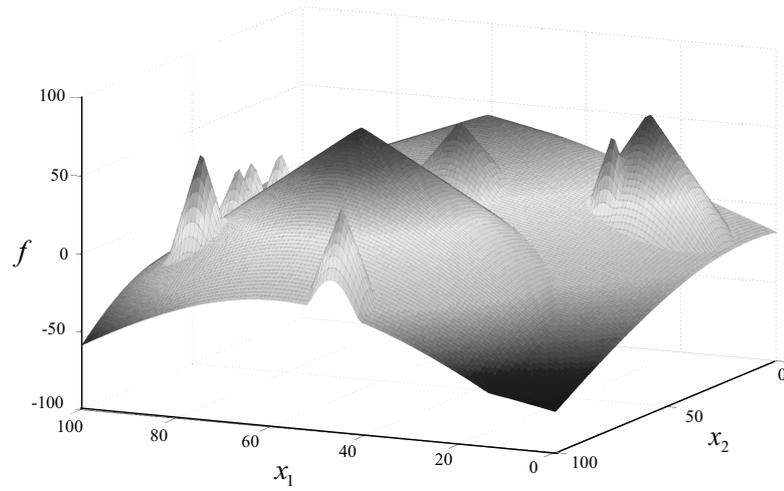
Dans le problème GDBG, des fonctions statiques (illustrées dans le tableau 4.1) ont été alternées, composées et combinées pour former six problèmes avec différents degrés de difficulté et représenter un large éventail des problèmes réels :

- $F1$: fonction à pics tournée (à 10 et 50 pics) ;
- $F2$: composition de la fonction Sphère ;
- $F3$: composition de la fonction Rastrigin ;
- $F4$: composition de la fonction Griewank ;
- $F5$: composition de la fonction Ackley ;
- $F6$: composition hybride de fonctions.

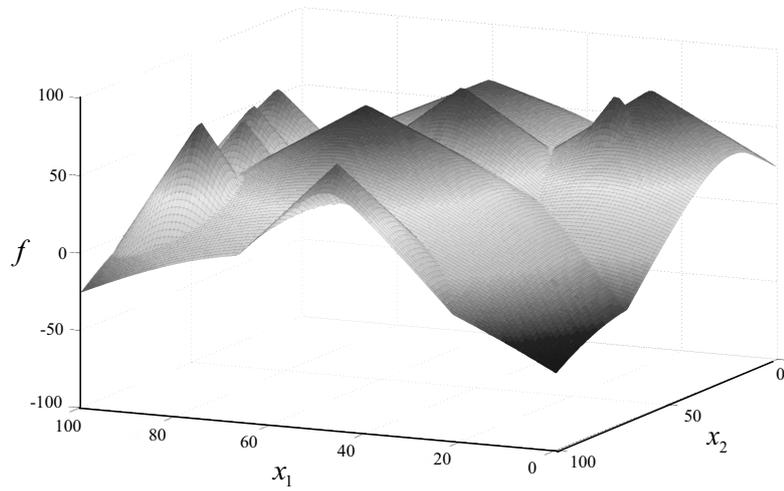
Une fonction composée peut être décrite sous la forme générale d'un DOP comme suit :

$$F = f(x, \phi, t) \quad (4.11)$$

où F désigne le problème d'optimisation, $f(\cdot)$ est la fonction objectif du problème à minimiser (ou à maximiser), x est une solution réalisable dans l'ensemble de solutions \mathbf{x} ,



(a) avant changement.



(b) après changement.

FIGURE 4.3 – Illustration du paysage de la fonction de test MPB.

t est le temps d'évolution et ϕ est un paramètre de contrôle du système, permettant de déterminer la répartition de la solution dans le paysage de la fonction objectif à chaque instant. Ceci permet de tester et de comparer l'adaptabilité des algorithmes dans différents types de problèmes dynamiques. Ce paramètre de contrôle peut être défini comme suit :

$$\phi(t+1) = \phi(t) \oplus \Delta\phi \quad (4.12)$$

où $\Delta\phi$ est la variation du paramètre ϕ et la $a \oplus b$ est la somme directe de a et b .

Alors, nous pouvons obtenir le nouvel environnement à l'instant $(t+1)$ comme suit :

$$F = f(x, \phi(t) \oplus \Delta\phi, t) \quad (4.13)$$

Dans le problème GDBG, il existe sept stratégies pour la modification du paramètre de contrôle $\phi(t)$, définies comme suit ([Changhe and Shengxiang, 2008](#); [Changhe et al., 2011](#)) :

- $T1$: un déplacement faible
- $T2$: un déplacement important
- $T3$: un déplacement aléatoire selon une distribution gaussienne
- $T4$: un changement chaotique
- $T5$: un changement récurrent
- $T6$: un changement récurrent bruité
- $T7$: un changement dans la dimension de l'espace de recherche

Tableau 4.1 – Fonctions statiques utilisées pour générer les fonctions de test du GDBG.

Nom	Fonction	Intervalle
Sphère	$f(\mathbf{x}) = \sum_{i=1}^d x_i^2$	$[-100; 100]^d$
Rastrigin	$f(\mathbf{x}) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5; 5]^d$
Weierstrass	$f(\mathbf{x}) = \sum_{i=1}^d \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - d \sum_{k=0}^{k_{\max}} [a^k \cos(\pi b^k)]$	$[-0.5; 0.5]^d$
Griewank	$f(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^d (x_i)^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-100; 100]^d$
Ackley	$f(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + e$	$[-32; 32]^d$

Par la combinaison de ces six fonctions de test ($F1$ étant utilisée avec 10 et 50 pics) avec les sept mécanismes de modification des paramètres de contrôle, nous obtenons un ensemble de quarante neuf scénarios dynamiques. Les moyennes des différentes valeurs, Équations (4.14)-(4.16), ainsi que l'écart-type de l'erreur absolue, Équation (4.17), sont calculées pour chaque scénarios comme suit :

$$Avg_{best} = \sum_{i=1}^{runs} \min_{j=1}^{N_{changes}} \frac{E_{ij}}{runs} \quad (4.14)$$

$$Avg_{mean} = \sum_{i=1}^{runs} \sum_{j=1}^{N_{changes}} \frac{E_{ij}}{runs \times N_{changes}} \quad (4.15)$$

$$Avg_{worst} = \sum_{i=1}^{runs} \max_{j=1}^{N_{changes}} \frac{E_{ij}}{runs} \quad (4.16)$$

$$STD = \sqrt{\frac{\sum_{i=1}^{runs} \sum_{j=1}^{N_{changes}} (E_{ij} - Avg_{mean})^2}{runs \times N_{changes}}} \quad (4.17)$$

où $E_{ij} = |f_j^* - \tilde{f}_j^*|$, f_j^* est la valeur de l'optimum global au cours du $j^{\text{ème}}$ paysage, \tilde{f}_j^* est la valeur de la meilleure solution trouvée au cours de la $i^{\text{ème}}$ exécution de l'algorithme testé, $runs$ est le nombre d'exécutions de l'algorithme testé et $N_{changes}$ est le nombre de changements.

4.3 Stratégie de réglage dynamique de la commande TSMPC

La stratégie de commande proposée pour le drone QTW est donnée par le schéma de principe de la figure 4.4, avec un étage de réglage en ligne des paramètres TSMPC à base d'algorithmes métaheuristiques QPSO. Ces derniers, présentant les outils de résolution d'un tel problème d'optimisation difficile, seront définis dans ce qui suit. Deux boucles, interne pour le contrôle de l'attitude et externe pour la position et l'altitude, sont implémentées. Les trajectoires de référence pour le tangage et le roulis sont générées moyennant le contrôleur des positions longitudinale x et transversale y .

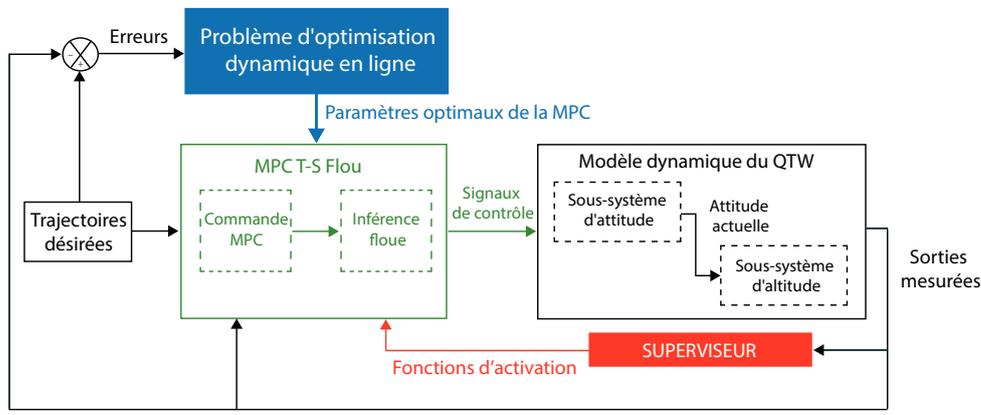


FIGURE 4.4 – Structure de contrôle proposée à base de l'optimisation dynamique des paramètres TSMPC en ligne.

La validation des résultats de mise en œuvre considère le modèle non linéaire du drone QTW modélisé sous forme T-S comme présenté dans la section 1.6.6. La stratégie de contrôle sous réglage en ligne des paramètres effectifs de commande a été introduite dans le chapitre 2. La loi de commande prédictive à appliquer au système est définie par :

$$u(k) = \sum_{i=1}^{N_{lti}} \mu_i(\theta(t)) u_i(\theta(t)) \quad (4.18)$$

Comme le modèle dynamique du drone QTW possède deux variables de prémisse θ et ϕ , le nombre des sous-modèles linéaires T-S est $N_{lti} = 2^2 = 4$. Ainsi, on définit :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^4 \mu_i(\theta) (A_i x(t) + B_i u(t)), \\ y(t) = \sum_{i=1}^4 \mu_i(\theta) (C_i x(t)). \end{cases} \quad (4.19)$$

Le modèle du véhicule QTW est soumis à des contraintes opérationnelles prises comme celles imposées dans (Hamouda et al., 2016) :

1. Contraintes sur les entrées :

$$\begin{cases} -0.5 < u_1 < 0.5 \\ -0.5 < u_2 < 0.5 \\ -0.5 < u_3 < 0.5 \end{cases}$$

2. Contraintes sur les sorties :

$$\begin{cases} -\frac{\pi}{2} < \phi < \frac{\pi}{2} \\ -\frac{\pi}{2} < \theta < \frac{\pi}{2} \\ -\frac{\pi}{2} < \psi < \frac{\pi}{2} \end{cases}$$

L'objectif de proposer une telle stratégie de réglage est de montrer l'intérêt de la mise à jour des paramètres de l'approche MPC et son effet sur les performances en rapidité et en précision de la réponse donnée en boucle fermée. La sélection de ces paramètres effectifs de contrôle peut devenir de plus en plus une tâche difficile et coûteuse en temps de conception pour des systèmes plus complexes et de grande dimension. En effet, les métaheuristiques proposées ont toujours montré leurs supériorités remarquables pour la résolution des problèmes de grande taille.

Pour cela, nous proposons la formulations du problème de réglage de tous les paramètres de la commande MPC pour tous les sous-modèles T-S, i.e. les horizons de prédiction \mathbf{N}_p et de contrôle \mathbf{N}_c ainsi que les facteurs de pondération $\boldsymbol{\lambda} > 0$, comme suit :

$$\begin{cases} \text{minimiser } J_1(\mathbf{x}, t) = \int_0^{+\infty} |e_i(y, t)| dt \\ \mathbf{x} = (\mathbf{N}_p, \mathbf{N}_c, \boldsymbol{\lambda}) \\ \text{sous contraintes :} \\ g_1(\mathbf{x}, t) = \mathbf{N}_c - \mathbf{N}_p \leq 0 \\ g_2(\mathbf{x}, t) = \boldsymbol{\lambda} > 0 \end{cases} \quad (4.20)$$

où $J(\cdot)$ est la fonction objectif du problème à minimiser, $g_i(\cdot)$ sont les contraintes non linéaires, voire non analytiques en les paramètres, de type inégalité, $\mathbf{x} = (\mathbf{N}_p, \mathbf{N}_c, \boldsymbol{\lambda})^T \in \mathbb{N}^4$ le vecteur des variables de décision, inconnus du problème, et $e_i(y, t)$ est l'erreur de poursuite entre la référence et la sortie réelle avec $i = \{\phi, \theta, \psi \text{ et } z\}$.

Face au problème d'optimisation formulé (4.20), nous proposons d'utiliser la technique de pénalisation suivante afin d'aboutir à une nouvelle expression de la fonction coût augmentée à minimiser sans contraintes :

$$J_2(\mathbf{x}, t) = \int_0^{+\infty} |e_i(y, t)| dt + \exp\left(1000 \frac{\mathbf{N}_c - \mathbf{N}_p}{\mathbf{N}_p}\right) \quad (4.21)$$

4.4 Résultats de simulation et discussion

Pour la mise en œuvre de l'approche de réglage proposée, nous retenons les paramètres de contrôle de l'algorithme mQPSO du tableau 4.2.

En partant des conditions initiales nulles et compte tenu des consignes désirées, les courbes présentant les variations temporelles des signaux de commande et de sorties asservies pour les dynamiques d'attitude et d'altitude sont données dans les figures 4.5, 4.6 et 4.7.

Les résultats d'optimisation relatifs au meilleur cas du problème (4.20) sont suggérés dans le tableau 4.3. Les figures 4.8(a) et 4.8(b) montrent la mise à jour des paramètres de la

Tableau 4.2 – Paramètres de contrôle d’algorithmes mQPSO.

Paramètres	Valeur
Nombre de pics	$5 \leq N \leq 100$
Fréq. de changement	5000
Nombre d’essais	5
Nombre de particules neutres	10
Nombre de particules chargées	10

commande prédictive TSMPC calculés en ligne, respectivement pour les sous-modèles T-S 1 et 2. D’après ces résultats, on remarque bien que les paramètres effectifs de l’approche MPC varient au cours de temps ce qui apporte à la commande un caractère adaptatif. Nous prenons l’exemple de sous-modèle T-S et à $t = 20s$. La méthode mQPSO peut favoriser la convergence et obtenir une bonne valeur de condition physique. Ces résultats montrent que l’approche dynamique proposée permet d’ajuster rapidement et avec efficacité remarquable les paramètres optimaux de la commande MPC.

Les performances en termes de précision et de rapidité pour le suivi des consignes sont satisfaisantes. L’approche proposée permet de stabiliser le drone autour des consignes désirées avec une dynamique de régulation rapide et précise.

Tableau 4.3 – Paramètres optimaux de la technique mQPSO-TSMPC obtenus en régime permanent.

Paramètres	Sous-modèle 1	Sous-modèle 2	Sous-modèle 3	Sous-modèle 4
N_p	42	38	43	19
N_c	32	28	3	9
λ	7	16	19	10

Considérons à présent le problème du suivi des trajectoires du véhicule convertible QTW. Dans cette étude, les résultats de mise en œuvre sont montrés dans les figures 4.9 et 4.10. Les trajectoires circulaires de référence retenues sont définies par les équations (4.4), (4.4) et (4.4) suivantes :

$$x_d(t) = 0.5 \sin(0.25t) \quad (4.22)$$

$$y_d(t) = 0.5 \cos(0.25t) \quad (4.23)$$

$$z_d(t) = 5t \quad (4.24)$$

Dans ce scénario, la poursuite d’une trajectoire hélicoïdale tridimensionnelle est considérée, figure 4.10. La réponse obtenue met en avant les hautes performances de la stratégie de commande TSMPC proposée avec réglage en ligne par l’algorithme mQPSO.

