الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Département : Génie électrique Et Informatique Industrielle

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en automatique

Spécialité
Automatique et informatique industrielle

Contribution à l'Optimisation Méta-Heuristique d'une Commande Robuste. Application à un Simulateur d'Hélicoptère

Réalisé par

Messah Ayoub
Ahmed Messaoud Walid

Présenté et soutenu publiquement le (04/07/2022)

Composition du Jury :

Président	Mme. BELHACHAT Faiza,	M.C.A	ENST
Promoteur	M. REZOUG Amar,	M.C.A	ENST
Co-promoteur	M. BAIZID Khlifa,	P.H.D	IIT, Italie
Examinateur	M .CHOUTRI Khair-Eddine,	M.C.B	USD-Blida
Examinateur	Mme. REBAI Karima,	M.C.B	ENST

Année universitaire 2021 –2022

Remerciements

Avant d'entamer la présentation de ce mémoire, nous tenons à remercier du fond du cœur le Bon Dieu de nous avoir accordé toute la patience, le courage, la volonté et la motivation qui nous ont permis d'achever ce travail.

Nous tenons tous d’abord à remercier nos promoteurs Monsieur REZOUG AMAR et Monsieur BAIZID KHELIFA pour leurs disponibilités, leurs confiances ainsi qu’aux précieux conseils pour la réalisation de ce projet de fin d’étude.

Nous remercions nos familles et nos amis qui ont toujours été là pour nous, pour le soutien inconditionnel et les encouragements qu’ils nous ont apportés.

Nous remercions nos enseignants de l'école nationale supérieure de Technologie pour leurs efforts voués à nous transmettre leur savoir, en particulier ceux du département Génie électrique et informatique industrielle.

Nous tenons à remercier vivement toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce de ce projet de fin d’étude.

Table des matières

LISTE DES FIGURES III

LISTE DES TABLEAUX VI

LISTE DES ABBREVIATIONS VII

INTRODUCTION GENERALE 1

CHAPITRE I : GENERALITES SUR LES HELICOPTERES DRONES ET MODELISATION DU SIMULATEUR AERO-QUANSER 3

I.1. INTRODUCTION 3
I.2. GENERALITES SUR LES HELICOPTERES DRONES 3
I.2.1. Les types des drones 3
I.2.2. Missions des hélicoptères drones 4
I.2.3. Principe de fonctionnement d'un hélicoptère 4
I.2.4. Forces agissantes sur un hélicoptère en vol 5
I.2.5. Modes de vol 6
I.3. DESCRIPTION DU AERO QUANSER 7
I.3.1. Moteur courant continue 8
I.3.2. Encodeurs 9
I.3.4. Panneau d'interface QFLEX 2 10
I.4. MODELE DYNAMIQUE DU SIMULATEUR AERO QUANSER 10
I.4.1. Les équations de mouvement 12
I.4.2. La représentation d'état 13
I.5. CONCLUSION 14

CHAPITRE II : NOTION DE BASE SUR LA COMMANDE ROBUSTE ET L'OPTIMISATION META-HEURISTIQUE 15

II.2. INTRODUCTION 15
II.2. THEORIE DE COMMANDE 15
II.2.1. Commande par PID 15
II.2.1.1 Action proportionnelle 15
II.2.1.2 Action intégrale 16
II.2.1.3. Action dérivée 16
II.2.1.4. Les effets du reglage indémpondant des parametres P,I et D 16
II.2.2. La commande par mode de glissement 17
II.2.2.1. Principe de la commande par mode de glissement 17
II.2.2.2. Choix de la surface de glissement 17
II.2.2.3. Condition de stabilité et d'attraction 18
II.2.2.4. Synthèse de la commande équivalente 19
II.2.2.5. Phénomène du broutement 20
II.2.3. Variantes de la commande par mode de glissement 21
II.2.3.1 Terminal sliding mode control (TSMC) 21
II.2.3.2. Nonsingular terminal sliding mode control (NTSMC) 22
II.2.3.3. Algorithme de super twisting (STW) 22
II.2.3. Commande adaptative 23
II.2.3.1. Commande adaptative indirecte 24
II.2.3.1. Commande adaptative directe 24
II.3. OPTIMISATION META-HEURISTIQUE 25
II.3.1. Grey Wolf Optimizer (GWO) 25
II.3.2. Wale optimization algorithm (WOA) 27
II.3.3. Salp swarm algorithm (SSA) 30
II.3.4. Ant Lion Optimizer (ALO) 32
II.3.5. Choix de la fonction objective 35
II.4. CONCLUSION 36

CHAPITRE III : SYNTHESE DES LOIS DE COMMANDE 37

III.1. INTRODUCTION 37
III.2. STRUCTURE DE COMMANDE PAR PID 37
III.3. SYNTHESE DES LOIS DE COMMANDE PAR MODE DE GLISSEMENT 38
III.3.1. Commande par mode glissant classique (SMC) 38
III.3.2. Commande par NTSMC 40
III.3.3. Commande par l'algorithme Super Twisting (STW) 41
III.3.4. Commande par nonsingular terminal super twisting (NTSTW) 42
III.4. SYNTHESE D'UNE LOI DE COMMANDE ADAPTATIVE 43
III.4.1. Preuve de la loi d'adaptation 45
III.4.2. Structure de la commande 46
III.4.3. Preuve de stabilité 47
III.5. CONCLUSION 51

CHAPITRE IV : SIMULATION ET DISCUSSIONS 52

VI.1. INTRODUCTION 52
IV.1. RESULTATS D'OPTIMISATION DES PARAMETRES DES DIFFERENTES LOIS DE COMMANDE 52
IV.1.1. Résultats de l'optimisation des paramètres de la commande PID 55
IV.1.2. Résultats de l'optimisation des paramètres de la commande SMC 57
IV.1.3. Résultats de l'optimisation des paramètres de commande NTSMC 60
IV.1.4. Résultats de l'optimisation des paramètres de la commande STW 62
IV.1.5. Résultats de l'optimisation des paramètres de la commande NTSTW 64
V.1.5 Comparaison entre les algorithmes d'optimisations 66
IV.2. COMPARAISON ENTRE LES DIFFERENTS TECHNIQUES DE COMMANDES 68
IV.3. RESULTAT DE SIMULATION DE LA COMMANDE ADAPTATIVE-OPTIMALE (ANTSWO) : 73
IV.4. CONCLUSION 82

CHAPITRE V : VISUALISATION DES RESULTATS DE SIMULATION SUR L'ENVIRONNEMENT ROS 83

V.1. INTRODUCTION 83
V.2. ARCHITECTURE DE ROS 83
V.3. UTILISATION DE RVIZ POUR LA VISUALISATION DU MODELE DU ROBOT AERO-QUANSER : 88
V.4. SIMULATION SUR LA PLATEFORME GAZEBO 91
V.5. CONCLUSION : 94

CONCLUSION GENERALE 95

BIBLIOGRAPHIE 96

ANNEXE 98

Liste des figures

Chapitre I

Figure I. 1 Classification des UAV basée sur les ailes et les rotors 3
Figure I. 5 Principe de fonctionnement d'un hélicoptère 5
Figure I. 6 Les forces agissantes sur un hélicoptère en vol.....6
Figure I. 7 Simulateur de vol Aero Quanser.....7
Figure I. 5 Schéma du circuit d'induit du moteur de l'un des propulseurs.....8
Figure I. 6 Encodeur rotatif 9
Figure I. 7 Les signaux générés par l'encodeur.....10
Figure I. 8 Les options du panneau d'interface QFLEX 2 10
Figure I. 9 Schéma synoptique d'une configuration d'hélicoptère à 2 degrés de liberté 11

Chapitre II

Figure II. 1 Structure parallèle d'un régulateur PID.....15
Figure II. 2 Surface de glissement.....17
Figure II. 3 Les signaux ueq et ud 19
Figure II. 4 Phénomène de broutement 20
Figure II. 5 Surface de glissement dans le cas de l'NTSMC 21
Figure II. 6 Plan de phase de l'évaluation de surface de glissement et sa dérivée 23
Figure II. 7 Schéma synoptique de la commande adaptative indirecte 24
Figure II. 8 Schéma synoptique de la commande adaptative directe 24
Figure II. 9 Comportement de chasse des loups gris : (A) chasser, approcher et traquer les proies (B-D) poursuivre, harceler et encercler (E) situation stationnaire et attaque.....25
Figure II. 10 (a) attaque de la proie (b) recherche d'autre proies 27
Figure II. 11 Pseudocode du GWO 27
Figure II. 12 Baleine à bosse en chasse.....28
Figure II. 13 La spirale de mise à jour des positions 29
Figure II. 14 Pseudocode du WOA 30
Figure II. 15 Une chaîne de salpes 31
Figure II. 16 Pseudocode du SSA 32
Figure II. 17 La larve de fourmilion et ses pièges à fourmis en forme de cône 33
Figure II. 18 Pseudocode d'ALO 35

Chapitre III

Figure III. 1 Structure .de la commande PID 37
Figure III. 2 Structure de la commande SMC 39
Figure III. 3 Structure de la commande NTSMC.....41
Figure III. 4 Structure de la commande STW 42
Figure III. 5 Structure de la commande NTSTW.....43
Figure III. 6 Structure de la commande ANTSTWO 44

Chapitre IV

Figure IV. 1 L'évolution de la fonction objective pour les meilleurs essais 55
Figure IV. 2 Réponses des angles de (a) tangage et (b) lacet en mode de régulation à l'aide de la commande PID optimisée par les algorithmes de GOW, WOA, SSA et ALO. 56
Figure IV. 3 Les signaux de commande générés par PID (a) tangage et (b) lacet 57
Figure IV. 4 Réponses des angles de (a) tangage et (b) lacet, en mode de régulation à l'aide de la commande SMC optimisée par les algorithmes de GOW, WOA, SSA et ALO. 58
Figure IV. 5 Les signaux générés par la commande SMC (a) tangage et (b) lacet 59
Figure IV. 6 Réponses des angles de (a) tangage et (b) lacet, en mode de régulation à l'aide de la commande NTSMC optimisée par les algorithmes de GOW, WOA, SSA et ALO. 61
Figure IV. 7 Les signaux de commande générés par NTSMC (a) tangage (b) lacet 62
Figure IV. 8 Réponses des angles de (a) tangage et (b) lacet en mode de régulation à l'aide de la commande STW optimisée par les algorithmes de GOW, WOA, SSA, ALO. 63
Figure IV. 9 Les signaux de commande générés par STW (a) tangage (b) lacet 64
Figure IV. 10 Réponses des angles de (a) tangage et (b) lacet, en mode de régulation à l'aide de la commande NTSTW optimisée par les algorithmes de GOW, WOA, SSA et ALO. 65
Figure IV. 11 Les signaux de commande générés par l'NTSTW (a) tangage (b) lacet 66
Figure IV. 12 Les réponses des angles de (a) tangage et (b) lacet, pour les commandes PID, SMC, NTSMC, STW et NTSTW optimisées 68
Figure IV. 13 Les signaux de commande des angles de (a) tangage et (b) lacet pour les commandes PID, SMC, NTSMC, STW, NTSTW optimisées. 69
Figure IV. 14 Les surfaces de glissements des angles de (a) tangage et de (b) lacet, pour les commandes SMC, NTSMC, STW et NTSTW optimisées 70
Figure IV. 15 Plan de phase (a) tangage et (b) lacet, surface et dérivée de surface (c) tangage et (d) lacet. 71
Figure IV. 16 Réponse indicielle du système (a) tangage (b) lacet dans le cas des commandes ANTSTWO et NTSTW 74
Figure IV. 17 Signaux de commande de ANTSTWO et NTSTW (a) tangage (b) lacet 75
Figure IV. 18 Evolution des gains adaptatifs de la commande ANTSTWO (Rejet de perturbation en mode régulation) 75
Figure IV. 19 Réponse en poursuite du système en l'application des commandes ANTSTWO et NTSTW (a) tangage (b) lacet 76
Figure IV. 20 Signaux de commande en poursuite (a) tangage (b) lacet 77
Figure IV. 21 Evolution des gains adaptatifs de la commande ANTSTWO (signale sinusoïdal) 78
Figure IV. 22 Réponse du système dans le cas d'un profil de vitesse (a) tangage et (b) lacet 79
Figure IV. 23 Signaux de la commande dans le cas du profil de vitesse (a) tangage (b) lacet 80
Figure IV. 24 Evolution des gains adaptatifs de la commande ANTSTWO (profile de vitesse) 81

Chapitre V

Figure V. 1 Schéma descriptif de l'architecture de communication publish/subscribe pour un seul topic 83
Figure V. 2 Joint revolute qui relie les Link Child et Link Parent 85
Figure V. 3 Exemple de la description URDF d'une partie de l'Aero Quanser 87
Figure V. 4 Assemblage du robot Aero_Quanser à l'aide de SolidWorks 87
Figure V. 5 Interface du plugin URDF exportée démontrant l'architecture arbre 88
Figure V. 6 Fichier launch pour le lancement d'Rviz 89
Figure V. 7 Principaux nodes et topics utilisés 89
Figure V. 8 Outil de commande joint_state_publisher_gui 90

Figure V. 9 Simulation sur l'environnement Rviz 90
Figure V. 10 Configuration de l'interface de transmission des joint yaw 91
Figure V. 11 Gazebo-Ros-control plugin code 92
Figure V. 12 Code définissant le type de contrôleur, le nom des joints et certains paramètres. 92
Figure V. 13 Méthode de commande.....93
Figure V. 14 Shéma bloc de la commande ANTSTWO dans le cas de l'application sur ROS 93
Figure V. 15 Simulation en condition réelle du robot sur Gazebo (a) position initiale et (b) position finale 94

Liste des tableaux

Tableau 1 Paramètres de Aero Quanser....12
Tableau 2 Les effets du réglage indépendant des paramètres P,I et D....16
Tableau 3 Les valeurs de la fonction objective, cas de l'optimisation des paramètres du PID....53
Tableau 4 Les valeurs de la fonction objective, cas de l'optimisation des paramètres de SMC....53
Tableau 5 Les valeurs de la fonction objective, cas de l'optimisation des paramètres de NTSMC....53
Tableau 6 Les valeurs de la fonction objective, cas de l'optimisation des paramètres de STW....53
Tableau 7 les valeurs de la fonction objective, cas de l'optimisation des paramètres de NTSTW....53
Tableau 8 Les paramètres optimaux de la commande PID....55
Tableau 9 Les paramètres optimaux de la commande SMC....57
Tableau 10 Les paramètres optimaux de commande NTSMC....60
Tableau 11 Les paramètres optimaux de la commande STW....62
Tableau 12 Les paramètres optimaux de la commande NTSTW....65
Tableau 13 Classement des algorithmes d'optimisation selon la valeur minimale de la fonction objective....66
Tableau 14 Classement des algorithmes d'optimisation selon le min de moyenne....67
Tableau 15 Comparaison entre les différentes lois de commande....72
Tableau 16 Tableau comparatif des performances des commandes NTSTW et ANTSTWO....75
Tableau 17 Etude comparative entre les commandes appliquées en poursuite de trajectoire (sinusoïdale)....78
Tableau 18 Etude comparative entre les commandes appliquées en poursuite de trajectoire (profile de vitesse)....81

Liste des abbreviations

GWO: Gray wolf optimizer

WOA: Whale optimization

ALO: Ant lion optimization

SSA: Salp swarm algorithm

DOF: Degree of freedom

UAV: Unmanned Aerial Vehicle

MIMO: Multiple-Input Multiple-Output

VTOL: Vertical take-off and landing

CG: center of gravity

IMU: Inertial measuring unit

LED: light-emitting diode

ITAE integral time absolute error

ISE integral time square error

ISCO: integral square command output

ROS: Robot Operating System

URDF: Unified Robots Description Format

SW: SolidWorks

CAO: Construction assisté par ordinateur

Introduction Générale

L'automatisation est un besoin qui se généralise dans tous les secteurs de l'industrie moderne. Elle permet de contrôler un système physique par de très hautes performances avec une dépense minimale d'énergie et de la matière première. Les avancées récentes de l'automatique sont dues aux progrès techniques réalisés en informatique industrielle, des moyens de calculs puissants et de la résolution de problèmes mathématiques complexes. Parmi les objectifs de l'automatique, l'élaboration de lois de commande simples et performantes pour conférer aux systèmes physiques des propriétés qu'ils n'avaient pas naturellement ou renforcer des propriétés qui existaient déjà. Par conséquent, différentes lois de commande et d'optimisation ont été développées pour les systèmes non linéaires.

L'élaboration (analyse et synthèse) de lois de commandes des systèmes a fait l'objet de nombreux travaux de recherche. Ces dernières peuvent être issues de la description mathématique d'un phénomène physique ou de l'expérience permettant de caractériser son comportement entrée-sortie. On peut noter qu'un modèle n'est réellement qu'une approximation de la réalité, et la commande désirée doit évidemment assurer des performances satisfaisantes, en prenant en considération les avantages, inconvénients, limites et contraintes de ces différentes techniques tout en s'appuyant sur un ensemble de critères :

- Mise en œuvre, nombre de paramètres à régler, facilité de réglage,
- Qualité du contrôle (l'asservissement) : précision, rapidité et stabilité,
- Robustesse (vis-à-vis des incertitudes paramétriques, des perturbations externes ainsi que les erreurs de modélisation),
- Point de vue énergétique : tensions, courants, puissance dissipée, instantanées, énergie consommée.

Certaines de ces méthodes ont déjà montré leurs efficacités dans des applications d'aéronautiques, notamment sur des avions et des hélicoptères. Cependant, avec le développement de l'ingénierie et de la science les systèmes deviennent de plus en plus complexes, les lois de commande deviennent elles-mêmes complexes et parfois difficiles à mettre en œuvre. Pour cela, des prototypes (ou simulateurs) ont été conçus, dans le but de tester de nouvelles lois de commande, qui ne peuvent être implémentées directement sur les systèmes réels.

Le simulateur de vol d'hélicoptère Aero Quanser [1] est l'un des simulateurs d'hélicoptère les plus récent. Ce simulateur est multi variables à deux entrées et deux sorties, caractérisé par des non linéarités fortes, un couplage complexe et une dynamique instable en boucle ouverte [2]. Ces propriétés vont permettre aux chercheurs de valider leurs lois de commande développées.

La synthèse d'une commande robuste pour ces systèmes est très importante pour réaliser des tâches efficacement. Cependant, lors de la modélisation, les paramètres des systèmes sont considérés statiques bien que, ces derniers peuvent varier par rapport au temps (perturbation interne) et l'environnement dans lequel il sera utilisé (perturbation externe). Mais aussi, le choix des paramètres de la commande est un problème majeur en automatique, puisqu'ils affectent grandement la robustesse et les performances de la commande. Afin d'y remédier à ce problème, nous avons fait appel aux algorithmes d'optimisation méta-heuristique pour trouver des paramètres optimaux pour les commandes synthétisées.

L'application d'algorithmes bio-inspirés pour l'optimisation de la commande robuste est une solution récente. Ces algorithmes sont un outil utile en raison de leurs capacités à résoudre des problèmes non linéaires. Parmi les méthodes d'optimisation les plus utilisées, nous pouvons trouver : les algorithmes génétiques (GA) [3], l'optimisation des colonies de fourmis (ACO) [4], l'optimisation des essaims de particules (PSO) [5], l'optimisation des colonies d'abeilles (BCO) [6], etc.

Ce projet de fin d'études est consacré à la modélisation analytique régissant la structure d'un drone, le modèle élaboré est obtenu en utilisant les lois de la physique. Les travaux réalisés portent essentiellement sur l'application d'une commande adaptative, robuste et optimale sur le système Aero-Quanser tout en comparant entre de différents algorithmes d'optimisation et les commandes, ceci pour avoir les meilleures performances et mettre en place une simulation 3D du robot pour le visualiser et suivre son comportement en condition réelle.

Les contributions de ce projet de fin d'étude sont résumées en quatre points :

- Comparaison entre les quatre algorithmes d'optimisation bio-inespéré à savoir, GWO [7], WOA [8], ALO [9] et SSA [10].
- Optimisation des commandes par les algorithmes cités dans le premier point.
- Développement d'une commande adaptative, robuste et optimisée, appelée « commande nonsingular terminal supertwisting adaptive optimale » ANTSTWO. Cette commande est obtenue en deux phases, premièrement une optimisation en se basant sur des algorithmes cités ci-dessus, puis une adaptation des gains en utilisant la théorie de Lyapunov.
- Visualisation du comportement du robot sur environnement virtuel réaliste à la base de ROS-GAZEBO.

Nous commencerons au chapitre 1 par l'analyse du robot Aero-Quanser et la modélisation de son comportement dynamique, ensuite, lors du Chapitre 2 nous présenterons la théorie de commandes et les algorithmes d'optimisation utilisés. Le chapitre 3 comprend la synthèse des commandes puis, dans le chapitre 4, nous les appliqueront en simulation sur l'Aero quanser. Afin d'avoir un bon rendu sur la réponse du système aux différentes commandes, dans Chapitre 5, nous visualiserons le comportement du robot sur un environnement virtuel en utilisant ROS-GAZEBO.

Chapitre I : Généralités sur les hélicoptères drones et Modélisation du simulateur Aero-Quanser

I.1. Introduction

Un simulateur à deux degrés de liberté (2-DOF) est un exemple d'UAV couramment utilisé comme une plateforme expérimentale de laboratoire, Pour tester pratiquement l'efficacité les lois de commandes proposées qui peuvent être appliquées sur un hélicoptère réel.

Parmi ces plates-formes à double rotor, on trouve Aero Quanser qui est un système MIMO non linéaire et instable qui présente une dynamique couplée entre le rotor principal et le rotor de queue [2].

Dans ce chapitre, nous allons d'abord présenter les hélicoptères et leurs principes de fonctionnement ainsi que leurs utilisations dans des applications pratiques. Puis, décrire les caractéristiques techniques de l'hélicoptère à deux degrés de liberté Aero Quanser. Enfin développer un modèle mathématique qui décrit la dynamique du système.

I.2. Généralités sur les hélicoptères drones

I.2.1. Les types des drones

Les drones sont classés en fonction de divers paramètres. Ils ont été classés en fonction du poids, de l'altitude et de la portée, des ailes et des rotors et de leur application [11]. Il n'existe pas de norme de classification unique pour les drones.

La figure I.1 illustre les divers types de drones classés en fonction de leurs ailes et de leurs rotors

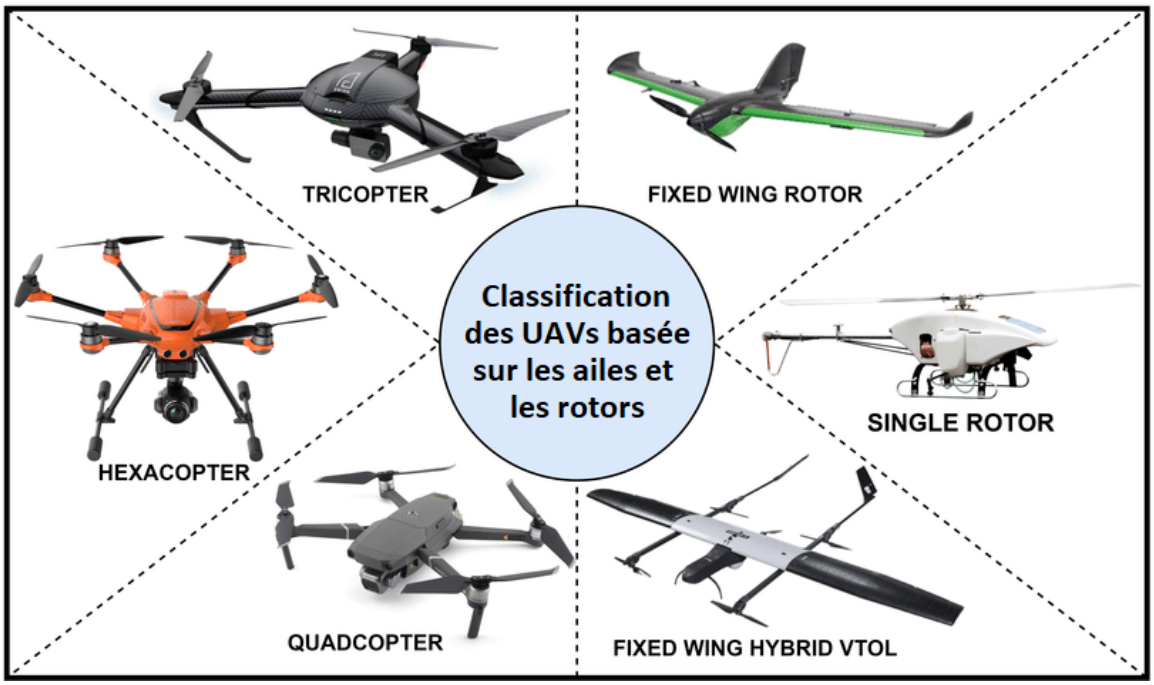


Figure I. 1 Classification des UAV basée sur les ailes et les rotors [11]

- Ailes fixes : drones qui ressemblent à une conception d'avion avec des ailes fixes.
- Rotor unique : drones qui ressemblent à un hélicoptère avec un rotor principal et un autre petit à la queue.

- Multi-rotor : drones qui ont plus d'un rotor. Les plus courants sont les tricoptères, les quadcoptères, les hexacoptères et les octacoptères.
- VTOL hybride à ailes fixes : drones hybrides avec un temps de vol plus long. Ils ont la stabilité des drones à ailes fixes ainsi que la capacité de planer, de décoller et d'atterrir verticalement. Ici, VTOL fait référence au décollage et à l'atterrissage verticaux.

I.2.2. Missions des hélicoptères drones

Voici les différentes missions pour les drones hélicoptères [12] :

- Ils sont rapides à déployer lors des missions de recherche et de sauvetage, ce qui permet un balayage rapide de la zone de recherche.
- Ils peuvent facilement être déployés dans des conditions atmosphériques qui empêcheraient normalement la recherche et le sauvetage par l'homme.
- Les hélicoptères drones sont utilisés dans le secteur militaire pour patrouiller et surveiller de vastes zones frontalières.
- Ils sont utilisés dans les missions d'inspections de grandes structures telles que des ponts et des barrages de manière rentable.
- Les hélicoptères drones peuvent construire des cartes topologiques plus précises que l'avion, parce que ils peuvent voler près de la terre et dans des secteurs très petits et contraints tout en portant des appareils photographiques ou des sondes pour construire des cartes en 3D de haute résolution et fortement détaillées.

I.2.3. Principe de fonctionnement d'un hélicoptère

Les pales du rotor principal tournent autour de l'axe du disque du rotor pour générer une poussée d'élévation suffisante pour contrer le poids de l'hélicoptère. Le fuselage comprend une poutre située à l'arrière, sur laquelle est montée un rotor de queue.

Le rotor de queue, également appelé rotor anti-couple en raison de sa fonction, celui-ci permet de compenser le moment due à la rotation du rotor principal selon son propre axe. En d'autres termes, il permet de contrôler la position de l'angle de lacet afin d'empêcher l'hélicoptère de tourner indéfiniment sur lui-même.

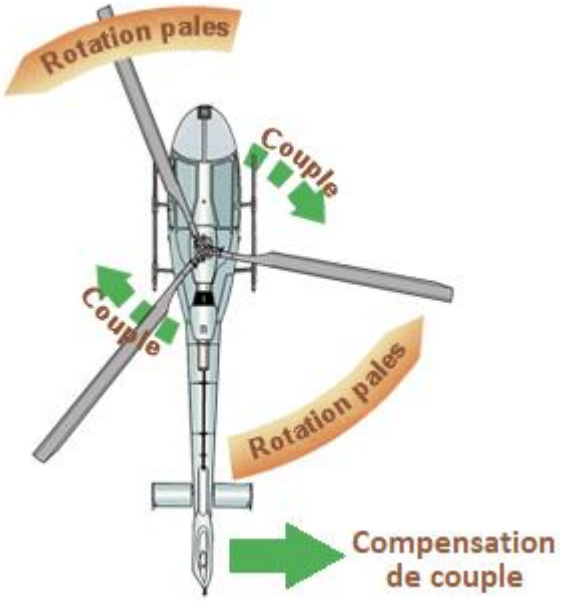


Figure I. 2 Principe de fonctionnement d'un hélicoptère [13]

I.2.4. Forces agissantes sur un hélicoptère en vol

Les forces agissant sur l'hélicoptère pendant le vol sont [13] :

$\overrightarrow{T_u}$: est La force de poussée ou la force d'avancement, produite par l'hélice principale. Elle s'oppose pour surmonter la force de la traînée $\overrightarrow{R_x}$.

$\overrightarrow{R_x}$: est la force de traînée causée par les perturbations de l'écoulement de l'air sur le rotor, le fuselage et d'autres objets entaché à l'hélicoptère. La traînée s'oppose à la poussée et propulse l'arrière parallèlement au vent relatif.

$m\vec{g}$: Le poids, ou m est la masse combinée de l'hélicoptère lui-même, de l'équipage, du carburant et de la cargaison ou des bagages. La masse tire l'hélicoptère vers le bas à cause de la force gravitationnelle, elle s'oppose à la portance et agit verticalement vers le bas en passant par le centre de gravité CG (centre de gravité) de l'hélicoptère.

$\overrightarrow{R_z}$: est la portance s'oppose à la force descendante du poids, elle est produite par l'effet dynamique de l'air agissant sur le ou les rotors, et agit perpendiculairement à la trajectoire de vol.

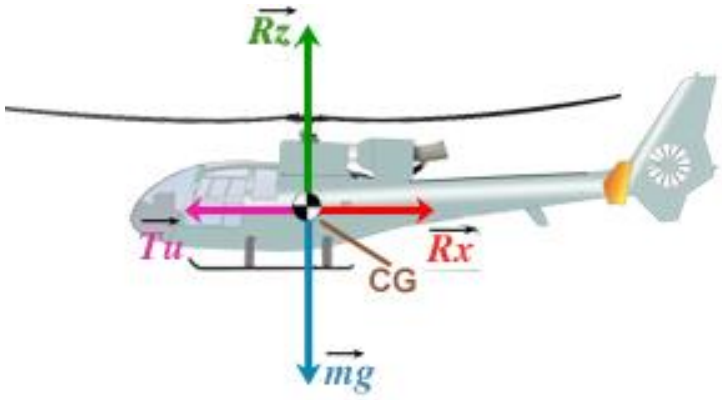


Figure I. 3 Les forces agissantes sur un hélicoptère en vol [13]

I.2.5. Modes de vol

L'hélicoptère exécute en principe trois sortes de vols [12,14] :

- Vol stationnaire.
- Vol vertical (ascendant ou descendant).
- Vol de translation (horizontalement ou incliné)

➢ **Le vol stationnaire :**

Il est réalisé lorsque l'hélicoptère reste à une certaine hauteur par rapport au sol en ayant une vitesse de translation pratiquement nulle. Dans ce mode de vol, les forces totales de portance et de poussée générées par le rotor principal s'exercent perpendiculairement à son plan de rotation (il n'y a pas de résultante aérodynamique).

Lorsqu'un hélicoptère vole sans vent, le plan de rotation du rotor est parallèle au sol, et la somme des forces du poids et de traînée sont exactement équilibrées par la somme des forces de poussée et de portance.

➢ **Le vol vertical**

En vol vertical, les composantes du poids et de la traînée sont combinées en un seul vecteur dirigé vers le bas (force totale de pesanteur). Et les composantes de portance et de poussée sont combinées en un seul vecteur dirigé vers le haut (force aérodynamique), suivant la résultante vectorielle des deux forces, le vol est ascendant ou descendant suivant si l'effet aérodynamique est supérieur ou inférieur au poids de l'appareil.

➢ **Le vol de translation**

Le vol de translation correspond à tout vol en avant, en arrière ou sur le côté. Pour la réalisation de ces déplacements, il faut incliner la résultante de la force aérodynamique de l'engin pour créer une force de traction horizontale, latérale ou les deux en même temps.

Le plan de rotation du rotor principal est incliné vers l'avant ou l'arrière pour le vol avant et arrière, pour le vol latéral, le plan de rotation du rotor principal est incliné dans la direction souhaitée.

I.3. Description du Aero Quanser

Le Quanser AERO [1] se compose de deux hélices, alimentées par des moteurs à courant continu, combinées avec un corps léger ce qui rend le système capable d'éffectuer des mouvements très réactifs. La base compacte du Aero Quanser comprend un amplificateur intégré avec un capteur de courant intégré, un dispositif d'acquisition de données intégré et un panneau d'interface QFLEX 2 interchangeable.

Les moteurs des hélices sont équipés d'encodeurs optiques. Les capteurs de courant et de tension des moteur peuvent être utilisés pour surveiller la consommation d'énergie du dispositif.

Le mécanisme à bague collectrice permet une rotation continue à 360°. Les angles des axes de tangage et de lacet sont mesurés à l'aide d'encodeurs optiques à haute résolution.

L'axe de tangage et de lacet peut être verrouillé indépendamment et l'angle des hélices peut être ajusté entre les positions horizontales et verticales. Cela permet aux utilisateurs de reconfigurer le Aero Quanser pour divers systèmes aérospatiaux (contrôle d'attitude 1 DOF, hélicoptère 2 DOF, demi-quadrotor).

La carte de l'unité de mesure inertielle (IMU) comprend des capteurs d'accéléromètre et de gyroscope, qui peuvent être utilisés pour l'estimation et la vérification de l'attitude et du lacet par rapport aux mesures de position directes des encodeurs.



Figure I. 4 Simulateur de vol Aero Quanser

L'Aero Quanser possède les caractéristiques techniques suivantes [1] :

I.3.1. Moteur courant continue

L'Aero Quanser dispose de deux modules propulseurs, chacun est actionné par un moteur à courant continu. Le schéma du circuit d'induit du moteur pour l'un des propulseurs est illustré à la figure I.5. Le moyeu est utilisé pour monter l'hélice sur le moteur et a un moment d'inertie J_h. Et l'hélice est fixée à l'arbre de la sortie avec un moment d'inertie J_p.

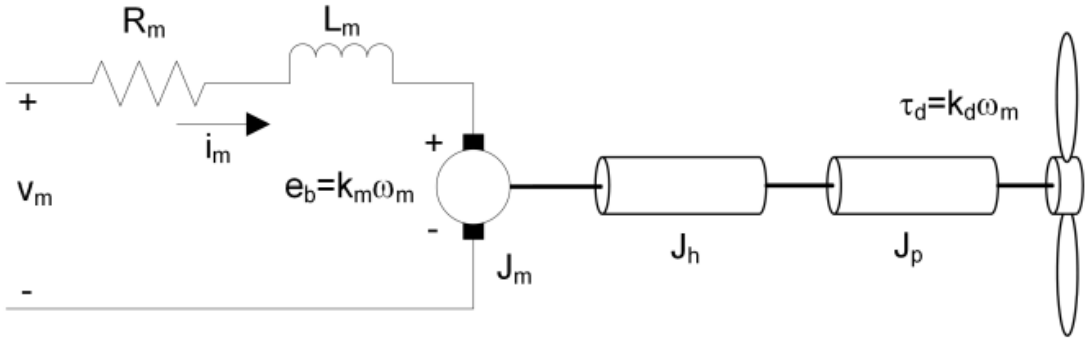


Figure I. 5 Schéma du circuit d'induit du moteur de l'un des propulseurs

Avec

e_b, J_m : force électromotrice et moment d'inertie du moteur.

R_m, L_m : résistance interne et inductance du moteur respectivement.

i_m, v_m : courant et tension du moteur.

τ_d : le couple de traînée (résistant).

La force électromotrice $e_b\ (t)$ dépend de la vitesse de l'arbre du moteur ω_m, et de la constante électromotrice du moteur k_m. Il s'oppose au passage du courant i_m. La tension force électromotrice est donnée par :

$$e_b\ (t) = k_m \omega_m(t)$$

Le couple exercé par la traînée et la résistance de l'air a été simplifié en un coefficient dérivé expérimentalement de la vitesse de l'hélice. Dans ce modèle, le couple de traînée τ_d qui s'oppose au couple moteur est donné par :

$$\tau_d = k_d \omega_m(t) \tag{I.1}$$

En utilisant la loi de Kirchoff, nous pouvons écrire l'équation suivante :

$$v_m(t) - R_m i_m(t) - L_m \frac{di_m(t)}{dt} - k_m \omega_m(t) = 0 \tag{I.2}$$

Comme l'inductance du moteur L_m est très inférieure à sa résistance, elle peut être ignorée. L'équation devient alors

$$v_m(t) - R_m i_m(t) - k_m \omega_m(t) = 0 \quad \text{(I.3)}$$

Le courant du moteur peut être trouvé comme

$$i_m(t) = \frac{v_m(t) - k_m \omega_m(t)}{R_m} \quad \text{(I.4)}$$

L'équation de l'arbre moteur est exprimée par

$$J_{eq} \dot{\omega}_m(t) = \tau_m(t) - \tau_d(t) \quad \text{(I.5)}$$

Où J_{eq} est le moment d'inertie total agissant sur l'arbre du moteur et τ_m est le couple appliqué par le moteur à courant continu. En fonction du courant appliqué, le couple moteur est donné par

$$\tau_m(t) = K_m i_m(t) \quad \text{(I.6)}$$

I.3.2. Encodeurs

Les encodeurs optiques incrémental rotatif sont utilisés pour mesurer la position angulaire des angles de tangage et de lacet de l'Aero Quansr.



Figure I. 6 Encodeur rotatif

L'encodeur a un disque marqué qui contient des cavités alternées, ce disque est relié à l'arbre du moteur à courant continu. Lorsque l'arbre tourne, la lumière d'une LED brille à travers les cavités et est captée par un capteur photoélectrique. Cela génère effectivement les signaux A et B illustrés à la figure I.7. Une impulsion d'index est déclenchée une fois pour chaque rotation complète du disque, qui peut être utilisée pour l'étalonnage ou la remise à l'origine d'un système.

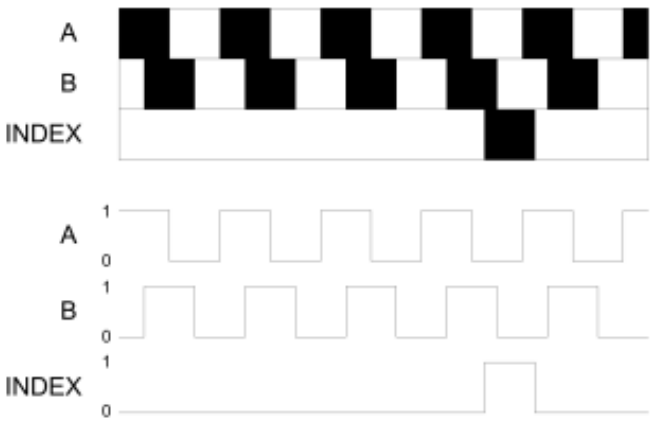


Figure I. 7 Les signaux générés par l'encodeur

I.3.4. Panneau d'interface QFLEX 2

L'Aero Quanser dispose de trois interfaces de communication interchangeables qui sont le panneau USB QFLEX 2, le panneau intégré et le panneau myRIO. Dans le cas du panneau USB, le logiciel de contrôle du système fonctionne sur PC via une connexion USB 2.0 standard. Le Quanser AERO USB peut être utilisé avec les logiciels MATLAB®/ Simulink® et Quanser QUARC, ou avec LabVIEW™ à l'aide du logiciel Quanser RCP.

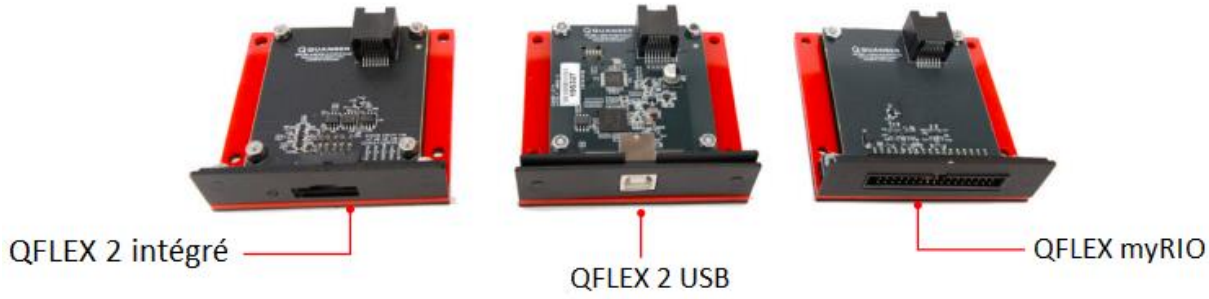


Figure I. 8 Les options du panneau d'interface QFLEX 2

I.4. Modèle dynamique du simulateur Aero Quanser

Les forces agissants sur le Quanser Aero dans une référenteile fixe sont illustrées dans la figure I.9. Lorsqu'une tension V_p est appliquée au moteur de tangage, la vitesse de rotation se traduit par une force F_p, qui agit normalement au corps à une distance r_p de l'axe de tangage. La rotation de l'hélice génère un couple autour de l'arbre du moteur du rotor de tangage qui entraîne le mouvement autour de l'axe de lacet. Donc, la rotation de l'hélice de tangage ne provoque pas seulement un mouvement autour de l'axe de tangage mais également autour de l'axe de lacet.

De même, le moteur du lacet provoque une force F_y qui agit sur le corps à une distance r_y de l'axe de lacet ainsi qu'un couple autour de l'axe de tangage.