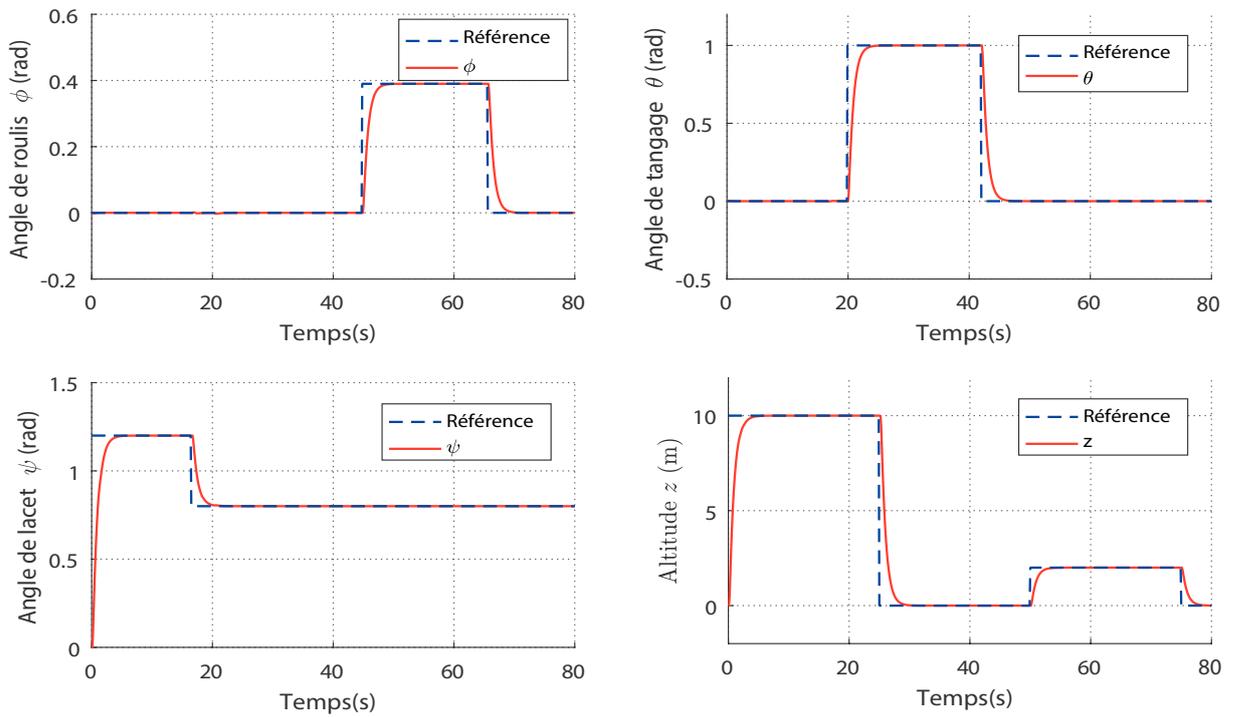


FIGURE 4.5 – Évolution des dynamiques d'attitudes et d'altitude en fonction du temps.

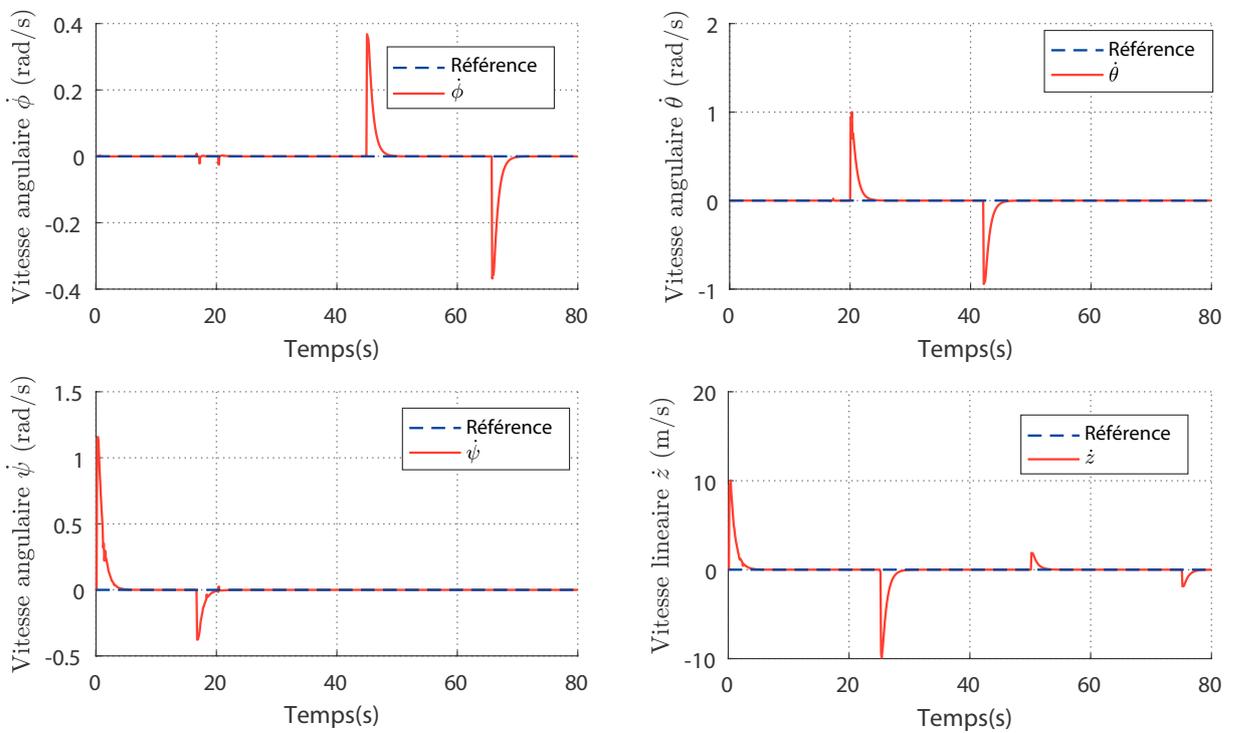


FIGURE 4.6 – Évolution des vitesses angulaires et linéaires en fonction du temps.

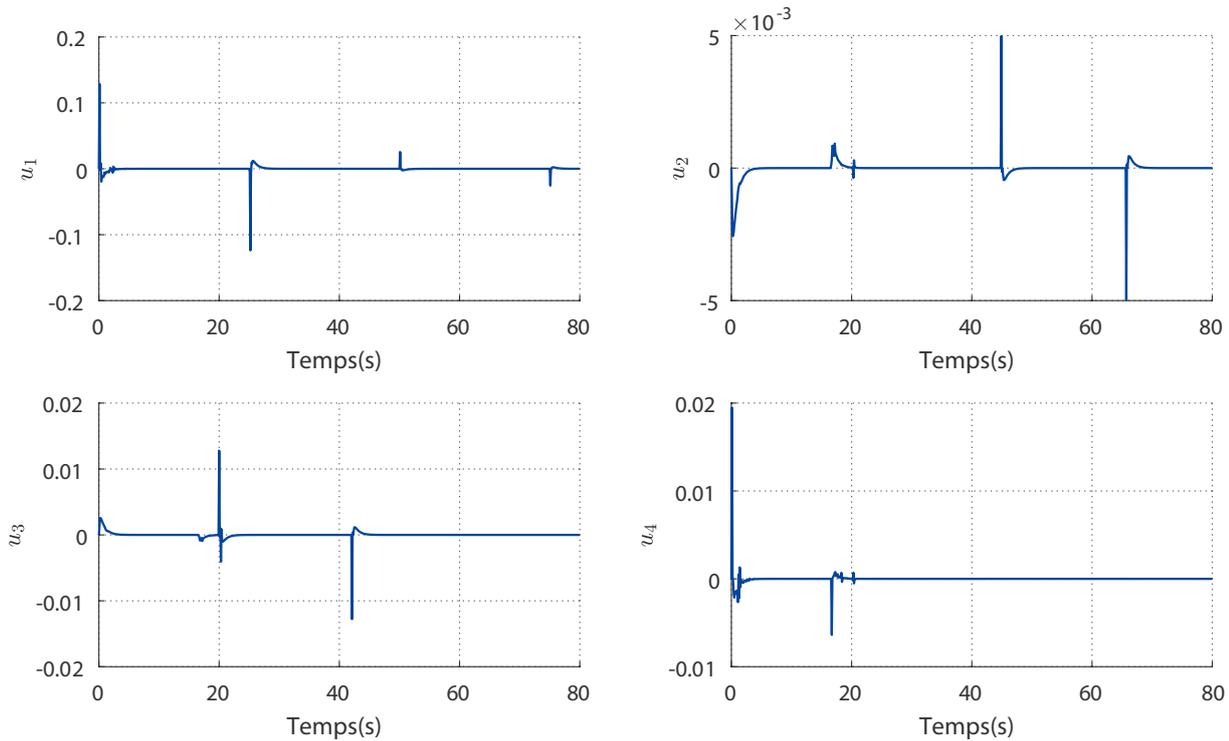


FIGURE 4.7 – Évolution des signaux de commande en fonction du temps.

4.4.1 Étude comparative

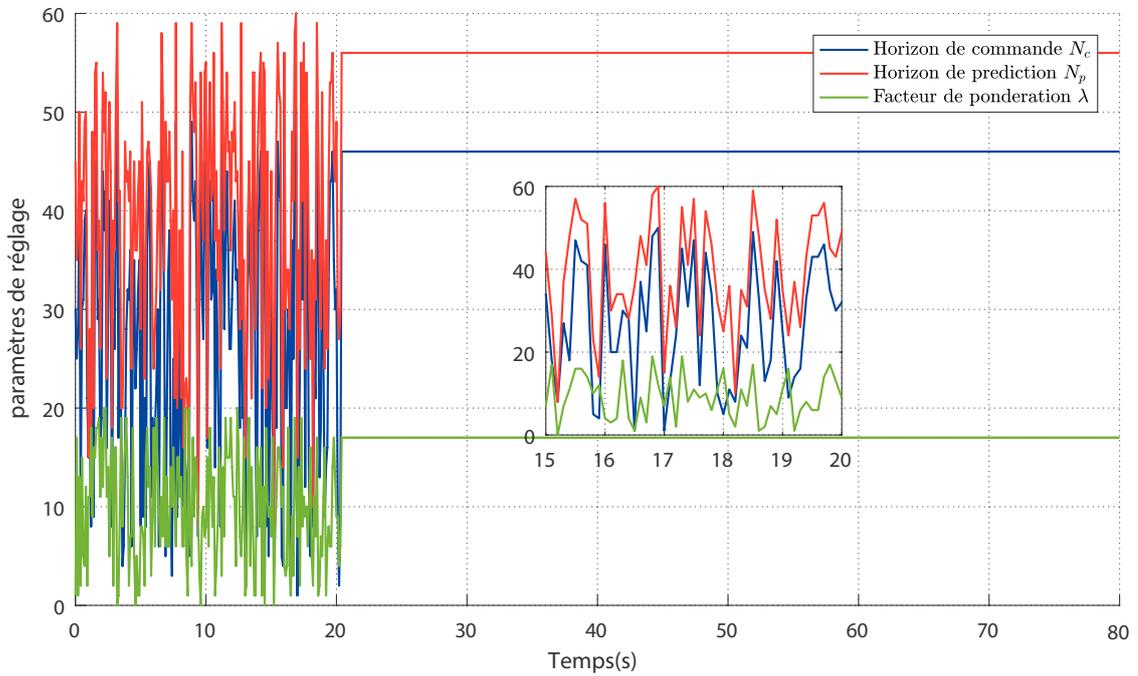
Afin de montrer l'efficacité et la validité de la stratégie de réglage proposée, nous menons dans cette partie, une étude comparative des performances obtenues avec d'autres correcteurs issus de la littérature.

- (a) le contrôle prédictif auto-adaptatif basé sur l'approche analytique désignée par ATSMPC (Turki et al., 2018a) ;
- (b) la compensation parallèle distribuée (PDC) (Sun and Liu, 2017) ;

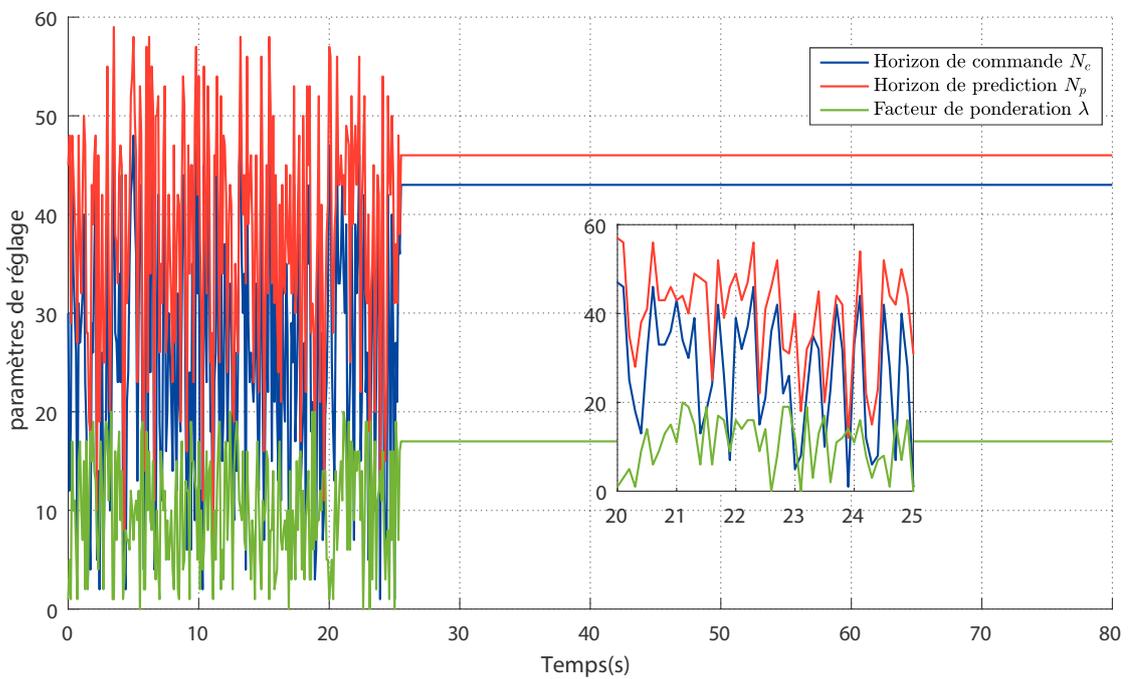
Les signaux de sortie et de commande sont présentés respectivement sur les figures 4.11 et 4.12. Des comparaisons de performances obtenues sont établies à partir des tableaux 4.4, 4.5 et 4.6.

D'après ces résultats, les temps de montée et d'établissement les plus rapides, pour toutes les dynamiques du vol, sont ceux obtenus avec l'approche de réglage en ligne proposée. En effet, grâce au choix approprié des paramètres de la stratégie MPC, nous proposons une amélioration du temps d'établissement d'environ dix fois comparée à la stratégie PDC proposée dans (Sun and Liu, 2017) et d'environ deux fois comparée à l'approche analytique ATSMPC (Turki et al., 2018a).

Si nous nous intéressons maintenant à la stabilité optimale du système en boucle fermée, nous constatons d'une part que le SDI calculé selon notre approche est le plus proche de l'unité. Ceci veut dire que la stabilité est la meilleure comparée aux autres. D'autre part, le dépassement minimal est garanti par notre approche qui est 40 fois plus petit pour



(a) Sous-modèle T-S : 1.



(b) Sous-modèle T-S : 2.

FIGURE 4.8 – Exemple d'évolution des paramètres de la TSMPC au cours du temps.

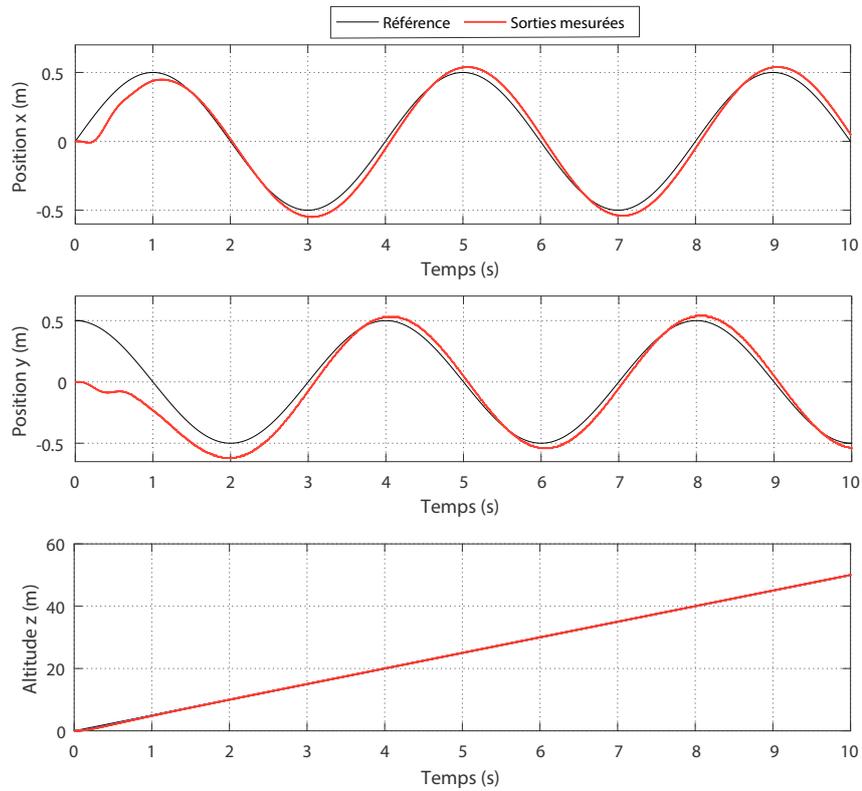


FIGURE 4.9 – Performances en poursuite de la commande TSMPC avec réglage par mQPSO.

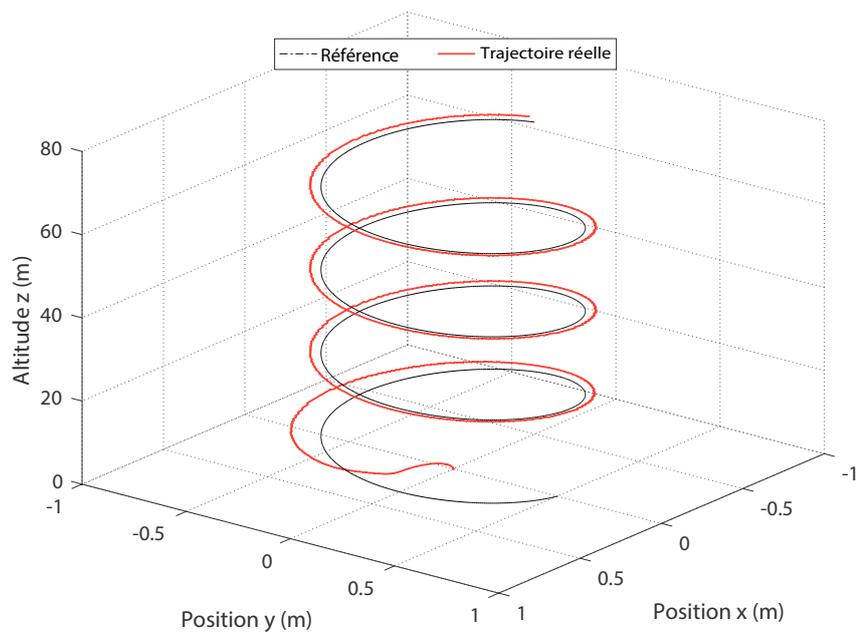


FIGURE 4.10 – Poursuite d'une trajectoire hélicoïdale dans l'espace.

les dynamiques d'altitude et roulis et d'environ 15 et 30 fois plus réduit respectivement pour les dynamiques de tangage et de lacet par rapport aux résultats de (Sun and Liu, 2017). Pour conclure, l'approche de réglage en ligne mQPSO-TSMPC proposée présente le meilleur compromis en termes d'énergie consommée et de rapidité. Les méthodes métaheuristiques et analytiques offrent toutes les deux des sorties stables avec un indice SDI presque identique et plus mieux que la commande PDC, plus particulièrement pour les dynamiques de roulis et tangage.

Tableau 4.4 – Comparaison des performances temporelles de la dynamique de roulis $\phi(t)$

Critères	PDC	ATSMPC	Approche proposée
RT (s)	4,216	3,233	1,785
ST (s)	32,285	5,546	3,347
PT (s)	16,201	40	39,7
OV (%)	39,502	0	$4,88e^{-13}$
MTE	0,304	$0,3053e^{-3}$	$3,458e^{-10}$
SDI	4,599	0,85 1	0,987
VARU	-0,243	-0,004	-0,458
CSE	245,485	156,229	98,875
CEE	3,540	6,521	8,257

Tableau 4.5 – Comparaison des performances temporelles de la dynamique de tangage $\theta(t)$

Critères	PDC	ATSMPC	Approche proposée
RT (s)	3,014	1,321	1,792
ST (s)	32,582	4,264	3,355
PT (s)	6,8	3,5	37,5
OV (%)	14,836	2,403	$2,930e^{-12}$
MTE	0,254	$0,657e^{-3}$	$5,687e^{-10}$
SDI	1,874	0,8551	0,965
VARU	-0,0033	-0,0045	-0,475
CSE	285,547	158,354	97,547
CEE	1,2207	7,5219	8,654

4.4.2 Analyse de robustesse de la méthode proposée

Examinons maintenant l'analyse de la robustesse de la commande adaptative proposée vis-à-vis des incertitudes paramétriques. Ainsi, nous considérons une incertitude sur le coefficient d'inertie du drone QTW donné comme suit :

$$\tilde{J} = J + \Delta J \quad (4.25)$$

Tableau 4.6 – Comparaison des performances temporelles de la dynamique de lacet $\psi(t)$

Critères	PDC	ATSMPC	Approche proposée
RT (s)	7,725	2,571	1,787
ST (s)	33,947	4,807	3,349
PT (s)	18,8	40	30,2
OV (%)	26,151	0	$7,77e^{-3}$
MTE	0,0033	$0,3053e^{-3}$	$8,674e^{-13}$
SDI	0,5992	0,8551	1,258
VARU	-0,0033	-0,0045	-0,874
CSE	222,4850	156,2298	80,158
CEE	0,9207	6,5219	10,247

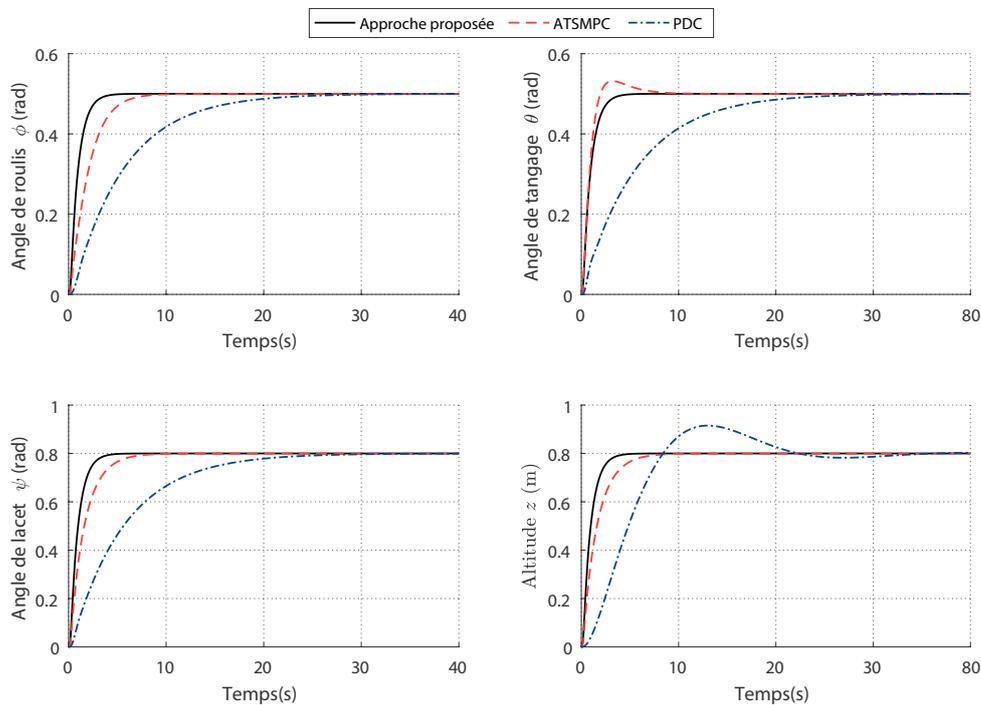


FIGURE 4.11 – Performances temporelles comparées des commandes PDC, ATSMPC et mQPSO-TSMPC : Signaux de sortie.

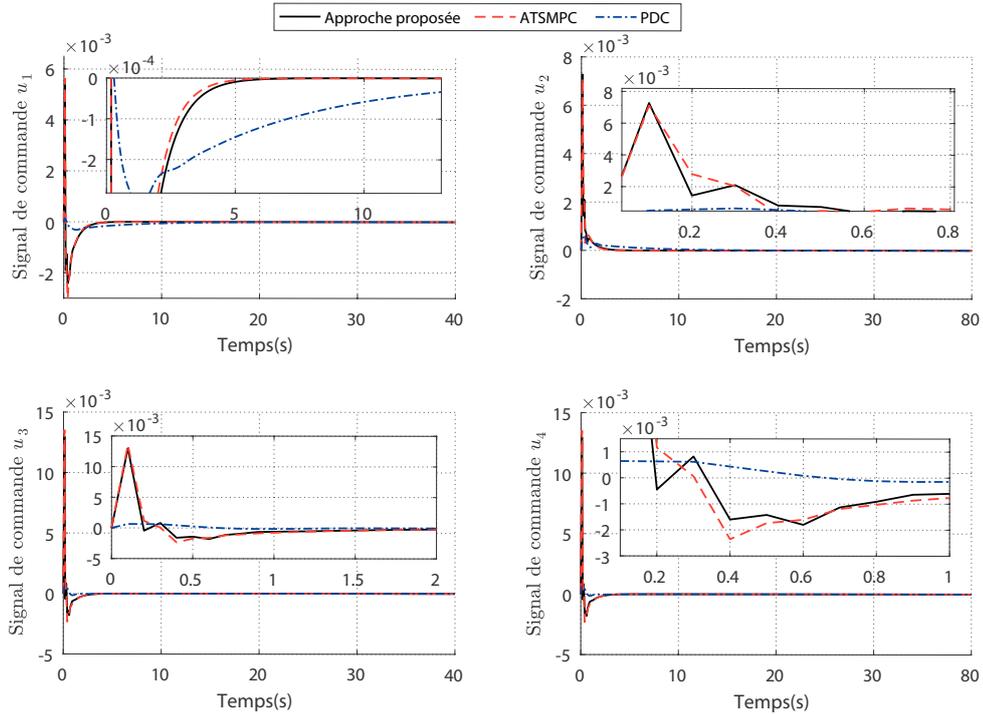


FIGURE 4.12 – Performances temporelles comparées des commandes PDC, ATSMPC et mQPSO-TSMPC : Signaux de commande.

où \mathbf{J} désigne la valeur nominale du coefficient d’inertie et $\Delta\mathbf{J}$ les incertitudes paramétriques.

Les figures 4.13, 4.14 et 4.15 montrent les réponses du système en boucle fermée des dynamiques de roulis, de tangage et de lacet avec un domaine borné d’incertitudes paramétriques.

On peut constater que la commande mQPSO-TSMPC proposée assure une bonne robustesse vis-à-vis d’incertitudes paramétriques et que la mise à jour de ses paramètres effectifs la rend plus performante et plus efficace face à d’éventuels défauts de ce type. La stabilité en boucle fermée est garantie pour les dynamiques de roulis, de tangage et de lacet avec une dégradation mineure des performances dans le domaine temporel.

La robustesse des correcteurs mQPSO-TSMPC est testée vis-à-vis des défauts d’actionneurs. En effet, nous considérons une perte d’efficacité de 20% sur les signaux de commande des quatre moteurs des hélices. La figure 4.16 montre la robustesse de la commande MPC adaptative à base de l’algorithme mQPSO. Ces contrôleurs, ayant désormais un comportement adaptatif, réduisent l’effet de ces défauts en ajustant, au cours de temps, les paramètres effectifs de la commande comme montré dans les figures 4.17, 4.18 et 4.19. En effet, le système contrôlé retourne suite à l’apparition des défauts à l’attitude désirée avec une dynamique amortie et relativement rapide, i.e. un temps de réponse de l’ordre de 0.9s pour les variables de tangage, de roulis et de lacet.

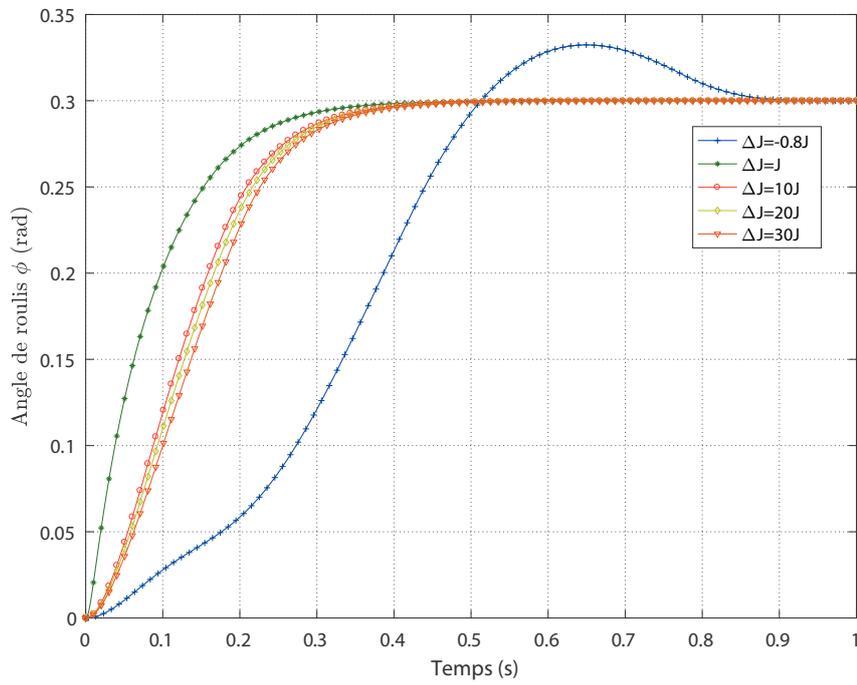


FIGURE 4.13 – Performance nominale et robuste de la dynamique de roulis.