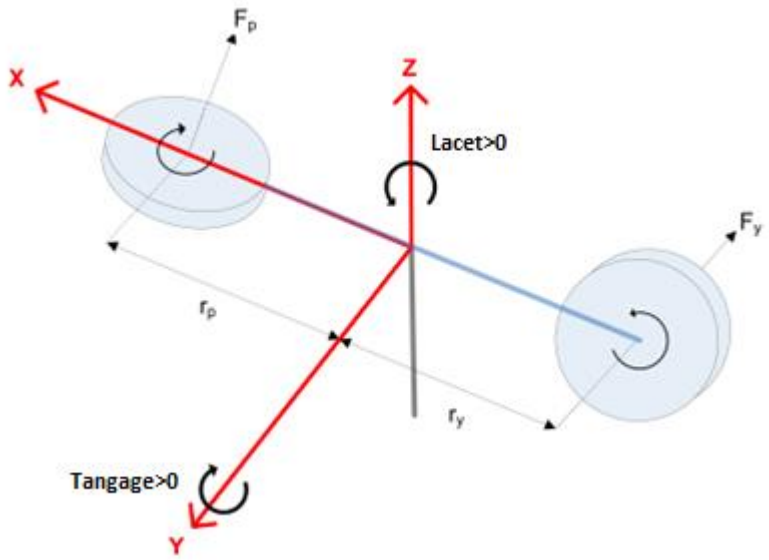


Figure I. 9 Schéma synoptique d'une configuration d'hélicoptère à 2 degrés de liberté

Le modèle non linéaire a été développé en tenant compte des hypothèses [15] :

- Les rotors principaux et arrière génèrent des couples dont l'un inffluence l'autre.
- Les rotors principaux et arrière sont de même taille et équidistants par rapport au centre de rotation.
- L' helice de tangage est parallèle au sol lorsque l'angle de tangage est nul.
- L'angle de tangage augmente positivement lorsque le rotor avant est déplacé vers le haut, le corps tourne dans le sens antihoraire autour de l'axe y et la tension du rotor avant est positive
- L'angle de lacet augmente positivement lorsque le corps tourne dans le sens antihoraire autour de l'axe z, dans ce cas la tension du rotor arrière est positive
- Les frottements dans le système sont du type visqueux

I.4.1. Les équations de mouvement

On a :

symbole	Description
J_p	Moment d'inertie autour de l'axe de tangage
J_y	Moment d'inertie autour de l'axe de lacet
D_p	constante de frottement de tangage
D_y	constante de frottement de lacet
K_{pp}	Gain de poussée de couple agissant sur l'axe de tangage à partir de l'hélice de tangage
K_{py}	Gain de poussée de couple agissant sur l'axe de tangage de l'hélice à lacet
K_{yp}	Gain de poussée de couple agissant sur l'axe de lacet à partir de l'hélice de tangage
K_{yy}	Gain de poussée de couple agissant sur l'axe de lacet à partir de l'hélice de lacet
l_c	Distance du centre de masse au corps fixe
m	Masse
g	Accélération de la gravitation

Tableau 1 Paramètres de Aero Quanser

Le centre de masse de corps fixe est représenté dans les cordonnées cartésiens par

$$\begin{aligned} x_c &= l_c \cos\psi \cos\theta \\ y_c &= l_c \sin\psi \cos\theta \\ z_c &= l_c \sin\theta \end{aligned} \tag{I.7}$$

Donc l'énergie potentielle totale (V) du système due à l'effet gravitationnel est :

$$Ep = mgl_c \sin\theta \tag{I.8}$$

L'énergie cinétique totale (T) du système est la somme des énergies cinétiques (T_1) et (T_2), de rotation agissant respectivement sur les axes de tangage et de lacet, ainsi que l'énergie cinétique (T_3) générée par le mouvement de translation du centre de masse [15].

$$Ec = Ec_1 + Ec_2 + Ec_3 \tag{I.9}$$

Où $Ec_1 = \frac{1}{2}J_p\dot{\theta}^2$, $Ec_2 = \frac{1}{2}J_y\dot{\psi}^2$ et $Ec_3 = \frac{1}{2}{ml_c}^2\left[\left(-\sin(\psi)\,\dot{\psi}\cos(\theta) - \cos(\psi)\sin(\theta)\,\dot{\theta}\right)^2 + \left(-\cos(\psi)\,\dot{\psi}\sin(\theta) + \sin(\psi)\sin(\theta)\,\dot{\theta}\right) + \left(\cos(\theta)\,\dot{\theta}\right)^2\right]$

Après simplification

$$Ec = \frac{1}{2}(J_p\dot{\theta}^2 + J_y\dot{\psi}^2 + {ml_c}^2\dot{\psi}^2 cos^2\theta + {ml_c}^2\dot{\theta}^2) \tag{I. 10}$$

L'équation de Lagrange est utilisée pour trouver les équations de mouvement des hélices de tangage et de lacet.

$$\frac{\partial}{\partial t}\left(\frac{dL}{d\dot{\theta}}\right) - \frac{dL}{d\theta} = Q_1$$
$$\frac{\partial}{\partial t}\left(\frac{dL}{d\dot{\psi}}\right) - \frac{dL}{d\psi} = Q_2 \quad \text{(I.11)}$$

Telle que

Q_1, Q_2 : sont les forces agissant respectivement sur l'axe de tangage et l'axe de lacet

L : la fonction lagrangienne, la différence entre les énergies cinétique et potentielle totales du système

$$Q_1 = \tau_p - D_p\dot{\theta}$$
$$Q_2 = \tau_y - D_y\dot{\psi} \quad \text{(I.12)}$$

Où

$$\tau_p = K_{pp}V_p + K_{py}V_y$$
$$\tau_y = K_{yp}V_p + K_{yy}V_y \quad \text{(I.13)}$$

τ_p, τ_y : sont les couples générés par les moteurs du tangage et du lacet, respectivement

Basé sur la formulation lagrangienne (I.5), les équations dynamiques peuvent être dérivées :

$$\left(J_p + ml_c{}^2\right)\ddot{\theta} = \left(K_{pp}V_p + K_{py}V_y - D_p\dot{\theta} - mgl_c cos(\theta) - ml_c{}^2\dot{\psi}^2 sin(\theta)cos(\theta)\right)$$
$$\left(J_y + ml_c{}^2 cos^2(\theta)\right)\ddot{\psi} = \left(K_{yp}V_p + K_{yy}V_y - D_y\dot{\psi} + 2ml_c{}^2\dot{\theta}\dot{\psi}sin(\theta)cos(\theta)\right) \quad \text{(I.14)}$$

I.4.2. La représentation d'état

Les équations de mouvement (I.8), peuvent être formulées dans l'espace d'état sous la forme $\dot{x} = f(x) + g(x)U$, considérant le vecteur d'état suivant

$$[\theta \quad \dot{\theta} \quad \psi \quad \dot{\psi}] = [x_1 \quad x_2 \quad x_3 \quad x_4] \quad \text{(I.15)}$$

Où x_1, x_2, x_3 et x_4 sont respectivement l'angle de tangage, la vitesse de tangage, l'angle de lacet et la vitesse de lacet. Donc la représentation de l'espace d'état est exprimée comme suit :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{1}{\left(J_p + ml_c{}^2\right)}\left(K_{pp}V_p + K_{py}V_y - D_p x_2 - mglcos(x_1) - ml_c{}^2 x_4{}^2 sin(x_1)cos(x_1)\right) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \dfrac{1}{\left(J_y + ml_c{}^2 cos^2(\theta)\right)}\left(K_{yp}V_p + K_{yy}V_y - D_y x_4 + 2ml_c{}^2 x_2 x_4 sin(x_1)cos(x_1)\right) \end{cases} \quad \text{(I.16)}$$

Posons $g_{11} = \frac{K_{pp}}{\left(J_p + ml_c{}^2\right)}$; $g_{12} = \frac{K_{py}}{\left(J_p + ml_c{}^2\right)}$; $g_{21} = \frac{K_{yp}}{\left(J_y + ml_c{}^2 cos^2(\theta)\right)}$; $g_{22} = \frac{K_{yy}}{\left(J_y + ml_c{}^2 cos^2(\theta)\right)}$

$$f_1 = \frac{-D_p x_2 - mgl_c cos(x_2) - ml^2 x_4{}^2 sin(x_1)cos(x_1)}{\left(J_p + ml_c{}^2\right)}$$

$$f_2 = \frac{-D_y x_4 + 2m{l_c}^2 x_2 x_4 sin(x_1) cos(x_1)}{b}$$

Donc le système est de la forme :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1 + g_{11} V_p + g_{11} V_y \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2 + g_{21} V_p + g_{22} V_y \end{cases} \qquad \text{(I.17)}$$

$$\begin{cases} y_1 = x_1 \\ y_2 = x_3 \end{cases}$$

La représentation d'état montre un système multi-variable, non linéaire avec un couplage complexe.

I.5. Conclusion

Dans ce chapitre, nous avons en premier lieu présenté des informations générales sur les hélicoptères (types, missions et modes de vol), puis décrit les caractéristiques techniques du simulateur d'hélicoptère à deux degrés de liberté appelé Aero Quanser. Ensuite, le formalisme de Lagrange Euler a été utilisé pour établir le modèle dynamique du système. Le modèle obtenu se caractérise par une nature couplée, complexe, non linéaire, et multi-variables.

Chapitre II : Notion De Base Sur La Commande Robuste et L’Optimisation Méta-Heuristique

II.2. Introduction

La conception d'une loi de commande pour l'Aero Quanser est une tâche délicate puisque la dynamique du système est non linéaire, multivariable et fortement couplée, et sujette aux incertitudes paramétriques à cause de la simplification du modèle. Ce chapitre présente les théories de commande utilisé dans ce mémoire. Les algorithmes d'optimisation méta-heuristiques utilisés pour trouver les paramètres optimaux des lois de commande sont présentés.

II.2. Théorie de Commande

II.2.1. Commande par PID

Un contrôleur proportionnel-intégral-dérivé (PID) est largement utilisé dans les systèmes industriels. Un contrôleur PID minimise l'erreur (la différence entre une consigne souhaitée et une grandeur mesurée de processus) en ajustant le processus à l'aide des trois actions P, I et D [16,17].

- Le terme P (qu'est proportionnel à l'erreur),
- Le terme I (qu'est proportionnel à l'intégrale de l'erreur),
- Le terme D (qu'est proportionnel à la dérivée de l'erreur).

Il existe plusieurs structures à partir de combinaison des actions telle que la structure parallèle II. 1.

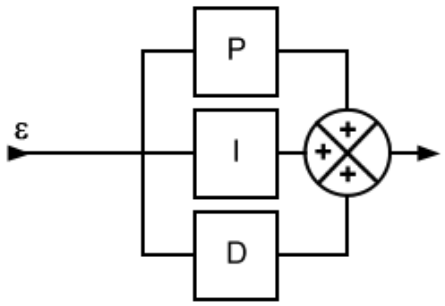


Figure II. 1 Structure parallèle d'un régulateur PID

Les paramètres du contrôleur sont le gain proportionnel K_p, le gain intégral K_i et le gain dérivé K_d. Les parties intégrale, proportionnelle et dérivée peuvent être interprétées comme des actions basées sur le passé, le présent et le futur de l'erreur respectivement.

II.2.1.1 Action proportionnelle

L'action proportionnelle produit une valeur de sortie proportionnelle à la valeur d'erreur actuelle. La réponse proportionnelle peut être ajustée en multipliant l'erreur par une constante K_p appelée gain proportionnel. Le terme proportionnel est donné par :

$$P(t) = K_p e(t) \tag{II.1}$$

L'erreur en régime permanent (en boucle fermée) diminue avec l'augmentation du gain proportionnel, mais cela a tendance à augmenter les oscillations, ce qui entraîne un risque de perte de la stabilité de système.

II.2.1.2 Action intégrale

L'action intégrale est proportionnelle à la fois à l'amplitude de l'erreur et à la durée de l'erreur. L'intégrale dans un contrôleur PID est la somme de l'erreur instantanée dans le temps, ce qui donne le décalage accumulé qui aurait dû être corrigé précédemment. L'erreur accumulée est ensuite multipliée par le gain intégral K_i et ajoutée à la sortie du contrôleur.

$$I(t) = K_i \int_0^t e(\tau) d\tau \qquad \text{(II.2)}$$

Le terme intégral accélère la convergence du processus vers la consigne et élimine l'erreur statique dans le régime permanent qui se produit avec un contrôleur proportionnel pur, mais il introduit des oscillations qui dégradent la stabilité du système.

II.2.1.3. Action dérivée

Le terme dérivé détermine l'évolution de l'erreur dans le temps et en multipliant ce taux de variation par le gain dérivé K_d, Le terme dérivé est donné par

$$D(t) = K_d \frac{de(t)}{dt} \qquad \text{(II.3)}$$

L'action dérivée anticipe le comportement du système et améliore ainsi le temps de stabilisation et la stabilité du système.

II.2.1.4. Les effets du réglage indépendant des parametres P,I et D

L'effet du réglage indépendant des paramètres P,I et D est résumé dans la tableau ci_dessou :

Réponse en boucle-fermée	Temps de monté	Dépassement	Temps de réponse.	Erreur en régime permanent	stabilité
augmentation de Kp	Diminuer	Augmenter	Petite augmentation	Diminuer	Dégrader
augmentation de Ki	Petite diminution	Augmenter	Augmenter	Nulle	Dégrader
augmentation de Kd	Petite diminution	Diminuer	Diminuer	changement mineur	Améliorer

Tableau 2 Les effets du réglage indépendant des parametres P,I et D

II.2.2. La commande par mode de glissement

La commande par mode de glissement (Sliding Mode Control, SMC) est une technique de commande non linéaire qui fait partie de la famille des commandes à structure variable, c.à.d. des commandes commutantes entre plusieurs lois de commande différentes [18]. L'importance de la commande par mode glissant réside dans : la réponse dynamique rapide, la simplicité de la conception et l'implantation, la robustesse par rapport aux incertitudes, aux variations des paramètres internes, aux erreurs de modélisation et aux perturbations externes.

II.2.2.1. Principe de la commande par mode de glissement

La commande SMC est conçue pour forcer la dynamique d'un système donné à suivre une surface particulière dans l'espace d'état, appelée surface de glissement. Une fois la surface de glissement atteinte, la commande par mode de glissement maintient la dynamique du système proche de la surface de glissement. Par conséquent, la conception de cette commande se fait en deux parties.

La première partie implique le choix d'une surface de glissement de telle sorte que le mouvement de glissement satisfait les spécifications désirées. La seconde concerne la sélection d'une loi de commande qui rendra la surface de commutation attractive pour l'état du système.

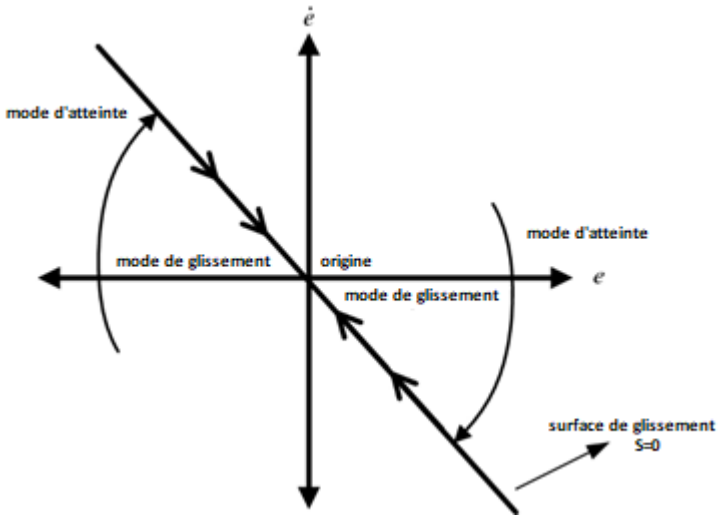


Figure II. 2 Surface de glissement

II.2.2.2. Choix de la surface de glissement

Le choix de la surface de glissement dépend de l'application et de l'objectif visé, dans la plupart des cas la surface est choisie pour faire tendre l'erreur vers 0. La surface de glissement est une fonction scalaire telle que la variable à régler glisse sur cette surface et tend vers l'origine du plan de phase.

Considérons le système non linéaire défini par les équations suivantes :

$$\begin{cases} x^n = f(x,t) + g(x,t)u \\ y = x \end{cases} \quad \text{(II.4)}$$

Où f et g sont des fonctions non linéaires, g est supposé inversible. L'erreur de poursuite e est définie par : $e = y_d - y$ Avec y_d est la consigne désirée.

La formule générale de la surface de glissement appelé surface de Slotine [19] est définie par :

$$S = \left(\frac{\partial}{\partial t} + \lambda\right)^{n-1} e(t) \tag{II.5}$$

Ou n représente le degré relatif du système par rapport à la sortie y, Il indique le nombre minimum de fois qu'il faut dériver la sortie $y(t)$ par rapport au temps, pour faire apparaitre l'entrée u.

II.2.2.3. Condition de stabilité et d'attraction

La loi de commande doit rendre la surface de glissement $S = 0$ attractive et assure ainsi l'apparition du mode glissant donc une fonction candidat de Lyapunov de type quadratique a été proposé pour la détermination de la région d'attraction

$$V(S) = \frac{1}{2}S^2 \tag{II.6}$$

La dérivée de la fonction donne

$$\frac{dV(S)}{dt} = S\dot{S} \tag{II.7}$$

Pour que la fonction candidate de Lyapunov garantisse l'existence du mode de glissement, il suffit d'assurer que

$$S\dot{S} < 0 \tag{II.8}$$

Cette condition n'est pas suffisante pour assurer une convergence en temps fini vers la surface. Une condition plus forte doit être respectée. Dans les modes glissants classique, on utilise généralement la condition d'attractivité non linéaire dite condition de η-attractivité

$$S\dot{S} < -\eta|S| \tag{II.9}$$

Dont $\eta > 0$, gain positif

Ce critère est toujours satisfait si la commande est de type

$$u_d = -Ksign(S) \tag{II.10}$$

Dans ce cas la dynamique du système convergent vers la surface de glissement dans un temps fini tr.

Preuve

$$\frac{dS}{dt} < -\eta \tag{II.11}$$

On intègre les deux côtés de l'inégalité

$$\int_0^{S(tr)} dS < \int_0^{tr} -\eta dt \tag{II.12}$$

Ce qui donne

$$[S]_0^{S(tr)} < [-\eta]_0^{tr} \tag{II.13}$$

Par conséquence

$$tr < \frac{S(0)}{\eta} \tag{II.14}$$

Où tr est le temps nécessaire à la convergence vers la surface de glissement, tant que le système n'a pas atteint la surface de glissement, la dynamique du système reste sensible à la variation des paramètres et peut être influencée par des perturbations extérieures, pour réduire le temps de convergence tr il suffit d'augmenter η c.à.d. augmenter l'amplitude de la command discontinue.

II.2.2.4. Synthèse de la commande équivalente

L'objectif de la loi de commande est de forcer le système vers la surface de glissement, puis de contraindre la trajectoire du système à rester dans la surface de glissement. Donc la loi de commande discontinue rend la surface localement attractive, c'est-à-dire au voisinage de la surface de glissement, toutes les trajectoires du système doivent être dirigées vers la surface.

Pendant le régime glissant, la commande discontinue a une fréquence de commutation en théorie infinie ; autrement dit la commande est discontinue à chaque instant. L'étude du système soumis à une telle commande est beaucoup plus complexe. La méthode théorique générale pour résoudre de telles équations différentielles discontinue est de remplacer par des inclusions différentielles [20], ou une autre approche plus concrète donnant une solution particulière par la méthode de commande équivalant. La nature analytique de cette méthode en fait un outil puissant à la fois pour l'analyse et la conception.

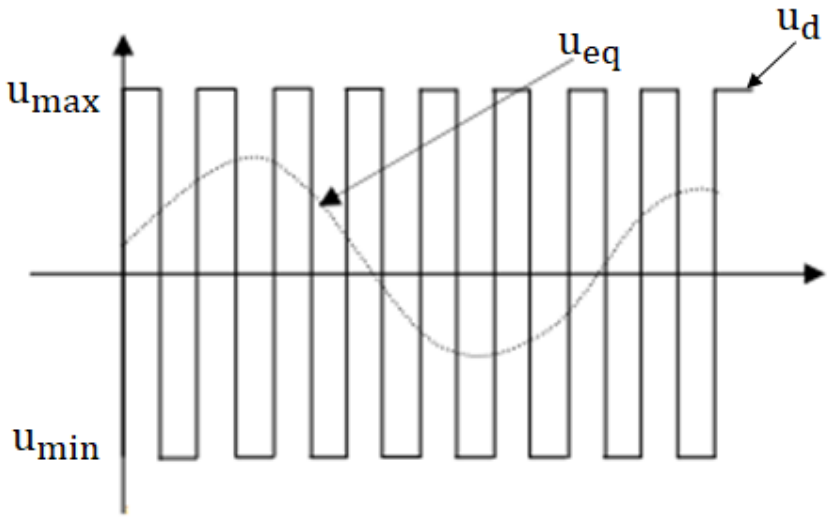


Figure II. 3 Les signaux u_{eq} et u_d

La méthode de commande équivalente peut être utilisée pour déterminer le mouvement du système limité à la surface de commutation $S(x) = 0$.

La première étape de l'approche de commande équivalente consiste à trouver l'entrée u_{eq} telle que la trajectoire d'état reste sur la surface de commutation $S(x) = 0$. L'existence du mode glissant implique que $S(x) = 0$, pour tout t ≥ t0,

$$\dot{S}(x) = 0 \qquad \text{(II.15)}$$

Remarque : la commande u_{eq} peut être interprétée comme étant la valeur moyenne que prend la commande u lors des commutations rapides entre u_{max} et u_{min}

II.2.2.5. Phénomène du broutement

Le phénomène de Broutement appelé aussi phénomène de Chattering, est caractérisé par de fortes oscillations autour de la surface de glissement due aux discontinuités appliquées à la commande qui commute à une fréquence infinie pendant le régime de glissement [21].

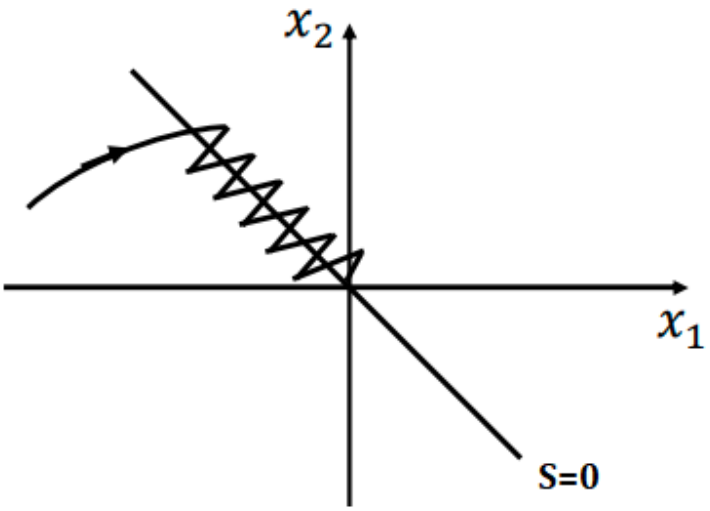


Figure II. 4 Phénomène de broutement

Cependant, pour une utilisation pratique, la fréquence de commutation des organes de commande est limitée (temps de commutation + temps de réponse des actionneurs). La discontinuité dans la commande engendre des oscillations hautes fréquences de la trajectoire du système autour de la surface de glissement. Ce phénomène a pour effet de retarder l'application de la commande permettant de ramener l'état du système sur la surface de glissement à partir du moment où il l'a quitté, c'est un des problèmes de la commande par mode glissant. Le phénomène de Chattering peut exciter des hautes fréquences dynamiques non modélisées et peut dégrader les performances du système et même conduire à l'instabilité, De plus le Chattering peut endommager les actionneurs et provoquer une détérioration prématurée des systèmes.

Pour résoudre ce problème plusieurs solutions ont été proposées qui sont :

- Utilisation des observateurs.
- Remplacement de la fonction sign par une fonction continue (couche limite).
- Utilisation des surfaces de glissements d'ordre supérieur.

II.2.3. Variantes de la commande par mode de glissement

II.2.3.1 Terminal sliding mode control (TSMC)

Le TSMC est une variante du contrôleur SMC, qui offre des propriétés supérieures telles qu'une convergence rapide en temps fini. Cette commande est particulièrement utile pour une commande à haute précision, car il accélère le taux de convergence au voisinage d'un point d'équilibre, la surface de glissement est données par [22] :

$$S = \dot{x} + \beta x^{\frac{q}{p}} \tag{II.16}$$

Où $\beta \in R^+$ et $p, q \in N$ avec $p > q$

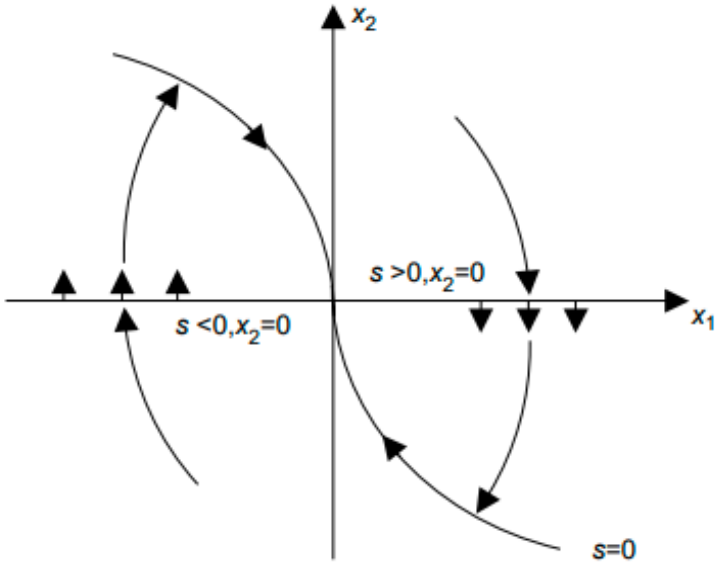


Figure II. 5 Surface de glissement dans le cas de l'NTSMC

Le TSMC a la même condition de convergence vers la surface de glissement que le SMC (voir II.2.2.3), et converge à la surface de glissement dans un temps fini tr donné par l'équation (II.14). Si $S(0) \neq 0$ les états du système convergent vers la surface de glissement dans un temps fini tr. Lorsque le mode de glissement $S = 0$ est atteint, la dynamique du système est déterminée par l'équation différentielle non linéaire suivante

$$\dot{x} + \lambda x^{\frac{q}{p}} = 0 \tag{II.17}$$

Donc

$$\frac{dx}{dt} = -\lambda x^{\frac{q}{p}} \tag{II.18}$$

On intègre les deux côtés de l'équation

$$\int_{x(tr)}^{x(ts+tr)} x^{-\frac{q}{p}}\,dx = \int_{tr}^{ts+tr} -\lambda\,dt \tag{II.19}$$

Ce qui donne

$$\left[\frac{1}{1-\frac{q}{p}}x^{1-\frac{q}{p}}\right]_{x(tr)}^{x(ts+tr)} = -[\lambda t]_{tr}^{ts+tr} \tag{II.20}$$

Où $x(ts+tr)=0$

Donc

$$-\frac{p}{p-q}x(tr)^{1-\frac{q}{p}} = -\lambda ts \tag{II.21}$$

Par conséquent

$$ts = \frac{p-q}{\lambda p}|x(tr)|^{1-\frac{q}{p}} \tag{II.22}$$

Où ts est le temps de glissement, il correspond au temps durant laquelle les trajectoires d'état sont confinées dans la surface de glissement S. Durant cette phase, le comportement du système ne dépond plus du système d'origine ni des perturbations, mais il est entièrement dominé par le choix de la surface de glissement ;

II.2.3.2. Nonsingular terminal sliding mode control (NTSMC)

La synthèse d'une commande équivalente nécessite une surface de glissement S différente à celle basée sur le TSMC car, dans le cas de TSMC, la surface est problématique car elle résulte des puissances négatives [23], provoquant une singularité (division par zéro dans la loi de commande). Cela conduit à un contrôle infini, ce qui est impossible pour les applications pratiques. Pour surmonter la singularité dans le TSMC, une alternative a été développée, et qui est devenue très populaire ces derniers temps, il s'agit de la surface TSMC non singulier (NTSM), défini par [24] :

$$S = |\dot{x}|^{\gamma} sign(\dot{x}) + \lambda x \tag{II.23}$$

Où $1 < \gamma < 2, \gamma$ peut également être exprimé sous la forme d'un nombre fractionnaire, c'est-à-dire γ = p/q où p et q sont des entiers impairs positifs afin que la différenciation peut être facilement effectuée.

II.2.3.3. Algorithme de super twisting (STW)

Les modes glissant d'ordre supérieur ont été introduits pour pallier au problème du chattering tout en gardant les propriétés de convergence en temps fini et de robustesse des commandes par modes glissants classiques. Parmi ces techniques, le super twisting est un algorithme très particulier qui a été développé par Levant [25]. En effet, contrairement aux autres algorithmes d'ordre supérieur, il ne requiert aucune information sur la dérivée de S.

Cette commande se décompose en un terme algébrique et un terme intégral. On peut considérer cet algorithme comme une généralisation non linéaire d'un PI [26].

$$w = -\frac{1}{2}\alpha|S|^{\frac{1}{2}}sign(S) - v$$
$$\dot{v} = \frac{1}{2}\,\beta\; sign(S) \qquad \text{(II.24)}$$

Dans cette approche, le terme discontinu n'apparaît plus directement dans l'expression de la commande synthétisée mais dans une de ses dérivées supérieures ce qui limite le phénomène de chattering dans la commande continue.

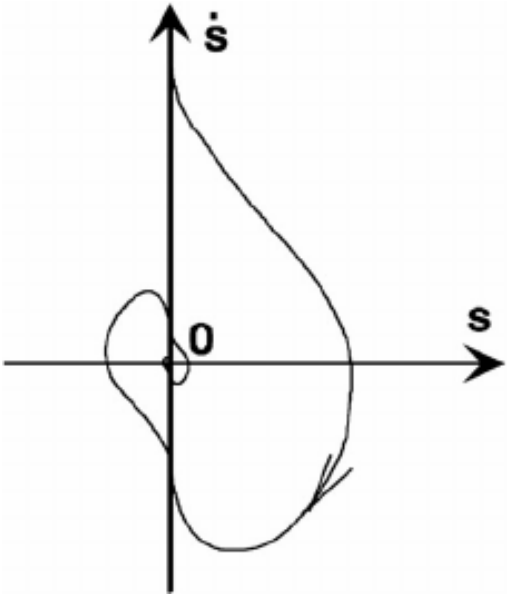


Figure II. 6 Plan de phase de l'évaluation de surface de glissement et sa dérivée

Remarque : Cet algorithme a été développé par Levant [26]. Pour un système ayant un degré relatif égal à un par rapport à la variable de glissement S.

II.2.3. Commande adaptative

La commande adaptative est une approche utilisée pour les systèmes ayant des paramètres incertains afin d'assurer des performances désirées. C'est aussi un ensemble de techniques pour l'ajustement automatique des contrôleurs en temps réel, afin d'atteindre et de maintenir un niveau souhaité de performances lorsque les paramètres du modèle dynamique du processus sont inconnus et/ou varient dans le temps [27].

Les algorithmes de commande adaptative peuvent être classés en deux groupes direct et indirect.

II.2.3.1. Commande adaptative indirecte

Le calcul des paramètres du régulateur passe par l'estimation en temps réel des paramètres du processus à commander. A chaque instant T (période d'échantillonnage), le système estimé est formé et traité comme s'il s'agissait du système réel à contrôler (à partir duquel sont calculés les paramètres du contrôleur). Cette approche est également appelée contrôle adaptatif explicite.

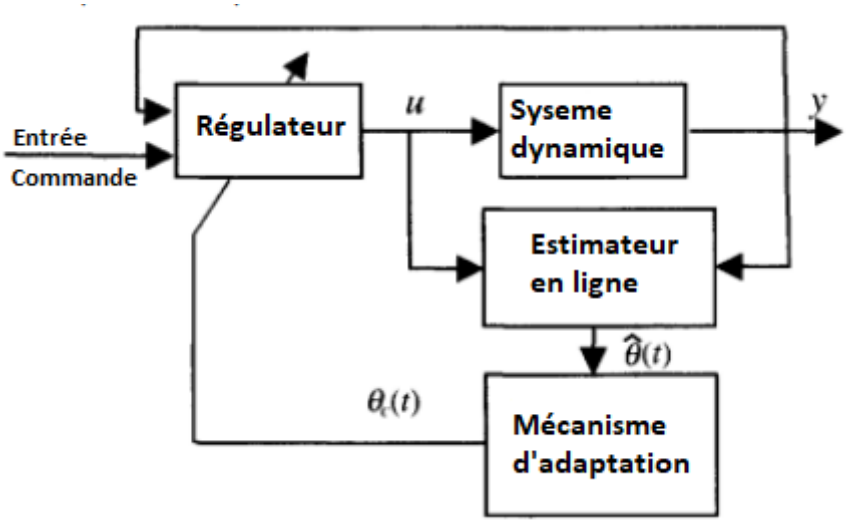


Figure II. 7 Schéma synoptique de la commande adaptative indirecte

II.2.3.1. Commande adaptative directe

Le processus à commander n'est pas identifié, et les paramètres du régulateur sont estimés directement (sans calculs intermédiaires impliquant des estimations de paramètres du système dynamique) selon une structure imposée par le concepteur. Cette approche est appelée aussi commande adaptative implicite. La structure de base du contrôle adaptatif direct est représentée dans la figure suivante.

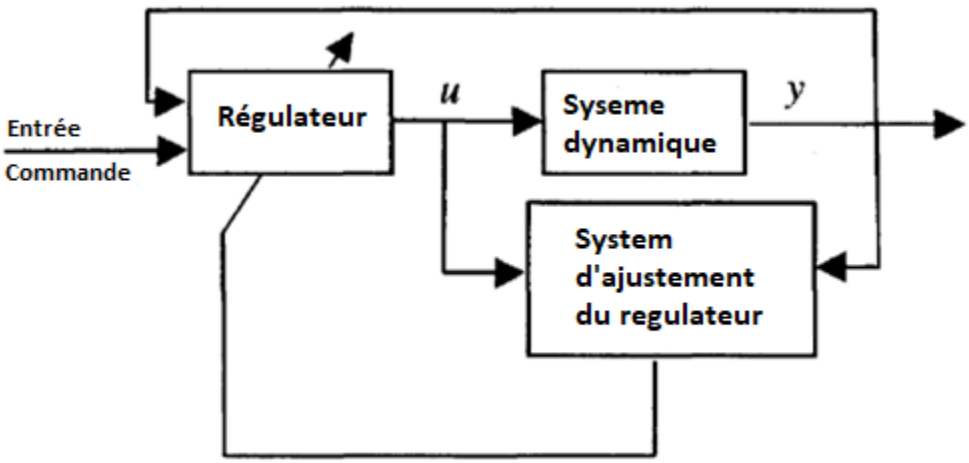


Figure II. 8 Schéma synoptique de la commande adaptative directe

II.3. Optimisation méta-heuristique

L'optimisation est l'art de prendre de bonnes décisions. Elle nous fournit un ensemble d'outils, principalement issus des domaines de l'informatique et des mathématiques, pour améliorer les performances d'un système.

Les algorithmes d'optimisation méta-heuristiques bio-inspirés ont pour objectif de résoudre les problèmes d'optimisation difficiles pour lesquels aucune méthode classique plus efficace n'est connue, Les méthode méta-heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global. Il existe un grand nombre de méthode allant de la simple recherche locale aux algorithmes de recherche globale complexes.

Lors de l'optimisation on définit la fonction objective par :

$$Fobj = f(x_1, x_2, \dots x_d) \tag{II.25}$$

Où $x_1, x_2, \dots$ sont les paramètres à optimisés.

II.3.1. Grey Wolf Optimizer (GWO)

Les loups gris vivent en société hiérarchisé, ils sont classés en alpha (α), bêta (β), delta (δ), et oméga (ω), et les loups passent par une étape de recherche ou exploration puis un étape d'exploitation ou de chasse, leur stratégie de chasse peut se résumer en trois étapes : la poursuite, l'encerclement et l'attaque. Après avoir suivi la proie, ils encerclent la proie jusqu'à ce que la proie arrête de bouger et attaquent enfin vers la proie.

Le Grey Wolf Optimizer a été développé on 2014 par Mirjalili [7]. Le GWO simule le processus de chasse des loups gris, il a été utilisé pour trouver des solutions sur des problèmes d'optimisation réels comme l'optimisation des systèmes hybrides d'énergie renouvelable [28].la répartition économique [29] et la localisation des nœuds capteurs [30].



Figure II. 9 Comportement de chasse des loups gris : (A) chasser, approcher et traquer les proies (B-D) poursuivre, harceler et encercler (E) situation stationnaire et attaque

Les proies représentent les solutions optimales, tandis que chaque loup représente une solution, les loups changent de position de façon à converger vers la solution optimale. Pour le comportement d'encerclement des loups, les équations (II.27) et (II.26) sont proposées.

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \quad \text{(II.26)}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \quad \text{(II.27)}$$

X représente la position du loup gris ; X_p représente la position de la proie, t représente l'itération en cours et $t+1$ représente l'itération suivante, A et C sont les coefficients dont les expressions sont données dans les équations (II.28) et (II.29).

$$\vec{A} = 2 \cdot \vec{a} \cdot \overrightarrow{r_1} - a \quad \text{(II.28)}$$

$$\vec{C} = 2 \cdot \overrightarrow{r_2} \quad \text{(II.29)}$$

Où a diminué linéairement de 2 à 0, r_1 et r_2 sont des valeurs aléatoires entre 0 et 1. En réalité, les loups voient leur proie mais dans GWO ce n'est pas possible. Puisqu'il n'y a pas de connaissance a priori sur le point optimal (proie) dans l'espace des solutions, le comportement de chasse est simulé avec les trois meilleures solutions trouvées par les n loups. Ainsi, les équations (II.30) ont été proposées pour décrire le comportement de chasse.

$$\begin{aligned}
\overrightarrow{D_\alpha} &= \left|\vec{C}.\overrightarrow{X_\alpha}(t) - \vec{X}(t)\right| \\
\overrightarrow{D_\beta} &= \left|\vec{C}.\overrightarrow{X_\beta}(t) - \vec{X}(t)\right| \\
\overrightarrow{D_\delta} &= \left|\vec{C}.\overrightarrow{X_\delta}(t) - \vec{X}(t)\right| \\
\overrightarrow{X_1}(t+1) &= \overrightarrow{X_\alpha} - \overrightarrow{A_1}.\overrightarrow{D_\alpha} \\
\overrightarrow{X_2}(t+1) &= \overrightarrow{X_\beta} - \overrightarrow{A_2}.\overrightarrow{D_\beta} \\
\overrightarrow{X_3}(t+1) &= \overrightarrow{X_\gamma} - \overrightarrow{A_3}.\overrightarrow{D_\gamma} \\
\vec{X} &= \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3}
\end{aligned} \quad \text{(II.30)}$$

L'algorithme divise le processus de recherche en exploration et exploitation :

- La recherche de proies (**exploration**) :

Le processus de recherche des loups gris suit la position de l'alpha (α), bêta (β), delta (δ), ces derniers diverge pour chercher et converge pour attaquer la proie. Dans GWO pour modéliser ce phénomène la valeur de A a été utilisé, si $|A| > 1$ le loup diverge de la proie (recherche d'autre proies) pour trouver une solution. Afin d'éviter une stagnation à un optimum local un autre paramètre aléatoire C est introduit.

- Attaque des proies (**exploitation**) :

L'attaque d'une proie est simulée en diminuant la valeur de a, ce qui rétrécit l'intervalle de variation de A ($[-a: a]$) au fil des itérations. Lorsque les valeurs aléatoires de $|A| < 1$, le loups est forcé d'attaquer vers la proie.

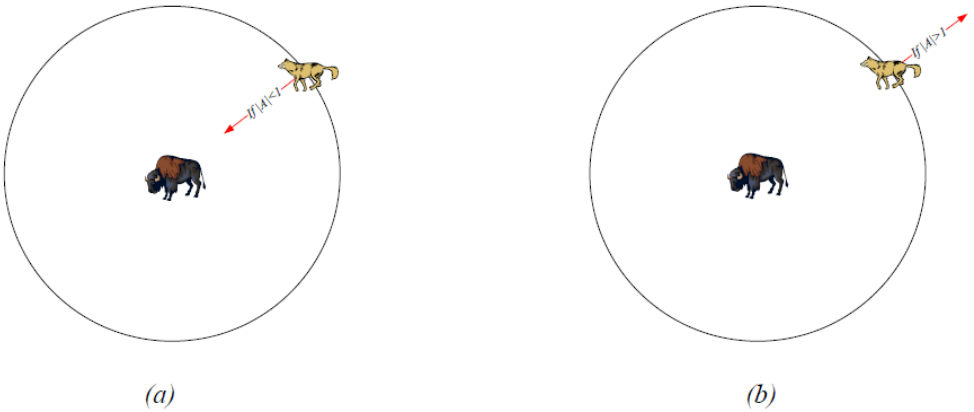


Figure II. 10 (a) attaque de la proie (b) recherche d'autre proies

Le pseudocode du GWO est donné dans la Figure II.11.

```
Initialiser la population des loups gris
Initialiser les paramètres A, a et C
Calculer chaque valeur de fitness des loups
Spécifiez la première, deuxième et troisième meilleures solutions
Tant que (t< max_iteration)
        Pour chaque loup
                Mettre à jour le poste
        FinPour
        Mettre à jour a, A, C
        Mettre à jour la condition physique de chaque loup
        Mettre à jour les première, deuxième et troisième meilleures solutions
        t = t + 1 ;/
Fait
```

Figure II. 11 Pseudocode du GWO

II.3.2. Wale optimization algorithm (WOA)

Même si les êtres humains, s'acceptent comme les créatures les plus intelligentes du monde, la plupart des animaux montrent des comportements intelligents. Les baleines en font partie, ce sont des mammifères qui respirent, ne dorment jamais, ont des comportements émotionnels et communiquent entre eux. Le processus de chasse des baleines à bosse est assez intéressant. Ils chassent leurs proies près de la surface de l'eau, créant des bulles.

Le **WOA** simule le comportement de chasse des baleines à bosse développé par Mirjalili en 2016 [8]. Même si les formulations de WOA sont assez similaires à celles de GWO, l'utilisation d'une spirale pour simuler le mécanisme d'attaque des baleines à bosse par filet à bulles et l'utilisation d'un optimale global sont les principales différences de WOA.

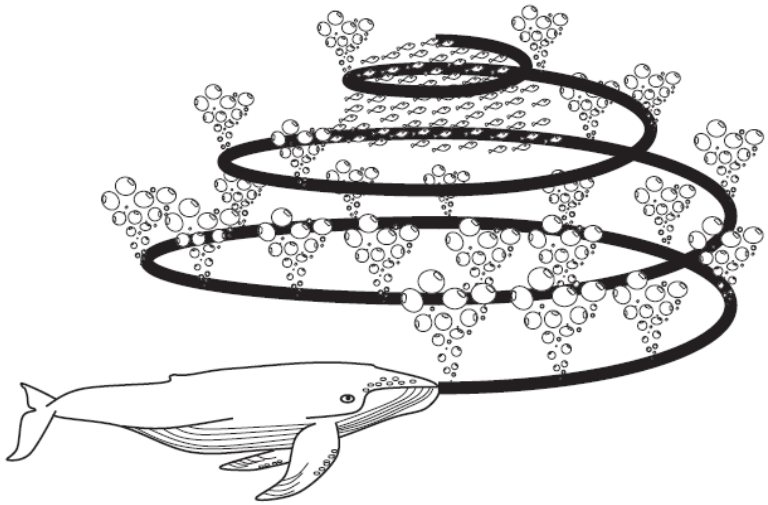


Figure II. 12 Baleine à bosse en chasse

Les proies des baleines à bosse sont les petits troupeaux de poissons et de krill (petit crustacé). Leur plan de chasse se compose de trois stratégies ; encerclant des proies, manœuvre d'alimentation en filet à bulles en spirale et recherche de proies. Ils reconnaissent leurs proies, créent des bulles en spirale et les encerclent comme des loups. Comme GWO, la proie est la meilleure solution optimale acceptée, puisque l'optimum global n'est pas connu a priori. Les positions de la population sont mises à jour vers la meilleure solution optimale comme indiqué dans les équations (II.31) et (II.32).

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_*}(t) - \vec{X}(t)\right| \quad \text{(II.31)}$$

$$\vec{X}(t+1) = \overrightarrow{X_*}(t) - \vec{A}.\vec{D} \quad \text{(II.32)}$$

D est la distance absolue entre la proie et la baleine.
A et C sont des vecteurs de coefficients calculés par les équations (II.33) et (II.34).
t et $t+1$ sont respectivement l'itération actuelle et la prochaine itération.
X est le vecteur position des baleines, X^* est la position de la meilleure solution atteinte jusqu'à cette itération et mise à jour à chaque itération.
a diminue linéairement de 2 jusqu'à 0, pour simule un comportement rétrécissant d'encerclement.
r_1 et r_2 sont deux vecteurs aléatoires.

$$\vec{A} = 2.\vec{a}.\overrightarrow{r_1} - a \quad \text{(II.33)}$$

$$\vec{C} = 2.\overrightarrow{r_2} \quad \text{(II.34)}$$

La stratégie du filet à bulles des baleines est également simulée dans l'algorithme comme indiqué dans l'équation **(**II.35), **(**II.36). Dans l'algorithme, les deux stratégies sont utilisées avec 50% de possibilité, les baleines choisissent entre le mécanisme d'encerclement rétrécissant ou le modèle en spirale pour mettre à jour leurs positions comme indiqué dans l'équation (II.37).

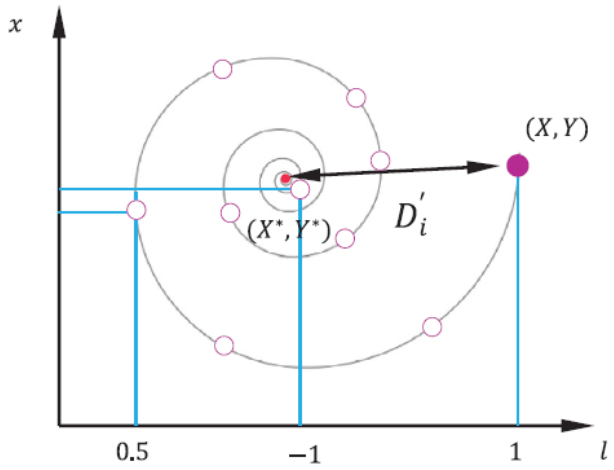


Figure II. 13 La spirale de mise à jour des positions

$$\overrightarrow{D'} = \left|\overrightarrow{X_*(t)} - \overrightarrow{X(t)}\right| \quad \text{(II.35)}$$

$$X(t+1) = D' e^{bl} \cos(2\pi l) + X_*(\mathrm{t}) \quad \text{(II.36)}$$

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_*}(t) - \vec{A}.\vec{D} & \text{si p} < 0.5 \\ \overrightarrow{D'} e^{bl} \cos(2\pi l) + \overrightarrow{X_*}(\mathrm{t}) & si\ p > 0.5 \end{cases} \quad \text{(II.37)}$$

Où

p est un nombre aléatoire dans [0,1].

D' indique la distance de la i ème baleine à la proie (meilleure solution obtenue à ce jour)

b est une constante pour définir la forme de la spirale logarithmique.

l est un nombre aléatoire dans [−1,1],

Le rétrécissement de l'encerclement et l'afflue continue des bulles en spirale sont tous les deux destinés à l'exploitation. Certains mouvements aléatoires sont également acceptés dans l'algorithme d'exploration, comme indiqué dans les équations (II.38)**,** (II.39).

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_{rand}}(t) - \vec{X}(t)\right| \quad \text{(II. 38)}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}}(t) - \vec{A}.\vec{D} \quad \text{(II.39)}$$

$\overrightarrow{X_{rand}}$ est un vecteur de position aléatoire (une baleine aléatoire) choisi parmi la population actuelle.

```
Initialiser la population de baleines X_i(i = 1, 2, ..., n)
Calculer le fitness de chaque agent de recherche
X* = le meilleur agent de recherche
Tant que (t < nombre maximum d'itérations)
        Pour chaque agent de recherche
        Mettre à jour a, A, C, l et p
                Si (p < 0,5)
                        Si (|A| < 1)
                                Mettre à jour la position de l'agent de recherche actuel par
                        l'équation (32)
                        Sinon si (|A| ≥ 1)
                                Sélectionnez un agent de recherche aléatoire ()
                                Mettre à jour la position de l'agent de recherche actuel par
                        l'équation (39)
                        Fin si
                Sinon (p > 0.5)
                        Mettre à jour la position de la recherche actuelle par l'Eq. (36)
                Fin si
        Fin pour
        Vérifiez si un agent de recherche dépasse l'espace de recherche et y remédier
        Calculer le fitness de chaque agent de recherche
        Mettre à jour X* s'il existe une meilleure solution t = t + 1
Fait
retourn X*
```

Figure II. 14 Pseudocode du WOA

II.3.3. Salp swarm algorithm (SSA)

Les salpes sont des créatures ressemblant à des méduses mais sont évolutivement très différents, ils ont des systèmes nerveux et digestifs complexes avec un cerveau, un cœur et des intestins. SSA simule les comportements de recherche de nourriture des salpes. Étant donné que l'algorithme est assez nouveau, quelques études utilisent cet algorithme. Une chaîne de salpes est visible sur la figure II.15. Les salpes créent généralement des chaînes et se déplacent en coopération.

Le ***SSA*** est l'un des récents algorithmes d'optimisation inspirés de la nature, développé par Mirjalili [9] pour la solution de problèmes d'optimisation continue. Étant donné que SSA n'a qu'un seul paramètre à adapter, il est assez facile à mettre en œuvre.

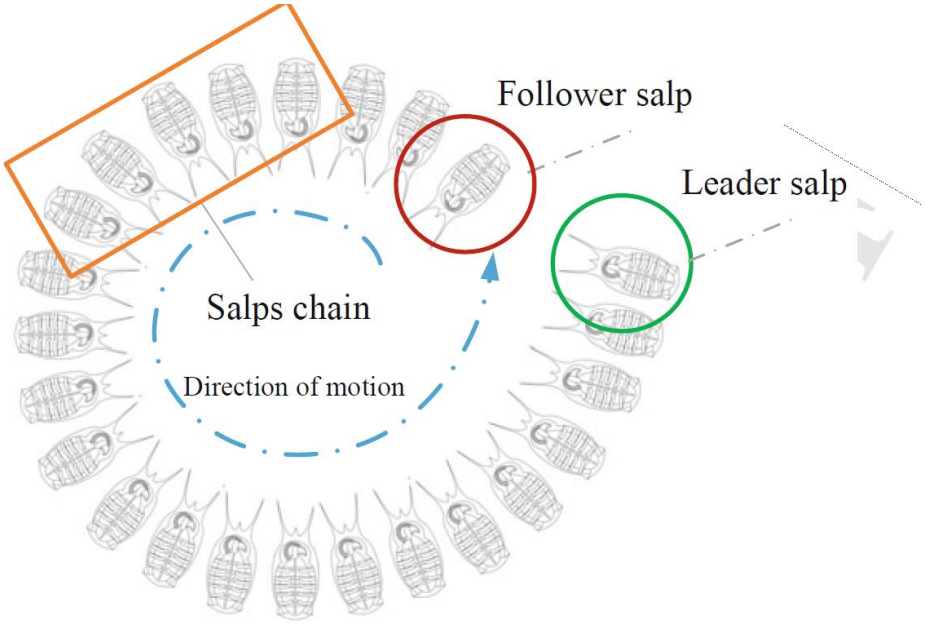


Figure II. 15 Une chaîne de salpes

En SSA, les salpes sont classées en deux groupes : salpe leader et suiveurs. Les suiveurs suivent la position du leader salpe. La position de la salpe leader est mise à jour avec l'équation (II.40).

$$x_j^1 = \begin{cases} F_j + c_1 * \left(ub_j - lb_j\right) * c_2 + lb_j; & si\ c_3 \geq 0 \\ F_j - c_1 * \left(ub_j - lb_j\right) * c_2 + lb_j; & si\ c_3 \geq 0 \end{cases} \quad \text{(II.40)}$$

x_j^1 Montre la position de la salpe leader dans la j ème dimension.

F_j Est la position de la source de nourriture dans la j ème dimension.

ub_j et lb_j représentent la borne supérieure et la borne inférieure de j ème dimension.

c_1 est calculé avec l'équation (II.41). c_2, et c_3 sont des nombres uniformes aléatoires entre 0 et 1.

$$c_1 = 2 * e^{-(\frac{4l}{L})^2} \quad \text{(II. 41)}$$

l est l'itération courante, L est le nombre d'itération. Les salpes suiveuses mettent à jour leurs positions avec l'équation (II.42).

$$x_j^1 = \frac{1}{2}at^2 + v_0 t \quad \text{(II. 42)}$$

t est le temps, mais dans la simulation t montre l'itération.

v_0 est la vitesse initiale.

a est l'accélération, qui est donnée avec l'équation (II.43) Puisque la différence entre l'itération Δt est 1.

$$a = \frac{\delta v}{\delta t} = \frac{v_{final} - v_0}{\Delta t} = \frac{v_{final} - v_0}{1} \quad \text{(II. 43)}$$

v_{final} est calculé avec l'équation (II.44).

$$v_{final} = \frac{x - x_0}{1} \quad \text{(II. 44)}$$

Si a est remplacé dans l'équation, les positions des salpes suiveuses peuvent être trouvées avec l'équation (II.45).

$$x_j^1 = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \quad \text{(II. 45)}$$

```
Créer des positions initiales pour n salpe en considérant lb,ub
Tant que les critères de résiliation ne sont pas satisfaits (comme le nombre d'itération)
        Calculer la fitness (aptitude de la fonction objective) de chaque salpe
        Trouver la meilleure valeur optimale F (La position du salpe leader)
        Mettre à jour c1
        Pour chaque salpes faire
                Si la salpe est leader
                        Mettre à jour la position de la salpe leader avec l'équation II.40
                Sinon
                        Mettre à jour la position actuelle de la salpe suiveuse avec l'équation II.45
                Fin
        Fin
Mettre à jour la population en utilisant les limites supérieure et inférieure des variables
Renvoyez les salpes qui ont violé les restrictions de délimitation.
Retoun F
```

Figure II. 16 Pseudocode du SSA

II.3.4. Ant Lion Optimizer (ALO)

Les Myrmeleontidae ou fourmilions est une famille d'insectes névroptères, ils ont une durée de vie près de trois ans et ils passent la majeure partie de leur vie sous forme de larves. ALO s'inspire du comportement de chasse des fourmilions. On voit des larves de fourmilions sur la Figure II. 17 (a) qui creusent des fosses en forme de cône dans le sable en se déplaçant le long d'un chemin circulaire et en jetant du sable sur la Figure II. 17 (b).

Après avoir construit des fosses en forme de cône, ils attendent que les fourmis tombent dans ce piège. Après que la fourmi ait été piégée dans cette fosse, la fourmi lion l'attrape et la chasse.

L'Ant Lion Optimizer (ALO) est l'un des récents algorithmes d'optimisation inspirés de la nature développés par Mirjalili spécialement pour la solution des problèmes d'optimisation continue en 2015 [10]. Depuis, différents types de problèmes d'ingénierie ont été résolus avec ALO tels que, le contrôle automatique de la génération d'un système multizone à l'aide d'un contrôleur PID [31], sélection de fonctionnalités pour l'apprentissage automatique [32], flux de puissance optimal avec amélioration de la stabilité de la tension et réduction de la perte de puissance [33], solution de problème de répartition de charge économique non convexe pour les systèmes électriques à petite échelle [34], planification d'itinéraires pour véhicule aérien sans pilote [35], détection de communauté dans un réseau complexe [36],ces travaux représentes quelque problèmes d'optimisation résolus avec succès en utilisant ALO.



a) Les fausses de fourmilions b) Une larve de fourmilion

Figure II. 17 La larve de fourmilion et ses pièges à fourmis en forme de cône

L'ALO imite l'interaction entre les fourmilions et les fourmis. Ainsi, les fourmis artificielles se déplacent sur l'espace de recherche et les fourmilions sont autorisés à les chasser. Étant donné que les fourmis se déplacent de manière stochastique dans la nature lorsqu'elles cherchent de la nourriture, une marche aléatoire est choisie pour modéliser le mouvement des fourmis. Dans cet algorithme, deux matrices initiales de dimension $n * d$ sont créées à la fois pour les fourmis et les fourmilières, comme indiqué dans l'équation (II 26). La matrice $n * d$ appartient aux fourmis, sauf les positions de n fourmis. La forme physique de chaque fourmi est enregistrée dans un vecteur $n * 1$.

Les fourmilions se cachent également quelque part dans l'espace de recherche. Afin de sauvegarder leurs positions et leurs valeurs de fitness, deux matrices $n * d$ et $n * 1$ sont utilisées. La matrice $n * d$ appartient aux fourmilières sauf les positions des n fourmilions. La forme physique de chaque fourmilière est enregistrée dans un vecteur $n * 1$. Dans ALO, chaque fourmilion correspond à une fourmi. Alors que les fourmis sont mobiles dans l'espace de recherche, les fourmilions sont immobiles. Les marches aléatoires des fourmis sont mises à jour avec l'équation (46).

$$X_i^t = \frac{(X_i^t - a_i) * (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \tag{II.46}$$

a_i et b_i sont respectivement le minimum et le maximum de la marche aléatoire de la $i$ème variable, c_i^t et d_i^t sont le minimum et le maximum de la marche aléatoire de la $i$ème variable pour l'itération t.

$$c_i^t = fourmilion_t^j + c^t \tag{II.47}$$

$$d_i^t = fourmilion_t^j + d^t \tag{II.48}$$

c et d sont mis à jour avec les équations (II.59) et (II.50) respectivement.

$$c^t = \frac{c^t}{I} \tag{II.49}$$

$$d^t = \frac{d^t}{I} \tag{II. 50}$$

Les marches aléatoires sont affectées par les pièges des fourmilions. Les fourmilions construisent des fosses en fonction de leurs valeurs de fitness. Les marches aléatoires sont

diminuées d'une itération à l'autre. Si une fourmi est plus apte qu'un fourmilion, alors la fourmi est attrapée par le fourmilion. Après avoir mangé cette fourmi, la fourmilière met à jour sa position afin d'augmenter ses chances d'attraper de nouvelles proies. Le meilleur fourmilion obtenu jusqu'à présent à chaque itération est conservé et considéré comme une élite.

Une marche aléatoire est choisie pour modéliser le mouvement des fourmis comme suit :

$$X(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_1) - 1) \\ , \dots\dots, cumsum(2r(t_n) - 1)] \quad \text{(II.51)}$$

Où cumsum calcule la somme cumulée, n est le nombre maximal d'itérations, t indique le pas de marche aléatoire (le nombre d'itération dans pour la plupart des études) et $r(t)$ est une fonction stochastique définie comme suit :

$$r(t) = \begin{cases} 1 \; Si \; rand > 0.5 \\ 0 \; Si \; rand \leq 0.5 \end{cases} \quad \text{(II.52)}$$

Où t montre le pas de marche aléatoire (itération dans la plupart des études) et $rand$ est un nombre aléatoire généré avec une distribution uniforme dans l'intervalle de [0, 1].

- Capture des proies et reconstruction de la fosse :

La dernière étape de la chasse est lorsqu'une fourmi atteint le fond de la fosse et est prise dans la mâchoire du fourmilion. Après cette étape, le fourmilion tire la fourmi à l'intérieur du sable et consomme son corps. Pour imiter ce processus, on suppose que la capture de proies se produit lorsque les fourmis deviennent plus aptes (elle va à l'intérieur du sable) et correspondant à un fourmilion. Un fourmilion doit alors mettre à jour sa position avec la dernière position de la fourmi chassée pour augmenter ses chances d'attraper de nouvelles proies. L'équation suivante est proposée à cet égard :

$$Antlion_j^t = Ant_i^t \; if \; (Ant_i^t) > f(Antlion_j^t) \quad \text{(II.53)}$$

Ant_j^t : indique la position de la *j* ème fourmi à la *t* ème itération

$Antlion_j^t$: indique la position de la *j* ème fourmilion à la *t* ème itération

- L'élitisme : est une caractéristique importante des algorithmes évolutionnaires qui leur permet de conserver la ou les meilleures solutions obtenues à n'importe quelle étape du processus d'optimisation. Dans cette étude, le meilleur fourmilion obtenu jusqu'à présent à chaque itération est conservé et considéré comme une élite. Étant donné que l'élite est le fourmilion le plus apte, elle devrait pouvoir affecter les mouvements de toutes les fourmis au cours des itérations. Par conséquent, on suppose que chaque fourmi marche au hasard autour d'un fourmilion sélectionné par la roulette et l'élite simultanément comme suit :

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad \text{(II.54)}$$

Ant_i^t: Indique la position de la *i* ème fourmi à la *t* ème itération.

R_A^t : est la marche aléatoire autour du fourmilion sélectionné par la roulette à la *t* ème itération,

R_E^t : est la marche aléatoire autour de l'élite à la t ème itération.

Le pseudocode de Ant Lion est donné dans la figure II.18

```
-Initialisez les positions des n fourmis et fourmilions
Calculez leurs valeurs de leur aptitude (fitness)
Trouvez une fourmilion d'élite
Tant que les conditions d'arrêt ne sont pas remplies
        Pour chaque fourmi
                Sélectionnez un fourmilion à l'aide de Roulette (wheel selection)
                Mettre à jour c et d en utilisant les équations Eqts. (II 49) et (II 50)
                Créez une marche aléatoire et normalisez-la à l'aide des équations. (II 51) et (II 46)
                Mettre à jour le poste eqt (II 54)
        FinPour
        Calculez les fitness de toutes les fourmis
        Remplacez un fourmilion par sa fourmi correspondante si elle devient plus apte Eq. (II 53)
        Mettre à jour l'élite si un fourmilion devient plus apte que l'élite
Fait
Retour Élite
```

Figure II. 18 Pseudocode d'ALO

II.3.5. Choix de la fonction objective

La fonction objective à minimiser suit le critère de performance ITAE qui est définie par l'intégrale du temps multipliée par l'erreur absolue :

$$ITAE = \int_{o}^{tf} t|e(t)|dt \tag{II.55}$$

L'indice de performance ITAE présente les avantages, (1) de minimiser le temps de stabilisation puisque le temps est inclus dans le critère ITAE, (2) de minimiser le dépassement maximal ceci à cause de la multiplication du temps par l'erreur absolue.

Afin d'éviter un grand signal de commande, qui peut condamner l'actionneur. La fonction objective utilisée a été prise comme la somme pondéré de deux indices de performance, qui sont ITAE et ISCO (l'intégrale de la commande au carrée) [37].

$$J = \int_{0}^{tf} (Qt|e| + Ru^2)dt \tag{II.56}$$

Où Q et R sont des coefficients constants positives

II.4. Conclusion

Dans ce chapitre, nous avons donné un aperçu théorique sur les commandes utilisées dans ce PFE (PID et mode de glissement), leurs propriétés, ainsi que les algorithmes d'optimisation GWO, WOA, ALO et SSA utilisés pour la détermination des paramètres optimaux des commandes.

Chapitre III : Synthèse des lois de commande

III.1. Introduction

Ce chapitre est consacré à la synthèse des lois de commande de l'angle de tangage et de lacet du Aero Quanser en se basant sur le modèle mathématique du système et les notions théoriques présentées dans les chapitres précédents. Ceci est afin de trouver une commande robuste qui réalise à la fois une convergence rapide de l'erreur de poursuite en temps fini et en évitant tous problèmes de singularité et aussi l'atténuation de l'amplitude du broutement en présence des perturbations. Plusieurs lois de commande ont été développées pour atteindre les performances et la robustesse souhaitées

III.2. Structure de commande par PID

Les deux angles de tangage et de lacet sont contrôlés indépendamment avec deux commandes PID, qui ont une structure de type parallèle. L'expression temporelle de la commande est donnée par :

$$U_{PID\theta}(t) = K_{p\theta}e_\theta(t) + K_{i\theta}\int_0^t e_\theta(\tau)d\tau + K_{d\theta}\frac{de_\theta(t)}{dt}$$
$$U_{PID\psi}(t) = K_{p\psi}e_\psi(t) + K_{i\psi}\int_0^t e_\psi(\tau)d\tau + K_{d\psi}\frac{de_\psi(t)}{dt} \quad \text{(III.1)}$$

Les paramètres des commandes $K_{p\theta}, K_{i\theta}, K_{d\theta}, K_{p\psi}, K_{i\psi}$ et $K_{d\psi}$ sont calculés à l'aide des algorithmes d'optimisation GOW, WOA, SSA et ALO, afin de minimiser une fonction objective. Le schéma général du contrôle est illustré dans la figure ci-dessous :

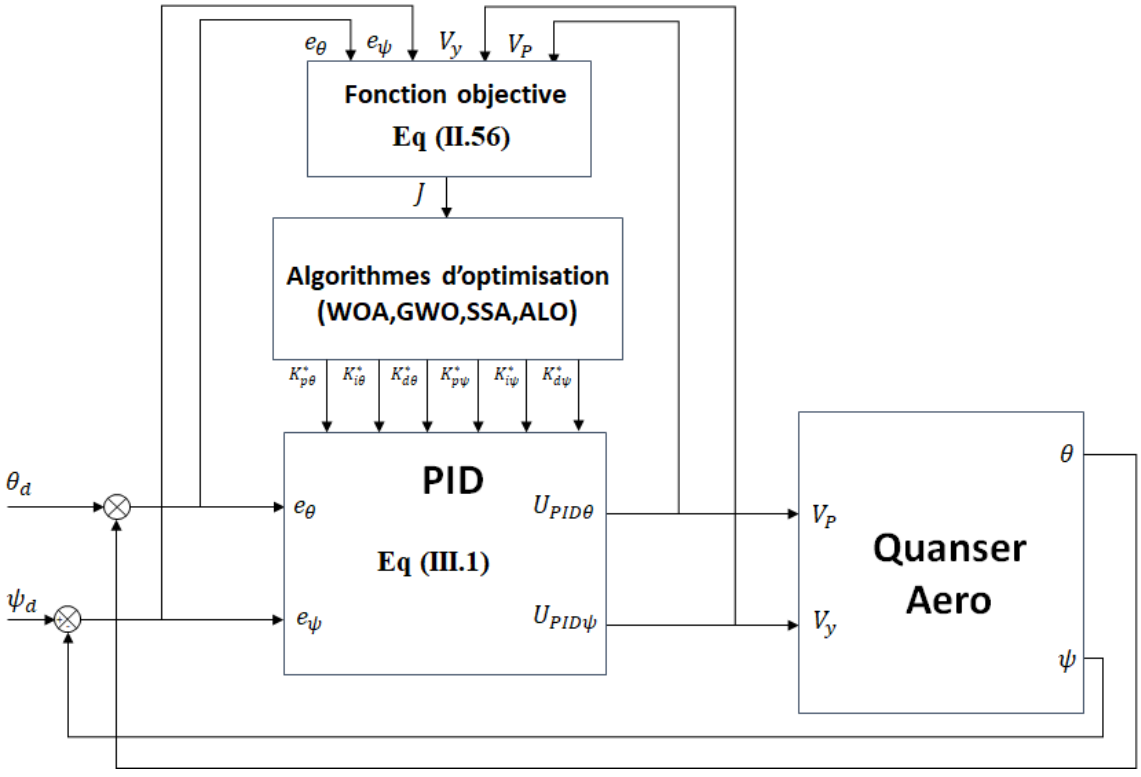


Figure III. 1 Structure .de la commande PID

III.3. Synthèse des lois de commande par mode de glissement

Plusieurs techniques de commande à structure variable ont été utilisées dans la conception de la loi de commande pour le Quanser Aero. Les paramètres de la surface de glissement et la loi de commande discontinue ont été optimisés via des algorithmes d'optimisation afin de minimiser la fonction objective J (II.56).

La commande totale appliquée sur chaque sous-système d'azimute et d'élévation est de la forme $V_{total} = V_d + V_{eq}$ pour les commandes SMS, NTSMC, STW et NTSTW, comme illustré dans les figures III.2, III.3, III.4 et III.5 respectivement.

III.3.1. Commande par mode glissant classique (SMC)

- *Synthèse de la commande équivalente*

D'abord on commence par le choix de la surface de glissement en utilisant la relation (II.5), le degré relatif associé à chaque sortie est égale 2 donc la surface est d'ordre 1 telle que :

$$\begin{aligned} S_\theta &= \lambda_1 x_1 + x_2 \\ S_\psi &= \lambda_2 x_3 + x_4 \end{aligned} \tag{III.2}$$

Où

$x_1, x_2, x_3\ et\ x_4$ sont les variables d'état du système,

λ_1, λ_2 sont des gains positifs réels.

On dérive les surfaces par rapport au temps, ce qui donne

$$\begin{aligned} [\lambda_1 \quad 1] \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= 0 \\ [\lambda_2 \quad 1] \begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= 0 \end{aligned} \tag{III.3}$$

La relation (III.3) peut être écrite sous une forme matricielle

$$\begin{bmatrix} \lambda_1 \dot{x}_1 + \dot{x}_2 \\ \lambda_2 \dot{x}_3 + \dot{x}_4 \end{bmatrix} = 0 \tag{III.4}$$

On replace (I.17) dans (III.4)

$$\begin{bmatrix} \lambda_1 x_2 + g_{11} V_p + g_{12} V_y + f_1 \\ \lambda_2 x_4 + g_{21} V_p + g_{22} V_y + f_2 \end{bmatrix} = 0 \tag{III.5}$$

Ce qui donne

$$\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} V_p \\ V_y \end{bmatrix} = \begin{bmatrix} -\lambda_1 x_2 - f_1 \\ -\lambda_2 x_4 - f_2 \end{bmatrix} \tag{III.6}$$

Où $g = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$

Les commandes V_{peq}et V_{yeq} existent si g est inversible c.à.d. $det(g) \neq 0$

$$\begin{bmatrix} V_{peq} \\ V_{yeq} \end{bmatrix} = -\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1} \begin{bmatrix} \lambda_1 x_2 + f_1 \\ \lambda_2 x_4 + f_2 \end{bmatrix} \tag{III.7}$$

La relation (III.7) peut être réécrite comme suit :

$$\begin{bmatrix} V_{peq} \\ V_{yeq} \end{bmatrix} = \frac{1}{g_{12}g_{21} - g_{11}g_{22}} \begin{bmatrix} g_{22} & -g_{12} \\ -g_{21} & g_{11} \end{bmatrix} \begin{bmatrix} \lambda_1 x_2 + f_1 \\ \lambda_2 x_4 + f_2 \end{bmatrix} \quad \text{(III.8)}$$

Donc l'expression de la commande équivalente pour l'SMC est donnée par

$$\begin{aligned} V_{peq} &= \frac{1}{g_{12}g_{21} - g_{11}g_{22}} \left(g_{22}(\lambda_1 x_2 + f_1) - g_{12}(\lambda_2 x_4 + f_2) \right) \\ V_{yeq} &= \frac{1}{g_{12}g_{21} - g_{11}g_{22}} \left(-g_{21}(\lambda_1 x_2 + f_1) + g_{11}(\lambda_2 x_4 + f_2) \right) \end{aligned} \quad \text{(III.9)}$$

- *Commande discontinue*

$$\begin{aligned} V_{d\theta} &= -K_1 sign(S_\theta) \\ V_{d\psi} &= -K_2 sign(S_\psi) \end{aligned} \quad \text{(III.10)}$$

- *Commande totale*

La commande totale est donnée par (III.11) et son schéma de contrôle est illustré par la figure III.2 :

$$\begin{aligned} V_p &= \left\{ \frac{1}{g_{12}g_{21} - g_{11}g_{22}} \left[g_{22}(\lambda_1 x_2 + f_1) - g_{12}(\lambda_2 x_4 + f_2) \right] \right\} \\ &\quad - K_1 sign(S_\theta) \\ V_y &= \left\{ \frac{1}{g_{12}g_{21} - g_{11}g_{22}} \left[-g_{21}(\lambda_1 x_2 + f_1) + g_{11}(\lambda_2 x_4 + f_2) \right] \right\} \\ &\quad - K_2 sign(S_\psi) \end{aligned} \quad \text{(III.11)}$$

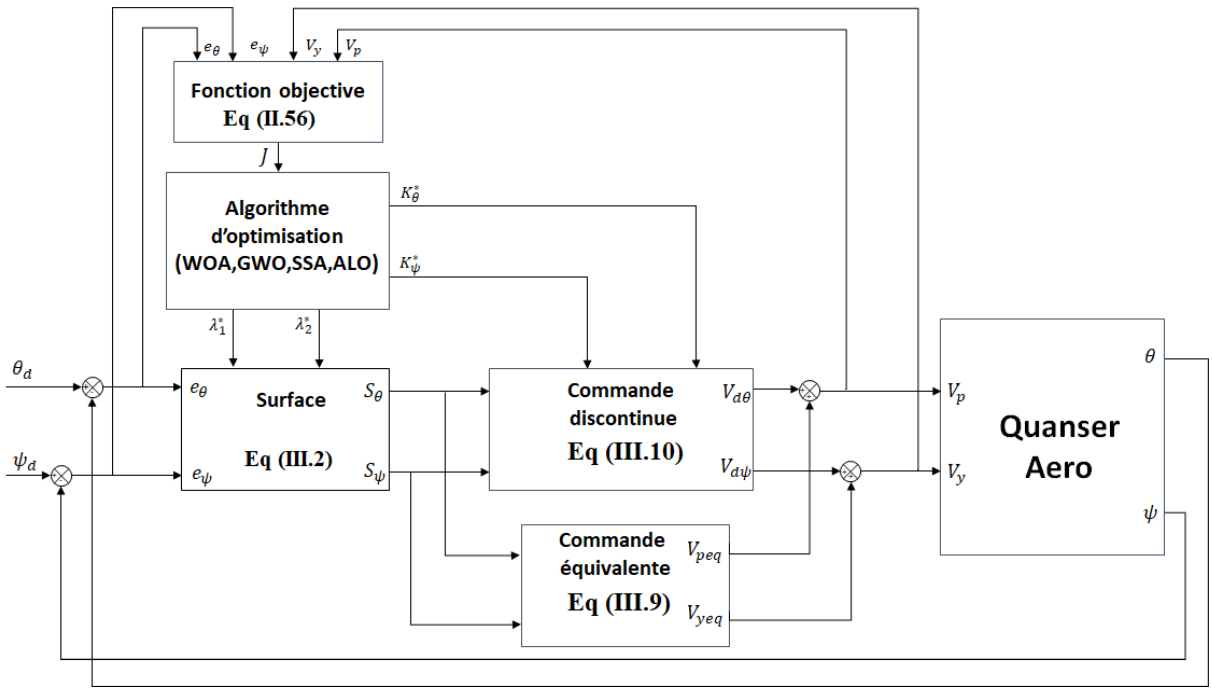


Figure III. 2 Structure de la commande SMC

III.3.2. Commande par NTSMC

- *Synthèse de la commande équivalente*

On utilise l'équation (II.22), les surfaces associées à chaque sous système est

$$\begin{aligned} S_\theta &= |x_2|^{\gamma_1} sign(x_2) + \lambda_1 x_1 \\ S_\psi &= |x_4|^{\gamma_2} sign(x_4) + \lambda_2 x_3 \end{aligned} \tag{III.12}$$

On dérive la relation (III.12), on obtient

$$\begin{aligned} [\lambda_1 \quad \gamma_1 |x_2|^{\gamma_1 - 1} sign(x_2)] \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = 0 \\ [\lambda_2 \quad \gamma_2 |x_4|^{\gamma_2 - 1} sign(x_4)] \begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = 0 \end{aligned} \tag{III.13}$$

Où encore

$$\begin{bmatrix} \lambda_1 \dot{x}_1 + \gamma_1 |x_2|^{\gamma_1 - 1} \dot{x}_2 sign(x_2) \\ \lambda_2 \dot{x}_3 + \gamma_2 |x_4|^{\gamma_2 - 1} \dot{x}_4 sign(x_4) \end{bmatrix} = 0 \tag{III.14}$$

On remplace (I.17) dans (III.14)

$$\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} V_p \\ V_y \end{bmatrix} = - \begin{bmatrix} f_1 + \lambda_1 {\gamma_1}^{-1} |x_2|^{2-\gamma_1} sign(x_2) \\ f_2 + \lambda_2 {\gamma_2}^{-1} |x_4|^{2-\gamma_2} sign(x_4) \end{bmatrix} \tag{III.15}$$

Les commandes $V_p$et V_y existent si g est inversible, elles sont données par (III. 16)

$$\begin{bmatrix} V_p \\ V_y \end{bmatrix} = - \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1} \begin{bmatrix} f_1 + \lambda_1 {\gamma_1}^{-1} |x_2|^{2-\gamma_1} sign(x_2) \\ f_2 + \lambda_2 {\gamma_2}^{-1} |x_4|^{2-\gamma_2} sign(x_4) \end{bmatrix} \tag{III.16}$$

Par conséquent

$$\begin{bmatrix} V_{peq} \\ V_{yeq} \end{bmatrix} = \frac{1}{g_{12} g_{21} - g_{11} g_{22}} \begin{bmatrix} g_{22} & -g_{12} \\ -g_{21} & g_{11} \end{bmatrix} \begin{bmatrix} f_1 + \lambda_1 {\gamma_1}^{-1} |x_2|^{2-\gamma_1} sign(x_2) \\ f_2 + \lambda_2 {\gamma_2}^{-1} |x_4|^{2-\gamma_2} sign(x_4) \end{bmatrix} \tag{III.17}$$

Donc l'expression de la commande équivalente, pour le cas de l'NTSMC est donnée par

$$\begin{aligned} V_{peq} = \frac{1}{g_{12} g_{21} - g_{11} g_{22}} \Big(& g_{22} \big(f_1 + \lambda_1 {\gamma_1}^{-1} |x_2|^{2-\gamma_1} sign(x_2) \big) \\ & - g_{12} \big(f_2 + \lambda_2 {\gamma_2}^{-1} |x_4|^{2-\gamma_2} sign(x_4) \big) \Big) \\ V_{yeq} = \frac{1}{g_{12} g_{21} - g_{11} g_{22}} \Big(& -g_{21} \big(f_1 + \lambda_1 {\gamma_1}^{-1} |x_2|^{2-\gamma_1} sign(x_2) \big) \\ & + g_{11} \big(f_2 + \lambda_2 {\gamma_2}^{-1} |x_4|^{2-\gamma_2} sign(x_4) \big) \Big) \end{aligned} \tag{III.18}$$

- *Commande discontinue*

La commande discontinue de l'NTSMC est la même que celle obtenue dans de l'SMC, elle est donnée par l'équation (III.10), sauf que la surface glissement est celle de l'équation (III. 12)

- *Commande totale*

La commande totale par NTSMC est donnée par (III.19) et son schéma est illustré par la figure III.3:

$$V_p = \left\{\frac{1}{g_{12}g_{21} - g_{11}g_{22}}\left[g_{22}\left(f_1 + \lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2)\right)\right.\right.$$
$$\left.\left. - g_{12}\left(f_2 + \lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4)\right)\right]\right\} - K_1 sign(S_\theta)$$
$$V_y = \left\{\frac{1}{g_{12}g_{21} - g_{11}g_{22}}\left[-g_{21}\left(f_1 + \lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2)\right)\right.\right.$$
$$\left.\left. + g_{11}\left(f_2 + \lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4)\right)\right]\right\} - K_2 sign(S_\psi) \quad \text{(III.19)}$$

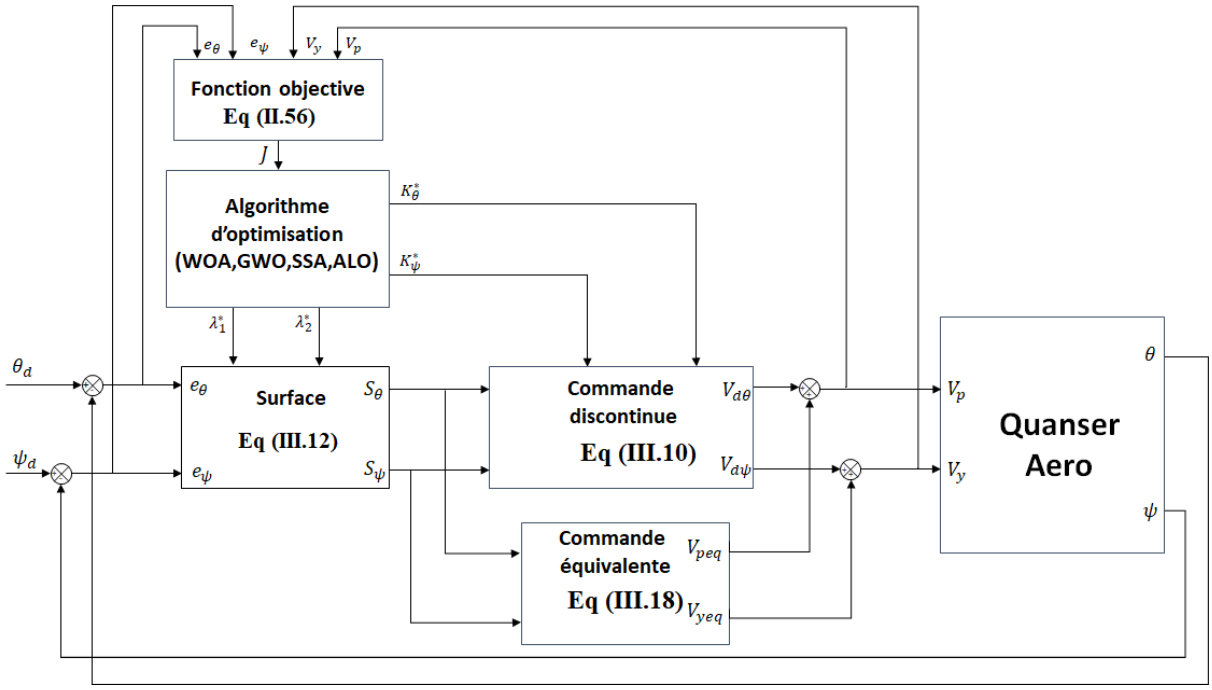


Figure III. 3 Structure de la commande NTSMC

III.3.3. Commande par l'algorithme Super Twisting (STW)

- *Synthèse de la commande équivalente*

La commande équivalente est la même que celle de l'SMC de l'équation (III.9).