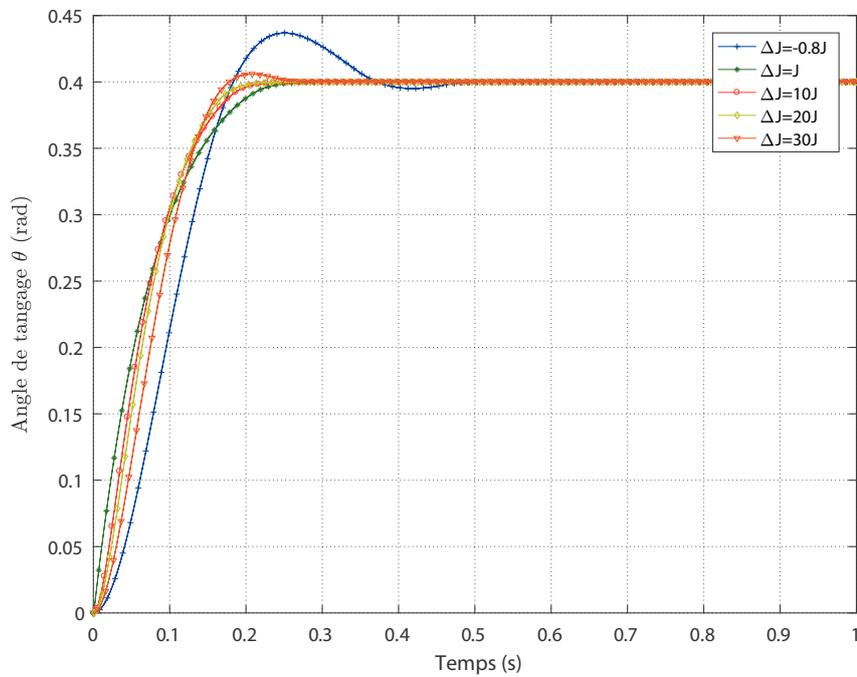


FIGURE 4.14 – Performance nominale et robuste de la dynamique de tangage.

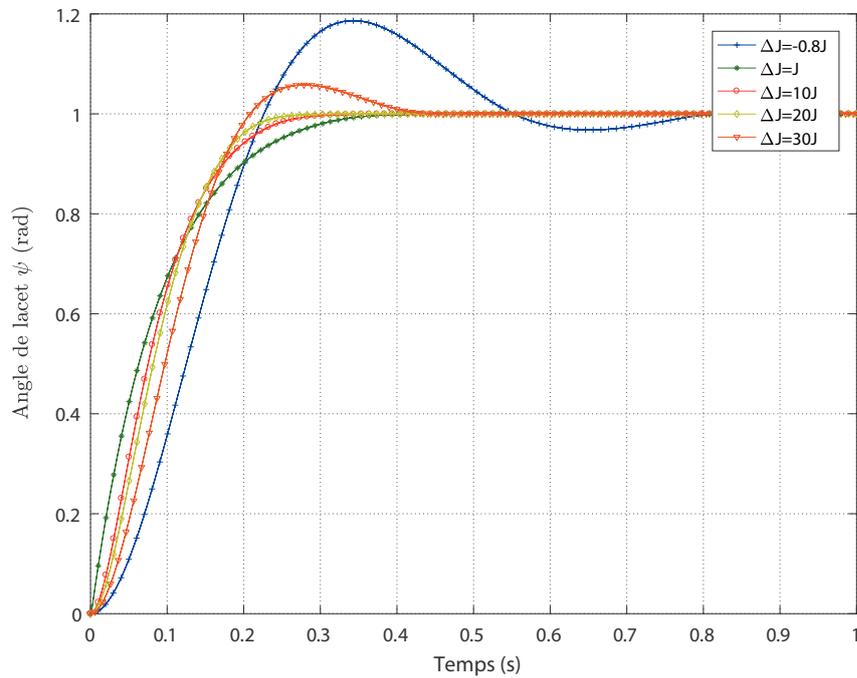


FIGURE 4.15 – Performance nominale et robuste de la dynamique d'attitude.

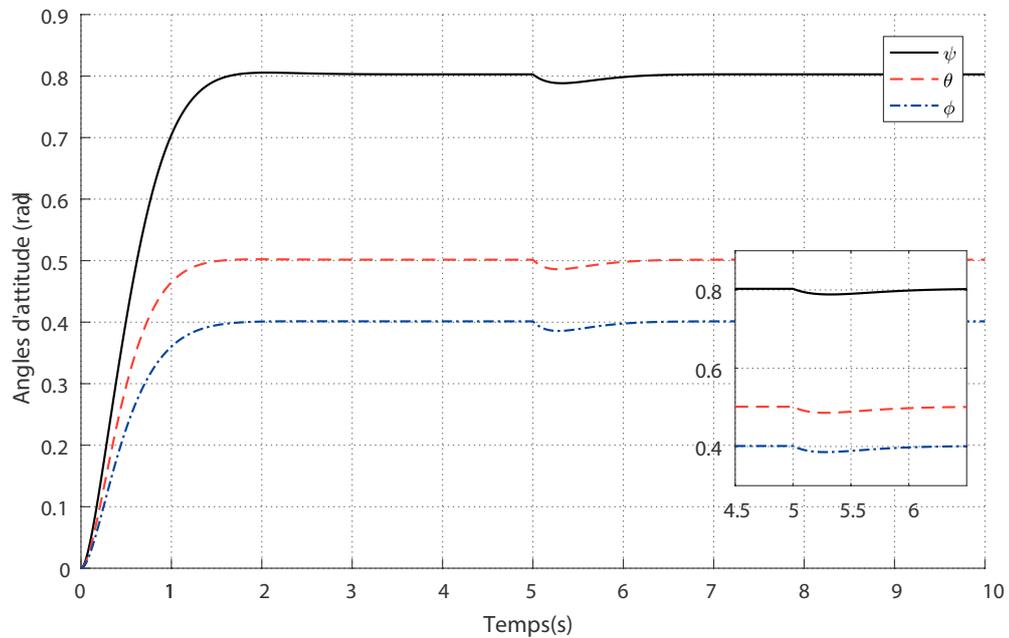


FIGURE 4.16 – Robustesse vis-à-vis d'un défaut de 20% sur la dynamique d'attitude.

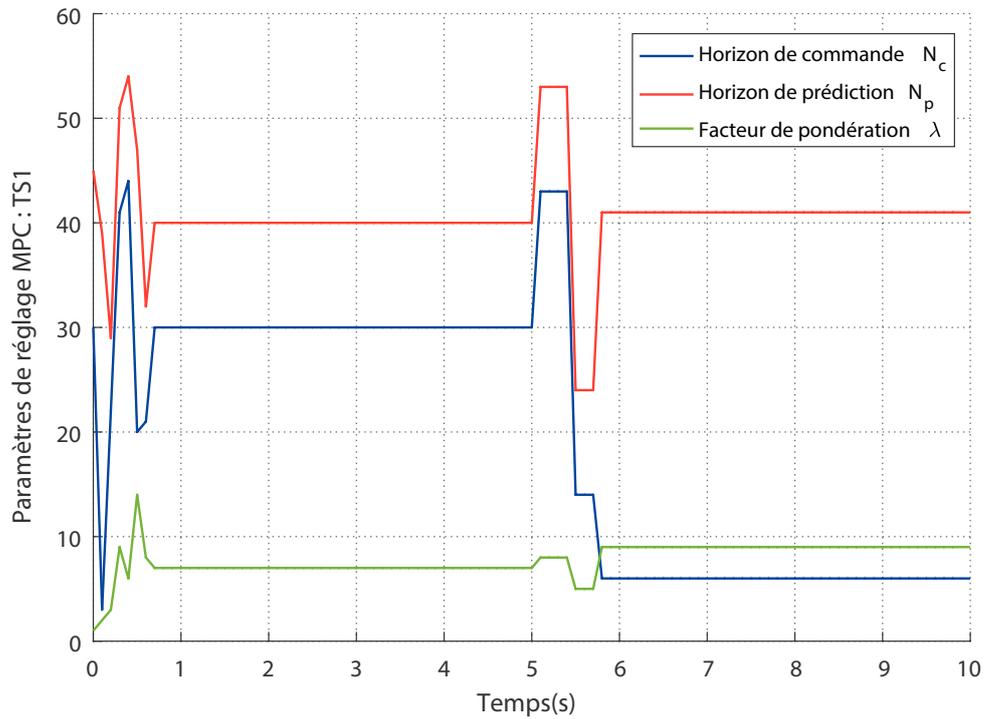


FIGURE 4.17 – Variation temporelle des paramètres de la technique MPC pour le sous-modèle T-S 1.

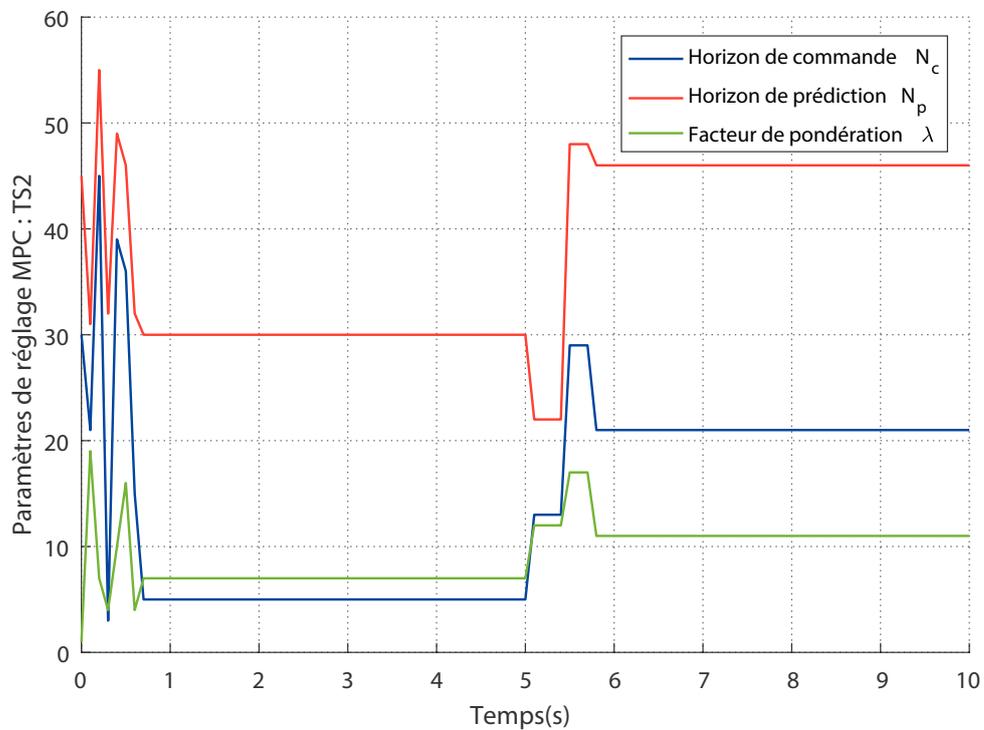


FIGURE 4.18 – Variation temporelle des paramètres de la technique MPC pour le sous-modèle T-S 2.

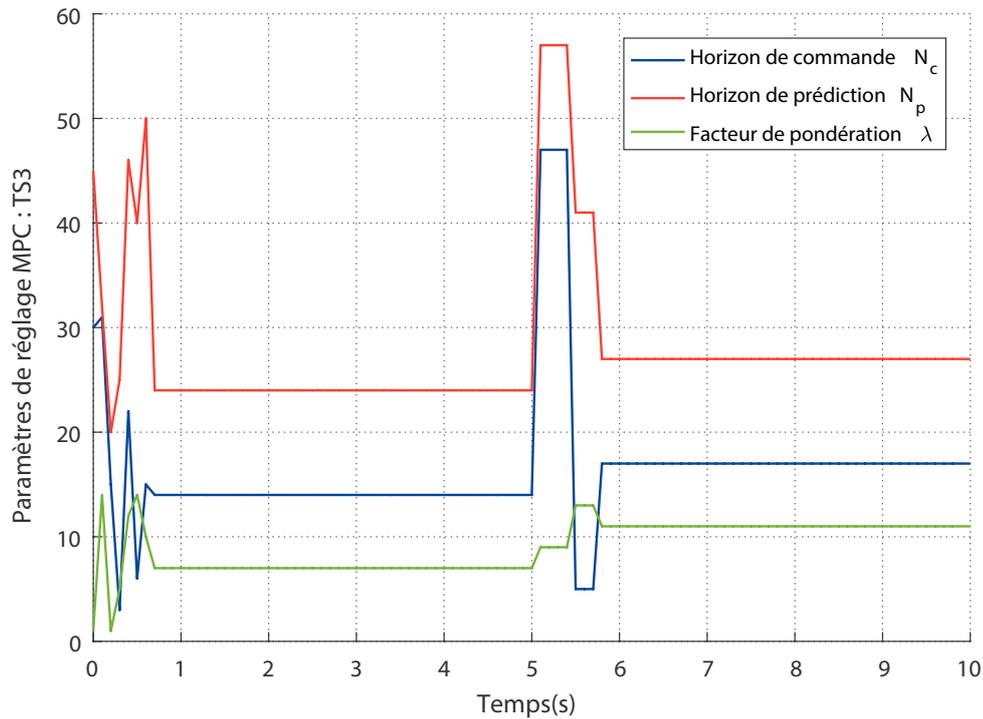


FIGURE 4.19 – Variation temporelle des paramètres de la technique MPC pour le sous-modèle T-S 3.

4.5 Conclusion

Dans ce chapitre, nous avons appliqué notre approche proposée pour le réglage adaptatif des paramètres de la commande prédictive MPC basée sur un algorithme métaheuristique dynamique mQPSO pour la stabilisation et la poursuite de trajectoires d'un drone convertible QTW. Nous avons comparé les résultats obtenus avec deux autres approches de commande. La première étant la commande MPC adaptative avec une technique analytique de réglage des paramètres ATSMPC. Quant à la seconde il s'agit de la commande PDC très utilisée pour les systèmes présentés sous forme T-S. Ces comparaisons sont toutefois effectuées afin de démontrer sur l'efficacité et la supériorité de l'outil de synthèse proposé.

Nous avons montré que la stabilisation du système en boucle fermée est possible sans avoir besoin de combiner des lois de commande PDC ou d'autres techniques de stabilisation de systèmes non linéaires décrits sous forme T-S. En plus, nous avons montré que l'approche de réglage proposée permet d'ajuster rapidement et avec une efficacité remarquable les paramètres effectifs de la commande MPC en assurant la stabilité optimale des sous modèles linéaires. La robustesse des correcteurs MPC optimisés est testée vis-à-vis des défauts d'actionneurs. En effet, nous avons considéré deux scénarios i.e. une perte d'efficacité sur les signaux de commande des quatre moteurs des hélices et des incertitudes paramétriques, pour montrer qu'un choix approprié des paramètres de la commande MPC d'améliorer la robustesse et les performances vis-à-vis d'éventuels défauts systèmes.

Conclusion générale et perspectives

Les travaux de recherche menés durant cette thèse ont visé à répondre au problème de la synthèse de lois de commande robuste et intelligente pour des aéronefs sans pilote de type convertible à ailes/rotors basculants QTW.

Nous avons commencé le mémoire, dans son premier chapitre, par l'introduction des connaissances de base sur les drones avec leur historique et leurs différentes catégories. Un intérêt particulier est porté aux véhicules ariens convertibles à ailes/rotors basculants. La famille QTW est plus particulièrement étudiée. Par la suite, un formalisme de modélisation de Newton-Euler a été retenu pour la modélisation. L'écriture, dans des référentiels adéquats, des forces et des moments aérodynamiques, a conduit à des modèles non linéaires dans l'espace d'état décrivant les différents modes de vol vertical, horizontal et de transition. Le modèle obtenu est alors complexe, non linéaire, multivariable et présentant une dynamique fortement couplée. Une structure de type Takagi-Sugeno (T-S) a été appliquée comme étant une technique de modélisation représentative et de contrôle. Cette technique permet l'écriture d'un système non linéaire sous forme quasi-LPV. Le modèle T-S résultant est obtenu à partir de l'approche par secteurs non linéaires qui peut représenter le comportement du système global.

Dans une deuxième partie, et dans l'objectif de préparer une alternative de comparaison, les dynamiques de position et d'attitude ont ensuite été stabilisées moyennant diverses approches de commande proposées, allant des techniques linéaires, comme les commandes LQR et MPC, jusqu'aux techniques non linéaires à savoir le backstepping avec action intégrale. Cette dernière a montré des performances élevées en termes de robustesse vis-à-vis de perturbations externes additives. Les performances en termes de poursuite de trajectoires en vol stationnaire sont également montrées. La commande prédictive est plus particulièrement étudiée et une formulation a été développée à base d'état et à temps discret en présence et en absence des contraintes opérationnelles et s'appuyant sur un modèle T-S du drone. L'effet des paramètres des commandes proposées a été étudié et présenté par simulations numériques. Ces paramètres effectifs de commande influencent d'une manière significative le comportement du système en boucle fermée, notamment, en termes de stabilité, rapidité et précision entre la sortie prédite et la référence. Leurs valeurs doivent être donc méticuleusement choisies.

Dans ce cadre, nous avons traité dans une troisième partie, après avoir présenté les algorithmes métaheuristiques de résolution proposée, le problème de réglage des paramètres effectifs des commandes optimales LQG, PID à supervision floue et MPC. Une nouvelle stratégie de commande LQG à paramètres optimisés a été établie pour la stabilisation des dynamiques d'attitude et d'altitude du drone QTW. Les résultats obtenus ont bien

montré la supériorité de l'approche proposée par rapport au cas LQG conventionnel. Dans un scénario de commande différent, nous avons opté pour la synthèse d'une commande stabilisante de type PID avec un mécanisme de supervision basé sur la logique floue. Le réglage et l'adéquation des paramètres de conception d'un tel système d'inférence floue sont accomplis à travers le recours aux métaheuristiques globales de type PSO. Les gains des correcteurs PID, implémentés pour les différentes dynamiques contrôlées de tangage, de roulis et de lacet, deviennent variables dans le temps. Ce caractère adaptatif de la commande proposée améliore davantage les performances de la boucle fermée en termes de robustesse vis-à-vis de défauts sur les actionneurs des hélices du véhicule QTW. Dans le même but de réduire davantage la complexité grandissante des méthodes de synthèse classiques, nous avons traité aussi le problème de réglage des paramètres effectifs de la commande prédictive à base de modèle MPC. Le problème du choix, souvent difficile, des facteurs de pondération et des horizons de contrôle et de prédiction, est contourné via la reformulation de la procédure de synthèse MPC sous forme d'un problème d'optimisation sous contraintes. Les résultats de mise en œuvre de ces approches PSO ont montré des hautes performances, en régulation comme en poursuite, avec une systématisation remarquable de la procédure de synthèse. Un gain remarquable en temps de conception de telles stratégies de commande non conventionnelles est ainsi montré et les inconvénients des méthodes classiques d'essais-erreurs se voient surmontés. Cependant, ces approches restent incapables de maintenir les performances désirées en cas des variations paramétriques ou d'éventuels défauts instantanés pouvant affecter le système à contrôler.

Comme solution à ces limitations, nous avons proposé, dans le quatrième chapitre, une nouvelle approche de réglage en ligne à base d'algorithmes métaheuristiques issus de l'optimisation dynamique, à savoir la technique mQPSO. Ces derniers sont capables de détecter un changement éventuel dans la fonction objectif des problèmes d'optimisation reformulés au cours de temps. Comparée, d'une part, à l'approche PDC et une approche de réglage analytique ATSMPC, d'autre part, notre approche de commande mQPSO-TSMPC a montré des performances élevées en termes de robustesse vis-à-vis de perturbations externes additives et d'incertitudes paramétriques. En plus, nous avons montré que l'approche de réglage proposée permet d'ajuster rapidement et avec une efficacité remarquable les paramètres effectifs de la commande MPC en assurant la stabilité des sous modèles linéaires. Les performances en termes de poursuite de trajectoires en vol stationnaire sont également montrées.

Les résultats obtenus dans nos travaux sont encourageants et prometteurs. La méthodologie de réglage mQPSO-TSMPC proposée peut être facilement appliquée pour l'implémentation non seulement des commandes linéaires étudiées mais aussi pour d'autres types de commandes non linéaires et plus complexes. L'ensemble des perspectives que nous estimons abordables seront présentées. Elles peuvent apporter soit des améliorations, soit des nouveautés aux contributions apportées par ce travail :

- L'apport des métaheuristiques dans la résolution de problèmes d'Automatique avancée et l'extension de ces algorithmes dans un formalisme d'optimisation dynamique multi-objectifs en particulier, pour la résolution de problèmes de diagnostic des défauts de systèmes non linéaires et de la planification de trajectoires de drones.
 - Mener une étude théorique sur les limites de robustesse de l'approche mQPSO-
-

TSMPC proposée vis-à-vis des défauts instantanés sur le système. La simulation numérique a montré qu'une amplitude et des incertitudes paramétriques élevées pouvaient déstabiliser le système en boucle fermée. Le contrôleur MPC dans ces conditions perd sa robustesse. Par conséquent, une étude théorique définissant les limites exactes de l'approche mQPSO-TSMPC par rapport à l'ampleur de défaut paraît intéressante.

- L'implémentation pratique de l'approche mQPSO-TSMPC sur un prototype réel afin de voir le véritable comportement de ce système dans la réalité et de montrer la faisabilité du travail de recherche réalisé sur le terrain pouvant montrer ainsi les performances réelles comparées à celles obtenues en simulation numérique.
-

Annexes

- **Annexe A : Matrice de transformation de vitesse**
- **Annexe B : Modèle Simulink de Quad Tilt Wing**
- **Annexe C : Code Source LabVIEW de modèle et des commandes**

Annexe A
Matrice de transformation de vitesse

$$\begin{aligned}
\begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_x^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_x^T R_y^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ c\phi\dot{\theta} \\ -s\phi\dot{\theta} \end{bmatrix} + \begin{bmatrix} c\theta & 0 & -s\theta \\ s\phi s\theta & c\phi & s\phi c\theta \\ s\theta & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ c\phi\dot{\theta} \\ -s\phi\dot{\theta} \end{bmatrix} + \begin{bmatrix} -s\theta \\ s\phi c\theta\dot{\psi} \\ c\phi c\theta\dot{\psi} \end{bmatrix}
\end{aligned}$$

D'où

$$= \begin{bmatrix} \dot{\phi} - s\theta \\ c\phi\dot{\theta} + s\phi c\theta\dot{\psi} \\ -s\phi\dot{\theta} + c\phi c\theta\dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = E(\alpha_i) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Finalement, on aura :

$$E(\alpha_i) = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \tag{A.1}$$

Annexe B

Modèle Simulink de Quad Tilt Wing

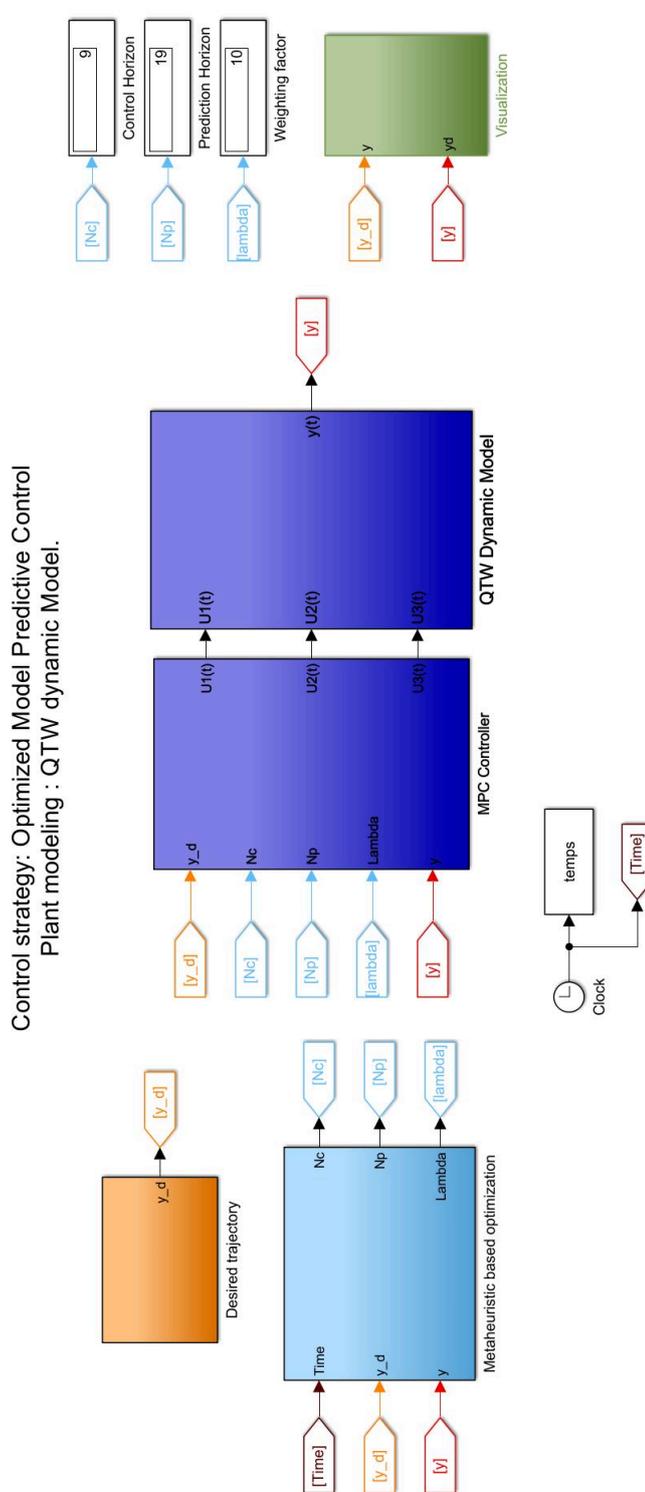


FIGURE B.1 – Stratégie de réglage dynamique de la commande TSMPC.

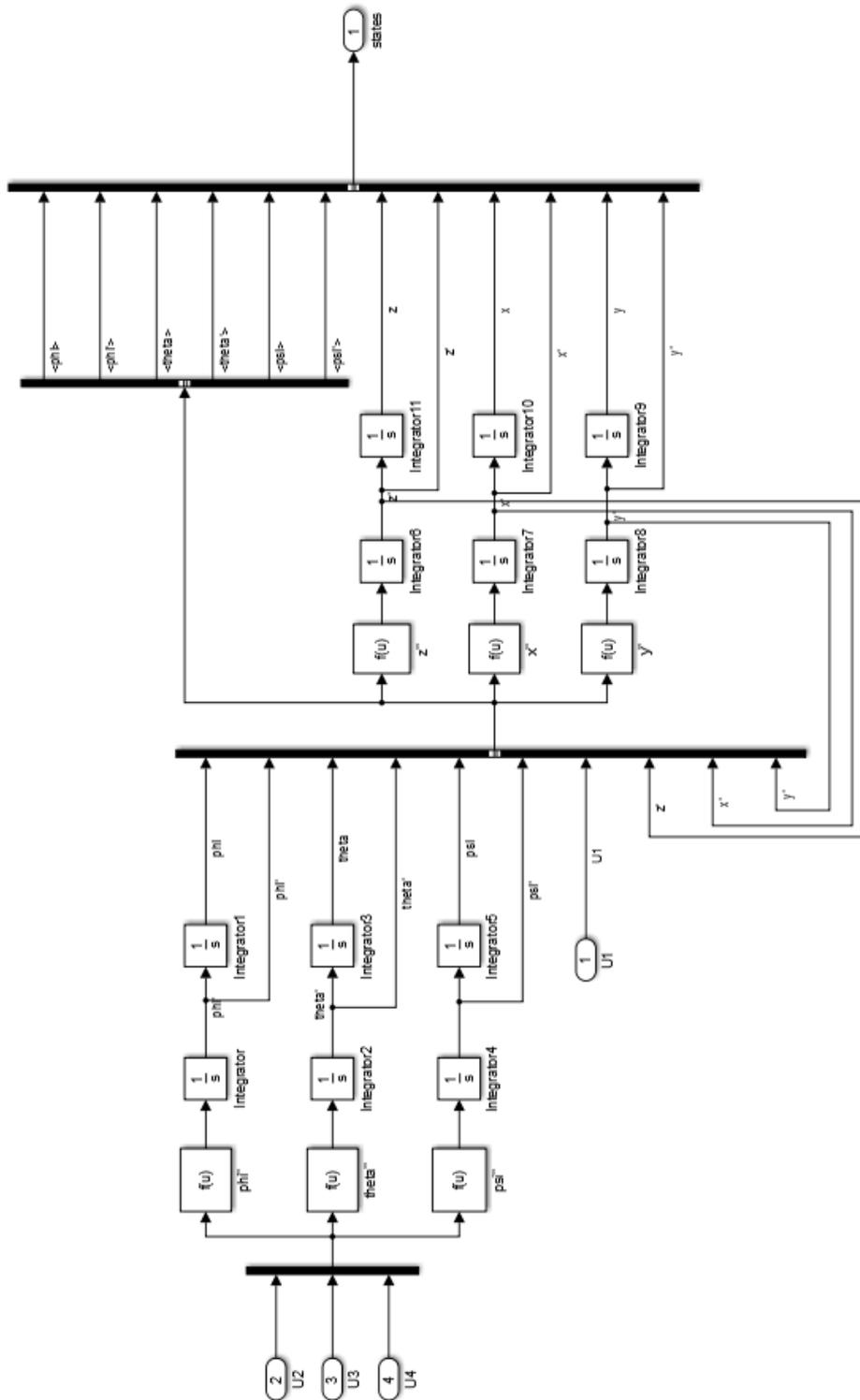


FIGURE B.2 – Modèle Dynamique de QTW en vol vertical.