- *Commande discontinue*

Utilisant la relation (II.24) ont peut déduire la commande discontinue pour chaque sous-système comme suit

$$V_{d\theta} = -\frac{\alpha_1}{2}|S_\theta|^{\frac{1}{2}}sign(S_\theta) - \int \frac{\beta_1}{2}\, sign\,(S_\theta)$$
$$V_{d\psi} = -\frac{\alpha_2}{2}|S_\psi|^{\frac{1}{2}}sign(S_\psi) - \int \frac{\beta_2}{2} sign\,(S_\psi) \quad \text{(III.20)}$$

- *Commande totale*

La commande totale représente la somme de la commande équivalente (III.9) et la commande discontinue (III.20) elle est donnée par :

$$V_p = \left\{\frac{1}{g_{12}g_{21} - g_{11}g_{22}}\left[g_{22}(\lambda_1 x_2 + f_1) - g_{12}(\lambda_2 x_4 + f_2)\right]\right\} - \frac{\alpha_1}{2}|S_1|^{\frac{1}{2}} sign(S_\theta) - \int \frac{\beta_1}{2}\, sign\,(S_\theta)$$

$$V_y = \left\{\frac{1}{g_{12}g_{21} - g_{11}g_{22}}\left[-g_{21}(\lambda_1 x_2 + f_1) + g_{11}(\lambda_2 x_4 + f_2)\right]\right\} - \frac{\alpha_2}{2}|S_2|^{\frac{1}{2}} sign(S_\psi) - \int \frac{\beta_2}{2} sign\,(S_\psi) \quad \text{(III.21)}$$

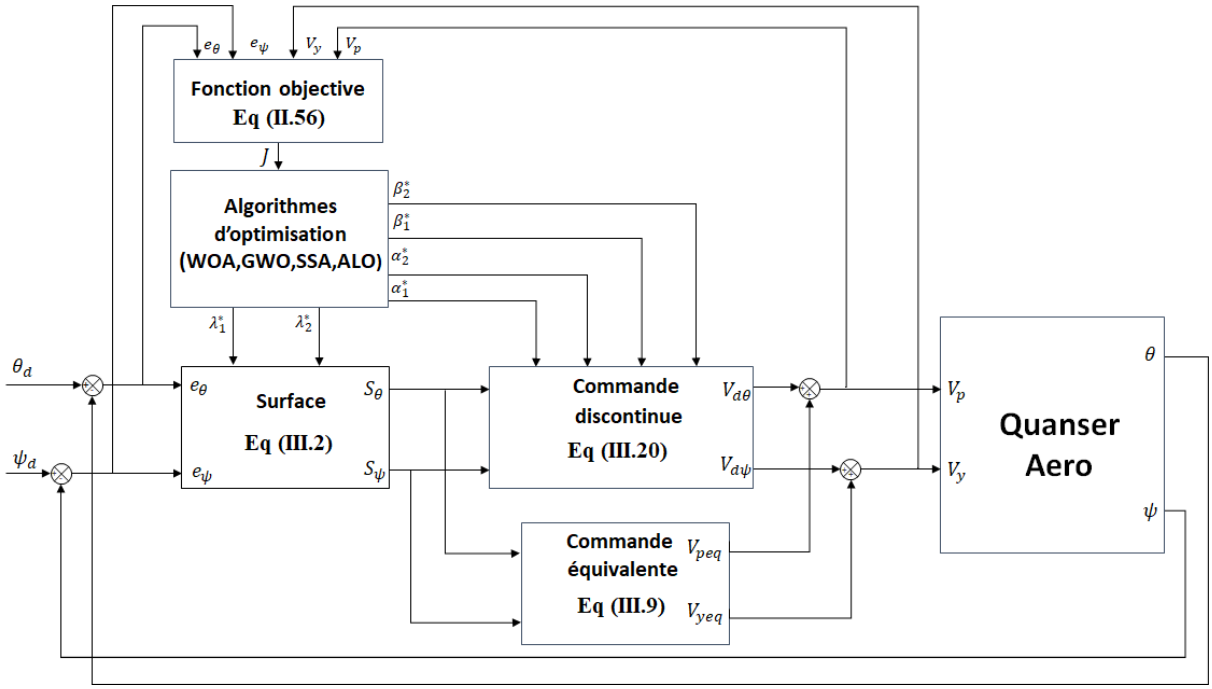


Figure III. 4 Structure de la commande STW

III.3.4. Commande par nonsingular terminal super twisting (NTSTW)

- *Synthèse de la commande équivalente*

La commande équivalente est la même que celle de l'NTSMC, donnée par (III.18).

- *Commande discontinue*

La commande discontinue dans ce cas est similaire à celle donnée par la relation (III.20)

- *Commande totale*

La commande totale représente la somme de la commande équivalente et la commande discontinue elle est donnée par (III.22) :

$$V_p = \left\{\frac{1}{g_{12}g_{21} - g_{11}g_{22}}\left[g_{22}\left(f_1 + \lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2)\right)\right.\right.$$
$$\left.\left. - g_{12}\left(f_2 + \lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4)\right)\right]\right\} - \frac{\alpha_1}{2}|S_1|^{\frac{1}{2}}sign(S_\theta)$$
$$- \int \frac{\beta_1}{2}\, sign\,(S_\theta)$$
$$V_y = \left\{\frac{1}{g_{12}g_{21} - g_{11}g_{22}}\left[-g_{21}\left(f_1 + \lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2)\right)\right.\right.$$
$$\left.\left. + g_{11}\left(f_2 + \lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4)\right)\right]\right\} - \frac{\alpha_2}{2}|S_2|^{\frac{1}{2}}sign(S_\psi)$$
$$- \int \frac{\beta_2}{2} sign\,(S_\psi)$$

(III.22)

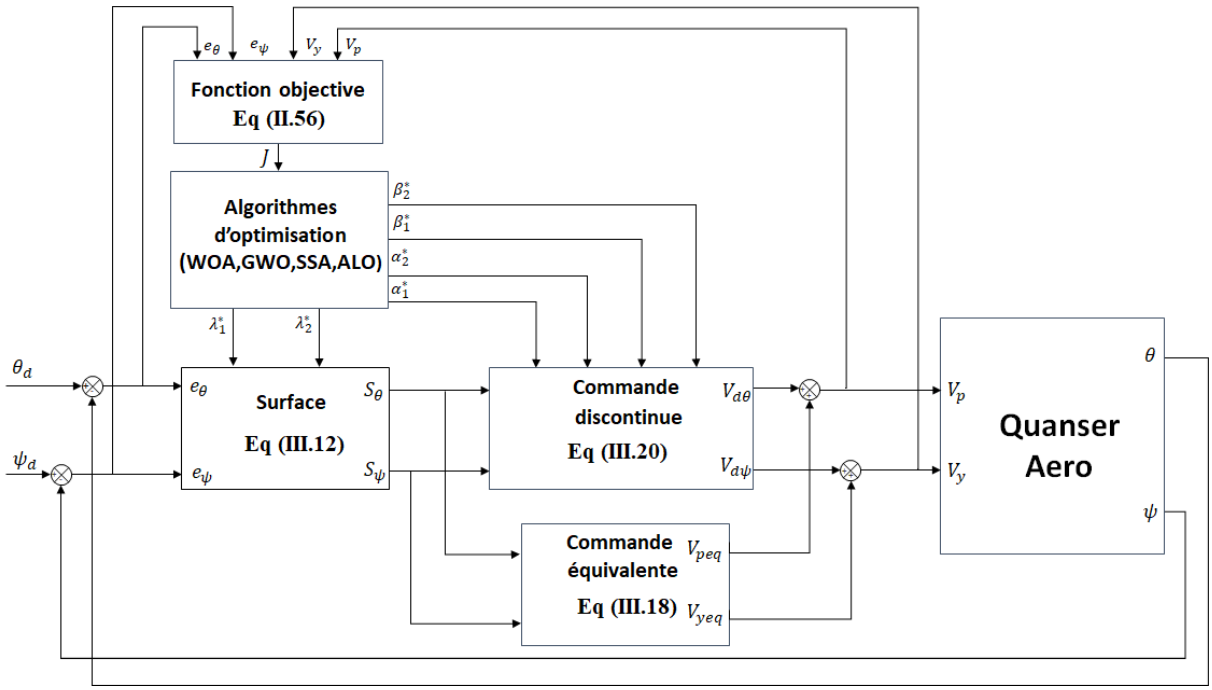


Figure III. 5 Structure de la commande NTSTW

III.4. Synthèse d'une loi de commande Adaptative

La commande des systèmes non linéaires incertains/ perturbés est une tâche difficile. Afin d'améliorer les performances notamment en termes de précision et remédier aux incertitudes, nous proposons une commande basée sur le super twisting adaptative [38], avec une surface de glissement de type NTSMC. Cette commande va permettre d'atténuer les broutements et garantir une convergence rapide en temps fini. Les gains de la commande Nonsingular Terminal Super-Twisting Adaptive (ANTSTW) sont donnés par la loi :

$$\dot{\alpha} = \begin{cases} \omega_1\sqrt{\frac{\gamma_1}{2}}sign(|S| - \mu), & if\ \alpha > \alpha_m \\ \eta, & if\ \alpha \le \alpha_m \end{cases} \qquad \beta = 2\varepsilon\alpha$$

(III.23)

Où $\omega_1, \eta, \varepsilon, \gamma_1, \alpha_m$ et μ sont des constantes positives arbitraires, $|S(0)| > \mu$ et $\alpha(0) > \alpha_m$

L'idée de la conception de la commande ANTSTW est d'augmenter dynamiquement les gains de contrôle $\alpha(t)$ et $\beta(t)$ jusqu'à ce que le mode glissement s'établi. Ensuite, les gains commenceront à diminuer. Cette réduction de gain doit être inversée dès que la variable glissante ou sa dérivée commence à s'écarter du point d'équilibre $\dot{S} = 0$. Ce qui permet de réduire l'amplitude du broutement puisque les gains ne sont pas surestimés mais ajustés aux incertitudes et aux perturbations. La structure de la commande ANTSTWO est illustrée par la figure (III.6).

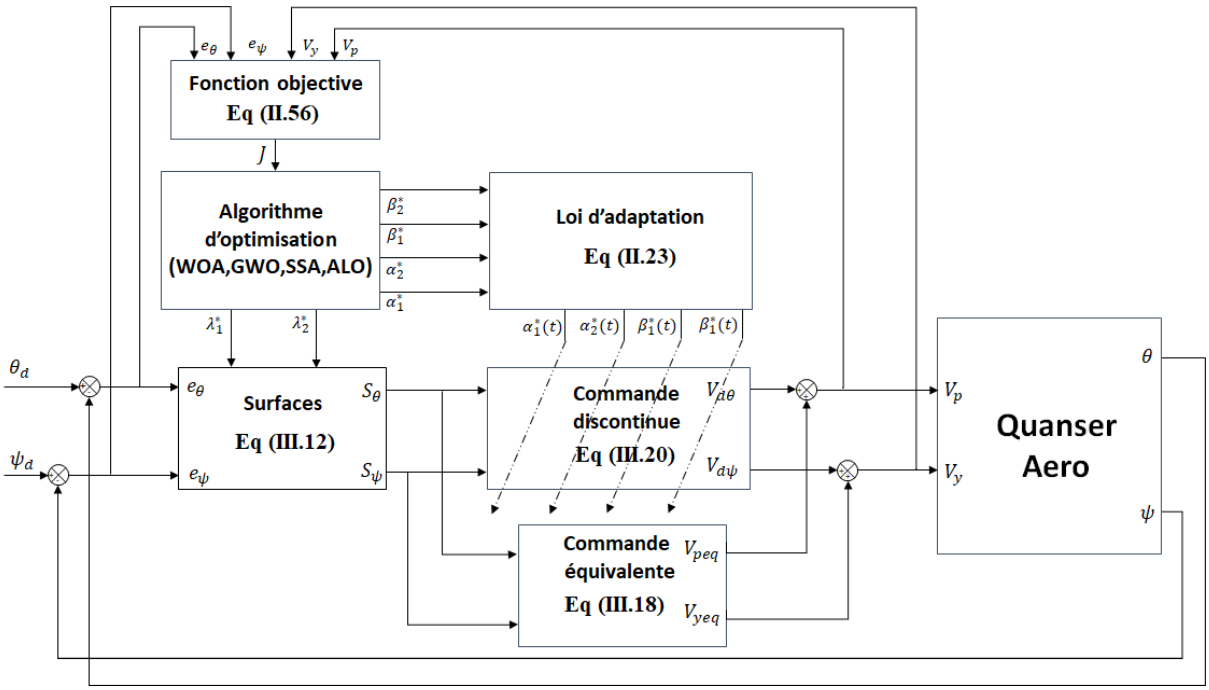


Figure III. 6 Structure de la commande ANTSTWO

Afin d'avoir une commande à la fois optimale et adaptative, nous avons initialisé les gains de la commande ANTSTW avec les paramètres optimaux de la commande NTSTW. Cette manière va permettre d'éviter le problème de recherche des paramètres de départ pour la commande adaptative. La commande ANTSTWO représente une véritable contribution dans ce travail. Les gains adaptatifs optimaux sont de la forme suivante :

$$\begin{aligned} \alpha_1^*(t) &= \alpha_1(t) + \alpha_1^* \\ \beta_1^*(t) &= \beta_1(t) + \beta_1^* \\ \alpha_2^*(t) &= \alpha_2(t) + \alpha_2^* \\ \beta_2^*(t) &= \beta_2(t) + \beta_2^* \end{aligned} \qquad \text{(III.24)}$$

Ou $\alpha_1^*, \beta_1^*, \alpha_2^*, \beta_2^*$ sont des gains optimaux de la commande ANTSTW, et $\alpha_1(t), \beta_1(t), \alpha_2(t), \beta_2(t)$ sont obtenus à partir de la lois d'adaptation de l'équation (III.23).

III.4.1. Preuve de la loi d'adaptation

Considérons le système non linéaire incertain suivant

$$\dot{x} = f(x,t) + g(x,t)u \tag{III.25}$$

Ou x est le vecteur d'état, u la commande, f représente la partie non linéaire différentiable. On suppose que :

- **Hypothèse 1.** Une variable glissante S est conçue pour que la dynamique du système (III.25) désirée soit atteinte dans le mode glissant $S\ (x,t)\ =\ 0$.
- **Hypothèse 2.** Le degré relatif du système (III.25) avec la variable glissante $S(x,t)$ égal à un par rapport à *u*, et la dynamique interne est stable.

Si on dérive *S*, on obtient :

$$\frac{dS}{dx}\frac{dx}{dt} = 0 \tag{III.26}$$

On remplace (III.25) dans l'équation (III.26), on a

$$\frac{dS}{dx}f(x,t) + \frac{dS}{dx}g(x,t)u = 0 \tag{III.27}$$

On ajout $\dot{S}$ aux deux côtés de l'égalité (III.27), on obtient

$$\dot{S} = \frac{dS}{dt} + \frac{dS}{dx}f(x,t) + \frac{dS}{dx}g(x,t)u \tag{III.28}$$

Posons $a(x,t) = \frac{dS}{dt} + \frac{dS}{dx}f(x,t)$ et $b(x,t) = \frac{dS}{dx}g(x,t)$

Donc

$$\dot{S} = a(x,t) + b(x,t)u \tag{III.29}$$

- **Hypothèse 3.** La fonction $b(x,t)$ est incertaine et peut être représentée comme suit

$$b(x,t) = b_0(x,t) + \Delta b(x,t) \tag{III.30}$$

Où $b_0(x,t) > 0$ est une fonction connue et $\Delta b(x,t)$ est une perturbation bornée, telle que

$$\left|\frac{\Delta b(x,t)}{b_0(x,t)}\right| = |\gamma(x,t)| \leq \gamma_1 < 1 \tag{III.31}$$

Avec γ_1 est une borne inconnue.

- **Hypothèse 4.** La fonction $a(x,t)$ est représentée par

$$a(x,t) = a_1(x,t) + a_2(x,t) \tag{III.32}$$

Avec les termes bornés suivants :

$$|a_1(x,t)| \leq \delta_1 |S|^{1/2} \qquad |a_2(x,t)| \leq \delta_2 \tag{III.33}$$

Où les bornes finies δ_1, $\delta_2 > 0$ existent mais ne sont pas connues.

Enfin, on obtient

$$\dot{S} = a(x,t) + b_1(x,t)\omega \tag{III.34}$$

Où $\omega = b_0(x,t)u,\ b_1(x,t) = 1 + \frac{\Delta b(x,t)}{b_0(x,t)}$

- **Hypothèse 5.**

De (III.34) et (III.31) on obtient :

$$1 - \gamma_1 \leq b_1(x,t) \leq 1 + \gamma_1 \tag{III.35}$$

Le problème est de ramener la variable glissante et sa dérivée à zéro en temps fini, en la présence des perturbations bornées, avec des bornes inconnues δ_1, δ_2, $\gamma_1 > 0$, au moyen d'une commande adaptative.

III.4.2. Structure de la commande

Soit la commande Super Twisting présentée par

$$\begin{aligned} \omega &= -\alpha|S|^{\frac{1}{2}}sign(S) + v \\ \dot{v} &= \beta\ sign(S) \end{aligned} \tag{III.36}$$

Où les gains adaptatifs sont

$$\begin{aligned} \alpha &= \alpha\big(S,\dot{S},t\big) \\ \beta &= \beta\big(S,\dot{S},t\big) \end{aligned} \tag{III.37}$$

La solution du système (III.34) se calcule au sens de (Filippov) [16]. En supposant que, la fonction incertaine $b_1(x,t)$ est une constante incertaine par morceaux. On peut réécrire l'équation (III.34) sous la forme (III.38) en utilisant et le (III.36):

$$\begin{aligned} \dot{S} &= -\alpha b_1(x,t)|S|^{\frac{1}{2}}sign(S) + \omega_* + a_1(x,t) \\ \dot{\omega}_* &= \frac{\beta b_1(x,t)}{2}sign(S) + \dot{a}_2(x,t) + b_1(x,t)v \\ \omega_*(0) &= 0 \end{aligned} \tag{III.38}$$

Où $\omega_* = a_2 + b_1(x,t)v$,

Supposons que le terme $\dot{b}_1 v$ soit borné avec une borne inconnue $\delta_3 \geq 0$, on peut écrire

$$\left|\dot{b}_1 v\right| \leq \frac{\left|\dot{b}_1\right|}{2}\int_0^t \beta d\tau \leq \delta_3 \tag{III.39}$$

Le gain adaptatif β est borné par une borne incertaine $\beta^* > 0$, Donc

$$\left|\dot{b}_1 v\right| \leq \frac{\left|\dot{b}_1\right|}{2}\beta^* t \leq \delta_3 \tag{III.40}$$

Finalement, les bornes de la fonction $\dot{\chi}(x,t) = \dot{a}_2(x,t) + b_1(x,t)v$ existent, mais elles sont inconnues,

$$|\dot{\chi}(x,t)| \leq \delta_2 + \delta_3 = \delta_4 \tag{III.41}$$

III.4.3. Preuve de stabilité

Soit le changement de variables suivant

$$z = [z_1 \quad z_2]^T = \left[|S|^{\frac{1}{2}} sign(S) \quad \omega_*\right]^T \tag{III.42}$$

Le système (III.38) peut être mis sous la forme :

$$\begin{aligned} \dot{z}_1 &= \frac{1}{2|z_1|}(-\alpha b_1 + z_2 + a_1(x,t)) \\ \dot{z}_2 &= -\frac{\beta b_1}{2|z_1|} + z_1 + \dot{\chi}(x,t) \end{aligned} \tag{III.43}$$

Où $\dot{\chi}(x,t) = \dot{a}_2(x,t) + \dot{b}_1(x,t)v$, l'équation (III.43) peut être réécrite sous la forme matricielle

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \frac{1}{2|z_1|}\begin{bmatrix} -\alpha b_1 & 1 \\ \beta b_1 & 0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \frac{1}{2|z_1|}\begin{bmatrix} 1 & 0 \\ 0 & 2|z_1| \end{bmatrix}\begin{bmatrix} a_1(x,t) \\ \dot{\chi}(x,t) \end{bmatrix} \tag{III.44}$$

A partir de l'hypothèse 4 et (III.44), nous pouvons écrire

$$\begin{aligned} a_1(x,t) &= \rho_1(x,t)|S|^{\frac{1}{2}} sign(S) = \rho_1(x,t) z_1 \\ \dot{\chi}(x,t) &= \frac{\rho_2(x,t)}{2} sign(S) = \frac{\rho_2(x,t)}{2}\frac{z_1}{|z_1|} \end{aligned} \tag{III.45}$$

Où $\rho_1(x,t)$, $\rho_2(x,t)$ sont des fonctions bornées, telles que :

$$0 < \rho_2(x,t) < \delta_1, \quad 0 < \rho_2(x,t) < 2\delta_4$$

L'équation (III.44) peut être réécrite en utilisant l'équation (III.45)

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \check{A}(z_1)\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{III.46}$$

Où

$$\check{A}(z_1) = \frac{1}{2|z_1|}\begin{bmatrix} -(\alpha b_1 - \rho_2(x,t)) & 1 \\ -(\beta b_1 - \rho_2(x,t)) & 0 \end{bmatrix}$$

Remarque importante : On peut observer que

- Si $z_1, z_2 \to 0$ dans un temps fini, alors $S, \dot{S} \to 0$ en temps fini.

- $|z_1| = |S|^{\frac{1}{2}}$ et $sign(z_1) = sign(S)$

Pour l'analyse de la stabilité du système (III.44), la fonction candidate de Lyapunov suivante est proposée

$$V(z_1, z_1, \alpha, \beta) = V_0 + \frac{1}{2\gamma_1}(\alpha - \alpha^*)^2 + \frac{1}{2\gamma_2}(\beta - \beta^*)^2 \tag{III.47}$$

Avec

$$V_0 = (\lambda + 4\varepsilon^2)z_1^2 + z_2^2 - 4\varepsilon z_1 z_1 = z^T P z$$
$$P = \begin{bmatrix} \lambda + 4\varepsilon^2 & -2\varepsilon \\ -2\varepsilon & 1 \end{bmatrix} \tag{III.48}$$

Où $\varepsilon, \lambda, \gamma_1, \gamma_2, \alpha^*, \beta^* > 0$, sont des constantes positives. Il est clair que la matrice P est définie positive si $\lambda > 0$ quelle que soit la valeur de ε. La dérivée de la fonction de Lyapunov est

$$\dot{V}(z_1, z_1, \alpha, \beta) = z^T\left[\check{A}^T(z_1)P + P\check{A}(z_1)\right]z + \frac{1}{\gamma_1}(\alpha - \alpha^*)\dot{\alpha} + \frac{1}{\gamma_2}(\beta - \beta^*)\dot{\beta} \tag{III.49}$$

Le 1[ier] terme de $\dot{V}_0$ peut s'écrire

$$\dot{V}_0 = z^T\left[\check{A}^T(z_1)P + P\check{A}(z_1)\right]z \leq -\frac{1}{2|z_1|}z^T\check{Q}z \tag{III.50}$$

La matrice symétrique $\check{Q}$ est calculée en tenant compte de (III.33) et (III.45)

$$\check{Q} = \begin{bmatrix} \check{Q}_{11} & \check{Q}_{12} \\ \check{Q}_{21} & 4\varepsilon \end{bmatrix} \tag{III.51}$$

Avec

$$\check{Q}_{11} = 2\lambda\alpha b_1 + 4\varepsilon b_1(2\varepsilon\alpha - \beta) - 2(\lambda + 4\varepsilon^2)\rho_1(x,t) + 4\varepsilon\rho_2(x,t)$$
$$\check{Q}_{12} = \check{Q}_{21} = (\beta b_1 - 2\varepsilon\alpha b_1 - \lambda - 4\varepsilon^2) + 2\varepsilon\rho_1(x,t) - \rho_2(x,t)$$

On prend

$$\beta = 2\varepsilon\alpha \tag{III.52}$$

La matrice $\check{Q}$ est une matrice définie positive, avec la valeur propre minimal $\lambda_{min}(\check{Q}) \geq 2\varepsilon$ si

$$\alpha > \frac{\delta_1(\lambda + 4\varepsilon^2) - \varepsilon(4\delta_4 + 1)}{\lambda(1 - \gamma_1)} + \frac{[2\varepsilon\delta_1 - 2\delta_4 - \lambda - 4\varepsilon^2]^2}{12\varepsilon\lambda(1 - \gamma_1)} \tag{III.53}$$

En utilisant les relations (III.52), (III.53) et (III.50), il est facile de prouver que

$$\dot{V}_0 \leq -rV_0^{1/2} \tag{III.54}$$

Où

$$r = \frac{\lambda_{min}^{\frac{1}{2}}(P)}{\lambda_{max}} \tag{III.55}$$

Donc

$$\dot{V}_0 \leq -\frac{1}{2|z_1|}z^T\check{Q}z \leq -\frac{2\varepsilon}{2|z_1|}z^Tz = -\frac{\varepsilon}{|z_1|}\|z\|^2 \tag{III.56}$$

Et

$$\lambda_{min}(P)\|z\|^2 \leq z^T P z \leq \lambda_{max}(P)\|z\|^2 \tag{III.57}$$

Où $\|z\|^2 = z_1^2 + z_2^2 = |S| + z_2^2$

$$|z_1| = |S|^{1/2} \leq \|z\|^2 \leq \frac{V_0^{1/2}}{\lambda_{min}^{1/2}(P)} \tag{III.58}$$

Par conséquent

$$\dot{V}_0(z) \leq -rV_0^{1/2}, \quad r = \frac{\lambda_{min}^{\frac{1}{2}}(P)}{\lambda_{max}} \tag{III.59}$$

Posons $\varepsilon_\alpha = \alpha - \alpha^*$ et $\varepsilon_\beta = \beta - \beta^*$,

$$\begin{aligned}\dot{V}(z_1, z_1, \alpha, \beta) &= \dot{z}^T P z + z^T P \dot{z} + \frac{1}{\gamma_1}\varepsilon_\alpha \dot{\alpha} + \frac{1}{\gamma_2}\varepsilon_\beta \dot{\beta} \\ &\leq \frac{1}{|z_1|} z^T \check{Q} z + \frac{1}{\gamma_1}\varepsilon_\alpha \dot{\alpha} + \frac{1}{\gamma_2}\varepsilon_\beta \dot{\beta} \\ &\leq -rV_0^{1/2} + \frac{1}{\gamma_1}\varepsilon_\alpha \dot{\alpha} + \frac{1}{\gamma_2}\varepsilon_\beta \dot{\beta} \\ = -rV_0^{\frac{1}{2}} - \frac{\omega_1}{\sqrt{2\gamma_1}}|\varepsilon_\alpha| - \frac{\omega_2}{\sqrt{2\gamma_2}}|\varepsilon_\beta| &+ \frac{1}{\gamma_1}\varepsilon_\alpha \dot{\alpha} + \frac{1}{\gamma_2}\varepsilon_\beta \dot{\beta} + \frac{\omega_1}{\sqrt{2\gamma_1}}|\varepsilon_\alpha| \\ &+ \frac{\omega_2}{\sqrt{2\gamma_2}}|\varepsilon_\beta|\end{aligned} \tag{III.60}$$

En tenant compte de l'inégalité bien connue $(x^2 + y^2 + z^2)^{1/2} \leq |x| + |y| + |z|$, Au vu de l'équation, on peut déduire

$$-rV_0^{\frac{1}{2}} - \frac{\omega_1}{\sqrt{2\gamma_1}}|\varepsilon_\alpha| - \frac{\omega_2}{\sqrt{2\gamma_2}}|\varepsilon_\beta| \leq -\eta_0\sqrt{V(z_1, z_1, \alpha, \beta)} \tag{III.61}$$

Ou $\eta_0 = min(r, \omega_1, \omega_2)$, prenons en considération l'équation (III.61), on peut réécrire l'équation (III.60) comme suit :

$$\begin{aligned}\dot{V}(z_1, z_1, \alpha, \beta) \leq -\eta_0\sqrt{V(z_1, z_1, \alpha, \beta)} + \frac{1}{\gamma_1}\varepsilon_\alpha \dot{\alpha} + \frac{1}{\gamma_2}\varepsilon_\beta \dot{\beta} + \frac{\omega_1}{\sqrt{2\gamma_1}}|\varepsilon_\alpha| \\ + \frac{\omega_2}{\sqrt{2\gamma_2}}|\varepsilon_\beta|\end{aligned} \tag{III.62}$$

On suppose que les gains adaptatifs $\alpha(t)$ et $\beta(t)$ sont bornés respectivement par α^* et β^*, qui donnent $\alpha(t) - \alpha^* < 0$ et $\alpha(t) - \beta^* < 0 \ \forall\, t > 0$.

$$\begin{aligned}\dot{V}(z_1, z_1, \alpha, \beta) \leq -\eta_0\sqrt{V(z_1, z_1, \alpha, \beta)} - |\varepsilon_\alpha|\left(\frac{1}{\gamma_1}\dot{\alpha} - \frac{\omega_1}{\sqrt{2\gamma_1}}\right) \\ - |\varepsilon_\beta|\left(\frac{1}{\gamma_2}\dot{\beta} - \frac{\omega_2}{\sqrt{2\gamma_2}}\right)\end{aligned} \tag{III.63}$$

Ce qui donne

$$\dot{V}(z_1, z_1, \alpha, \beta) \leq -\eta_0 \sqrt{V(z_1, z_1, \alpha, \beta)} + \xi \quad \text{(III.64)}$$

Avec

$$\xi = -|\varepsilon_\alpha| \left(\frac{1}{\gamma_1} \dot{\alpha} - \frac{\omega_1}{\sqrt{2\gamma_1}} \right) - |\varepsilon_\beta| \left(\frac{1}{\gamma_2} \dot{\beta} - \frac{\omega_2}{\sqrt{2\gamma_2}} \right) \quad \text{(III.65)}$$

- Pour $|S| \geq \mu$ et $\alpha(t) \geq \alpha_m$, à partir de l'équation (III.24)

$$\dot{\alpha} = \omega_1 \sqrt{\frac{\gamma_1}{2}} \quad \text{(III.66)}$$

Donc l'équation (III.65) va devenir

$$\xi = -|\varepsilon_\beta| \left(\frac{1}{\gamma_2} \dot{\beta} - \frac{\omega_2}{\sqrt{2\gamma_2}} \right) \quad \text{(III.67)}$$

En choisissant encore $\varepsilon = \frac{\omega_2}{\omega_1} \sqrt{\frac{\gamma_2}{\gamma_1}}$, et on dérive (III. 52)

$$\dot{\beta} = 2\varepsilon \dot{\alpha}$$
$$\dot{\beta} = \varepsilon \omega_1 \sqrt{2\gamma_1} \longrightarrow \dot{\beta} = \omega_2 \sqrt{\frac{\gamma_2}{2}} \quad \text{(III.68)}$$

Donc le terme ξ déviant nul et

$$\dot{V}(z_1, z_1, \alpha, \beta) \leq -\eta_0 \sqrt{V(z_1, z_1, \alpha, \beta)} \quad \text{(III.69)}$$

La stabilité au sens de Lyapunov est garantie, et le gain α augmente suivant la relation (III.70)

$$\alpha(t) = \alpha(0) + \omega_1 \sqrt{\frac{\gamma_1}{2}} \quad \text{(III.70)}$$

- Pour $|S| < \mu$ donc $\alpha(t)$ se calcule

$$\dot{\alpha} = \begin{cases} -\omega_1 \sqrt{\frac{\gamma_1}{2}}, & if\ \alpha > \alpha_m \\ \eta, & if\ \alpha \leq \alpha_m \end{cases} \quad \text{(III.71)}$$

Dès que $\alpha(t)$ devient inférieur ou égal à α_m, sa valeur commence immédiatement à augmenter de sorte que

$$\alpha(t) = \alpha_m + \eta t \quad \text{(III.72)}$$

III.5. Conclusion

Dans ce chapitre nous avons synthétisé les différentes lois de commande, à savoir, PID, SMC, NTSMC, STW et NTSTW pour le Quanser Aero. Le choix des commandes se fait dans l'objectif de trouver la commande la plus robuste vis-à-vis les perturbations externes, les erreurs de modélisation et qui garantit en même temps les meilleures performances possibles en la présence d'une dynamique non linéaire fortement couplée. Une nouvelle approche de commande robuste appelée ANTSTWO a été synthétisée, cette commande est à la fois optimale et adaptative.

CHAPITRE IV : Simulation et Discussions

VI.1. Introduction

Ce chapitre aura pour objectif de présenter les résultats de simulation de l'application des commandes synthétisées dans le chapitre III.

En premier lieu, les résultats de l'optimisation de la fonction objective permettant d'améliorer la rapidité et la consommation l'énergie seront présentés, ceci est afin effectuer une comparaison entre les algorithmes d'optimisation. En seconde lieu, on appliquera les différentes lois de commande sur le modèle dynamique du système (vu en détail dans le chapitre I). En troisième lieu, la nouvelle commande proposée sera simulée et comparée avec les commandes les plus performantes issues de l'optimisation.

IV.1. Résultats d'optimisation des paramètres des différentes lois de commande

Pour trouver le meilleur algorithme à adopter pour le système Aero Quanser, les quatre algorithmes méta-heuristiques (GWO, WOA, SSA et ALO) ont été utilisés pour chaque commande (PID, SMC, NTSMC, STW et NTSTW) en régulation afin de trouver les paramètres optimaux des commandes qui minimisent la fonction objective. L'expérience a été répétée dix fois pour chaque algorithme, ce qui va permettre d'éviter les minimums locaux et la convergence prématurée des algorithmes.

Nous avons utilisé la fonction objective donnée dans sa forme générale par l'équation (II.56), dans notre cas cette fonction est donnée comme suit :

$$J = \int_0^{tf} \left([t \quad t] Q \begin{bmatrix} |e_\theta| \\ |e_\psi| \end{bmatrix} + [V_p \quad V_y] R \begin{bmatrix} V_p \\ V_y \end{bmatrix} \right) dt$$

Les coefficients R et Q sont choisis de telle manière à garantir la précision, la rapidité et le temps de stabilité minimum sans que les entrées de commande V_p et V_y ne dépassent pas leurs valeurs maximales -24 et 24 volts respectivement.

Dans les expériences réalisées nous avons choisi les valeurs suivantes $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $= \begin{bmatrix} 0.0015 & 0 \\ 0 & 0.0015 \end{bmatrix}$, $tf = 20$. Donc la fonction J devient

$$J = \int_0^{tf} t|e_\theta| + t|e_\psi| + 0.0015{V_p}^2 + 0.0015{V_y}^2 dt$$

Chaque algorithme a été exécuté 10 fois sous les mêmes conditions suivantes :

- Le nombre d'agents de recherche : 30 agents,
- Le nombre d'itérations : 50 itérations,
- La même fonction objective J,
- Un PC dont les caractéristiques techniques sont : Processor Intel(R) Core (TM) i5-7200U CPU @ 2.50GHz, 8G de RAM

Les résultats de l'optimisation de la fonction objective J sont présentés dans les tableaux ci-dessous

Algorithme	meilleur J	Mauvaise J	Moyenne 10	Meilleur essai
GWO	**1,43020131**	**1,43302296**	**1,43098411**	6
WOA	1,44231994	1,54634437	1,49162604	6
SSA	1,43031	1,4467172	1,43549007	5
ALO	1,43219026	1,45242009	1,43832748	8

Tableau 3 Les valeurs de la fonction objective, cas de l'optimisation des paramètres du PID

Algorithme	meilleur J	Mauvaise J	Moyenne	Meilleur essai
GWO	2,8636291	**2,9222534**	**2,9040744**	6
WOA	**2,8463551**	3,5661384	2,9782770	9
SSA	2,8925872	5,2094203	3,4732022	6
ALO	3,0146715	4,6544082	3,4154672	4

Tableau 4 Les valeurs de la fonction objective, cas de l'optimisation des paramètres de SMC

Algorithme	meilleur J	Mauvaise J	Moyenne 10	Meilleur essai
GWO	**2,80034044**	**2,85683905**	**2,81685419**	4
WOA	2,85374485	2,95269486	2,87405766	1
SSA	2,8317953	7,03750848	3,60071992	1
ALO	2,81980906	5,63474808	3,54016723	8

Tableau 5 Les valeurs de la fonction objective, cas de l'optimisation des paramètres de NTSMC

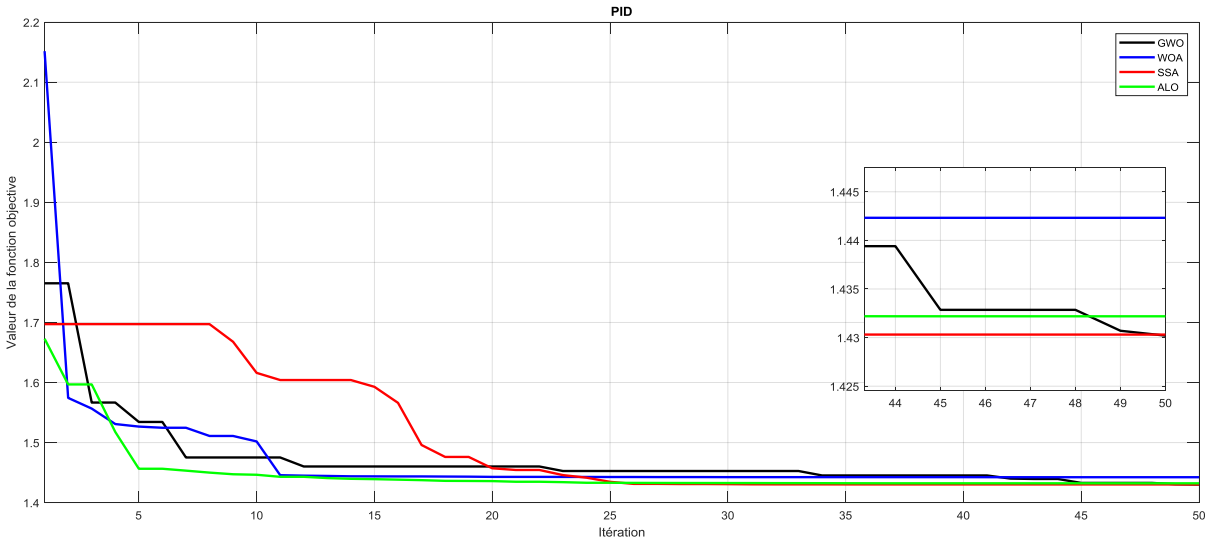
Algorithme	meilleur J	Mauvaise J	Moyenne 10	Meilleur essai
GWO	**2,51585141**	**2,52280305**	**2,51784165**	1
WOA	2,51922217	2,59324303	2,53927148	7
SSA	2,51776391	2,69928311	2,55970367	8
ALO	2,52657595	2,64169054	2,55694725	8

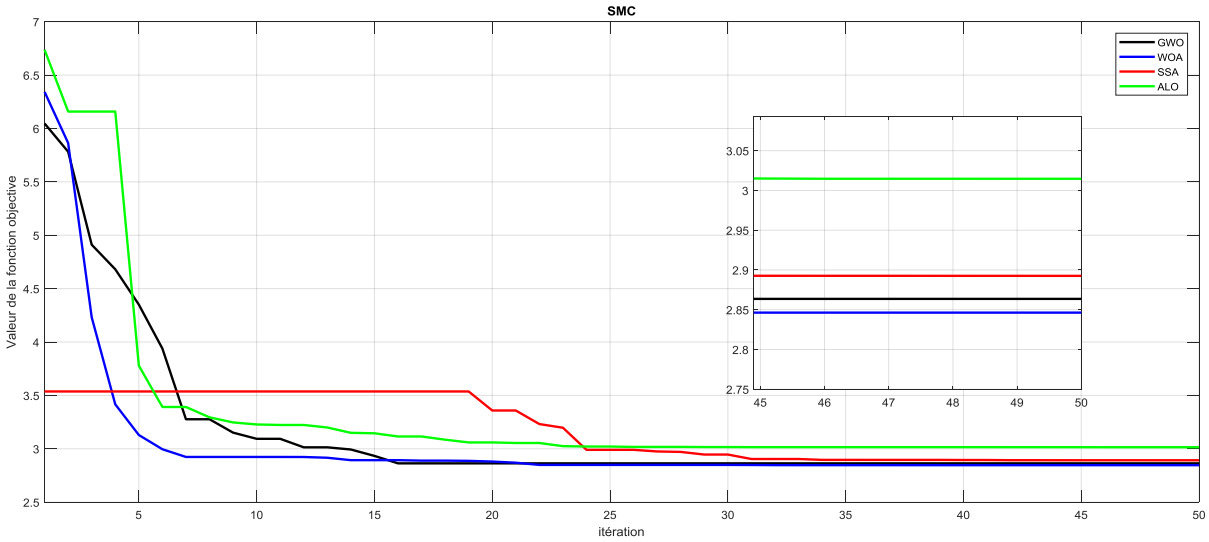
Tableau 6 Les valeurs de la fonction objective, cas de l'optimisation des paramètres de STW

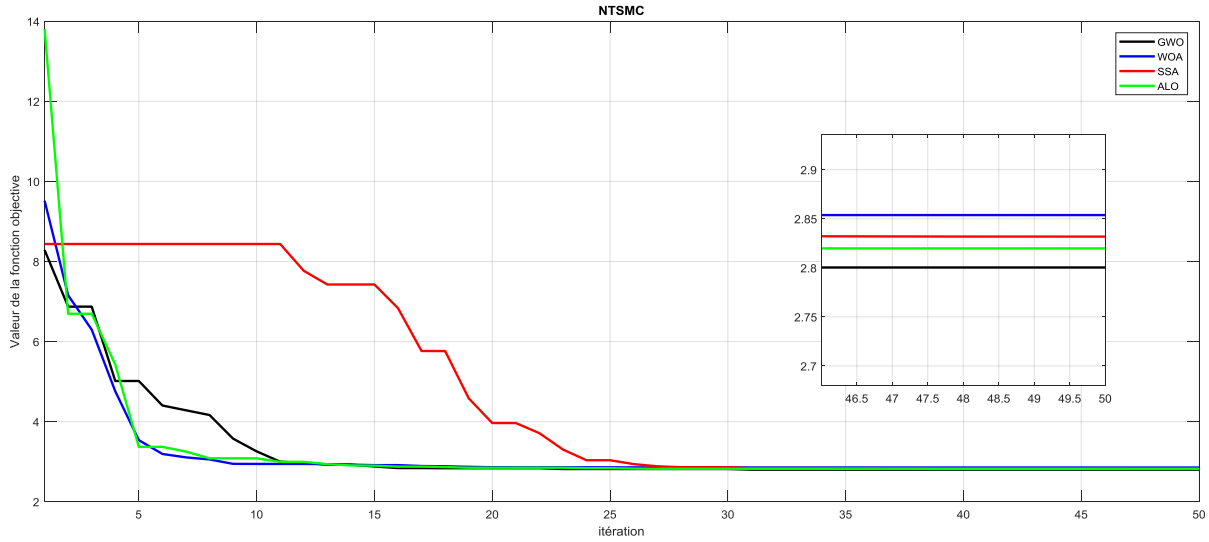
Algorithme	meilleur J	Mauvaise J	Moyenne 10	Meilleur essai
GWO	**2,51631223**	**2,52738979**	**2,518**	3
WOA	2,5231548	2,56126206	2,53344082	7
SSA	2,52004681	2,65526602	2,55480261	3
ALO	2,53813251	2,68847387	2,58276923	5

Tableau 7 les valeurs de la fonction objective, cas de l'optimisation des paramètres de NTSTW

Dans les tableaux 3 à 7, nous avons présenté trois valeurs de la fonction objective tels que, la meilleure valeur et le numéro de l'expérience où elle a été obtenue, la plus mauvaise valeur, ainsi que la moyenne des valeurs du J sur les 10 essais. Ces chiffres vont permettre de trancher d'une manière efficace sur quel est le meilleur algorithme par rapport à cette étude sachant que les expériences se font d'une façon aléatoire.


PID
GWO
WOA
SSA
ALO
Valeur de la fonction objective
Itération


SMC
GWO
WOA
SSA
ALO
Valeur de la fonction objective
itération


NTSMC
GWO
WOA
SSA
ALO
Valeur de la fonction objective
itération

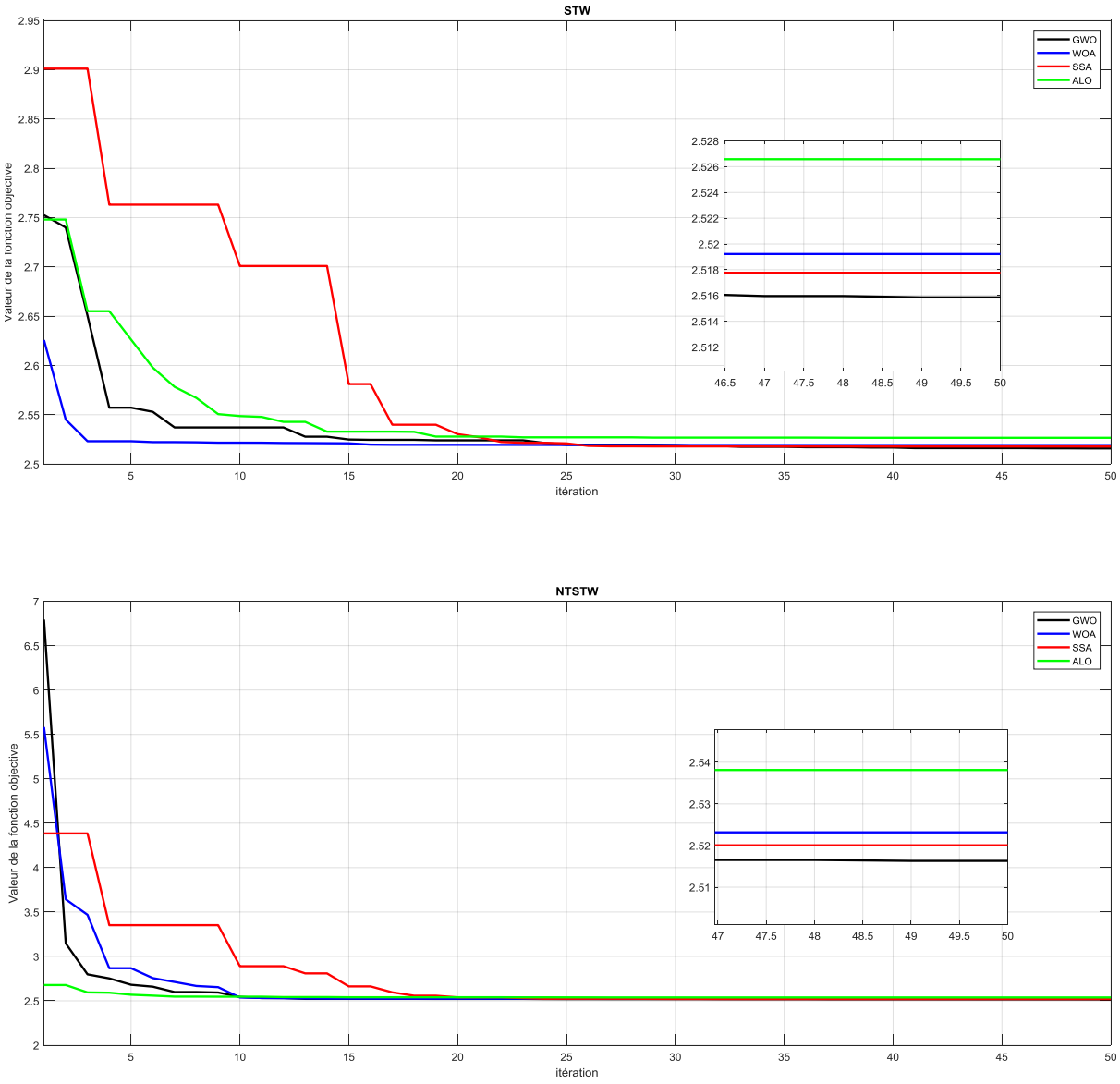


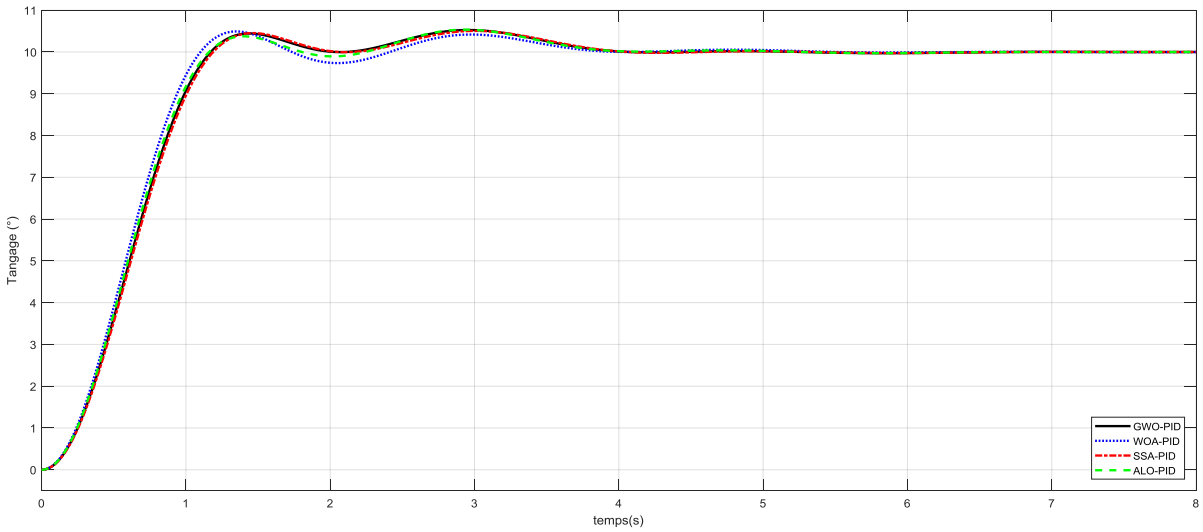
Figure IV. 1 L'évolution de la fonction objective pour les meilleurs essais

IV.1.1. Résultats de l'optimisation des paramètres de la commande PID

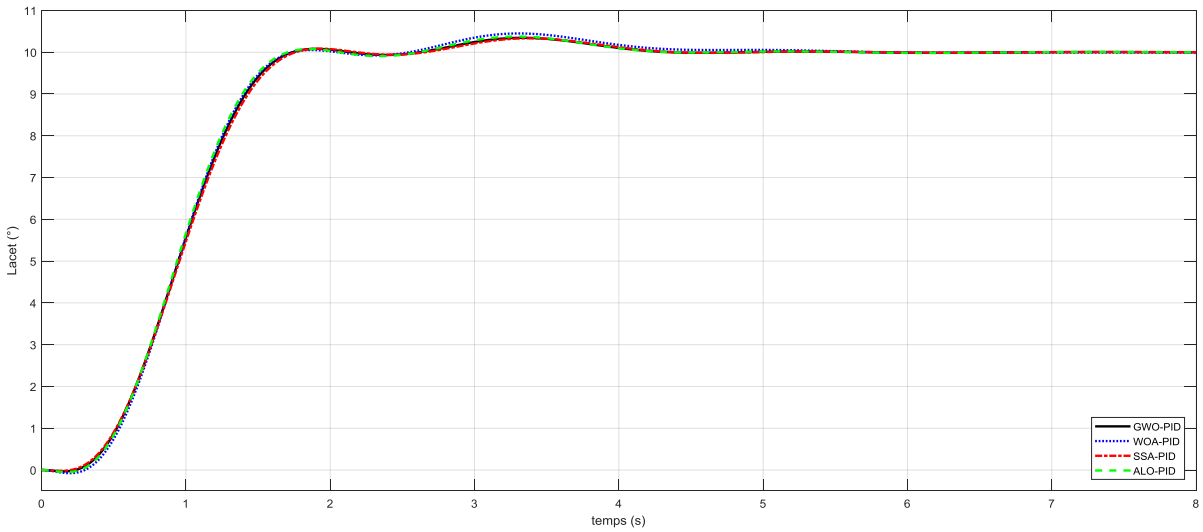
L'intervalle de recherche est choisi entre $[0 \quad 60]$ pour tous les paramètres du contrôleur, les résultats d'optimisation obtenus sont résumés dans les tables ci-dessous.

Algorithme	$K_{p\theta}$	$K_{i\theta}$	$K_{d\theta}$	$K_{p\psi}$	$K_{i\psi}$	$K_{d\psi}$
GWO	56,336	59,702	31,342	60	42,880	32,459
WOA	60	60	31,0639	60	43,5795	34,236
SSA	54,741	58,447	30,466	59,999	42,570	32,595
ALO	58,145	59,831	31,921	59,999	43,315	32,383

Tableau 8 Les paramètres optimaux de la commande PID

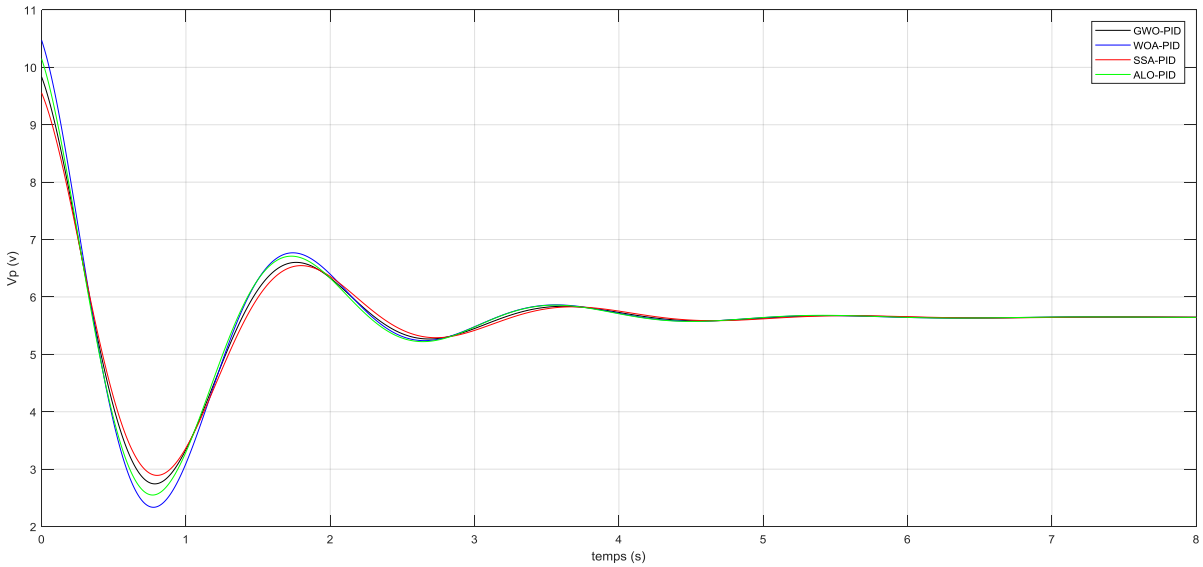


(a)

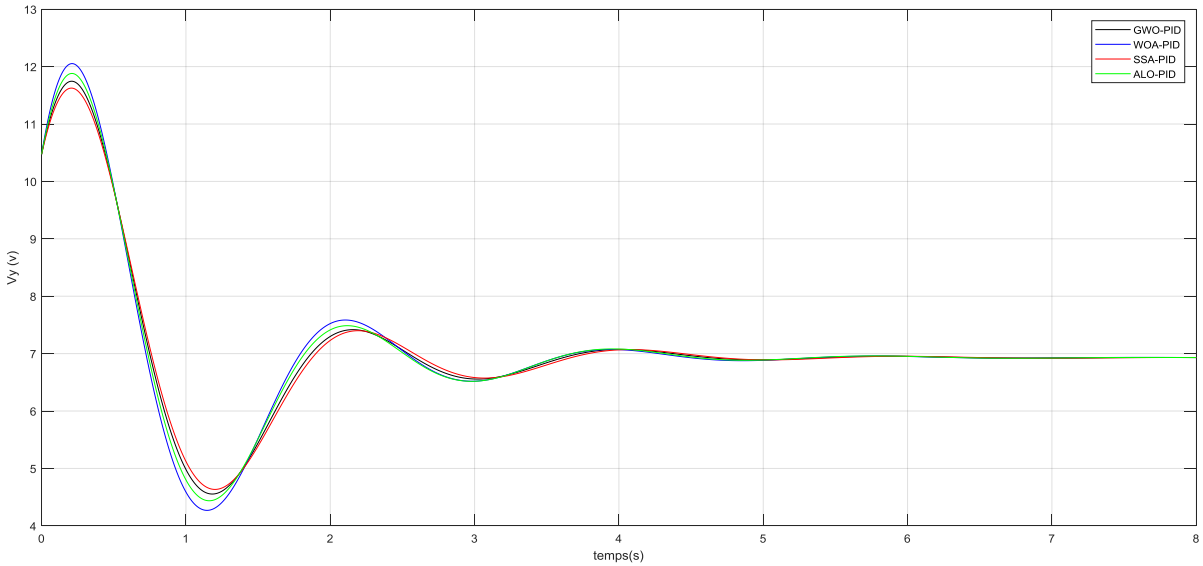


(b)

Figure IV. 2 Réponses des angles de (a) tangage et (b) lacet en mode de régulation à l'aide de la commande PID optimisée par les algorithmes de GOW, WOA, SSA et ALO.



(a)



(b)

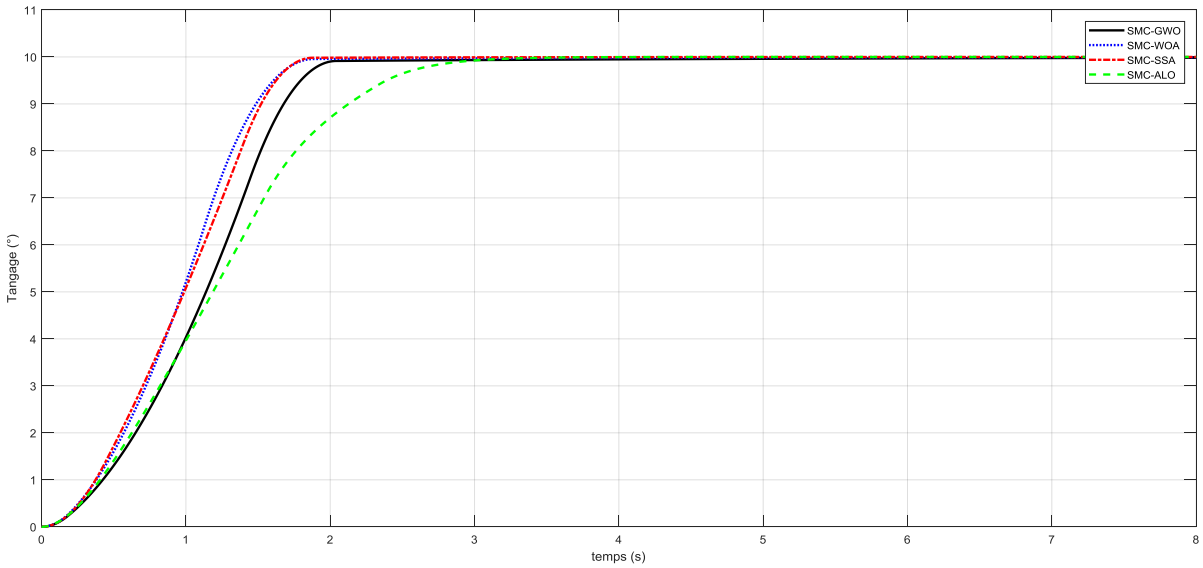
Figure IV. 3 Les signaux de commande générés par PID (a) tangage et (b) lacet

IV.1.2. Résultats de l'optimisation des paramètres de la commande SMC

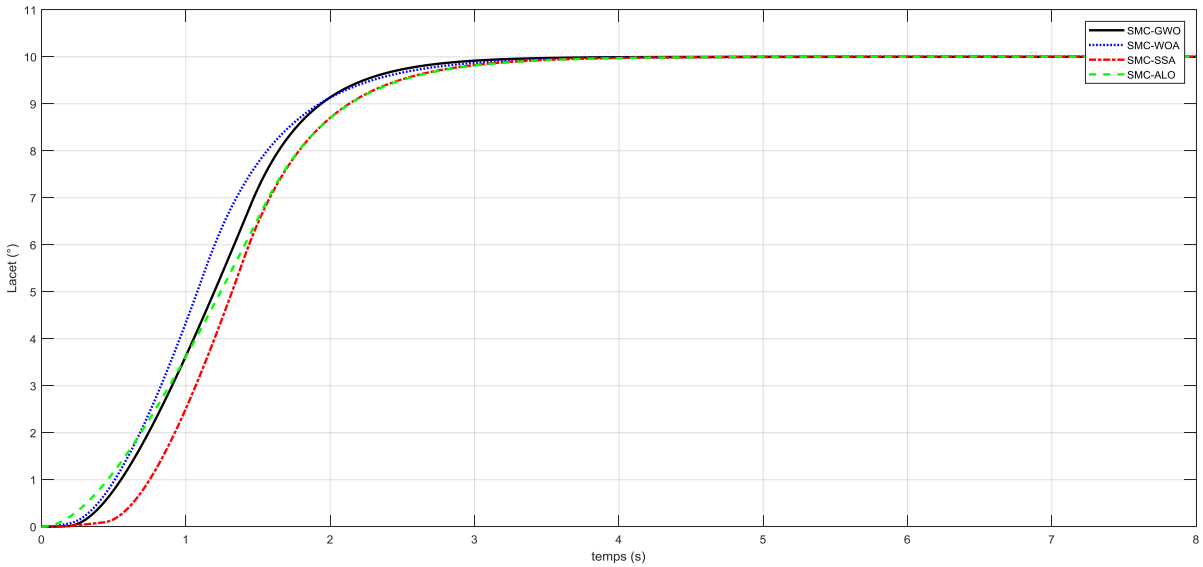
Les paramètres optimisés sont K_1 , K_2 et λ_1, λ_2 contrairement au PID ici, on a 4 paramètres à optimiser. L'intervalle de recherche est choisi comme suit $[0 \quad 30]$ pour tous les paramètres de la commande SMC, les paramètres obtenus sont illustrés dans la table ci-dessous.

Algorithmes	K_1	K_2	λ_1	λ_2
GWO	1,350	1,820	0,257	2,354
WOA	1,160	2,248	0,325	1,907
SSA	1,446	2,0213	0,674	2,001
ALO	0,461	2,935	3,105	1,945

Tableau 9 Les paramètres optimaux de la commande SMC

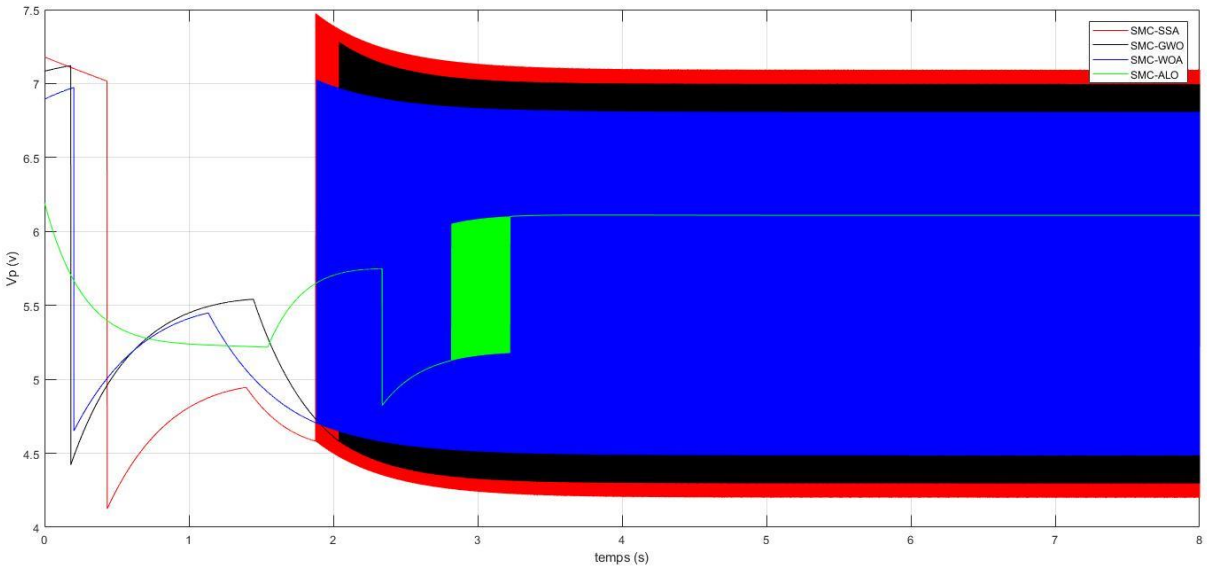


(a)

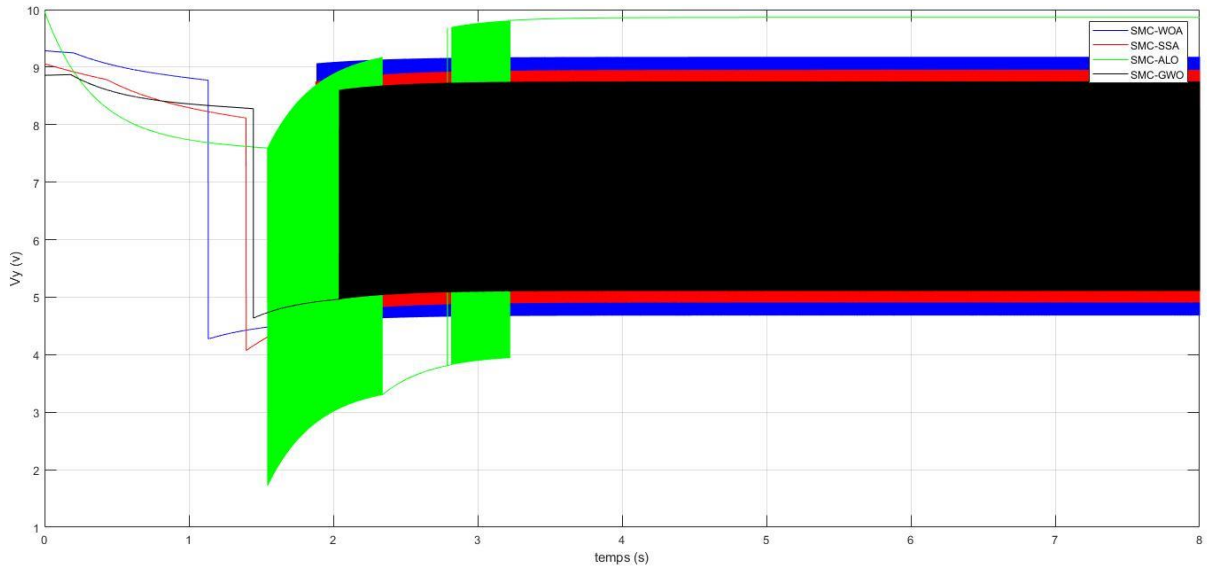


(b)

Figure IV. 4 Réponses des angles de (a) tangage et (b) lacet, en mode de régulation à l'aide de la commande SMC optimisée par les algorithmes de GOW, WOA, SSA et ALO.



(a)



(b)

Figure IV. 5 Les signaux générés par la commande SMC (a) tangage et (b) lacet

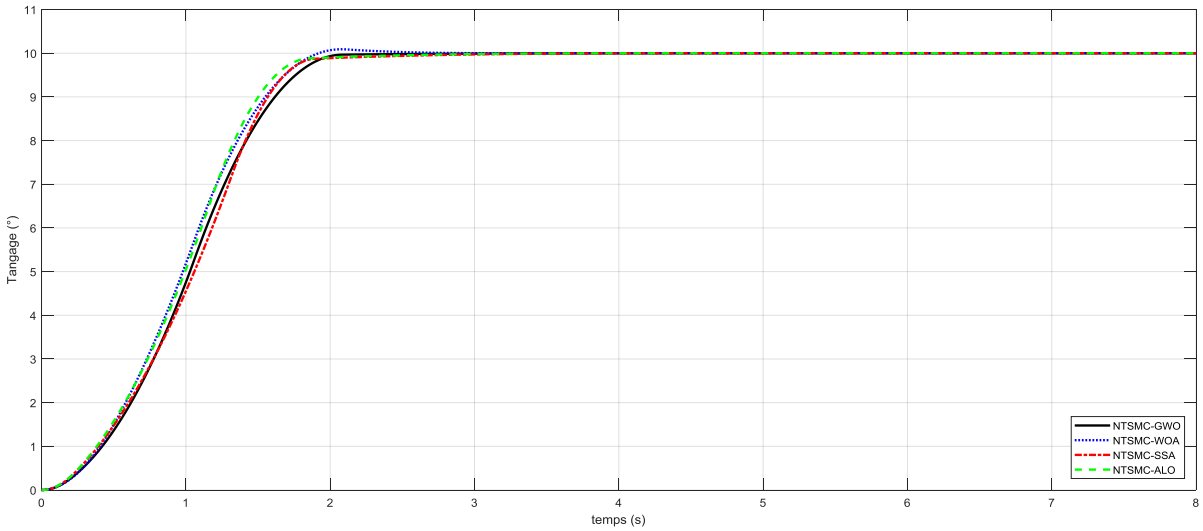
IV.1.3. Résultats de l'optimisation des paramètres de commande NTSMC

Dans le cas de la commande NTSMC quatre paramètres sont optimisés, à savoir K_1 , K_2 et λ_1, λ_2, donc c'est les mêmes paramètres que ceux de l'SMC. L'intervalle de recherche est choisi entre $[0 \quad 30]$ pour tous les paramètres, les paramètres obtenus sont ci-dessous.

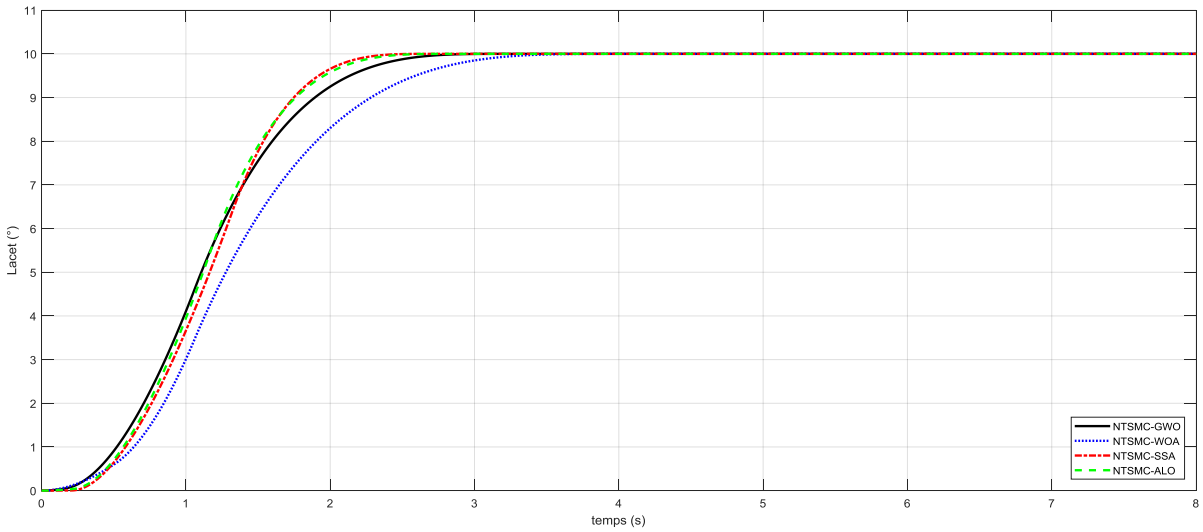
Algorithmes	K_1	K_2	λ_1	λ_2
GWO	0,860	1,950	0,0619	0,752
WOA	0,683	2,0420	0,280	0,534
SSA	1,508	1,956	0,100	1,209
ALO	0,461	2,935	3,105	1,945

Tableau 10 Les paramètres optimaux de commande NTSMC

La figure IV.6 et IV.7 représentent respectivement, les réponses du système et les signaux de commande appliqués sur le Quanser Aero.

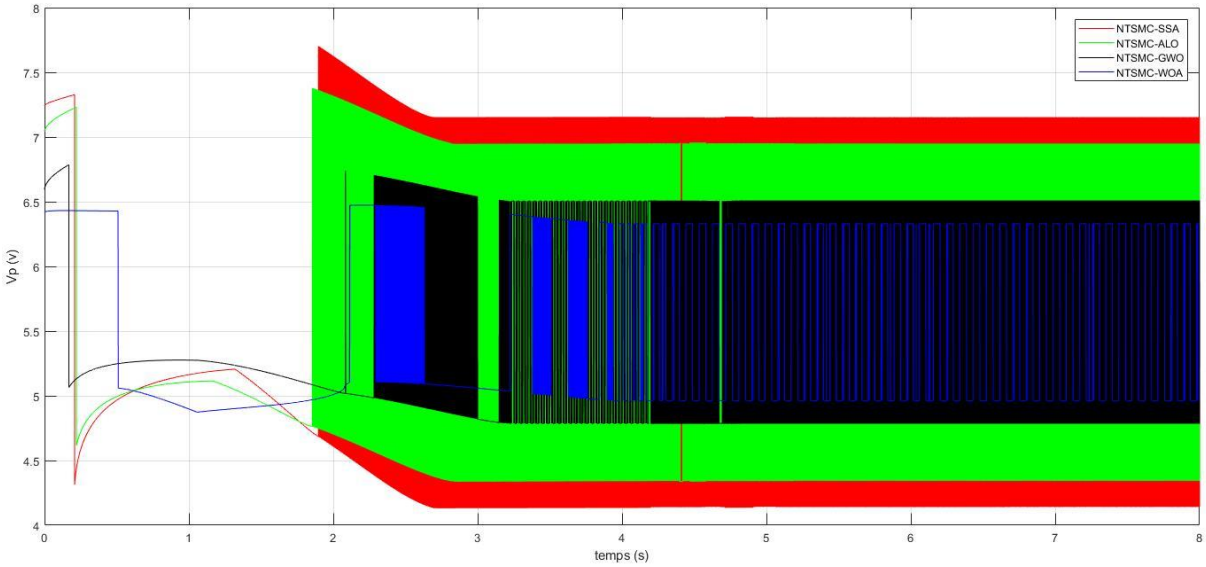


(a)

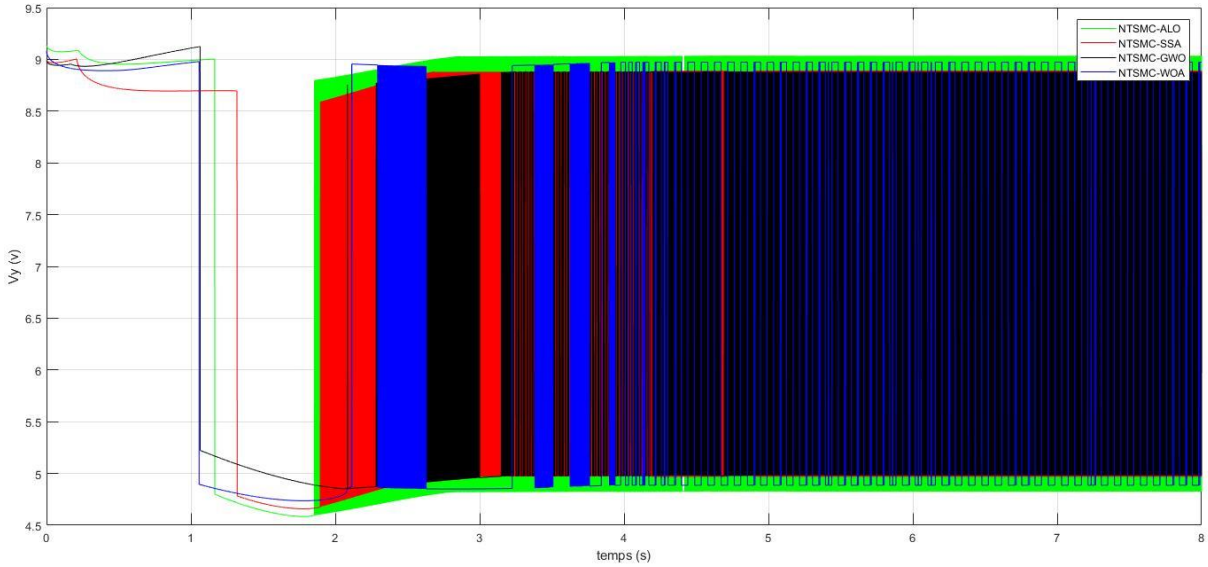


(b)

Figure IV. 6 Réponses des angles de (a) tangage et (b) lacet, en mode de régulation à l'aide de la commande NTSMC optimisée par les algorithmes de GOW, WOA, SSA et ALO.



(a)



(b)

Figure IV. 7 Les signaux de commande générés par NTSMC (a) tangage (b) lacet

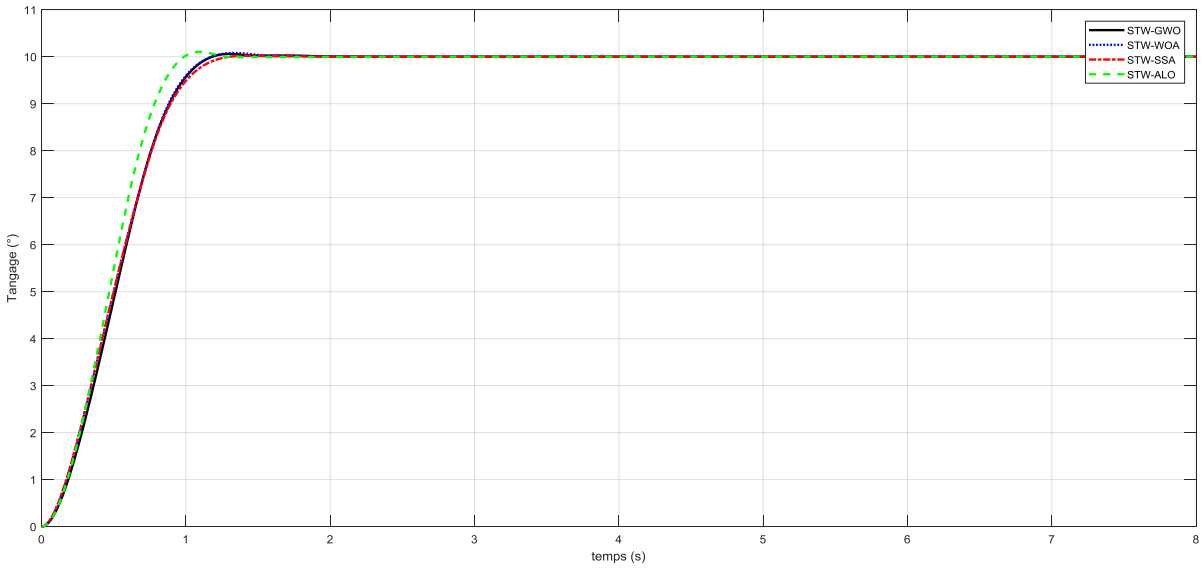
IV.1.4. Résultats de l'optimisation des paramètres de la commande STW

Dans le cas de cette commande, six (6) paramètres ont été optimisés à savoir $\alpha_1, \alpha_2, \beta_1, \beta_2, \lambda_1$ et λ_2 . Il est à rappeler ici que, cette commande est la somme de la commande équivalente et le super-twisting. L'intervalle de recherche est choisi entre $[0 \quad 60]$, pour tous les paramètres de la commande, les paramètres obtenus sont résumés dans le tableau ci-dessous.

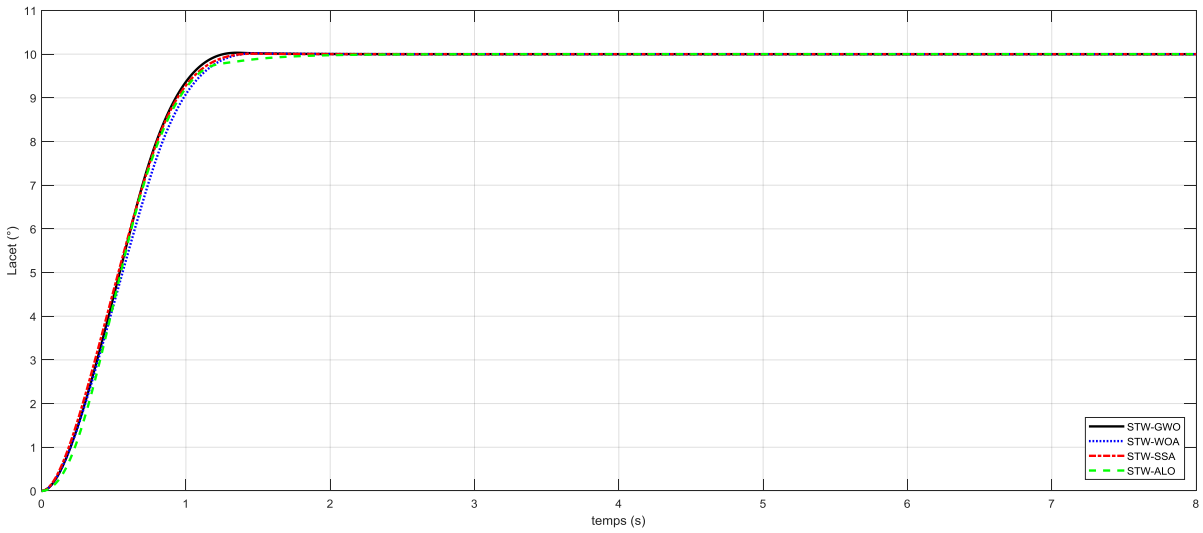
Algorithmes	α_1	α_2	β_1	β_2	λ_1	λ_2
GWO	2,389	33,478	1,6374	18,484	4,640	3,628
WOA	4,892	37,072	1,235	11,364	5,444	3,903
SSA	3,21263	37,334	1,054	11,547	5,515	4,254
ALO	17,366	30,663	0,173	0,422	0,804	2,995

Tableau 11 Les paramètres optimaux de la commande STW

Les figures IV.8 et IV.9 représentent respectivement, les réponses du système et les signaux de commande appliqués sur l'Aero Quanser en mode de régulation.