Annexe C

Code Source LabVIEW de modèle et des commandes

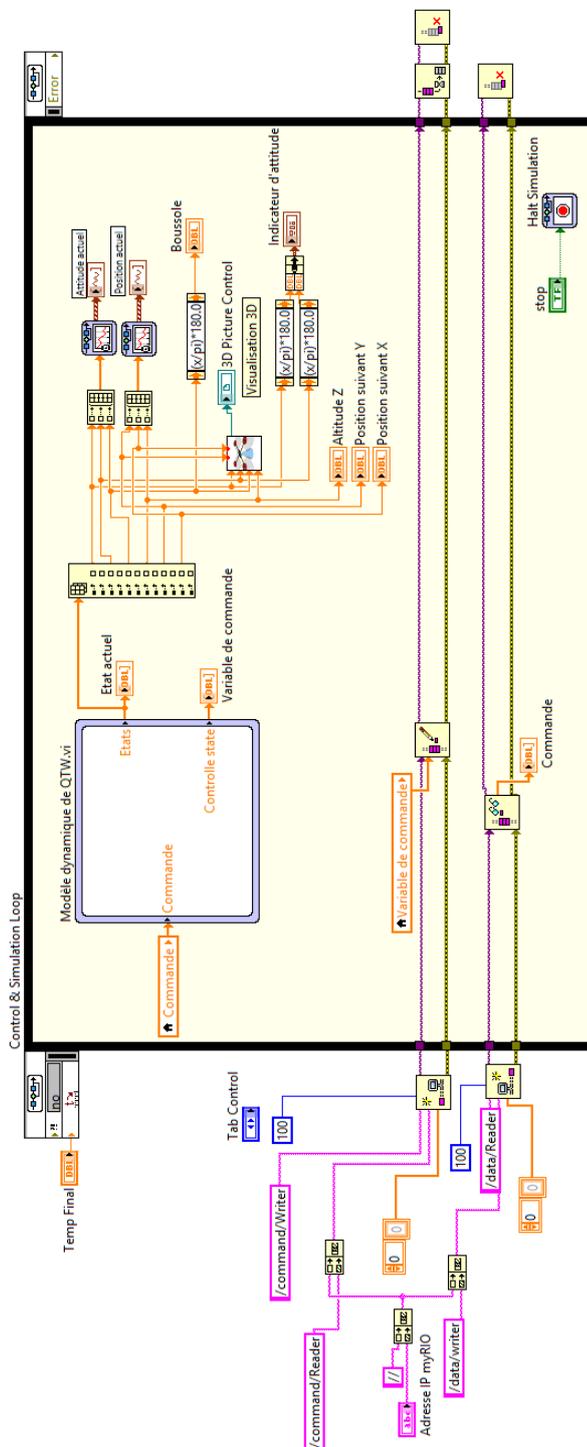


FIGURE C.1 – Code source LabVIEW du modèle QTW.

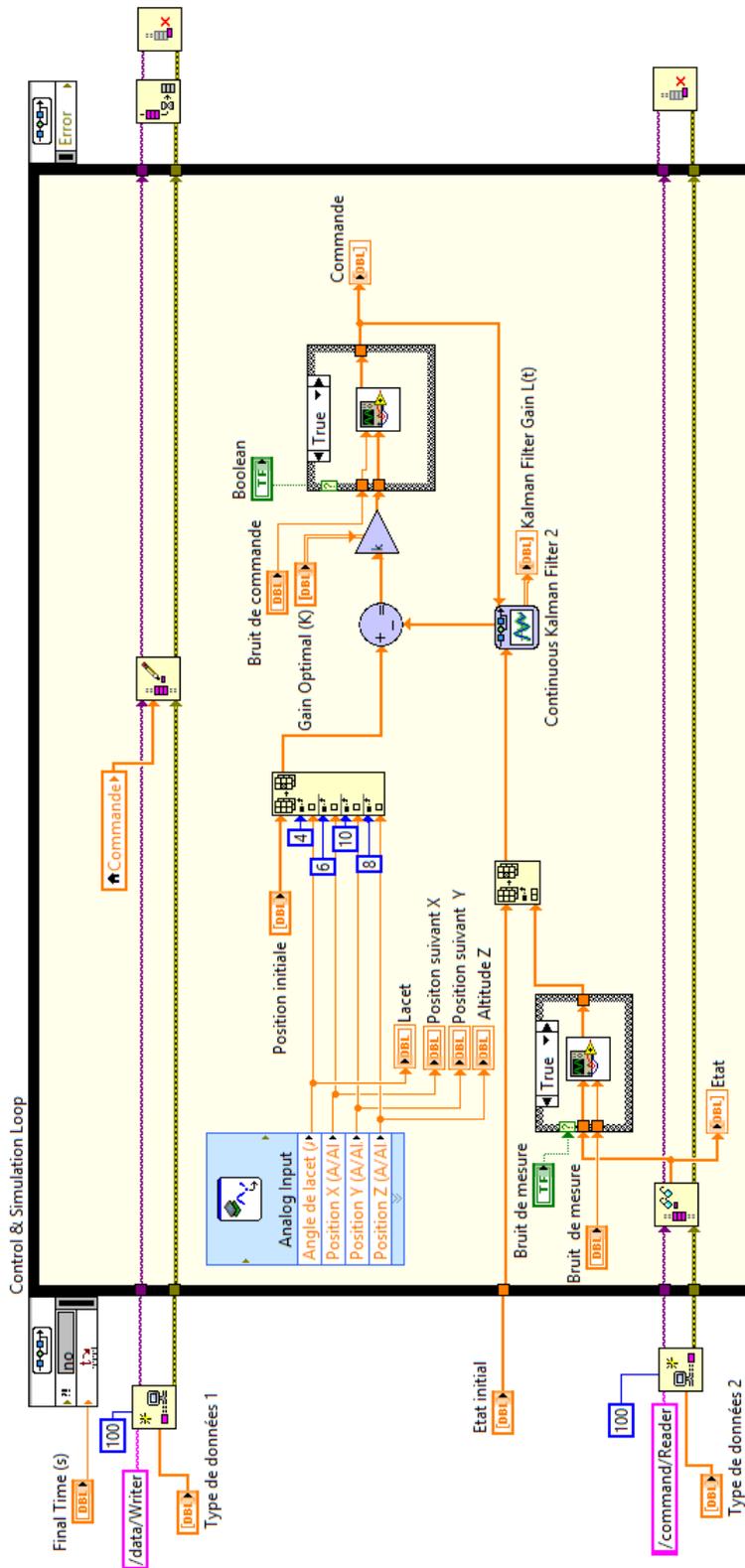


FIGURE C.2 – Code source LabVIEW pour la co-simulation commande LQG.

Liste des publications

Articles de revues internationales et nationales avec comité de lecture

1. **K. BenKhoud**, S. Bouallègue, and M. Ayadi, (2018). Design and co-simulation of a fuzzy gain-scheduled PID controller based on particle swarm optimization algorithms for a quad tilt wing unmanned aerial vehicle. *Transactions of the Institute of Measurement and Control*, 40(14), 3933–3952. [THOMSON REUTERS IF= 1,956]
2. **K. BenKhoud** and S. Bouallègue, (2018). Dynamics Modeling and Advanced Metaheuristics Based LQG Controller Design for a Quad Tilt Wing UAV. *International Journal of Dynamics and Control*, 6(2) :630–651. [Indexed In Scopus]
3. S. Bouallègue and **K. BenKhoud**, (2016). Integral Backstepping Control Prototyping for a Quad Tilt Wing Unmanned Aerial Vehicle. *International Review of Aerospace Engineering*, 9(5) :152–161. [Indexed In Scopus]

Articles de conférences internationales et nationales avec comité de lecture

1. M. Turki, **K. Benkhoud**, N. Langlois, and A. Yassine, (2018). Analytical Tuning of a TSMPC Dedicated to Nonlinear MIMO Systems. In *Proceedings of the 2018 European Control Conference*, pages 1367–1372, Limassol. [IEEE Xplore].
 2. **K. BenKhoud**, S. Bouallègue, and M. Ayadi (2017), Rapid control prototyping of a Quad Tilt Wing unmanned aerial vehicle. In *Proceedings of 2017 International Conference on Control, Automation and Diagnosis*, pages 423–428, Hammamet. [IEEE Xplore].
 3. **K. BenKhoud** and S. Bouallègue, (2016). Model Predictive Control Design for a convertible Quad Tilt-Wing UAV. In *Proceedings of the fourth International Conference on Control Engineering and Information Technology*, pages 1–6, Hammamet. [IEEE Xplore]
 4. **K. BenKhoud** and S. Bouallègue, (2016). Modeling and LQG Controller Design for a Quad Tilt-Wing UAV. In *Proceedings of the third International Conference on Automation, Control Engineering and Computer Science*, pages 198–204, Hammamet.
-

Communications orales

1. **K. BenKhoud**, S. Bouallègue, M. Ayadi and N. Langlois, *Diagnostic et commande prédictive tolérante aux défauts d'un drone convertible de type QTW*, Communication orale interne IRSEEM, Rouen, France, 20 Juin 2017.
2. M. Turki, **K. Benkhoud**, N. Langlois, and A. Yassine, (2018). *Analytical Tuning of a TSMPC dedicated to nonlinear MIMO systems*, Groupe de Travail Commande Prédictive Non linéaire (CPNL), Châtillon, 04 Juin 2018.

Posters

1. **K. BenKhoud**, S. Bouallègue, M. Ayadi and N. Langlois, (2017), *Diagnostic et commande prédictive tolérante aux défauts d'un drone convertible de type QTW*, Journée des Doctorants de l'IRSEEM (JDD), Rouen, France, 30 Novembre 2017.
 2. **K. BenKhoud**, S. Bouallègue, M. Ayadi and N. Langlois, (2019), *Optimisation dynamique par multi-essais particuliers de lois de commande prédictive pour un drone convertible QTW*, Journée des Doctorants de l'IRSEEM (JDD), Rouen, France, 27 Mars 2019.
-

Références bibliographiques

- Abdenmour, R. B., Borne, P., and Ksouri, M. (2001). *Identification et commande numérique des procédés industriels*, volume 11. Editions Technip.
- Abdessameud, A. and Tayebi, A. (2013). *Motion Coordination for VTOL Unmanned Aerial Vehicles : Attitude Synchronisation and Formation Control*. Springer, London.
- Akhenak, A. (2004). *Conception d'observateurs non linéaires par approche multi-modèle application au diagnostic*. Thèse de Doctorat, Institut National Polytechnique de Lorraine.
- Alba, E., Nakib, A., and Siarry, P. (2013). *Metaheuristics for Dynamic Optimization*. Springer-Berlin, Germany.
- Alexis, K., Nikolakopoulos, G., and Tzes, A. (2010). Constrained-control of a quadrotor helicopter for trajectory tracking under wind-gust disturbances. In *Proceedings of the 15th IEEE Mediterranean Electrotechnical Conference*, pages 1411–1416, Valletta.
- Allouche, B. (2016). *Modélisation et commande des robots nouvelles approches basées sur les modèles Takagi-Sugeno*. Thèse de Doctorat, Université de Valenciennes et du Hainaut-Cambresis.
- Argentim, L., Rezende, W., Santos, P., and Aguiar, R. (2013). Pid, lqr and lqr-pid on a quadcopter platform. In *Proceedings of the International Conference on Informatics, Electronics & Vision*, Dhaka.
- Aström, K. and Hägglund, T. (1995). *PID Controllers : Theory, Design, and Tuning*. Instrument Society of America, Research Triangle Park, North Carolina.
- Austin, R. (2010). *Unmanned Aircraft Systems : UAVs Design, Development and Deployment*. John Wiley & Sons Ltd, London.
- Bagheri, P. and Khaki-Sedigh, A. (2010). An anova based analytical dynamic matrix controller tuning procedure for fopdt models. *Amirkabir International Journal of Modeling, Identification, Simulation & Control*, 42(2) :55–64.
- Balchen, J., Ljungquist, D., and Strand, S. (1989). State space model predictive control of a multi stage electro-metallurgical process. *Modeling*, 10(1) :35–51.
-

- Banerjee, P. T., Roychoudhury, J., Das, S., and Abraham, A. (2010). Hybrid intelligent predictive control system for high speed bldc motor in aerospace application. In *Proceedings of the 3rd International Conference on Emerging Trends in Engineering and Technology (ICETET)*, pages 258–262.
- Bara, G. (2001). *Estimation d'état des systèmes linéaires à paramètres variants*. Thèse de Doctorat, Vandoeuvre-les-Nancy, INPL.
- BenKhoud, K. and Bouallègue, S. (2016a). Model Predictive Control Design for a convertible Quad Tilt-Wing UAV. In *Proceedings of the 4th International Conference on Control Engineering & Information Technology*, pages 1–6, Hammamet.
- BenKhoud, K. and Bouallègue, S. (2016b). Modeling and LQG Controller Design for a Quad Tilt-Wing UAV. In *Proceedings of the 3rd International Conference on Automation, Control Engineering and Computer Science*, pages 198–204, Hammamet.
- BenKhoud, K. and Bouallègue, S. (2018). Dynamics Modeling and Advanced Metaheuristics Based LQG Controller Design for a Quad Tilt Wing UAV. *International Journal of Dynamics and Control*, 6(2) :630–651.
- BenKhoud, K., Bouallègue, S., and Ayadi, M. (2017). Rapid control prototyping of a quad-tilt-wing unmanned aerial vehicle. In *Proceedings of 2017 International Conference on Control, Automation and Diagnosis*, pages 423–428, Hammamet.
- BenKhoud, K., Bouallègue, S., and Ayadi, M. (2018). Design and co-simulation of a fuzzy gain-scheduled PID controller based on particle swarm optimization algorithms for a quad tilt wing unmanned aerial vehicle. *Transactions of the Institute of Measurement and Control*, 40(14) :3933–3952.
- Bertrand, S. (2007). *Commande de drone miniature à voilure tournante*. Thèse de Doctorat, Office National d'Etudes Et Recherches Aérospatiales, Nice.
- Bezzaoucha, S. (2013). *Commande tolérante aux défauts de systèmes non linéaires représentés par des modèles de Takagi-Sugeno*. Thèse de Doctorat, Université de Lorraine.
- Blackwell, T. and Branke, J. (2004). Multi-swarm optimization in dynamic environments. In *Applications of Evolutionary Computing*, pages 489–500, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Blackwell, T. and Branke, J. (2006). Multi-swarms, exclusion and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 10(4) :459–472.
- Boeing (2016). V-22 Osprey Guide book. Technical report. Accessed on July 2016.
- Bouabdallah, S. and Siegwart, R. (2007). Full Control of a Quadrotor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–158, San Diego, California.
-