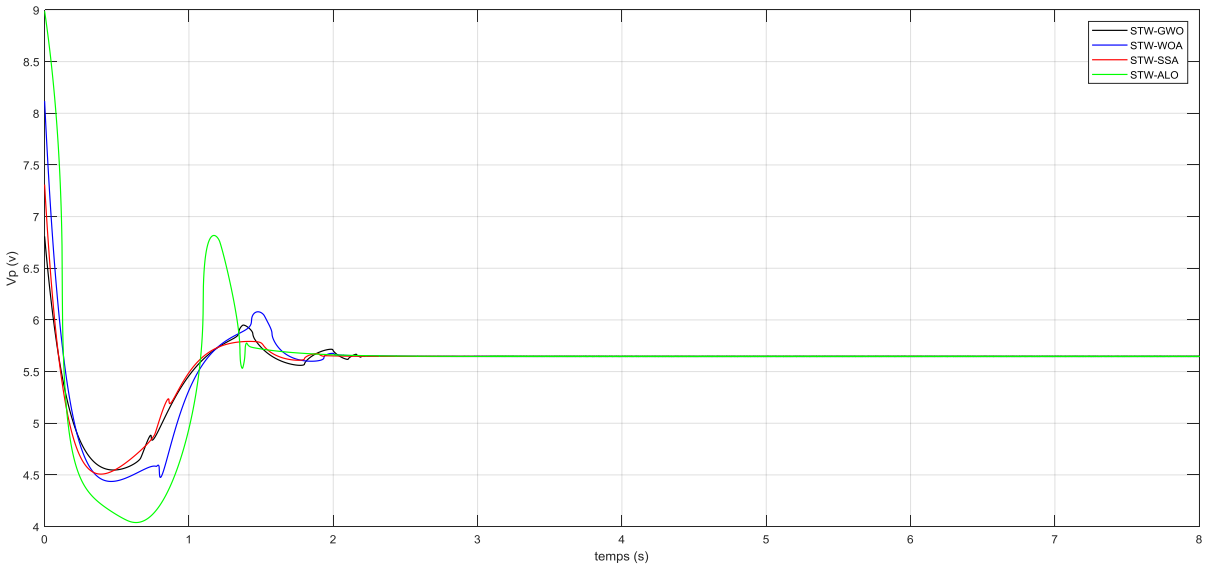
(a)



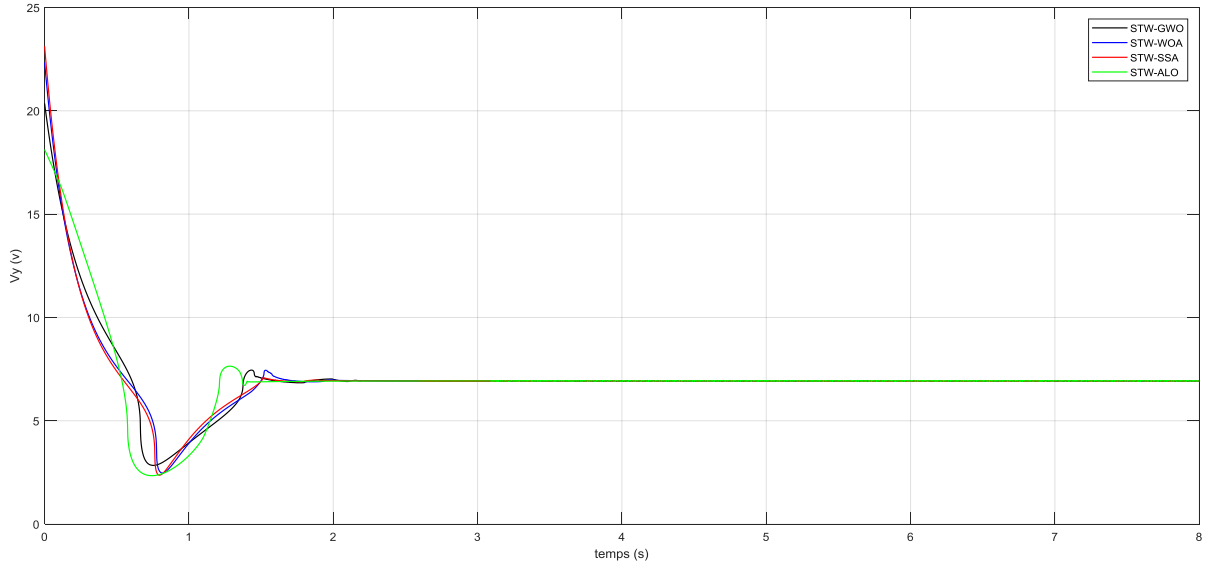
(b)

Figure IV. 8 Réponses des angles de (a) tangage et (b) lacet en mode de régulation à l'aide de la commande STW optimisée par les algorithmes de GOW, WOA, SSA, ALO.

Les signaux de commande de la commande super-twisting sont illustrés dans la figure IV.8



(a)



(b)

Figure IV. 9 Les signaux de commande générés par STW (a) tangage (b) lacet

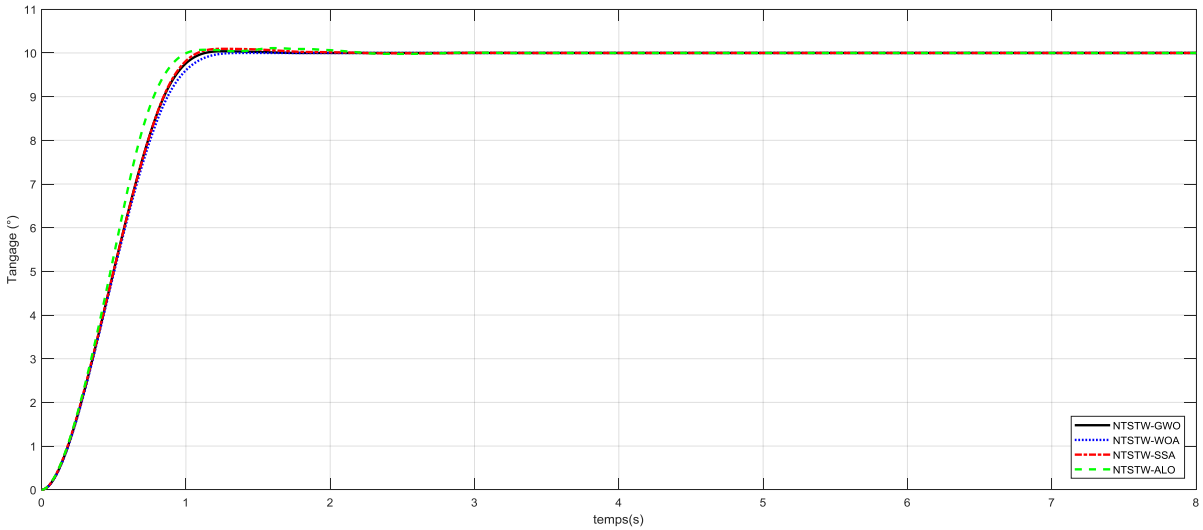
IV.1.5. Résultats de l'optimisation des paramètres de la commande NTSTW

Les paramètres à optimiser dans cette commande sont les même que ceux de la commande STW (i.e. $\alpha_1, \alpha_2, \beta_1, \beta_2, \lambda_1$ et λ_2). Il est à noter que p et q sont choisis arbitrairement et ils sont égaux a 7 et 5 respectivement. L'intervalle de recherche est choisi entre $[0 \quad 50]$ pour tous les paramètres de commande, les résultats obtenus sont ci-dessous.

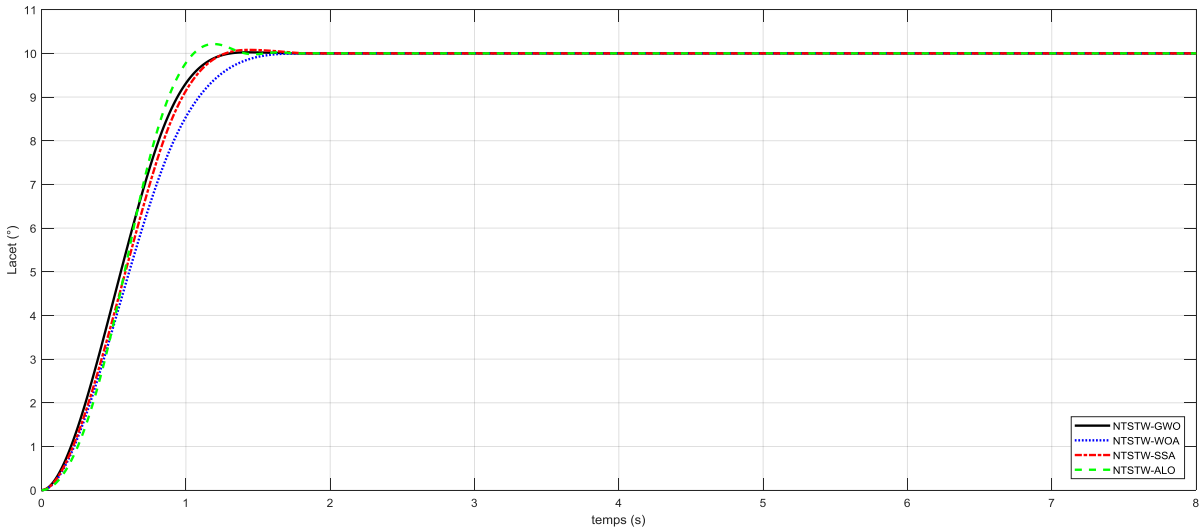
Algorithmes	α_1	α_2	β_1	β_2	λ_1	λ_2
GWO	4,109	45,472	0,0440	3,422	2,934	2,264
WOA	7,999	50	0	0,644	2,639	1,576
SSA	7,802	47,943	1,325	7,328	3,774	2,191
ALO	12,685	48,845	5,053	40,896	5,668	2,134

Tableau 12 Les paramètres optimaux de la commande NTSTW

Les figures IV.10 et IV.11 représentent respectivement, les réponses du système et les signaux de commande appliqués sur l'Aero Quanser en mode de régulation.

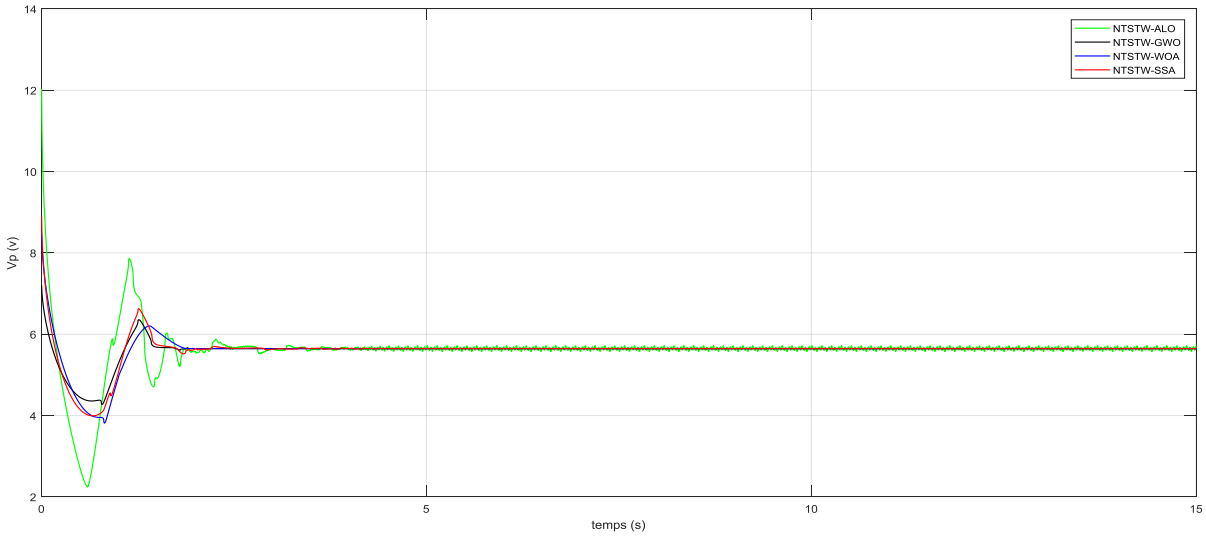


(a)

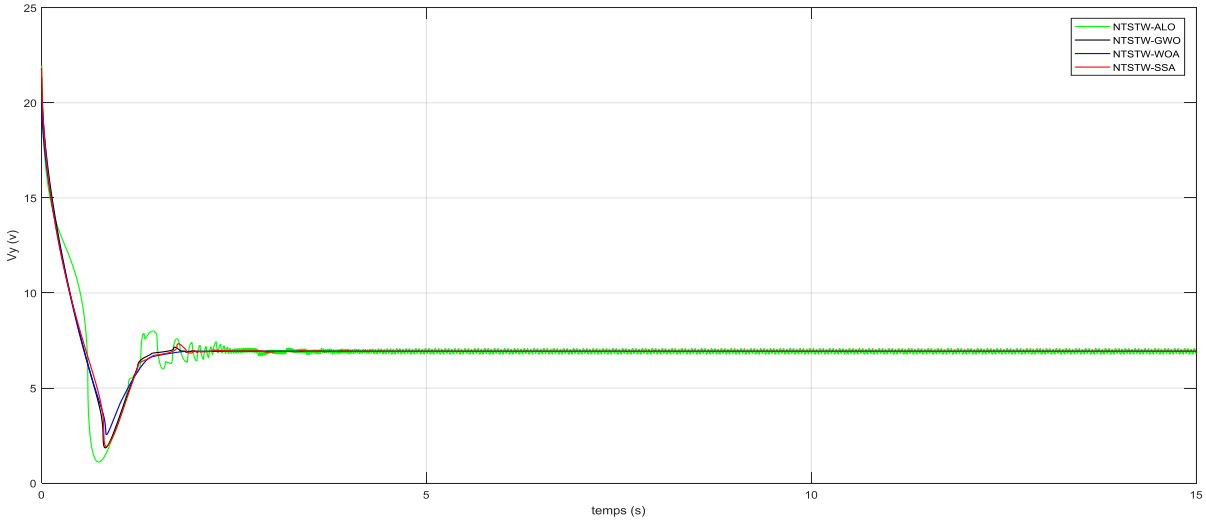


(b)

Figure IV. 10 Réponses des angles de (a) tangage et (b) lacet, en mode de régulation à l'aide de la commande NTSTW optimisée par les algorithmes de GOW, WOA, SSA et ALO.



(a)



(b)

Figure IV. 11 Les signaux de commande générés par l'NTSTW (a) tangage (b) lacet

V.1.5 Comparaison entre les algorithmes d'optimisations

Un classement des algorithmes d'optimisation en fonction des valeurs minimales de la fonction objective *J* est présenté dans le tableau ci-dessous pour les différentes commandes.

Commandes	GWO	WOA	ALO	SSA
PID	1	4	3	2
SMC	2	1	4	3
NTSMC	1	3	2	2
HOSMC	1	3	4	2
NTSMC_HOSMC	1	3	4	2

Tableau 13 Classement des algorithmes d'optimisation selon la valeur minimale de la fonction objective

Le tableau ci-dessous donne un classement des moyennes de la fonction objective pour les différents algorithmes et pour les différentes commandes appliquées.

Commandes	GWO	WOA	ALO	SSA
PID	1	4	3	2
SMC	1	2	3	4
NTSMC	1	2	3	4
HOSMC	1	2	3	4
NTSMC_HOSMC	1	2	4	3

Tableau 14 Classement des algorithmes d'optimisation selon le min de moyenne

- *Discussion et interprétation des résultats*

D'après les tableaux (13,14), les mauvais résultats, de point de vue valeur minimale de la fonction objective, sont obtenus par les algorithmes WOA et ALO, tandis que le GWO a performé des valeurs minimales de la fonction objective dans la plupart des cas. De plus, l'algorithme GWO a également été le meilleur en termes de la meilleure moyenne pour les 10 essais. L'algorithme WOA a permet d'obtenir une valeur raisonnable de la moyenne des essais de la fonction objective. D'autre part, la valeur moyenne la plus élevée est donnée par les algorithmes SSA et ALO. Sur cette base, nous pouvons confirmer que l'algorithme GWO a montré des performances relativement supérieures par rapport aux algorithmes WOA, SSA et ALO.

Une particularité est remarquée dans ces résultats concernant l'algorithme SSA, cet algorithme commence à converger qu'après un certain nombre d'itérations, le cas le plus visible est celui de l'optimisation de la commande SMC, dont on voit un commencement de convergence après 18 itérations, ceci représente 36 % du nombre totale d'itérations, ce qui n'est pas négligeable.

La figure (IV.1) montre les meilleures valeurs de la fonction objective par itération pour les quatre algorithmes d'optimisation. Dans les cinq commandes, on remarque que SSA est le plus lent par rapport aux autres algorithmes d'optimisation (par exemple, dans le cas deu PID et NTSMC, il arrive à son minimum, après 26 itérations ceci représente plus de 50% du nombre d'itérations). On remarque que WOA et ALO convergent rapidement. Par contre, l'algorithme GWO a présenté une bonne vitesse de convergence au minimum de la fonction objective (par exemple, dans le cas de SMC et NTSMC, il arrive à son minimum juste après 16 et 10 itérations respectivement, ceci représente 32% et 20 % de la totalité d'itérations).

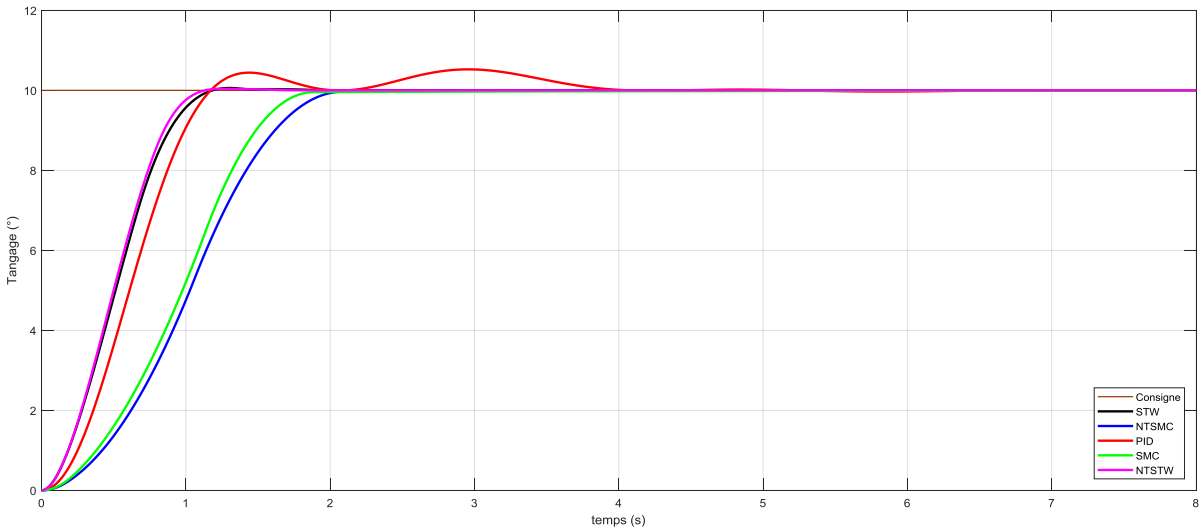
Une remarque importante est à ajouter ici, c'est que l'utilisation de 50 itérations pour tous les algorithmes va permettre d'avoir un degré de confiance élevé sur la qualité des résultats. Car on remarque que, les valeurs de J pour tous les résultats arrivent à leurs minimums bien avant les 50 itérations et pour les 10 essais. Le cas qui présente le plus de minimisation jusqu'à près de la $50^{émé}$ itération est celui de GWO dans le cas du PID. En revanche, on a déjà conclu par les autres résultats que le GWO est le meilleur algorithme donnant la minimisation de la fonction objective.

IV.2. Comparaison entre les différents techniques de commandes

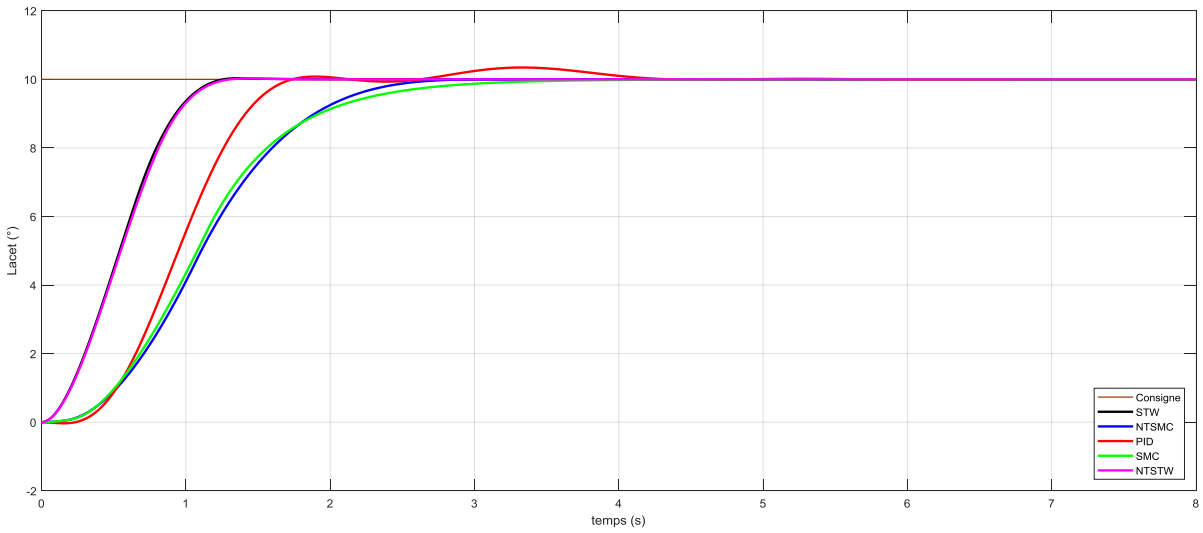
La comparaison entre les performances des lois de commande se base sur plusieurs indices comme le temps de réponse et de stabilisation, le dépassement, l'erreur statique (précision). Le indice ISCO $= \int u^2\, dt$ interprète la consommation d'énergie et ISE $= \int e^2 dt$ interprète la précision en régime transitoire.

➢ Mode régulation

Afin de comparer les lois de commande optimisées, nous traçons les meilleures réponses des angles (selon la valeur de la fonction objective voir la figure IV.12), les surfaces de glissement (figure IV.13) et les tensions des commandes pour les sorties en tangage et lacet (figure IV.14), les paramètres optimaux (les optimaux des optimaux) sont trouvés dans la section précédente.

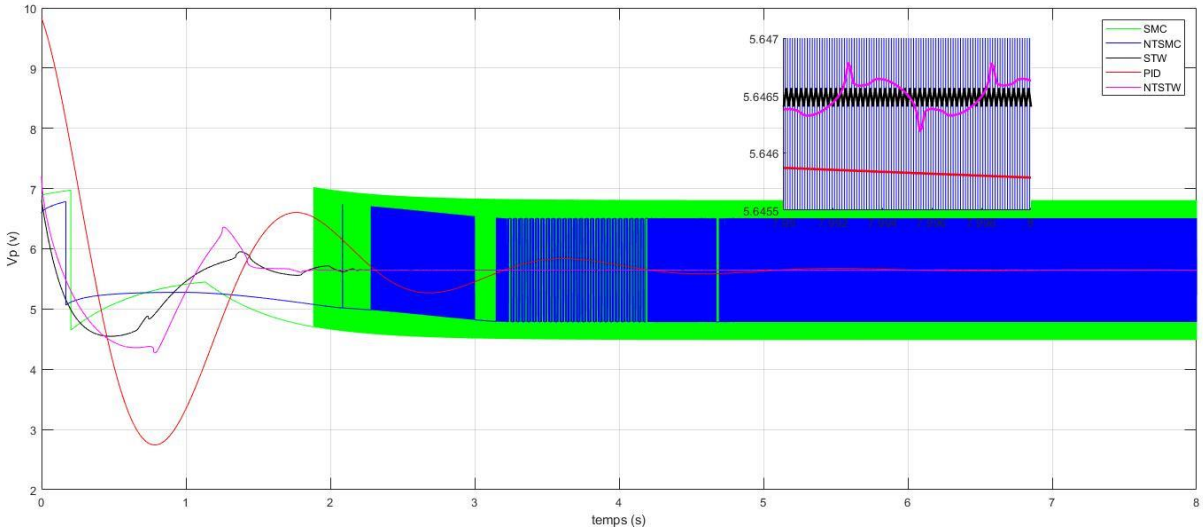


(a)

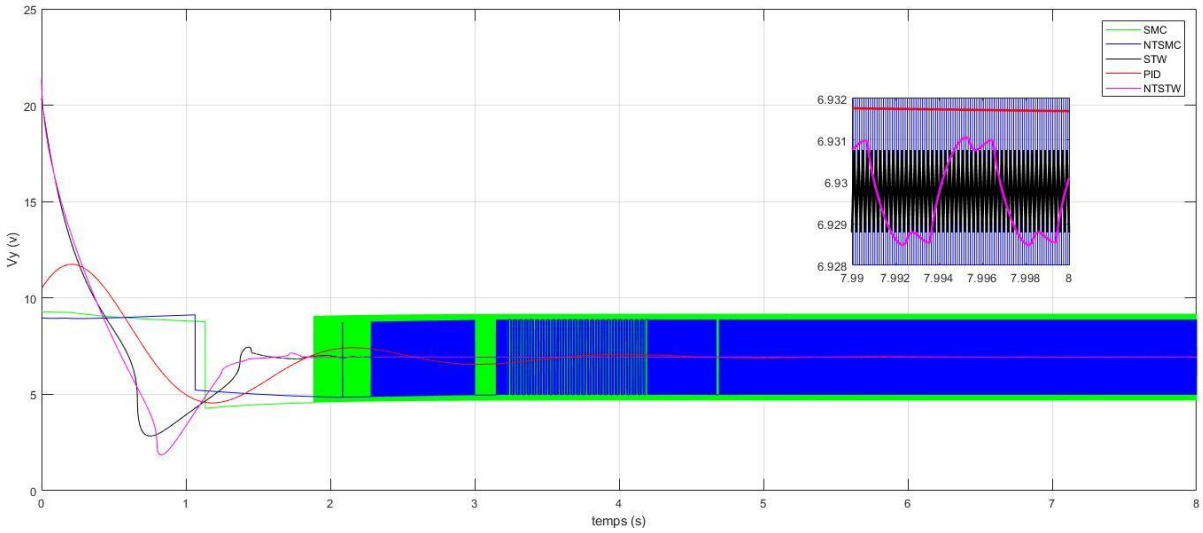


(b)

Figure IV. 12 Les réponses des angles de (a) tangage et (b) lacet, pour les commandes PID, SMC, NTSMC, STW et NTSTW optimisées

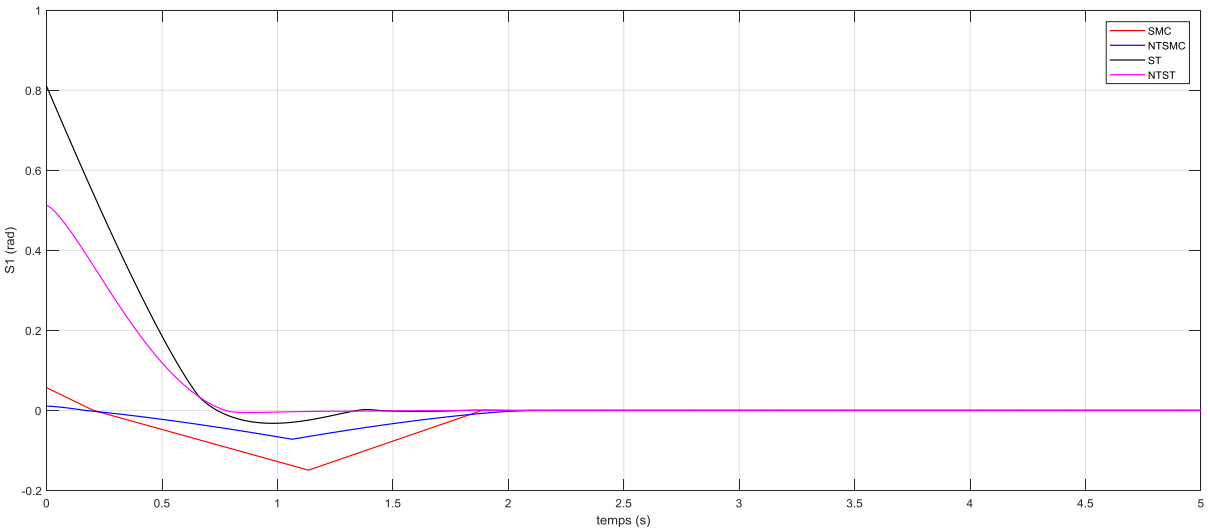


(a)

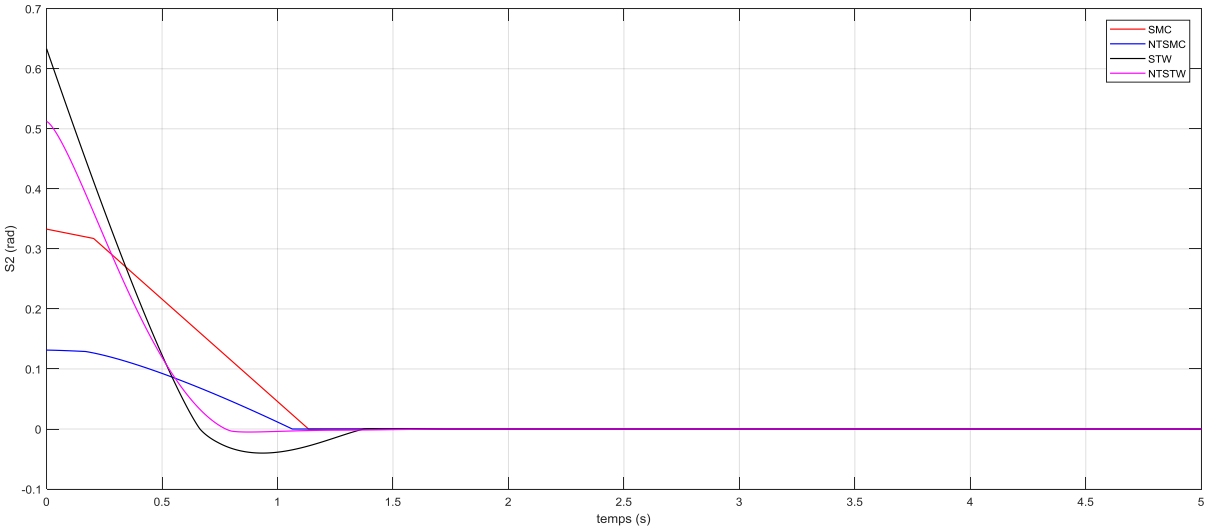


(b)

Figure IV. 13 Les signaux de commande des angles de (a) tangage et (b) lacet pour les commandes PID, SMC, NTSMC, STW, NTSTW optimisées.



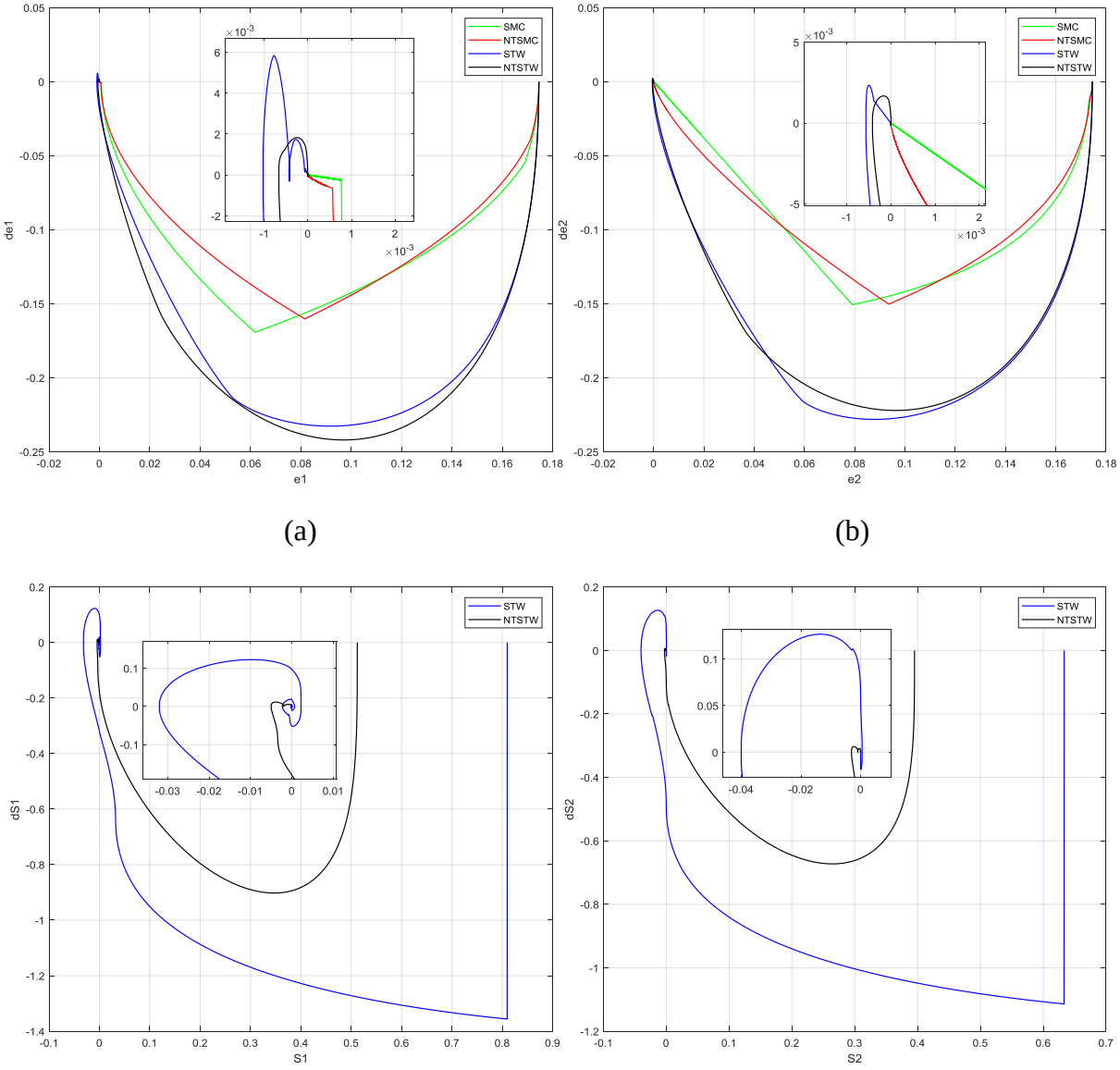
(a)



(b)

Figure IV. 14 Les surfaces de glissements des angles de (a) tangage et de (b) lacet, pour les commandes SMC, NTSMC, STW et NTSTW optimisées

La figure IV.15 ci-dessous présente le plan de phase des commandes SMC, NTSMC, STW et NTSTW. En plus, le plan de phase des commandes STW et NTSTW en surface de glissement et sa drivée sont aussi données pour les deux commandes STW et NTSTW, ces graphes sont plus parlants parce que la commande STW est calculée à la base de la surface de glissement et sa dérivée.



(a) (b)

(c) (d)

Figure IV. 15 Plan de phase (a) tangage et (b) lacet, surface et dérivée de surface (c) tangage et (d) lacet.

Le tableau 15 résume les performances des meilleures commandes.

	Commande	RT (s)	ST (s)	OS (%)	Erreur statique (°)	ISCO
Tangage	PID	0.7407	3.5829	5.2507	3.526 E-08	637.881
	SMC	1.1066	1.7226	**0**	2.630 E-05	656.778
	NTSMC	1.1970	1.8914	**0**	3.766 E06	645.318
	STW	0.7010	1.0720	0.5920	**4.067 E-12**	**631.540**
	NTSTW	**0.6646**	**1.0132**	0.3835	7.552 E-12	631.878
Lacet	PID	0.8927	3.7892	3.4461	4.748 E-08	**987.272**
	SMC	1.4185	2.7693	0.0011	1.005 E-04	1076.112
	NTSMC	1.3726	2.3906	**8 e-06**	9.311 E-12	1051.912
	STW	**0.7299**	**1.1279**	0.3241	**5.214 E-12**	1002.800
	NTSTW	0.7374	1.1457	0.2388	1.469 E-09	1003.799

Tableau 15 Comparaison entre les différentes lois de commande

- *Discussion et interprétation des résultats*

Le tableau ci-dessus résume les meilleurs preformances obtenues en utilisant les algorithmes d'optimisation. Ce tableau va permettre de comparer les commandes avec plus de précision. Nous remarquons qu'on terme de rapidité les deux commandes STW et NTSTW ont abouti aux valeurs minimales de RT et ST avec un léger avantage pour le NTSTW en tangage, dont les valeurs sont 0.6646 *sec* et 1.0132 *sec* réspectivement. Par contre dans le cas du lacet l'avantage est pour le STW avec les valeurs de RT=0.7299 *sec* et ST= 1.127 *sec,* ce qu'est globalment, une amélioration signéficative de ces preformances dans le cas de ces commandes par rapport aux PID, SMC et NTSMC. Contrairment à cela, on remarque que le SMC et le NTSMC présentent un dépassment nul ce qui est logique, sachant que ces commandes ne contiennent pas d'actions intégrales.

Globalement les dépassements trouvés sont très légers et ils n'affectent pas le comportement du système. Dans le cas de STW et NTSTW, l'erreur en régime permanent a été améliorée d'une valeur signéficative notamment en tangage en envirant 100%. La consomation d'énergie est presque pareille dans tous les cas, la différence est quasiment négligable. Par contre, les formes des signaux contiennent clairement du brouttement dans le cas de SMC et NTSMC, la commande STW contient de légéres fluctuations, mais qui peuvent affecter les actionneurs du robot à long terme. Le PID ne présente aucune oscillation et il est précis à cause de la correction de l'action intégrale. La commande NTSTW est la meilleure dans ce contexte, car il n'y a pas de brouttement visible dans cette commande ceci est interpreté par l'influance de la commande équivalante basée sur le NTSMC sur la partie super-twisting.

Nous pouvons voir à travers le plan de phase (Figure IV.15) que, toutes les quatre commandes SMS, NTSMC, STW et NTSTW ont convergé au point d'équilibre (0,0) avec de différentes trajectoires, l'avantage des commandes STW et NTSTW est leurs capacités de garantir la convergence de S et $\dot{S}$ à « zero », cette dernière a été confirmée a travers les courbes (c) et (d) avec un avantage de rapidité de convergence à la commande NTSTW.

IV.3. Résultat de simulation de la commande adaptative-optimale (ANTSWO) :

Afin d'examiner la robustesse et les performances de la commande ANTSWO, des tests de rejet de perturbation et de suivi de trajectoire sont effectués. Les paramètres de la commande adaptative sont choisis comme suit :

Pour le sous-système d'élévation (Tangage) :

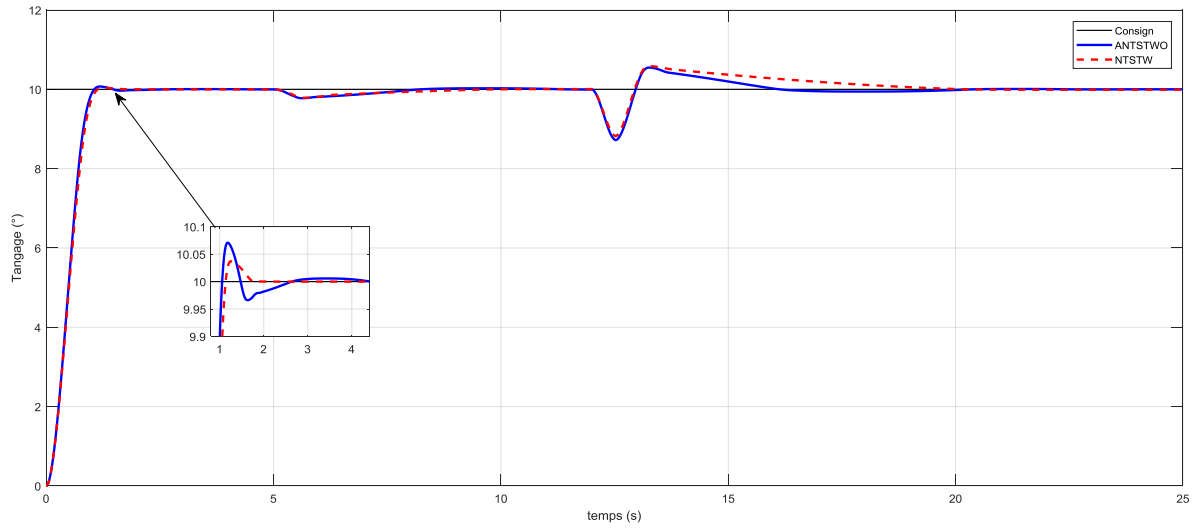
$\omega_1 = 0.001\,, \eta = 0.0001, \varepsilon = 0.02\,, \gamma_1 = 2, |S(0)| > \mu = 0.02$ et $\alpha(0) > \alpha_m = 0.0001$

Pour le sous-système d'azimut (Lacet) :

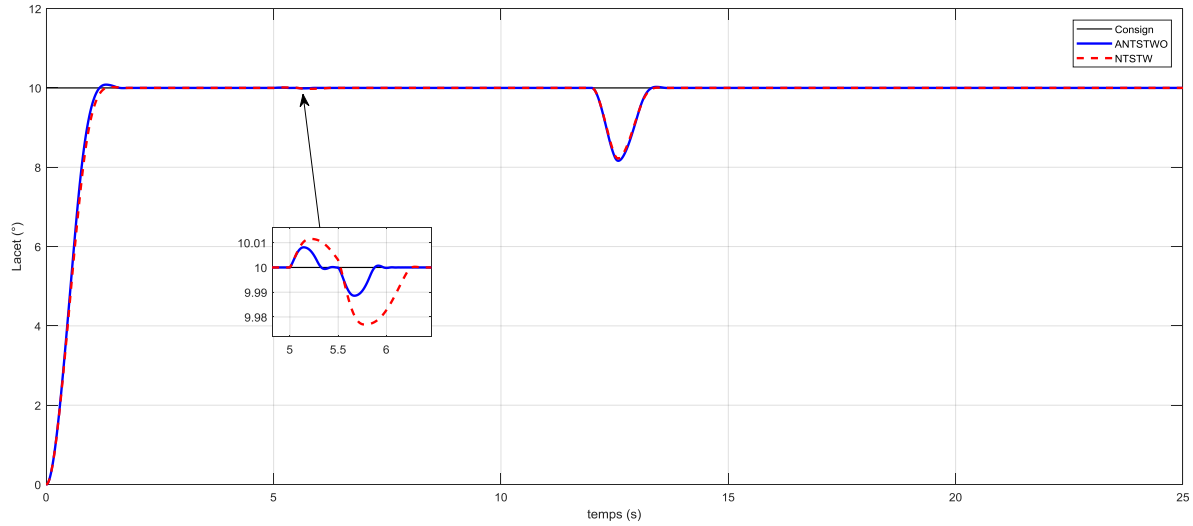
$\omega_1 = 0.001\,, \eta = 0.0001, \varepsilon = 0.05\,, \gamma_1 = 2, |S(0)| > \mu = 0.06$ et $\alpha(0) > \alpha_m = 0.0001$

- *Rejet de perturbation*

Dans cette expérience, nous avons simulé le système pour une réponse en régulation, le système a été perturbé deux fois avec une perturbation sous forme d'impulsion sur une durée égale à 0.5 *Sec*. La première perturbation a été appliquée sur l'axe de tangage à t=5 *Sec* et la seconde perturbation est appliquée sur l'axe de lacet à t=12 *Sec*. Ceci va permettre d'examiner les deux commandes vis-à-vis de l'effet de couplage. Les résultats graphiques de la réponse du système, les signaux des différentes commandes, l'évolution des gains adaptatifs ainsi qu'un tableau résumant les performances obtenues sont donnés respectivement dans les figures IV.16, IV.17, IV.18 et 16.