- Bouallègue, S. and BenKhoud, K. (2016). Integral Backstepping Control Prototyping for a Quad Tilt Wing Unmanned Aerial Vehicle. *International Review of Aerospace Engineering*, 9(5) :152–161.
- Bouallègue, S. and Fessi, R. (2016). Rapid Control Prototyping and PIL Co-Simulation of a Quadrotor UAV Based on NI myRIO-1900 Board. *International Journal of Advanced Computer Science and Applications*, 7(6) :26–35.
- Bouallègue, S., Haggège, J., Ayadi, M., and Benrejeb, M. (2012). PID-type Fuzzy Logic Controller Tuning Based on Particle Swarm Optimization. *Engineering Applications of Artificial Intelligence*, 25 :484–493.
- Bouallègue, S., Haggège, J., and Benrejeb, M. (2011). Particle Swarm Optimization-Based Fixed-Structure \mathcal{H}_∞ Control Design. *International Journal of Control, Automation and Systems*, 9(2) :258–266.
- Bouallègue, S., Toumi, F., Haggège, J., and Siarry, P. (2015). *Complex System Modeling and Control through Intelligent Soft Computations : Studies in Fuzziness and Soft Computing*, chapter Advanced Metaheuristics-based Approach for Fuzzy Control Systems Tuning. Springer-Verlag, Germany. DOI 10.1007/978-3-319-12883-2-22.
- Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in System and Control Theory*, volume 15. Society for Industrial and Applied Mathematics.
- Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
- Branke, J. (2002). *Evolutionary Optimization in Dynamic Environments*. Springer, US. DOI 10.1007/978-1-4615-0911-0.
- Branke, J., Kaussler, T., Schmidt, C., and Schmeck, H. (2000). A multi-population approach to dynamic optimization problems. In Parmee, I., editor, *Evolutionary Design and Manufacture*, pages 299–307, London. Springer London.
- Cao, S., Rees, N., and Feng, G. (1999). Analysis and design of fuzzy control systems using dynamic fuzzy-state space models. *IEEE Transactions on Fuzzy Systems*, 7(2) :192–200.
- Cao, Y., Lam, J., and Sun, Y. (1998). Static output feedback stabilization : an ilmi approach. *Automatica*, 34(12) :1641–1645.
- Castillo, P., Lozano, R., and Dzul, A. (2004). *Modeling and Control of Mini-Flying Machines*. Springer, London.
- Cazy, N. (2016). *Commande prédictive pour la réalisation de tâches d’asservissement visuel successives*. Thèse de Doctorat, Rennes 1.
- Cetinsoy, E., Dikyar, S., Hancer, C., Oner, K., Sirimoglu, E., Unel, M., and Aksit, M. (2012). Design and Construction of a Novel Quad Tilt-Wing UAV. *Mechatronics*, 22(6) :723–745.
-

- Chadli, M. (2002). *Stabilité et commande de systèmes décrits par des multimodèles : Approche LMI*. Thèse de Doctorat, Institut National Polytechnique de Lorraine-INPL.
- Chadli, M., Didier, M., and Ragot, J. (2001). On the stability analysis of multiple model systems. In *Proceedings of the 2001 European Control Conference (ECC)*, pages 1894–1899, Porto.
- Chadli, M., Didier, M., and Ragot, J. (2002). An LMI formulation for output feedback stabilization in multiple model approach. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 311–316, Las Vegas,.
- Chadli, M. and Karimi, R. (2013). Robust observer design for unknown inputs takagi-sugeno models. *IEEE Transactions on Fuzzy Systems*, 21(1) :158–164.
- Changhe, L. and Shengxiang, Y. (2008). A generalized approach to construct benchmark problems for dynamic optimization. pages 391–400.
- Changhe, L., Shengxiang, Y., and David, P. (2011). Benchmark generator for the iee wcci-2012 competition on evolutionary computation for dynamic optimization problems.
- Clarke, D., Mohtadi, C., and Tuffs, P. (1987). Generalized predictive control part 1. the basic algorithm. *Automatica*, 23(2) :137–148.
- Clerc, M. and Kennedy, J. (2002). The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, 6(1) :58–73.
- Derouiche, M., Bouallègue, S., Haggège, J., and Sandou, G. (2016). LabVIEW Perturbed Particle Swarm Optimization Based Approach for Model Predictive Control Tuning. In *Proceedings of the 4th IFAC International Conference on Intelligent Control and Automation Sciences*, pages 359–364, Reims.
- Dimitrov, D., Sherikov, A., and Wieber, P. (2011). A sparse model predictive control formulation for walking motion generation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2292–2299.
- Dinh, Q., Savorgnan, C., and Diehl, M. (2011). Real-time sequential convex programming for nonlinear model predictive control and application to a hydro-power plant. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5905–5910.
- Djeghaba, M. (2012). *Modélisation multimodèle et commande prédictive d’une station d’épuration*. Thèse de Doctorat, Université Badji Mokhtar-Annaba.
- Dréo, J., Pétrowski, A., Siarry, P., and Taillard, E. (2006). *Metaheuristics for Hard Optimization : Methods and Case Studies*. Springer-Verlag, Berlin.
-

- Drogies, S. and Geest, D. (1999). Predictive Control : Propositions for the Design Methodology. In *Proceedings of the 1999 American Control Conference*, pages 647–651, San Diego, CA.
- D.Yazdani, Nasiri, B., Sepas-Moghaddam, A., and Meybodi, M. R. (2013). A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. *Applied Soft Computing*, 13(4) :2144 – 2158.
- Eberhart, R. and Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory. In *Proceedings of The 6th International Symposium on Micro Machine and Human Science*, pages 39–43, Nagoya.
- Eberhart, R. and Shi, Y. (2001). Particle Swarm Optimization : Developments, Applications and Resources. In *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 81–86, Seoul.
- Erginer, B. and Altug, E. (2012). Design and Implementation of a Hybrid Fuzzy Logic Controller for a Quadrotor VTOL Vehicle. *International Journal of Control, Automation, and Systems*, 10(1) :61–70.
- Escareno, J., Salazar, S., and Lozano, R. (2006). Modeling and Control of a Convertible VTOL Aircraft. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 69–74, San Diego, CA.
- Escareno, J., Stone, H., Sanchez, A., and Lozano, R. (2007). Modeling and Control Strategy for the transition of a Convertible UAV. In *Proceedings of European Control Conference (ECC07)*, pages 120–125, Kos, Greece.
- Eskandar, H., Sadollah, A., Bahreininejad, A., and Hamdi, M. (2012). Water cycle algorithm : A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures*, 110-111 :151–166.
- Fantoni, I. (2007). *Vers l'autonomie des véhicule aériens*. Mémoire d'Habilitation à Diriger des Recherches, École Doctorale de l'Université de Technologie, Compiène.
- Fessi, R. and Bouallègue, S. (2016). Modeling and Optimal LQG Controller Design for a Quadrotor UAV. In *Proceedings of the 3rd International Conference on Automation, Control Engineering and Computer Science*, pages 264–270, Hammamet.
- Fletcher, R. (1981). Practical methods of optimization : Vol. 2 : Constrained optimization. *JOHN WILEY & SONS, Inc.*
- García Carrillo, L., Lozano, R., Pégard, C., and Dzul López, A. (2013). *Quad Rotorcraft Control*. Springer.
- Garriga, L. J. and Soroush, M. (2010). Model predictive control tuning methods : A review. *Industrial and Engineering Chemistry Research*, 49(8) :3505–3515.
-

- Gasso, K. (2000). *Identification des systèmes dynamiques non-linéaires : approche multi-modèle*. Thèse de Doctorat, Vandoeuvre-les-Nancy, INPL.
- Gendreau, M. and Potvin, J. (2010). *Handbook of Metaheuristics*. Springer Verlag, New York.
- Gerardo, F. and Lozano, R. (2013). Transition flight control of the Quad-tilting Rotor Convertible MAV. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 789–784, Atlanta, USA.
- Gerrardo, F. and Lozano, R. (2014). A Nonlinear Control Law for Hover to Level Flight for the Quad Tilt-rotor UAV. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control (IFAC)*, pages 11055–11059, Cap Town, South Africa.
- Granado, E. (2004). *Commande prédictive à base de programmation semi définie*. Thèse de Doctorat, INSA de Toulouse.
- Greco, C., Menga, G., Mosca, E., and Zappa, G. (1984). Performance improvements of self-tuning controllers by multistep horizons : The musmar approach. *Automatica*, 20(5) :681–699.
- Guerra, T., Kruszewski, A., Vermeiren, L., and Tirmant, H. (2006). Conditions of output stabilization for nonlinear models in the takagi–sugeno’s form. *Fuzzy Sets and Systems*, 157(9) :1248–1259.
- Guerra, T. and Vermeiren, L. (2001). Control laws for takagi–sugeno fuzzy models. *Fuzzy sets and systems*, 120(1) :95–108.
- Gupta, R., Kalabić, V., Cairano, S. D., Bloch, A., and Kolmanovsky, I. (2015). Constrained spacecraft attitude control on so (3) using fast nonlinear model predictive control. In *Proceedings of the American Control Conference (ACC)*, pages 2980–2986.
- Hamouda, L. (2015). *Sur la synthèse de commandes prédictives tolérantes aux défauts à base de modèles TS flous*. Thèse de Doctorat, École Nationale d’ingénieurs de Tunis et Université de Rouen.
- Hamouda, L., Ayadi, M., and Langlois, N. (2016). Fuzzy fault tolerant predictive control for a diesel engine air path. *International Journal of Control, Automation and Systems*, 14(2) :443–451.
- Hamouda, L., Bennouna, O., Ayadi, M., and Langlois, N. (2013). Quasis-lpv model predictive reconfigurable control for constrained nonlinear systems. In *Proceedings of the Conference on Control and Fault-Tolerant Systems*, pages 590–595, Nice.
- Hamouda, L., Bennouna, O., Ayadi, M., and Langlois, N. (2014). Fuzzy model predictive reconfigurable control for nonlinear systems subject to actuators faults. In *Proceedings of the International Conference on Automation and Computing*, pages 140–145, Bedfordshire.
-

- Hasanjani, R., Javadi, S., and Nadooshan, R. (2015). Dc motor speed control by self-tuning fuzzy pid algorithm. *Transactions of the Institute of Measurement and Control*, 37(2) :164–176.
- Helicopter, B. (2016). Eagle EYE Pocket Guide. Technical report, www.bellhelicopter.com. Accessed on July 2016.
- Honggui, H. and Junfei, Q. (2014). Nonlinear model-predictive control for industrial processes : An application to wastewater treatment process. *IEEE Transactions on Industrial Electronics*, 61(4) :1970–1982.
- Hu, P., Rong, L., Liang-lin, C., and Li-xian, L. (2011). Multiple swarms multi-objective particle swarm optimization based on decomposition. *Procedia Engineering*, 15 :3371–3375.
- Huang, D. and Nguang, S. (2007). Static output feedback controller design for fuzzy systems : An ILMI approach. *Information Sciences*, 177(14) :3005–3015.
- Huang, Y. and Jadbabaie, A. (1999). Nonlinear h infinity control an enhanced quasi-lpv approach. *IFAC Proceedings Volumes*, 32(2) :2754–2759.
- Ichalal, D. (2009). *Estimation et diagnostic de systèmes non linéaires décrits par un modèle de Takagi-Sugeno*. Thèse de Doctorat, Institut National Polytechnique de Lorraine-INPL.
- Ichalal, D., Marx, B., Ragot, J., and Maquin, D. (2009). Simultaneous state and unknown inputs estimation with pi and pmi observers for takagi sugeno model with unmeasurable premise variables. In *Proceedings of the 17th Mediterranean Conference on Control and Automation*, pages 353–358, Thessaloniki.
- Ikeda, Y., Nakajima, T., and Chida, Y. (2012). Attitude control of spacecraft by nmpe with consideration of singularity avoidance of cmg. In *Proceedings of the 51st IEEE Conference on Decision and Control(CDC)*, pages 1733–1739.
- Instruments, N. (2009). LabVIEW Control Design User Manual. Technical Report 371057G-01, www.ni.com.
- Instruments, N. (2013). NI myRIO-1900 : User Guide and Specifications. Technical Report 376047A-01, www.ni.com.
- Janson, S. and Middendorf, M. (2004). *Workshops on Applications of Evolutionary Computation EvoWorkshops 2004*, chapter A hierarchical particle swarm optimizer for dynamic optimization problems. Springer-Heidelberg, Germany. DOI 10.1007/978-3-540-24653-4-52.
- Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation*, 9(3) :303–317.
-

- Johansen, T. A., Shorten, R., and Murray-Smith, R. (2000). On the interpretation and identification of dynamic takagi-sugeno fuzzy models. *IEEE Transactions on Fuzzy systems*, 8(3) :297–313.
- Johansson, M. (1999). *Piecewise linear control systems*. Thèse de Doctorat, Lund Institute of Technology, Sweden.
- Kalabic, U., Gupta, R., Cairano, S. D., Bloch, A., and Kolmanovsky, I. (2014). Constrained spacecraft attitude control on so (3) using reference governors and nonlinear model predictive control. In *Proceedings of the American Control Conference (ACC)*, pages 5586–5593.
- Kanellakopoulos, I., Kokotovic, P., and Morse, A. (1991). Systematic Design of Adaptive Controllers for Feedback Linearizable Systems. *IEEE Transactions on Automatic Control*, 36 :1241–1253.
- Kar, I. (1999). Design of static output feedback controller for uncertain systems. *Automatica*, 35(1) :169–175.
- Kau, S. W., Lee, H., Yang, C., Lee, C., Hong, L., and Fang, C. (2007). Robust infinity fuzzy static output feedback control of ts fuzzy systems with parametric uncertainties. *Fuzzy sets and systems*, 158(2) :135–146.
- Kendoul, F., Fantoni, I., and Lozano, R. (2005). Modeling and control of a small autonomous aircraft having two tilting rotors. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8144–8149, Seville, Spain.
- Kennedy, J. and Eberhart, R. (1995). A New Optimizer using Particle Swarm Theory. In *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pages 39–43, Nagoya.
- Kerrigan, E. and Maciejowski, J. (2000). *Soft constraints and exact penalty functions in model predictive control*. University of Cambridge.
- Khatoon, S., Gupta, D., and Das, L. (2014). PID & LQR Control for a Quadrotor : modeling and simulation. In *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics*, pages 796–802, New Delhi.
- Kim, E. and Lee, H. (2000). New approaches to relaxed quadratic stability condition of fuzzy control systems. *IEEE Transactions on Fuzzy systems*, 8(5) :523–534.
- Koehl, A. (2012). *Modélisation, observation et commande d’un drone miniature à birotor coaxial*. Thèse de Doctorat, Université de Lorraine.
- Koenemann, J., Prete, A. D., Tassa, Y., Todorov, E., Stasse, O., Bennewitz, M., and Mansard, N. (2015). Whole-body model-predictive control applied to the hrp-2 humanoid. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3346–3351.
-

- Kordestani, J. K., Ranginkaman, A. E., Meybodi, M. R., and Novoa-Hernández, P. (2019). A novel framework for improving multi-population algorithms for dynamic optimization problems : A scheduling approach. *Swarm and Evolutionary Computation*, 44 :788 – 805.
- Krstic, M., Kanellakopoulos, I., and Kokotovic, P. (1995). *Nonlinear and Adaptive Control Design*. John Wiley & Sons, New York.
- Larminat, P. (2013). *Analysis and Control of Linear Systems*. John Wiley & Sons, New York.
- Leith, D. and Leithead, W. (2002). Global reconstruction of nonlinear systems from families of linear systems. *IFAC Proceedings Volumes*, 35(1) :435–440.
- Lin, C., Wang, Q. G., and Lee, T. (2005). Improvement on observer-based h infinie control for t–s fuzzy systems. *Automatica*, 41(9) :1651–1656.
- Lino, Y., Tomida, K., Fujiwara, H., and Takagi, Y. (1993). A New Input-Output Constrained Model Predictive Control with Frequency-domain Tuning Technique and its Application to an Ethylene Plant. In *Proceedings of the International Conference on Industrial Electronics, Control, and Instrumentation*, pages 457–462, Maui HI.
- Lo, J. and Lin, M. (2003). Robust h/sub/spl infin//nonlinear control via fuzzy static output feedback. *IEEE Transactions on Circuits and Systems I : Fundamental Theory and Applications*, 50(11) :1494–1502.
- Lozano, R., editor (2010). *Unmanned Aerial Vehicles : Embedded Control*. John Wiley & Sons, ISTE, UK.
- Luenberger, D. G. and Yinyu, Y. (1984). *Linear and nonlinear programming*, volume 2. Springer.
- Luo, W., Sun, J., Bu, C., and Liang, H. (2016). Species-based particle swarm optimizer enhanced by memory for dynamic optimization. *Applied Soft Computing*, 47 :130 – 140.
- Ma, H., Sh.Shen, Yu, M., Yang, Z., Fei, M., and Zhou, H. (2019). Multi-population techniques in nature inspired optimization algorithms : A comprehensive survey. *Swarm and Evolutionary Computation*, 44 :365 – 387.
- Magni, L., Raimondo, D., and Allgöwer, F., editors (2009). *Nonlinear Model Predictive Control : Towards New Challenging Applications*, chapter Particle Swarm Optimization Based NMPC : An Application to District Heating Networks. Springer Verlag, Berlin Heidelberg.
- Mamboundou, J. and Langlois, N. (2011). Indirect adaptive model predictive control supervised by fuzzy logic. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 2979–2986, Taipei.
-