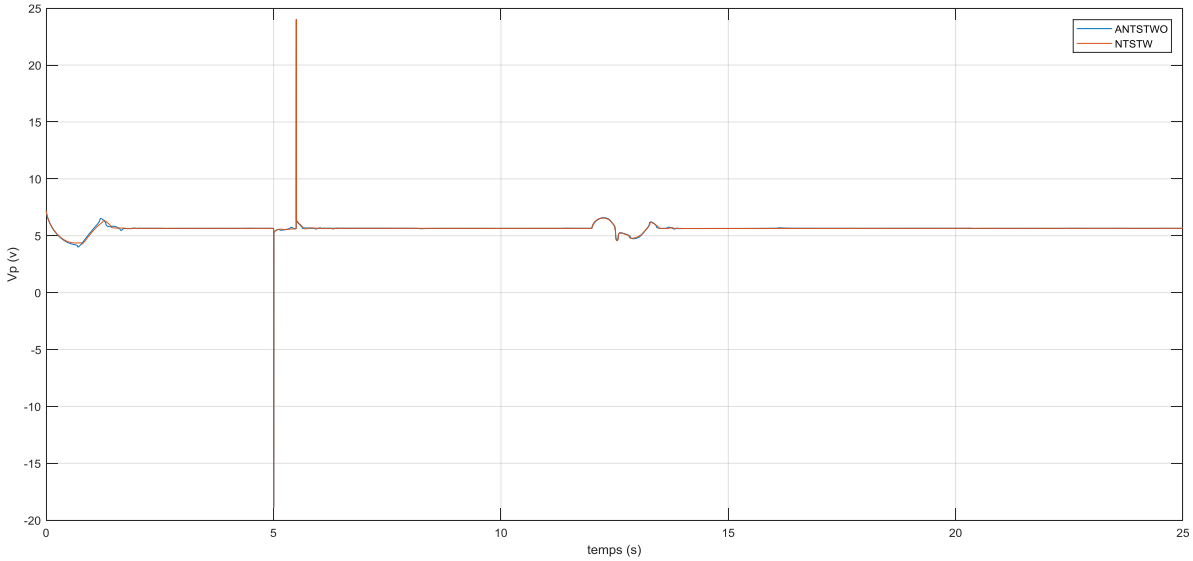


(a)

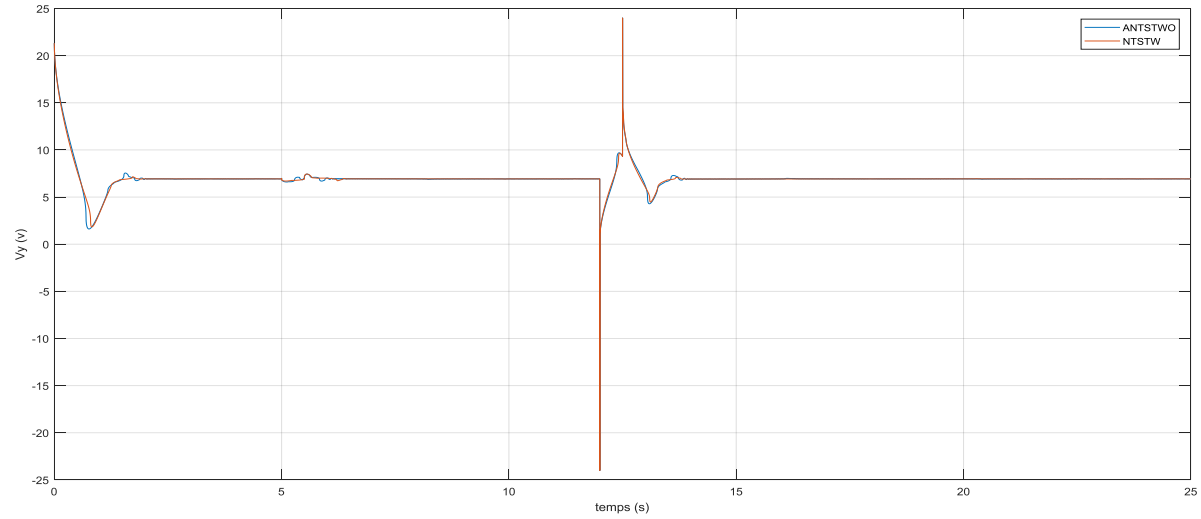


(b)

Figure IV. 16 Réponse indicielle du système (a) tangage (b) lacet dans le cas des commandes ANTSTWO et NTSTW



(a)



(b)

Figure IV. 17 Signaux de commande de ANTSTWO et NTSTW (a) tangage (b) lacet

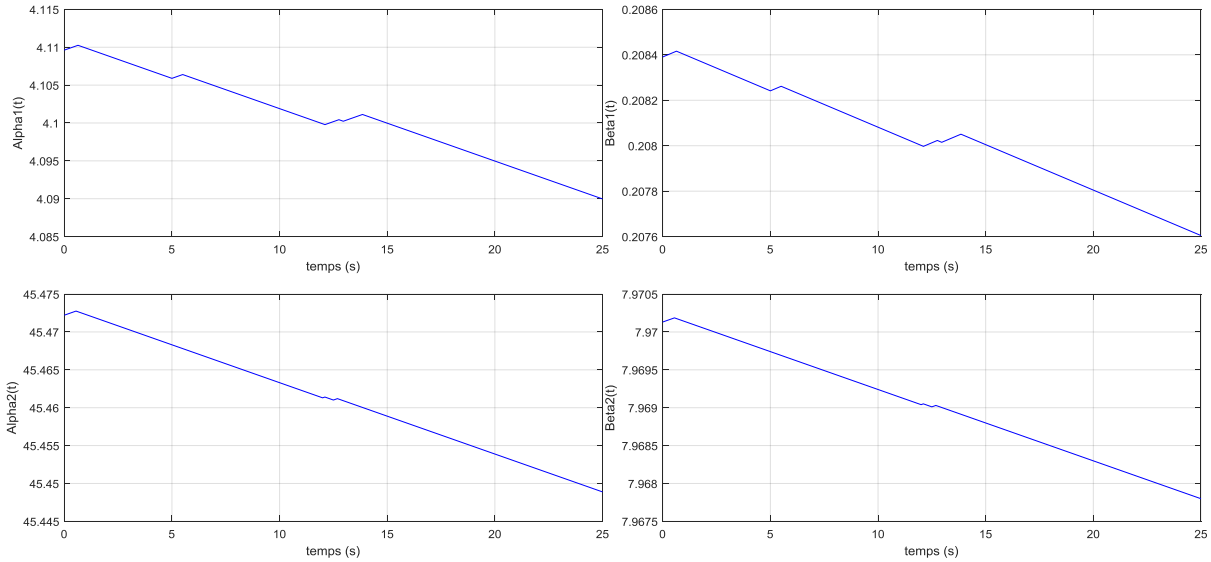


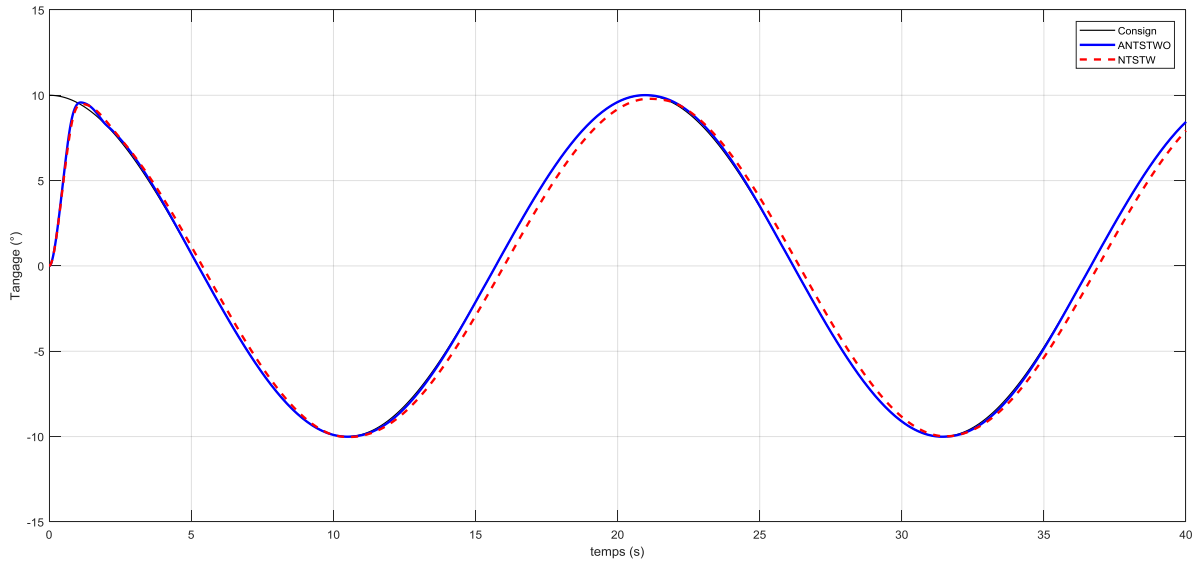
Figure IV. 18 Evolution des gains adaptatifs de la commande ANTSTWO (Rejet de perturbation en mode régulation)

Angle	Commande	ISE	ISCO
Tangage	NTSTW	0.0121	**792.08**
	ANTSTW	**0.0118**	792.42
Lacet	NTSTW	0.0130	**1252.05**
	ANTSTW	**0.0127**	1256.41

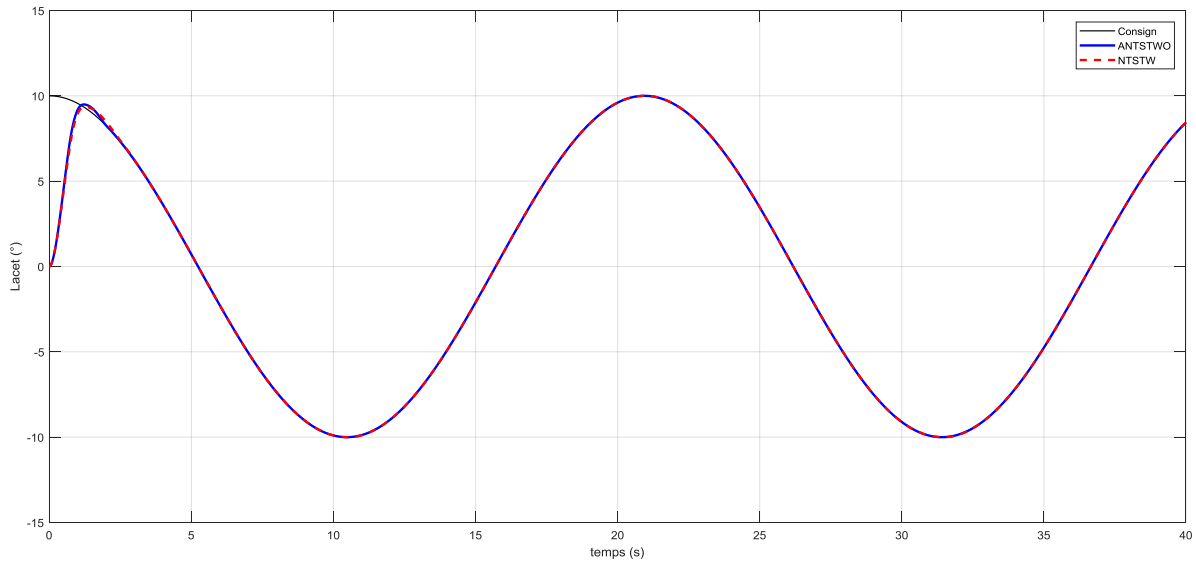
Tableau 16 Tableau comparatif des performances des commandes NTSTW et ANTSTWO

➢ *Poursuit de trajectoire sinusoïdale*

Dans cette expérience nous avons testé les deux commandes NTSTW et ANTSTWO en mode poursuite de trajectoire sinusoïdale. Les résultats graphiques de la réponse du système, les signaux des différentes commandes, l'évolution des gains adaptatifs ainsi qu'un tableau résumant les performances obtenues sont donnés respectivement dans les figures IV.19, IV.20, IV.21 et 17.

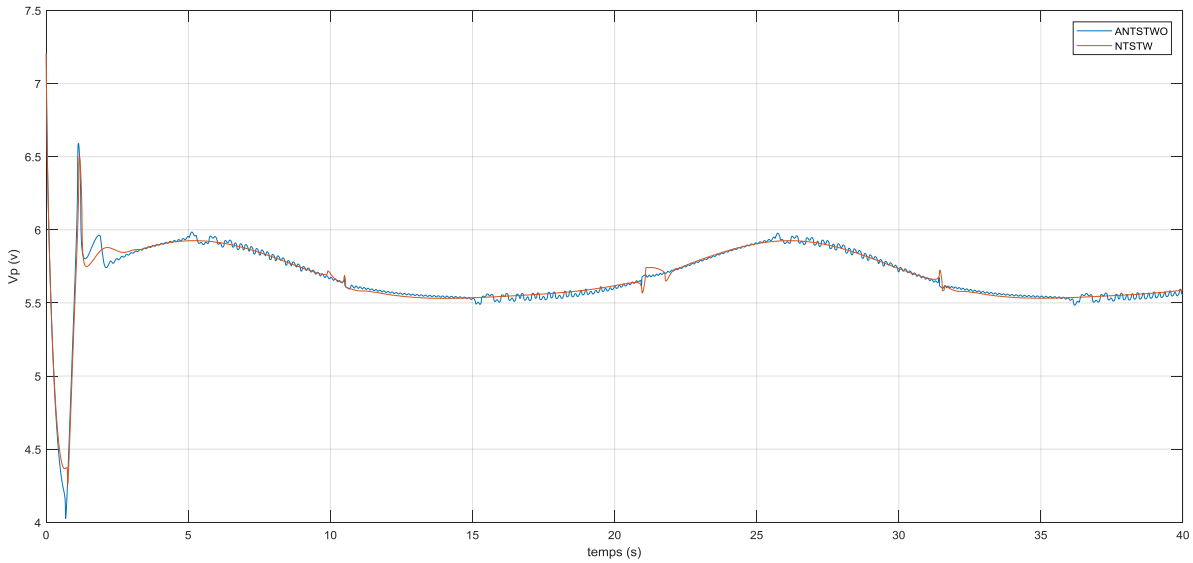


(a)

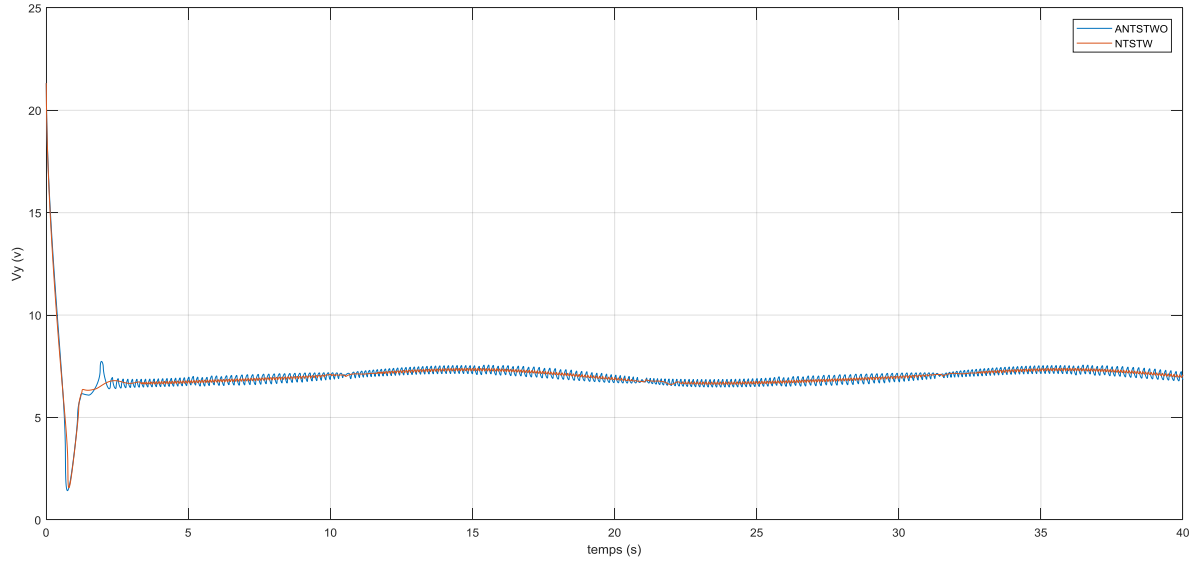


(b)

Figure IV. 19 Réponse en poursuite du système en l'application des commandes ANTSTWO et NTSTW (a) tangage (b) lacet



(a)



(b)

Figure IV. 20 Signaux de commande en poursuite (a) tangage (b) lacet

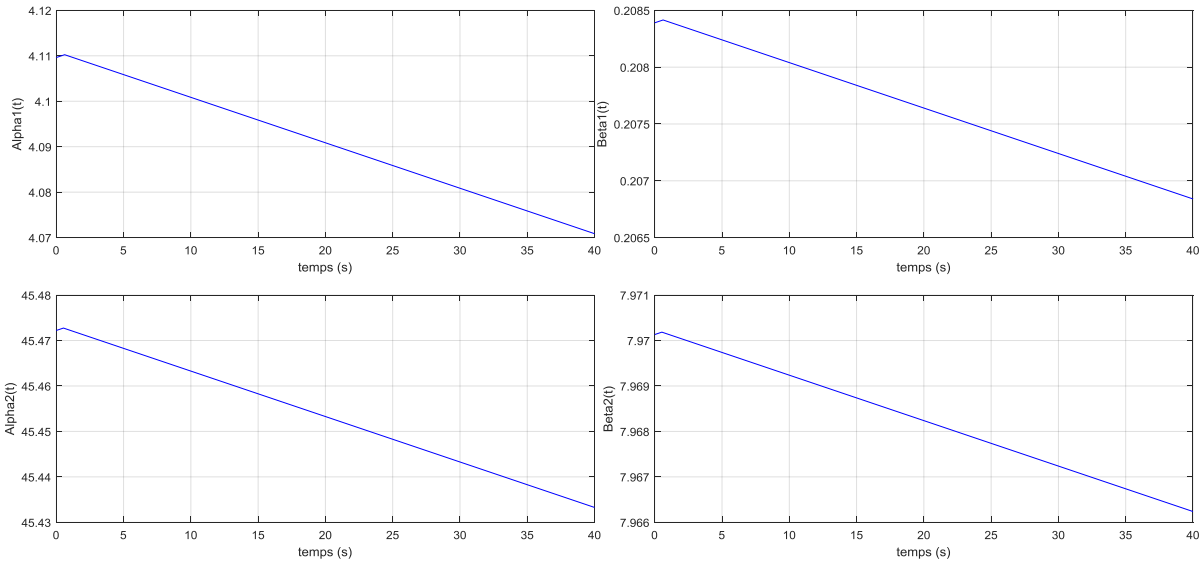


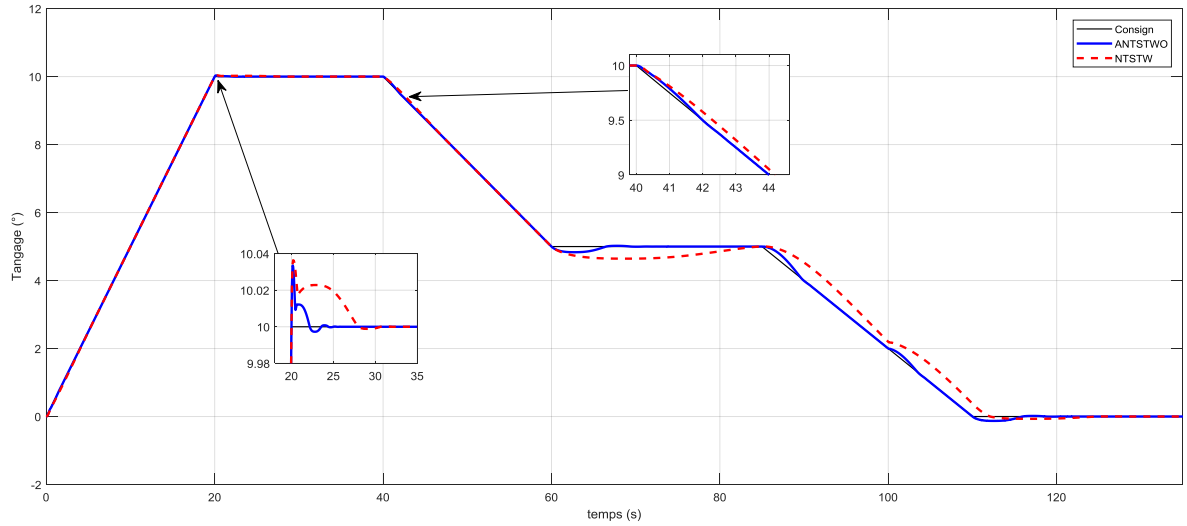
Figure IV. 21 Evolution des gains adaptatifs de la commande ANTSTWO (signal sinusoïdal)

Angle	Commande	ISE	ISCO
Tangage	NTSTW	0.0141	1292.05
	ANTSTW	**0.0109**	**1291.78**
Lacet	NTSTW	0.0122	**2000.31**
	ANTSTW	**0.0119**	2003.64

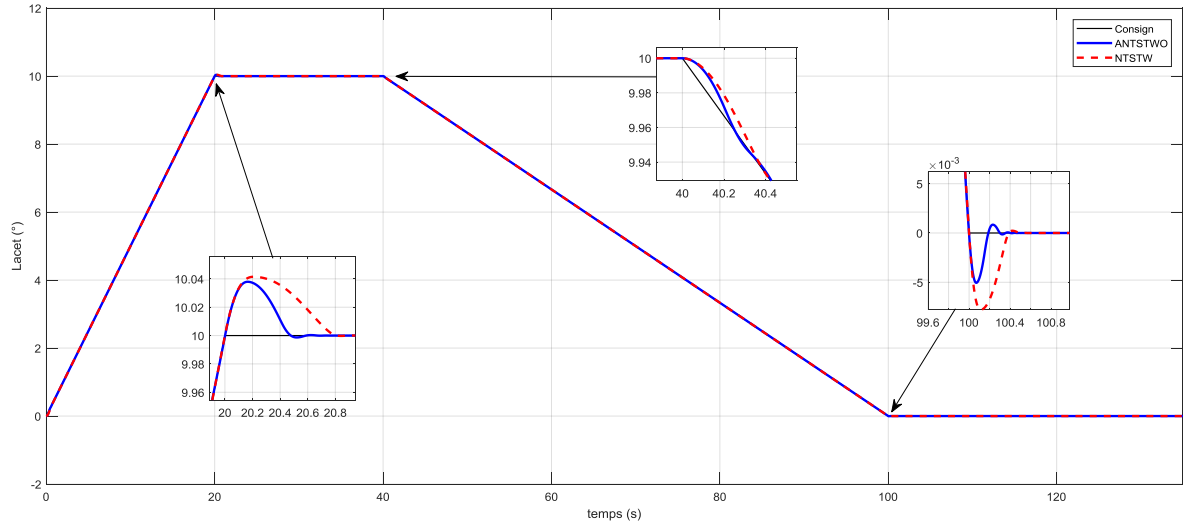
Tableau 17 Etude comparative entre les commandes appliquées en poursuite de trajectoire (sinusoïdale)

- *Poursuite de trajectoire en profile de vitesse*

Dans cette partie, nous avons testé les commandes ANTSTWO et ANTSTW pour une trajectoire qui contient une variation de la position sous une forme connue dans la littérature par la nomination profile de vitesse. Nous avons choisi un scénario qui permet de mettre en examen les commandes et voir leurs efficacités vis-à-vis du couplage. Les résultats graphiques de la réponse du système, les signaux des différentes commandes, l'évolution des gains adaptatifs ainsi qu'un tableau résumant les performances obtenues sont donnés respectivement dans les figures IV.22, IV.23, IV.24 et 18.

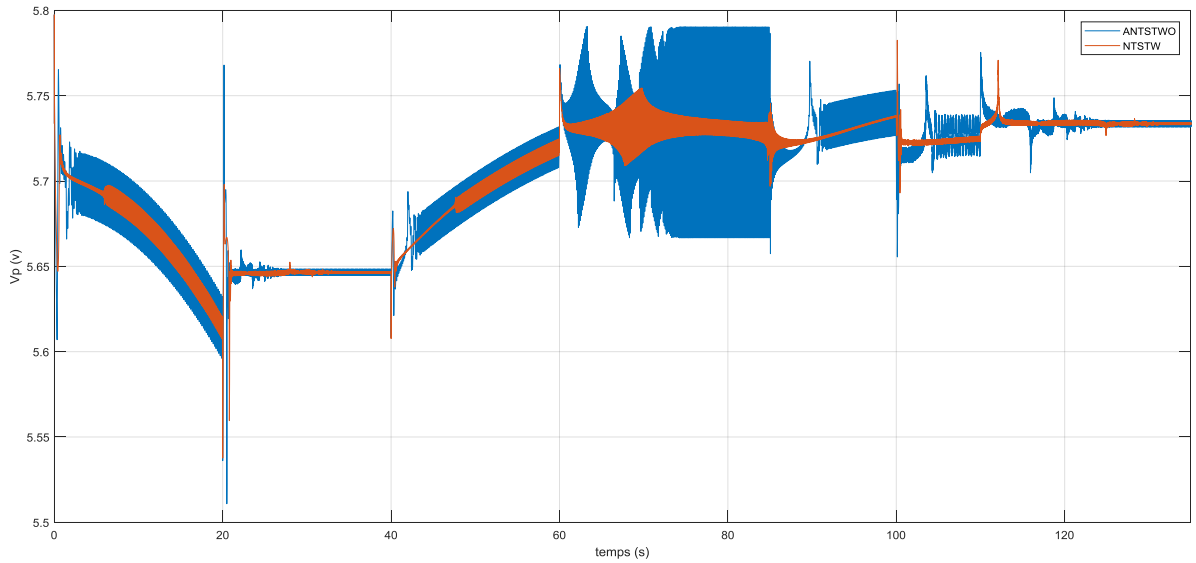


(a)

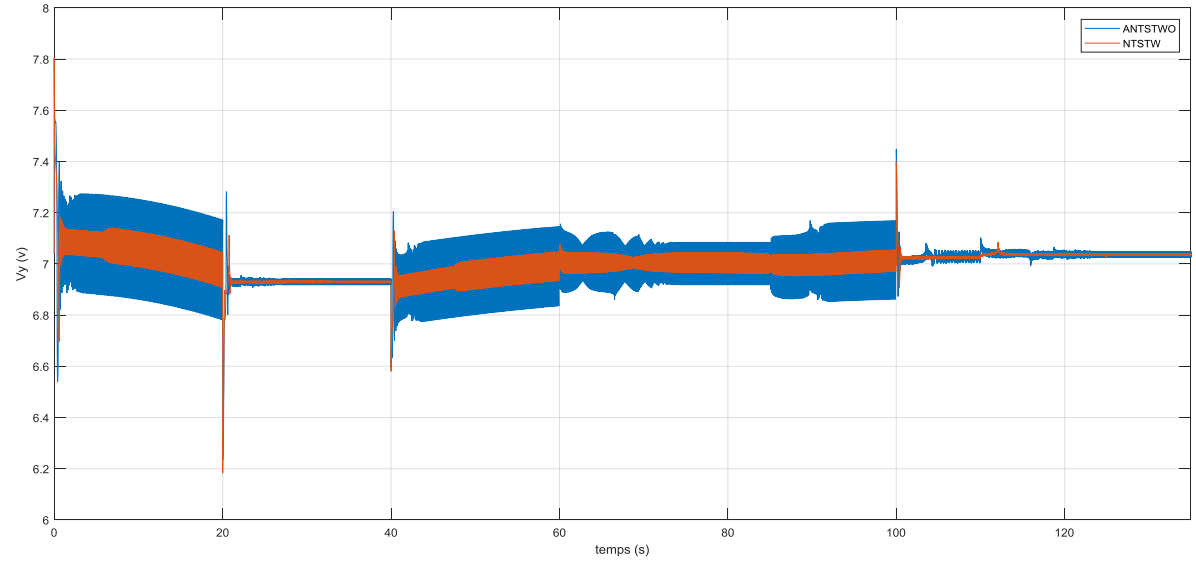


(b)

Figure IV. 22 Réponse du système dans le cas d'un profil de vitesse (a) tangage et (b) lacet



(a)



(b)

Figure IV. 23 Signaux de la commande dans le cas du profil de vitesse (a) tangage (b) lacet

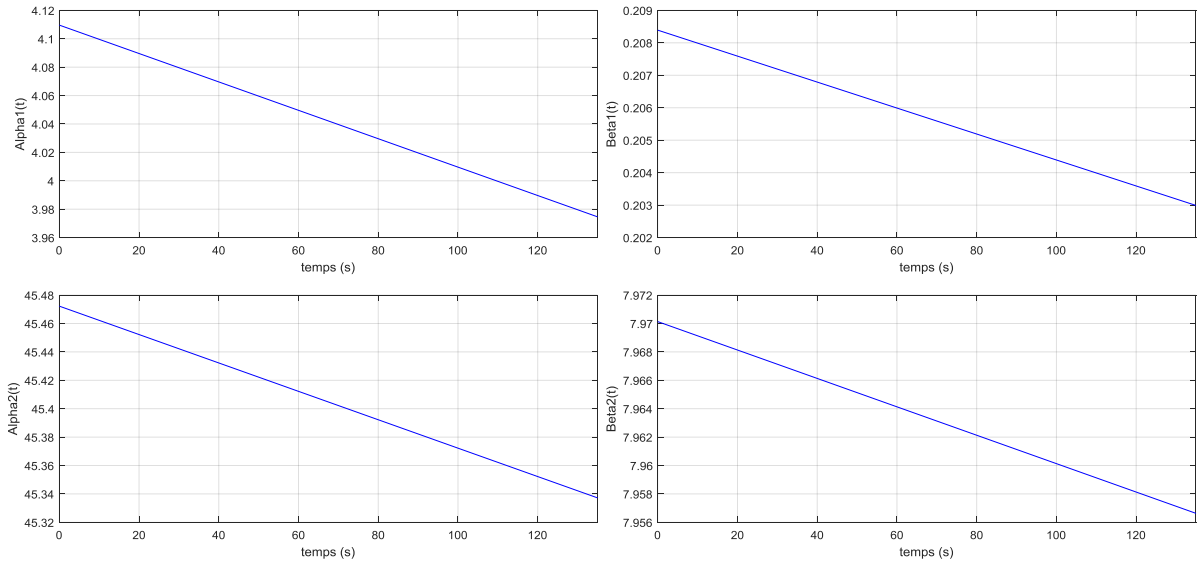


Figure IV. 24 Evolution des gains adaptatifs de la commande ANTSTWO (profile de vitesse)

Angle	Commande	ISE	ISCO
Tangage	NTSTW	0.0020	**4227.22**
	ANTSTW	**8.872 e-05**	4227.31
lacet	NTSTW	4.718 e-07	**6367.30**
	ANTSTW	**2.272 e-07**	6368.24

Tableau 18 Etude comparative entre les commandes appliquées en poursuite de trajectoire (profile de vitesse)

- *Discussion et interprétation des résultats*

La figure IV.16 vérifie le principe de la commande ANTSWO, i.e. lorsque le système est éloigné de la consigne, les gains adaptatifs augmentent, ce qu'est remarqué entre 0 et 0.7 *sec*, et lorsque la consigne est atteinte, les gains commencent à diminuer jusqu'à ce qu'une nouvelle perturbation survenue, à cet effet les gains recommencent à s'augmenter. Dans notre simulation (celle de la régulation) on peut visualiser ceci à l'instant 5 *sec* et 12 *sec*.

La commande ANTSTWO gère l'effet de couplage et de rejet de perturbations plus effectivement que la commande NTSTW, comme le montre la figure IV. 16. Le système corrige sa position et revient plus rapidement à la consigne. D'après le tableau (16), la commande ANTSTWO a minimisé l'indice ISE de 2.54% et 2.3% pour les positions de tangage et de lacet respectivement par rapport à la commande NTSTW, en raison de la diminution de l'amplitude du broutement, mais la consommation d'énergie a augmenté de 0,34 % pour lacet par la commande ANTSTWO, cette petite consommation est due à l'amélioration de la précision.

En mode poursuite de trajectoire, la commande ANTSTWO est également plus performante que NTSTW sur les deux trajectoires (sinusoïdal et profile de vitesse). Selon les figures (IV.19, IV.22). Nous pouvons voir que la commande NTSTW a du mal à suivre la trajectoire surtout en angle de tangage en raison de l'effet gravitationnel et du couplage, spécialement dans le cas de la trajectoire profile de vitesse, car le couple généré par le moteur du lacet affecte l'axe de tangage considérablement. La commande ANTSTWO a réussi à surmonter le problème, mais il en résulte une augmentation de l'amplitude des fluctuations, qui est interprété aussi par l'augmentation de l'indice ISCO. Pendant ce temps, l'indice ISE a diminué considérablement de 343,6 % et de 29,36 % par la commande ANTSTWO par rapport NTSTW à pour les deux trajectoires de tangage. Pour le cas du lacet, il n'y a pratiquement pas d'amélioration de ces indices (voir tableau 17 et 18).

IV.4. Conclusion

Nous avons présenté dans ce chapitre une étude comparative entre les différentes lois de commande et les algorithmes d'optimisation afin de trouver la meilleure commande à appliquer sur Aero Quanser en termes de performances et robustesse.

Les résultats ont montré que GWO est l'algorithme le plus performant par rapport au algorithmes WOA, SSA et ALO de point de vue solution optimale (minimum) de la fonction objective imposée, dont cette dernière a été choisi soigneusement afin de garantir un compromis entre centaines performances, ceci est valable quel que soit le type de commande et/ou le nombre de paramètres à optimisés.

Ensuite, les résultats obtenus montrent que la commande NTSTW est plus performante que les commandes PID, SMC, NTSMC et STW dans la plupart des aspects.

Enfin, la commande NTSTW a été comparée à la commande ANTSTWO soumise à de nombreux tests de robustesse. La commande originale appelée ANTSTWO que nous avons proposé dans ce PFE a montré une supériorité significative sur tous les plans (i.e. performances et robustesse) en mode régulation et en poursuite des trajectoires.

Chapitre V : Visualisation des résultats de simulation sur l’environnement ROS

V.1. Introduction

Les simulateurs ont joué un rôle essentiel dans la recherche en robotique pour tester rapidement et efficacement des nouvelles applications, stratégies de fabrication et des algorithmes de commande. Un simulateur en robotique est utilisé afin de créer des applications pour un ou plusieurs robots sans dépendre de la machine réelle, ce qui permet d'économiser du temps et de l'argent. Dans certains cas, ces applications peuvent être transférées sur le robot réel sans grandes modifications. Robot Operating System (ROS) [39] est un ensemble d'outils informatiques sous forme de logiciels libres open source (bibliothèques de packages et outils), pour aider à créer des applications robotiques. Dans ce chapitre nous avons construit un modèle 3D de l'Aero Quanser puis le simulé sur l'environnement Gazebo en fonction des consignes de Matlab/ Simulink.

V.2. Architecture de ROS

La philosophie de ROS consiste à créer un logiciel qui pourrait fonctionner pour de différents robots de même forme en faisant de petits changements dans le code. Le fonctionnement de ROS repose essentiellement sur :

1- l'architecture de communication Publish (publier) /Subscribe (abonner) qui divise les appareils clients du réseau en 2 catégories, Publishers (les éditeurs) et Subscribers (les abonnés) où (figure V.1):

- **Les publishers :** sont des appareils ou des nodes (nœuds) qui entrent des données dans le réseau.
- **Les subscibers :** sont les terminaux ou nodes qui reçoivent les données.
- **Les messages** : ce sont des données à envoyer sous la forme de différents **topics** (thèmes).
- **Topics** : représentent les structures des messages et les données à communiquer.
- **Les services :** gèrent le processus d'abonnement et publication des nodes aux topics. Les requêtes des nodes pour les interactions demandes/réponses.

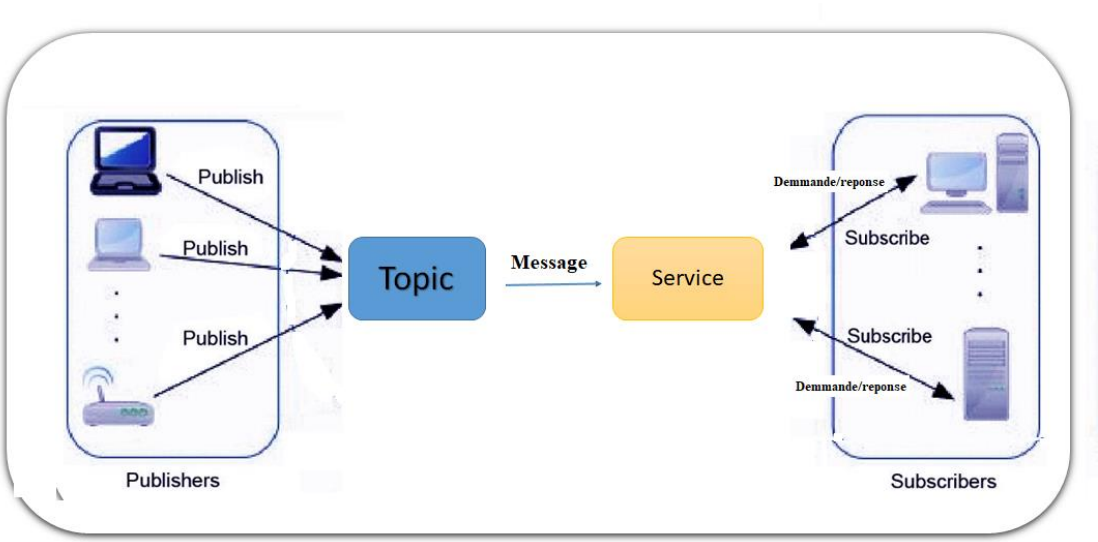


Figure V. 1 Schéma descriptif de l'architecture de communication publish/subscribe pour un seul topic

Le modèle de Publish/subscribe permet une communication très flexible de plusieurs-à-plusieurs, pour se faire, les nodes envoient des demandes aux services pour recevoir des messages du topic et les services répondent dès leurs disponibilités.

2- Le ROS Master est le cœur de ROS, puisqu'il fournit le nom d'enregistrement et consultation des graphes de calcul, il peut être considéré comme un interrupteur marche-arrêt de l'équipement électrique, il partage les topics et gère les communications entre les nodes et, l'échange de messages et le lancement des services.

Il est intéressant de souligner que ROS fournit suffisamment d'outils pour le traitement et l'analyse de données, la visualisation 3D (RViz), la journalisation en temps réel des expériences sur ces robots et les exécutés hors ligne avec (rosbag/rxbag), le traçage des données (rxplot) et la visualisation de l'ensemble du réseau ROS structure (rxgraphe).

Développement d'un fichier URDF

L'URDF (Unified Robots Description Format) est un format XML pour décrire un modèle de robot, il fournit une description du robot en termes de propriétés cinématiques, dynamiques et visuelles. Le fichier est constitué d'une arborescence alternant des éléments de Link (composant) et joint (articulation). Le fichier URDF contient également :

- La balise <**robot**> contient le nom du robot, qui est affiché dans tous les sous-systèmes du ROS et sa description.