- Mamboundou, J. and Langlois, N. (2012). Robustness analysis of indirect adaptive model predictive control supervised by fuzzy logic. In *2012 IEEE International Conference on Industrial Technology*, pages 284–291.
- Marcelin, D. (2010). *Commande prédictive généralisée non linéaire à temps continu des systèmes complexes*. Thèse de Doctorat, Université de Rouen.
- menadefense.net (2013). L’incroyable histoire du drone amel. <https://www.menadefense.net/algerie/lincroyable-histoire-du-drone-amel/>. (consulté le 13 octobre 2018).
- Mills, S. (2019). *The Dawn of the Drone*. Casemate Publishers, US. ISBN 978-1-61200-789-2.
- Mirzaei, M., Poulsen, N., and Niemann, H. (2012). Robust model predictive control of a wind turbine. In *Proceedings of the American Control Conference (ACC)*, pages 4393–4398.
- Morère, Y. (2001). *Mise en oeuvre de lois de commande pour les modèles flous de type Takagi-Sugeno*. Thèse de Doctorat, Valenciennes.
- Morrison, R. W. (2004). *Designing Evolutionary Algorithms for Dynamic Environments*. Springer, Germany. DOI 10.1007/978-3-662-06560-0.
- Mousavi, Y. and Alfi, A. (2015). A memetic algorithm applied to trajectory control by tuning of fractional order proportional-integral-derivative controllers. *Applied Soft Computing*, 36 :599–617.
- Munson, K. (1988). *World unmanned aircraft*. Jane’s Information Group, London.
- Munson, K. (1995). *Jane’s Unmanned Aerial Vehicles and Targets*. JANE’S UNMANNED AERIAL VEHICLES AND TARGETS. Jane’s Information Group.
- Nagoudi-Layerle, K. (2011). *Application de la commande prédictive reconfigurable sur le circuit d’air du moteur Diesel*. Thèse de Doctorat, Rouen.
- Nagy, A., Schutz, G. M. G., and Ragot, J. (2010). Modélisation d’un réacteur biologique à l’aide d’une structure multimodèle. méthode analytique d’obtention d’un multimodèle. *Journal Européen des Systèmes Automatisés (JESA)*, 44(4-5) :423–443.
- Nguyen, H. (2012). *Commande sous contraintes pour des systèmes dynamiques incertains : une approche basée sur l’interpolation*. Thèse de Doctorat, Supélec.
- Nguyen, T. T., Yang, S., and Branke, J. (2012). Evolutionary dynamic optimization : A survey of the state of the art. *Swarm and Evolutionary Computation*, 6 :1 – 24.
- Nonami, K. (2007). Prospect and Recent Research & Development for Civil Use Autonomous Unmanned Aircraft as UAV and MAV. *Journal of System Design and Dynamics*, 1 :120–128.
-

- Nonami, K., Kendoul, F., Suzuki, S., Wang, W., and Nakazawa, D. (2010). *Autonomous Flying Robots : Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer, Tokyo.
- Ohtake, H., Tanaka, K., and Wang, O. (2003). Fuzzy modeling via sector nonlinearity concept. *Integrated Computer-Aided Engineering*, 10(4) :333–341.
- Oner, K., Cetinsoy, E., Unel, M., Aksit, M., Kandemir, I., and Gulez, K. (2008). Dynamic model and control of a New Quadrotor Unmanned Aerial Vehicle with Tilt Wing Mechanism. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 2(9) :12–17.
- Oner, K., Unel, M., Cetinsoy, E., Aksit, M., Sirimoglu, E., Gulez, K., and Hancer, C. (2012). Mathematical modeling and vertical flight control of a Tilt-Wing UAV. *Turkish journal of electrical engineering and computer sciences*, 20 :149–157.
- Oner, K. T. (2009). Modeling and control of a new Unmanned Aerial Vehicle (SUAVI) with Tilt Wing Mechanism. Mémoire de Mastère, Université de Sabanci, Istanbul.
- Orjuela, R., B. Marx, B., Ragot, J., and Maquin, D. (2008). State estimation for nonlinear systems using a decoupled multiple model. *International Journal of Modelling, Identification and Control*, 4(1) :59–67.
- Orjuela, R., B. Marx, B., Ragot, J., and Maquin, D. (2009). Observer design for nonlinear systems described by multiple models. *IFAC Proceedings Volumes*, 42(8) :18–23.
- Osborne, S. (2007). Transitions between hover and level flight for a tailsitter UAV. Mémoire de Mastère, Brigham Young, Provo, UT.
- Oudghiri, M. (2008). *Commande multi-modèles tolérante aux défauts : Application au contrôle de la dynamique d'un véhicule automobile*. Thèse de Doctorat, Université de Picardie Jules Verne.
- Oudghiri, M., Chadli, M., and Hajjaji, A. E. (2007). One-step procedure for robust output h8 fuzzy control. In *Proceedings of the Mediterranean Conference on Control & Automation*, pages 1–6.
- Ozsoydan, F. B. and Baykasoglu, A. (2019). Quantum firefly swarms for multimodal dynamic optimization problems. *Expert Systems with Applications*, 115 :189 – 199.
- Papachristos, C., Alexis, K., and Tzes, A. (2013). Hybrid Model Predictive Flight Mode Conversion Control of Unmanned Quad-Tilt Rotors. In *Proceedings of the European Control Conference (ECC)*, pages 17933–1798, Zürich, Switzerland.
- Parrott, D. and Li, X. (2004). A particle swarm model for tracking multiple peaks in a dynamic environment using speciation. In *Proceedings of the 2004 Congress on Evolutionary Computation, CEC 2004*, pages 98–103, Portland.
-

- Piotr, S., Bogumil, U., AbdelMeguid, H., and Paluszczyszyn, D. (2010). Model predictive control for energy and leakage management in water distribution systems. In *Proceedings of the UKACC International Conference on Control*. IET.
- Pirouzmand, F. and Ghahramani, N. (2013). Robust model predictive control based on mras for satellite attitude control system. In *Proceedings of the 3rd International Conference on Control, Instrumentation and Automation (ICCIA)*, pages 53–58.
- Poinsot, D. (2008). *Commande d'un drone en vue de la conversion vol rapide-vol stationnaire*. Thèse de Doctorat, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse.
- Rakotomamonjy, T. (2006). *Modélisation et contrôle du vol d'un micro drone à ailes battantes*. Thèse de Doctorat, Université Paul Cézanne, Marseille.
- Richalet, J., Ata-Doss, S. E., Arber, C., Kuntze, H., Jacubasch, A., and Schill, W. (1987). Predictive functional control-application to fast and accurate robots. *IFAC Proceedings Volumes*, 20(5) :251–258.
- Richalet, J., Rault, A., Testud, J., and Papon, J. (1978). Model predictive heuristic control. *Automatica*, 14(5) :413–428.
- Rowe, C. and Maciejowski, J. (2000). Tuning MPC using H-infinity Loop Shaping. In *Proceedings of the 2000 American Control Conference*, pages 1332–1336, Chicago IL.
- Safran, E. (2014). Sperwer : le drone tactique polyvalent. <https://www.safran-electronics-defense.com/fr/terrestre/drones-tactiques/systeme-sperwer>. (consulté le 13 octobre 2018).
- Sala, A. (2009). On the conservativeness of fuzzy and fuzzy-polynomial control of nonlinear systems. *Annual Reviews in Control*, 33(1) :48–58.
- Sandou, G. (2009). Metaheuristic Strategy for the Hierarchical Predictive Control of Large Scale Energy Networks. *Journal of Control Engineering and Applied Informatics*, 11 :32–40.
- Shi, Y. and Eberhart, R. (1999). Empirical Study of Particle Swarm Optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1945–1950, Washington.
- Siarry, P. and Michalewicz, Z., editors (2008). *Advances in Metaheuristics for Hard Optimization*. Springer Verlag, Berlin.
- Sinthipsomboon, K., Hunsacharoonroj, I., Khedari, J., and Pongaen, W. (2011). A Hybrid of Fuzzy and Fuzzy Self-tuning PID Controller for Servo Electro Hydraulic System. In *Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications*, pages 220–225, Beijing.
-

- Snyder, D. (2000). The Quad Tiltrotor : Its Beginning and Evolution. In *Proceedings of the 56th Annual Forum, American Helicopter Society*, Virginia Beach, Virginia.
- Soeterboek, R. (1992). *Predictive Control : A unified approach*. Prentice-Hall, Inc.
- Sun, J., Lai, C.-H., and Wu, X.-J. (2012). *Particle Swarm Optimisation : Classical and Quantum Perspectives*. CRC Press, Taylor & Francis Group, New York.
- Sun, M. P. and Liu, J. J. (2017). Tracking Control of a Quad-rotor UAV Based on TS Fuzzy Model. In *Proceedings of the 36th Chinese Control Conference*, Dalian.
- S.Yang, Ong, Y. S., and Jin, Y. (2007). *Evolutionary Computation in Dynamic and Uncertain Environments*. Springer-Berlin, Germany. DOI 10.1007/978-3-540-49774-5.
- Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions. Systems, Man and Cybernetics*, 15 :116–132.
- Tanaka, K., Ikeda, T., and Wang, H. (1996). Robust stabilisation of uncertain nonlinear systems via fuzzy control : quadratic stability, hinf control theory and lms. *IEEE Transactions on Fuzzy Systems*, 4(1) :1–12.
- Tanaka, K., Ikeda, T., and Wang, H. (1998a). Fuzzy regulators and fuzzy observers : relaxed stability conditions and lmi-based designs. *IEEE Transactions on fuzzy systems*, 6(2) :250–265.
- Tanaka, K., Nishimura, M., and Wang, H. (1998b). Multi-objective fuzzy control of high rise/high speed elevators using lmis. In *Proceedings of the American Control Conference, 1998. Proceedings of the 1998*, volume 6, pages 3450–3454.
- Tanaka, K. and Sugeno, M. (1992). Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems*, 45 :135–156.
- Tanaka, K., Tsuyoshi, H., and Wang, H. (2003). A multiple lyapunov function approach to stabilization of fuzzy control systems. *IEEE Transactions on fuzzy systems*, 11(4) :582–589.
- Tanaka, K. and Wang, O. (2001). *Fuzzy control system design and analysis. a Linear Matrix Inequality approach*. John Wiley and Sons Inc.
- Tran, Q. N., Scholten, J., Ozkan, L., and Backx, A. (2014). A model-free approach for auto-tuning of model predictive control. *IFAC Proceedings Volumes*, 47(3) :2189–2194. 19th IFAC World Congress.
- Turki, M., Benkhoud, K., Langlois, N., and Yassine :, A. (2018a). Analytical Tuning of a TSMPC Dedicated to Nonlinear MIMO Systems. In *Proceedings of the 2018 European Control Conference*, pages 1367–1372, Limassol.
-

- Turki, M., Oukkacha, I., Langlois, N., Yassine, A., Camara, M., and Dakyo, B. (2018b). A New Tuning Approach for MPC Applied to a Disturbed DC Motor. In *5th International Conference on Control, Decision and Information Technologies (CoDIT)*, Paris.
- Wang, H., Tanaka, K., and Griffin, M. (1996). An approach to fuzzy control of nonlinear systems : Stability and design issues. *IEEE transactions on fuzzy systems*, 4(1) :14–23.
- Wang, L. (2009). *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media.
- Wang, X., Gao, X.-Z., and Zenger, K. (2015). *Identification et commande numérique des procédés industriels*. Springer Briefs in Computational Intelligence.
- Weinmann, A. (2012). *Uncertain models and robust control*. Springer Science & Business Media.
- Wills, G. and Heath, P. (2005). Application of barrier function based model predictive control to an edible oil refining process. *Journal of Process Control*, 15(2) :183–200.
- Wu, H. N. (2008). An ilmi approach to robust h2 static output feedback fuzzy control for uncertain discrete-time nonlinear systems. *Automatica*, 44(9) :2333–2339.
- Wudhichai, A., Nguang, S., and Shi, P. (2004). H infinity output feedback control design for uncertain fuzzy singularly perturbed systems : an lmi approach. *Automatica*, 40(12) :2147–2152.
- Xiaodong, L. and Qingling, Z. (2003). New approaches to h infinie controller designs based on fuzzy observers for ts fuzzy systems via lmi. *Automatica*, 39(9) :1571–1582.
- Xiaohong, Y., Shaoyuan, L., Ning, L., Yi, Z., and Wenjian, C. (2014). Energy-saving-oriented control strategy for vapor compression refrigeration cycle systems. In *Proceedings of the IEEE 9th Conference on Industrial Electronics and Applications (ICIEA)*, pages 1561–1566.
- Xinchao, Z. (2010). A Perturbed Particle Swarm Algorithm for Numerical Optimization. *Applied Soft Computing*, 10 :119–124.
- Yang, S. and Yao, X. (2013). *Evolutionary Computation for Dynamic Optimization Problems*. Springer-Berlin, Germany. DOI 10.1007/978-3-642-38416-5.
- Yang, X. (2010). *Engineering Optimization : an introduction with metaheuristic applications*. John Wiley & Sons, New Jersey.
- Yoneyama, J. (2008). H infinity output feedback control for fuzzy systems with immeasurable premise variables : Discrete-time case. *Applied Soft Computing*, 8(2) :949–958.
- Youssef, T., Chadli, M., Karimi, H., and Wangd, R. (2017). Actuator and sensor faults estimation based on proportional integral observer for ts fuzzy model. *Journal of the Franklin Institute*, 354(6) :2524–2542.
-

-
- Zhang, J., Li, R., and Zhang, P. (2001). Stability analysis and systematic design of fuzzy control systems. *Fuzzy sets and Systems*, 120(1) :65–72.
- Zhao, Z.-Y., Tomizuka, M., and Isaka, S. (1993). Fuzzy Gain Scheduling of PID Controllers. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(5) :1392–1398.
- Zhou, J. and Wen, C. (2008). *Adaptive Backstepping Control of Uncertain Systems*. Lecture Notes in Control and Information Sciences 372, Springer, New York.
- Zhou, K., Doyle, J., and Glover, K. (1996). *Robust and Optimal Control*. Prentice Hall, New Jersey.
-