- La balise <**link**> contient le nom du Link et représente un composant du robot, il décrit sa masse, son inertie et son centre de masse accompagné d'une description de la partie visuelle du modèle (tag <**visual**>), qui contient des informations sur la géométrie de l'objet <**géométrie**> (Valeurs possibles - <**cylindre**>, <**box**>, <**sphere**> et <**mesh**>), la balise contient aussi la position du composant et des informations supplémentaires telles que la couleur, la texture, et d'autres.

Toutes les propriétés dynamiques de chaque Link sont exprimées par rapport à son qualité de Link parent. La figure V.2 schématise un link revolute et la relation entre le child et le parent :

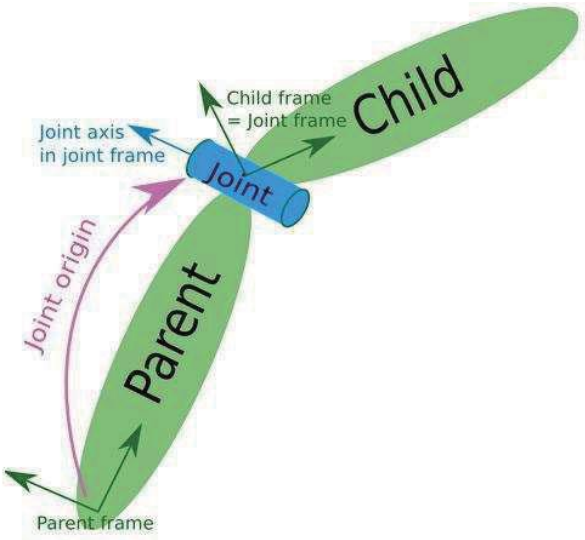


Figure V. 2 Joint revolute qui relie les Link Child et Link Parent

- La balise <**origin**> ajuste la position initiale d'un Link dans l'espace.

- La balise <**inertial**> définit la masse du Link et le décalage du centre de masse. Elle est nécessaire pour simuler la gravité.

- La balise <**collision**> initie les paramètres de collision, où ROS crée une couche invisible au-dessus de l'objet visuel, qui est une coque pouvant entrer en collision avec d'autres objets.

- La balise <**joint**> est une connexion qui relie les deux objets <Link> appelés Link Parent et Link Child. La balise <joint> contient les noms des deux Link, l'emplacement du joint, les informations sur les contrôleurs et le type de connexion. Le type de joint [40] peut être l'un des suivants :

- **Fixe** : comme son nom l'indique, ce type de joints est fixe et il n'a aucun degré de liberté de mouvement.
- **Revolute** : c'est une articulation rotoide dont le mouvement de rotation s'effectue autour d'un axe sortant de l'articulation, l'angle de rotation à une limite supérieure et une limite inférieure.
- **Continuous** : contrairement à l'articulation Revolute celle-ci n'a pas de limitation et peut effectuer des rotations de 360 degrés.
- **Prismatic** : Ce type d'articulation coulisse le long d'un axe dans les limites spécifiées.
- **Planar** : Dans ce type d'articulation, le mouvement est perpendiculaire à l'axe spécifié.
- **Floating** : il ne s'agit pas exactement d'un joint car tous les six degrés de liberté sont libres

La figure V.3 présente une partie de code URDF développé pour l' Aero quanser :

```
<robot
  name="Nom_du_robot">
   <link
    name="Nom_du_Link">
    <inertial>
      <origin
        xyz="0 0 0"
        rpy="0 0 0" />
      <mass
        value="0.103908177017482" />
       <inertia
        ixx="2.1809893405232E-05"
        ixy="-3.30493388769738E-22"
        ixz="0"
        iyy="1.5910939605802E-05"
        iyz="1.89896668640322E-22"
        izz="2.1809893405232E-05" />
    </inertial>
    <visual>
      <origin
        xyz="0 0 0"
        rpy="0 0 0" />
      <geometry>
        <mesh
          filename="package://Aero_Quanser/meshes/Link_01.STL" />
      </geometry>
      <material
        name="">
        <color
          rgba="0.898039215686275 0.917647058823529 0.929411764705882 1" />
      </material>
    </visual>
    <collision>
      <origin
        xyz="0 0 0"
        rpy="0 0 0" />
      <geometry>
        <mesh
          filename="package://Aero_Quanser/meshes/Link_01.STL" />
      </geometry>
    </collision>
  </link>
  <joint
    name="Joint_01"
    type="continuous">
</joint>
```

```
    <origin
      xyz="-0.00107015680451168 -0.178 0.0192600680485297"
      rpy="1.5707963267949 0 -0.521522624384189" />
    <parent
      link="base_link" />
    <child
      link="Link_01" />
    <axis
      xyz="0 1 0" />
</robot>
```

Figure V. 3 Exemple de la description URDF d'une partie de l'Aero Quanser

Création de l'Aéro Quanser par solidWorks

Au lieu de créer manuellement le robot, on peut passer par SolidWorks pour créer les différentes parties du robot et faire leurs assemblages, puis utiliser l'outil dynamique 'SW to URDF exporter'. Ce plugin fonctionne de telle manière qu'une pièce SolidWorks est convertie en un seul élément de Link URDF, tandis que lorsqu'un assemblage SolidWorks est exporté, un joint est détecté au niveau des points de liaisons toutes en définissant le type de connections. Il est important de citer ici que, lors de l'assemblage, on ajoute les connecteurs (articulations) entre les différentes parties tout en définissant leurs positions et orientations.

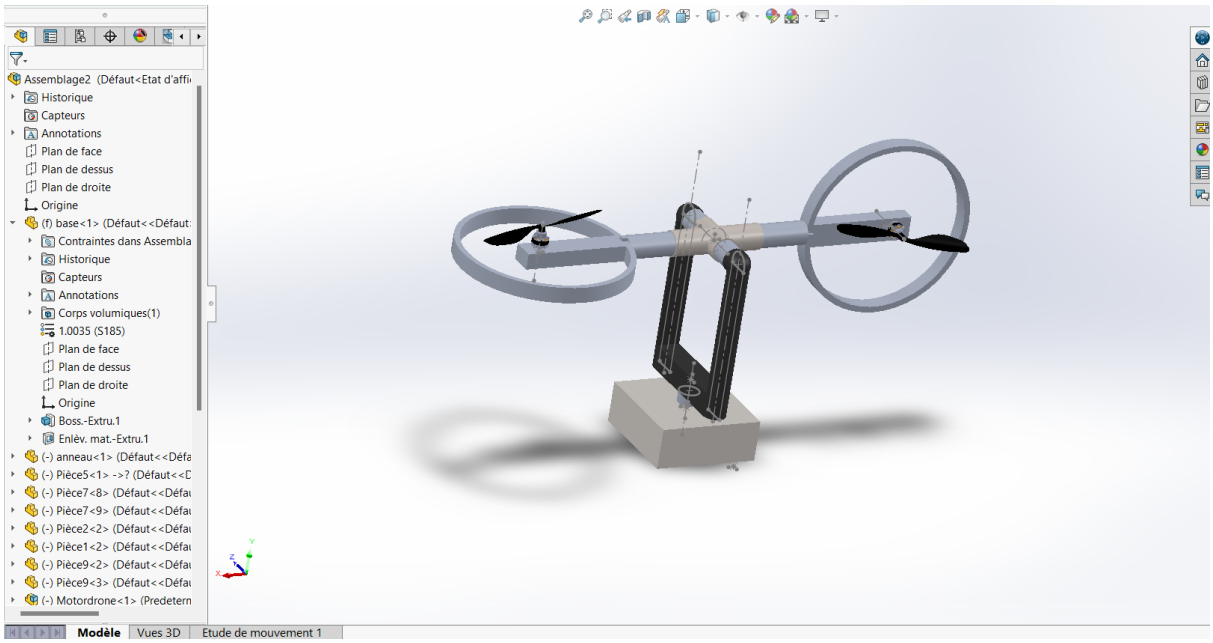


Figure V. 4 Assemblage du robot Aero_Quanser à l'aide de SolidWorks

Cette approche garantit une obtention directe des modules décrits selon URDF à partir du plugin d'exportation SolidWorks vers URDF. Un haut niveau de personnalisation peut être atteint avec plusieurs combinaisons de modules, en faisant varier non seulement leurs ordres de connexion, comme le montre la Figure V.4, mais aussi leurs tailles, leurs matériaux et leurs formes.

La valeur ajoutée d'un logiciel de CAO (Conception assistée par ordinateur) réside dans la possibilité d'exporter les fichiers de maillage (mesh files) et l'estimation des paramètres dynamiques tels que la masse, les coordonnées du centre de masse et le tenseur d'inertie.

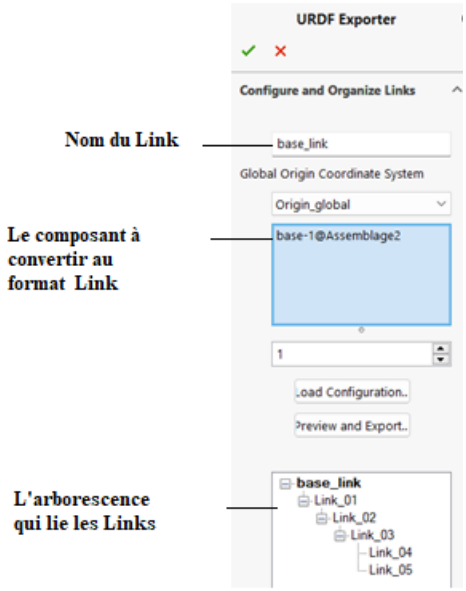


Figure V. 5 Interface du plugin URDF exportée démontrant l'architecture arbre

Le fichier URDF est le format de description standard dans les bibliothèques ROS, il peut être téléchargé dans des simulateurs ROS et des planificateurs de mouvement tels que Gazebo et MoveIt pour suivre en temps réel l'évolution du robot dans l'espace 3D.

V.3. Utilisation de Rviz pour la visualisation du modèle du robot Aero-Quanser :

RViz est un outil de visualisation extrêmement utile de ROS, qui peut être utilisé pour afficher les sujets communiqués entre les nodes. RViz a une interface utilisateur graphique (GUI) permettre aux utilisateurs de configurer et de modifier les robots.

Pour l'implémentation du robot Aero-Quanser, nous utilisons le fichier 'launch' (figure V.6) qui contient divers paramètres comme la description du robot, les packages utilisés et le module 'gui'.

```
<launch>

  <arg
    name="model" />
  <param
    name="robot_description"
    textfile="$(find Aero_Quanser)/urdf/Aero_Quanser.urdf" />
  <node
    name="joint_state_publisher_gui"
    pkg="joint_state_publisher_gui"
    type="joint_state_publisher_gui" />
  <node
    name="robot_state_publisher"
    pkg="robot_state_publisher"
    type="robot_state_publisher" />
  <node
    name="rviz"
    pkg="rviz"
    type="rviz"
    args="-d $(find Aero_Quanser)/urdf.rviz" />
</launch>
```

Figure V. 6 Fichier launch pour le lancement d'Rviz

A l'exécution du ficher launch, il sera créé les nodes essentiels pour établir un réseau ROS, qui peut contrôler la position de chaque joint (articulation) et fournit une méthode simple pour vérifier la connexion commune entre les Links.

La figure V.7 ci-dessous présente les principaux nodes et topics utilisés pour transporter les informations.

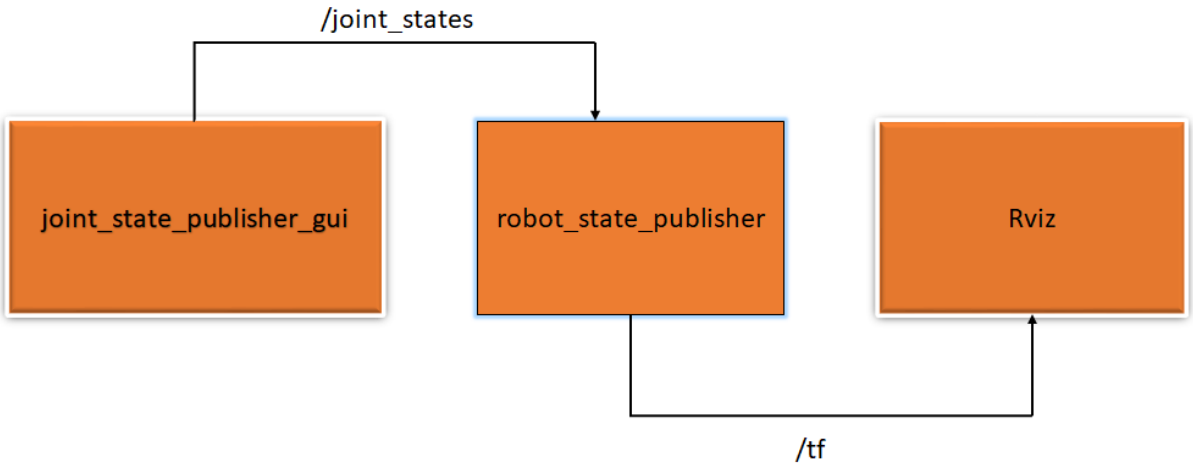


Figure V. 7 Principaux nodes et topics utilisés

Joint state Publisher *gui* : ce node lit la description des paramètres du robot et publie toutes les informations sur les joints non fixes telles que, la position, la vitesse et l'accélération à une

fréquence à spécifier. Il fournit également une interface graphique de contrôle pour définir la position des joints.

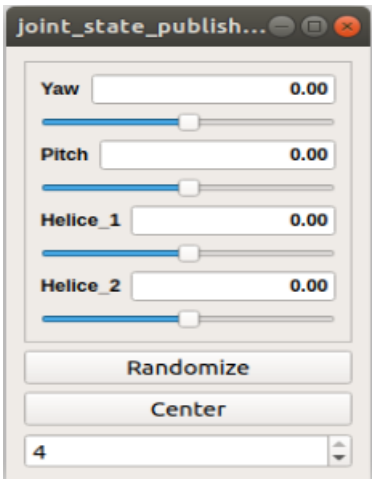


Figure V. 8 Outil de commande joint_state_publisher_gui

Robot state Publisher : ce node diffuse l'état du robot à la bibliothèque de transformation ***tf*** pour ce faire il possède en interne un modèle cinématique du robot, ainsi compte tenu des positions des joints du robot, 'Robot state Publisher' peut calculer et diffuser la pose 3D de chaque maillon du robot.

tf : la bibliothèque tf a été conçue pour fournir un moyen standard pour traquer la position de tous les joints dans les différents repères de coordonnés du système (les repères liant les Links connectés) en fonction du temps. La figure V.9 présente l'Aéro Quanser visualisé par l'usage de Rviz.

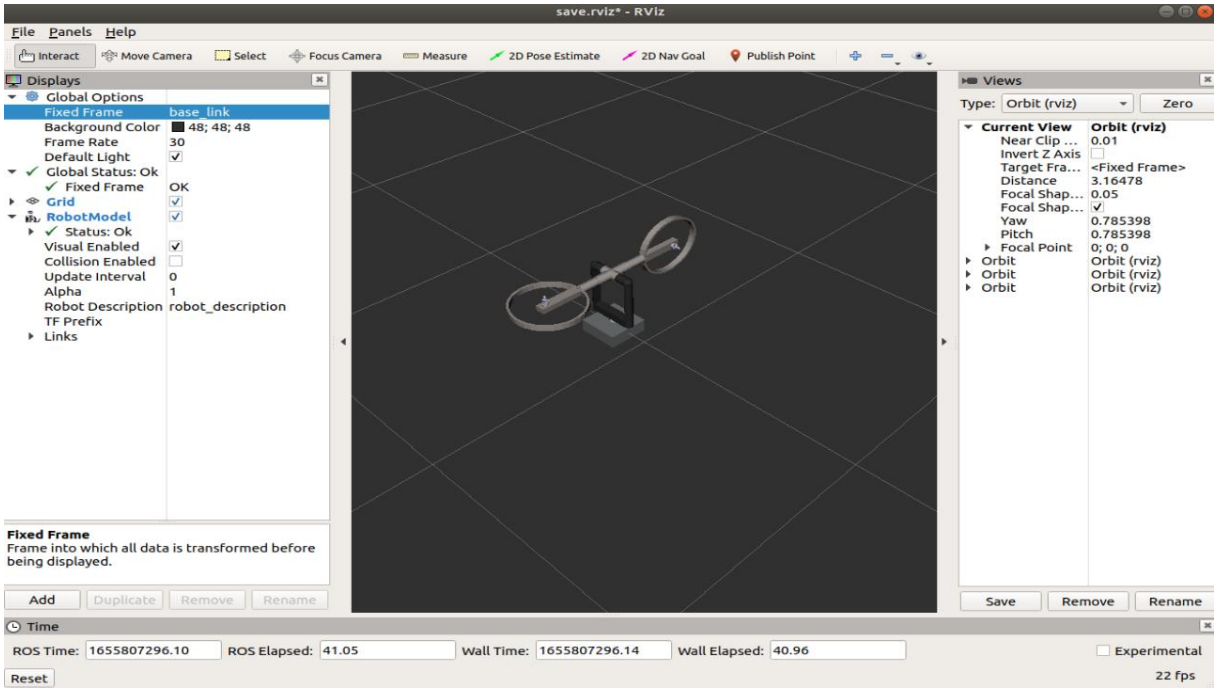


Figure V. 9 Simulation sur l'environnement Rviz

V.4. Simulation sur la plateforme Gazebo

Gazebo [41] est un simulateur virtuel 3D et multi-robots qui permet de simuler avec précision la cinématique et la dynamique d'un robot ou plusieurs robots, objets ou capteurs dans des environnements complexes, intérieurs et/ou extérieurs, réalistes et en 3D. De nombreux chercheurs ont utilisé Gazebo pour développer et exécuter des expériences uniquement dans un environnement simulé, ceci va permettre de dire que Gazebo est proche de la réalité.

Il y a une grande différence entre la visualisation RViz et Simulation Gazebo. Rviz est utilisé pour afficher les positions des Links, mais Gazebo peut être considéré comme une copie expérimentale d'un robot réel dans un monde virtuel.

Une transmission est un élément de contrôle qui transforme les efforts et les flux appliqués sur le joint de telle sorte que la puissance produite reste constante. La transmission met en œuvre un effort et un flux bidirectionnel uniforme sur une interface partagée à travers les types de transmission (effort, vélocité (vitesse), position et force). Les contrôleurs de Gazebo ont besoin de l'interface de transmission correspondante dans le fichier URDF.

On peut configurer un contrôleur d'effort d'un joint nommé Yaw par le bloc de code ci-dessous.

```
    <transmission name="Yaw">
    <type>transmission_interface/SimpleTransmission</type>
    <joint name="Yaw">
      <hardwareInterface>hardware_interface/EffortJointInterface</hardwareI
nterface>
    </joint>
    <actuator name="motor_Yaw">
      <hardwareInterface>hardware_interface/EffortJointInterface</hardwareI
nterface>
      <mechanicalReduction>1</mechanicalReduction>
    </actuator>
  </transmission>
```

Figure V. 10 Configuration de l'interface de transmission des joints yaw

En plus des balises de transmission, il est nécessaire d'ajouter un plugin Gazebo à notre URDF qui analyse réellement la transmission, charge les interfaces matérielles et le contrôleur approprié à la gestion. Le plugin Gazebo-ros-control est très simple, bien qu'il soit également extensible via une architecture de plugin supplémentaire, pour créer des interfaces matérielles de robot personnalisées entre le ros-Control et Gazebo (Le package peut être créé et exécuté séparément puis affilié au système). l'insertion du bloc de code ci-dessous active le plugin Gazebo-ros-control afin de nous permettre d'exécuter une liste de services de gestion du contrôleur, qui peut être utilisé pour lister, démarrer, arrêter ou changer de contrôleur.

```
<!-- ros_control plugin -->
<gazebo>
  <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
    <robotNamespace>/Aero_Quanser</robotNamespace>
    <robotSimType>gazebo_ros_control/DefaultRobotHWSim</robotSimType>
  </plugin>
</gazebo>
```

Figure V. 11 Gazebo-Ros-control plugin code

Afin de commander notre robot il est essentiel d'ajouter un contrôleur, nous utiliserons dans cette partie un contrôleur de position. Dans le fichier de configuration, nous définissons les types de contrôleurs, les noms communs et les paramètres du PID pour suivre les consignes sur les angles de tangage et de lacet lors de la visualisation.

```
Aero_Quanser:
  # Publish all joint states -----------------------------------
  joint_state_controller:
    type: joint_state_controller/JointStateController
    publish_rate: 50

  # Position Controllers ---------------------------------------
  joint1_position_controller:
    type: effort_controllers/JointPositionController
    joint: Yaw
    pid: {p: 60.0, i: 30.0, d: 50.0}
  joint2_position_controller:
    type: effort_controllers/JointPositionController
    joint: Pitch
    pid: {p: 60.0, i: 30.0, d: 50.0}
```

Figure V. 12 Code définissant le type de contrôleur, le nom des joints et certains paramètres.

Le message de commande directement depuis le terminal est un moyen simple pour contrôler notre robot, il suffit d'ouvrir un terminal pour envoyer un topic avec format spécifié aux joints. De cette façon, nous pouvons effectuer certaines tâches simples, par exemple, envoyer une instruction simple pour ramener toutes les joints à la position initiale.

```
rostopic pub /Aero_Quanser/joint1_position_controller/command
std msgs/Float64 "data: 0.0"
```

Afin de visualiser l'ensemble, nous calculerons les positions à partir de la plateforme Matlab/Simulink puis les transférer à partir des topics /Aero_Quanser/joint_position_controller/command au contrôleur de Gazebo et suivre avec un retour l'état des joints à partir du topic /Aero_Quanser/joint1_position_controller/state.

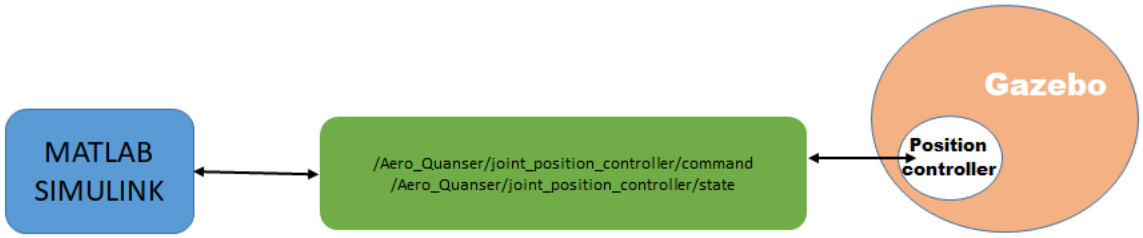


Figure V. 13 Méthode de commande

Nous utiliserons le toolbox Ros de Simulink sur Matlab pour envoyer les commandes au contrôleur de position de Gazebo. La figure V. 14 présente la commande ANTSTWO que nous avons proposé dans le cas de son application sur ROS. La figure V. 15 illustre le résultat de la simulation, la première figure représente la position du départ du robot et la deuxième sa position finale.

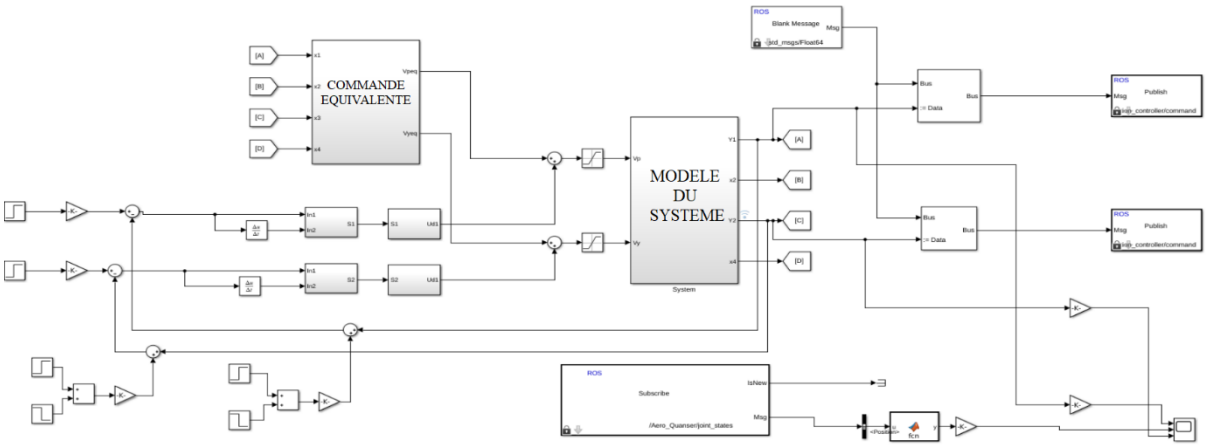
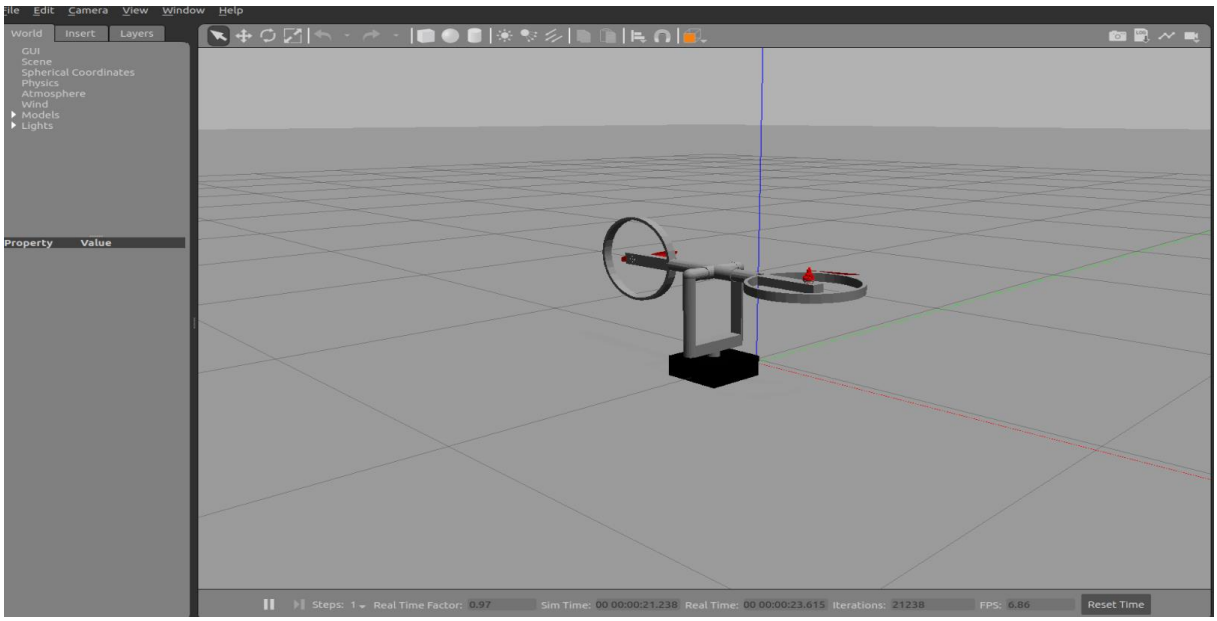
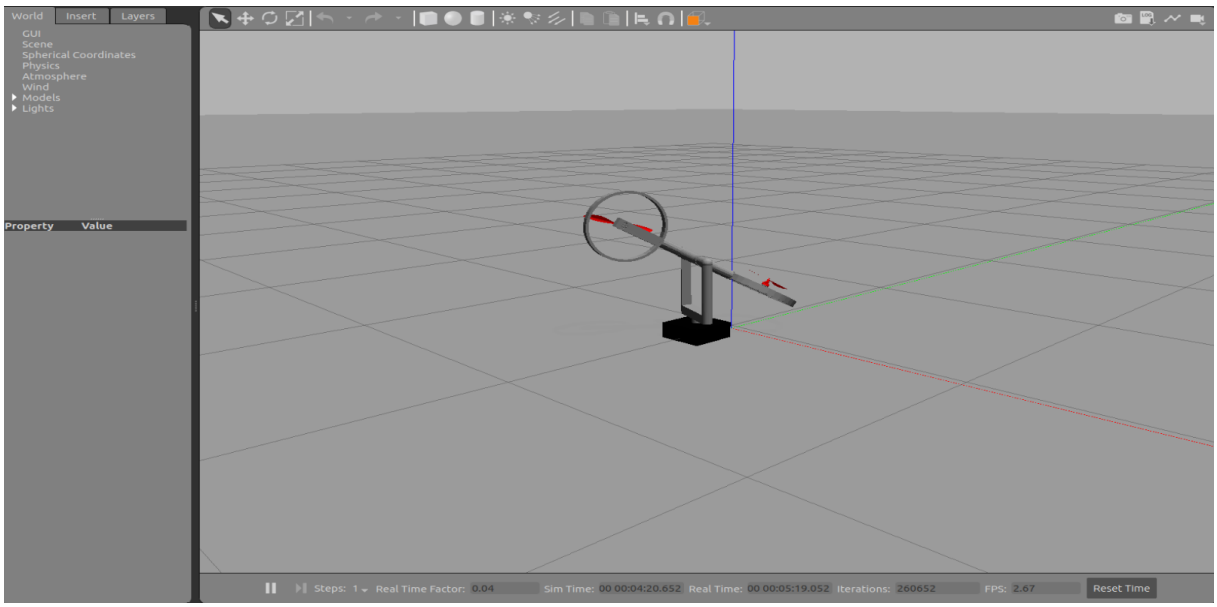


Figure V. 14 Schéma bloc de la commande ANTSTWO dans le cas de l'application sur ROS



(b)



(b)

Figure V. 15 Simulation en condition réelle du robot sur Gazebo (a) position initiale et (b) position finale

V.5. Conclusion :

A travers ce chapitre, nous avons utilisé l'environnement ROS pour la création d'un modèle 3D du robot Aero-Quanser, vérifié sa structure, ces connections et ces propriétés dynamiques sur la plateforme Rviz. Ensuite, nous avons simulé le système sur la plateforme Gazebo pour visualiser son comportement à la commande ANTSTWO proposée dans un environnement virtuel proche de la réalité.

Conclusion Générale

Ce projet de fin d'étude s'insère dans la synthèse des lois de commande robuste, optimale et adaptative pour les systèmes avioniques, il s'agit de proposer une ou plusieurs commandes en utilisant la théorie de la commande par mode de glissement d'ordre 1, et 2 classique ou basée sur des surfaces non linéaires appelée commande par terminal sliding mode. Le système étudié est l'Aero Quanser qui est un simulateur d'hélicoptère permettant de se rapprocher à la réalisé des hélicoptères d'avantage. Ce dernier est un système multi-variables à deux entrées et deux sorties, caractérisé par des non linéarités fortes, un couplage complexe et une dynamique instable en boucle ouverte, ce qui rend sa commande compliquée.

Nous avons étudié plusieurs lois de commande telles que, la commande par le régulateur PID et les commandes par mode de glissement SMC, NTSMC, STW, NTSTW et ANTSTWO. Les paramètres de ces commandes ont été trouvés par les quatre algorithmes méta-heuristiques à savoir, GWO, WOA, ALO et SSA. Les commandes optimisées ont été appliquées sur le simulateur d'Hélicoptère Aero-Quanser.

En détail, la première partie fut une présentation du simulateur de vol Aero Quanser avec ses caractéristiques et son modèle dynamique pour mieux appréhender son fonctionnement.

Par la suite nous avons synthétisé les différentes lois de commande à savoir PID, SMC, NTSMC, STW et NTSTW et évalué les réponses obtenues dans le but de trouver la commande la plus robuste vis-à-vis les perturbations externes, les erreurs de modélisation et qui garantit au même temps les meilleures performances sur l'Aero Quanser.

La troisième partie est consacrée à l'étude et à la comparaison des quatre algorithmes méta-heuristiques à savoir, GWO, WOA, ALO et SSA, ces algorithmes ont été utilisés pour la détermination des paramètres des commandes synthétisées. Nous avons conclu que, la plupart des commandes optimisées par GWO sont les meilleurs en performances. Nous avons trouvé que la commande NTSTW dont les paramètres sont optimisés par le GWO est la meilleure commande.

Dans notre quête sur la robustesse et l'optimisation, nous avons proposé une nouvelle approche de commande appelée ANTSTWO, cette commande est à la fois optimale et adaptative. La commande ANTSTWO est une nouvelle approche adaptative, optimale et aussi robuste. Cette approche a permet de régler le problème de choix des paramètres de la commande, et elle a prouvé ses qualité vis-à-vis une poursuite des trajectoires (sinusoïdale et profile de vitesse) ainsi que vis-à-vis d'une perturbation externe.

Pour finaliser notre travail, nous avons utilisé l'environnement ROS pour la création d'un modèle 3D du robot Aero-Quanser, la vérification de sa structure, de ces connections et de ces propriétés dynamiques sur la plateforme Rviz. Par la suite, nous avons simulé l'Aero-Quanser sur la plateforme Gazebo pour visualiser sa réponse à la commande ANTSTWO et suivre son comportement dans un environnement virtuel proche de la réalité.

Plusieurs perspectives peuvent être envisagées :

1- Avec la possibilité de l'acquisition de ce système au niveau de l'école, la première action systématique est l'application expérimentale de ces commandes sur le système réel.
2- L'optimisation méta-heuristique multicritères est aussi envisageable dans le futur.

Bibliographie

[1] Quanser Inc, Quanser AERO Laboratory Guide, Quanser: Arkham, Canada, 2016.
[2] SCHLANBUSCH, Siri Marte. *Adaptive Backstepping Control of Quanser 2DOF Helicopter: Theory and Experiments*. 2019. Thèse de maîtrise. Universitetet i Agder; University of Agder.
[3] ALLIOT, Jean-Marc et DURAND, Nicolas. Algorithmes génétiques. *Centre d'Etudes de la Navigation Aérienne*, 2005.
[4] DORIGO, Marco, BIRATTARI, Mauro, et STUTZLE, Thomas. Ant colony optimization. *IEEE computational intelligence magazine*, 2006, vol. 1, no 4, p. 28-39.
[5] MARINI, Federico et WALCZAK, Beata. Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 2015, vol. 149, p. 153-165.
[6] TEODOROVIĆ, Dušan. Bee colony optimization (BCO). In: *Innovations in swarm intelligence*. Springer, Berlin, Heidelberg, 2009. p. 39-60.
[7] MIRJALILI, Seyedali, MIRJALILI, Seyed Mohammad, et LEWIS, Andrew. Grey wolf optimizer. *Advances in engineering software*, 2014, vol. 69, p. 46-61.
[8] MIRJALILI, Seyedali et LEWIS, Andrew. The whale optimization algorithm. *Advances in engineering software*, 2016, vol. 95, p. 51-67.
[9] MIRJALILI, Seyedali, GANDOMI, Amir H., MIRJALILI, Seyedeh Zahra, *et al.* Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 2017, vol. 114, p. 163-191.
[10] MIRJALILI, Seyedali. The ant lion optimizer. *Advances in engineering software*, 2015, vol. 83, p. 80-98.
[11] CHAMOLA, Vinay, KOTESH, Pavan, AGARWAL, Aayush, *et al.* A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques. *Ad hoc networks*, 2021, vol. 111, p. 102324.
[12] MARTINI, Adnan. *Modélisation et commande de vol d'un hélicoptère drone soumis à une rafale de vent*. 2008. Thèse de doctorat. Université Paul Verlaine-Metz.
[13] MÉCANIQUE DU VOL. lavionnaire [en ligne]. [sans date] [consulté le 22 mai 2022]. Disponible sur : https://www.lavionnaire.fr/HelicoMecaVol.php
[14] C. Rotaru, and M. Todorov, "Helicopter Flight Physics", in Flight Physics - Models, Techniques and Technologies. London, United Kingdom: IntechOpen, 2017
[15] ABDELMAKSOUD, Sherif I., MAILAH, Musa, et ABDALLAH, Ayman M. Practical real-time implementation of a disturbance rejection control scheme for a twin-rotor helicopter system using intelligent active force control. *IEEE Access*, 2020, vol. 9, p. 4886-4901.
[16] ÅSTRÖM, Karl Johan. Control system design lecture notes for me 155a. Department of Mechanical and Environmental Engineering University of California Santa Barbara, 2002, vol. 333.
[17] ÅSTRÖM, Karl Johan. Control system design lecture notes for me 155a. Department of Mechanical and Environmental Engineering University of California Santa Barbara, 2002, vol. 333.
[18] V. Utkin, "Variable structure systems with sliding modes," in *IEEE Transactions on Automatic Control*, vol. 22, no. 2, pp. 212-222, April 1977, doi: 10.1109/TAC.1977.1101446.
[19] J.J.E. Slotine and J.A. Coetsee, « Adaptive sliding controller synthesis for non-linear systems» , Int.J. Control, N°6,p.1631-1651, 1986.
[20] FILIPPOV, Aleksei Fedorovich. *Differential equations with discontinuous righthand sides: control systems*. Springer Science & Business Media, 2013.
[21] BREGEAULT, Vincent. *Quelques contributions à la théorie de la commande par modes glissants*. 2010. Thèse de doctorat. Ecole Centrale de Nantes (ECN).
[22] M. Zak, "Terminal attractors for addressable memory in neural network," Phys. Lett. A, vol. 133, nos. 1/2, pp. 18–22, 1988.
[23] YU, Xinghuo, FENG, Yong, et MAN, Zhihong. Terminal sliding mode control–An overview. *IEEE Open Journal of the Industrial Electronics Society*, 2020, vol. 2, p. 36-52.
[24] Y. Feng, X. Yu, and Z. Man, "Non-singular adaptive terminal sliding mode control of rigid manipulators," Automatica, vol. 38, no. 12, pp. 2159–2167, 2002
[25] LEVANT, Arie. Higher-order sliding modes, differentiation and output-feedback

control. *International journal of Control*, 2003, vol. 76, n [2]o 9-10, p. 924-941.
[26] BOUYEKNI, A., TALEB, R., BOUDJEMA, Z., *et al.* Commande par mode glissant d'ordre 2 pour une capture maximale d'énergie d'une turbine éolienne. *Journal of Renewable Energies*, 2018, vol. 21, no 1, p. 19-26.
[27] IOANNOU, Petros A. et SUN, Jing. *Robust adaptive control*. Courier Corporation, 2012.
[28] YAHIAOUI, Adel, FODHIL, Fadhila, BENMANSOUR, Khelifa, *et al.* Grey wolf optimizer for optimal design of hybrid renewable energy system PV-Diesel Generator-Battery: Application to the case of Djanet city of Algeria. *Solar Energy*, 2017, vol. 158, p. 941-951.
[29] NADERI, Ehsan, AZIZIVAHED, Ali, NARIMANI, Hossein, *et al.* A comprehensive study of practical economic dispatch problems by a new hybrid evolutionary algorithm. *Applied Soft Computing*, 2017, vol. 61, p. 1186-1206.
[30] ARORA, Sankalap et KAUR, Ranjit. Nature inspired range based wireless sensor node localization algorithms. 2017.
[31] SAIKIA, Lalit Chandra et SINHA, Nidul. Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller. *International Journal of Electrical Power & Energy Systems*, 2016, vol. 80, p. 52-63.
[32] EMARY, Eid, ZAWBAA, Hossam M., et HASSANIEN, Aboul Ella. Binary ant lion approaches for feature selection. *Neurocomputing*, 2016, vol. 213, p. 54-65.
[33] TRIVEDI, Indrajit N., JANGIR, Pradeep, et PARMAR, Siddharth A. Optimal power flow with enhancement of voltage stability and reduction of power loss using ant-lion optimizer. *Cogent engineering*, 2016, vol. 3, no 1, p. 1208942.
[34] KAMBOJ, Vikram Kumar, BHADORIA, Ashutosh, et BATH, S. K. Solution of non-convex economic load dispatch problem for small-scale power systems using ant lion optimizer. *Neural Computing and Applications*, 2017, vol. 28, no 8, p. 2181-2192.
[35] YAO, Peng et WANG, Honglun. Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle. *Soft Computing*, 2017, vol. 21, no 18, p. 5475-5488.
[36] BABERS, Ramadan, GHALI, Neveen I., HASSANIEN, Aboul Ella, *et al.* Optimal community detection approach based on Ant Lion Optimization. In: *2015 11th international computer engineering conference (ICENCO)*. IEEE, 2015. p. 284-289.
[37] DAS, Saptarshi, PAN, Indranil, DAS, Shantanu, *et al.* A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices. *Engineering Applications of Artificial Intelligence*, 2012, vol. 25, no 2, p. 430-442.
[38] SHTESSEL, Yuri, TALEB, Mohammed, et PLESTAN, Franck. A novel adaptive-gain supertwisting sliding mode controller: Methodology and application. *Automatica*, 2012, vol. 48, no 5, p. 759-769.
[39] ROS. Home page. ROS [en ligne]. [Sans date] [consulté le 22 mai 2022]. Disponible sur : https://www.ros.org/
[40] LINNER, Thomas, SHRIKATHIRESAN, Alaguraj, VETRENKO, Maxim, et al. Modeling and operating robotic environments using Gazebo/ROS. In: Proceedings of the 28th international symposium on automation and robotics in construction (ISARC2011). 2011. p. 957-962.
[41] QIAN, Wei, XIA, Zeyang, XIONG, Jing, et al. Manipulation task simulation using ROS and Gazebo. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). IEEE, 2014. p. 2594-2598.
[42] KIM, Seok-Kyoon et AHN, Choon Ki. Performance-Boosting Attitude Control for 2-DOF Helicopter Applications via Surface Stabilization Approach. *IEEE Transactions on Industrial Electronics*, 2021, vol. 69, no 7, p. 7234-7243.

Annexe

1- Les valeurs numériques des différents paramètres du Aero Quanser sont données par :

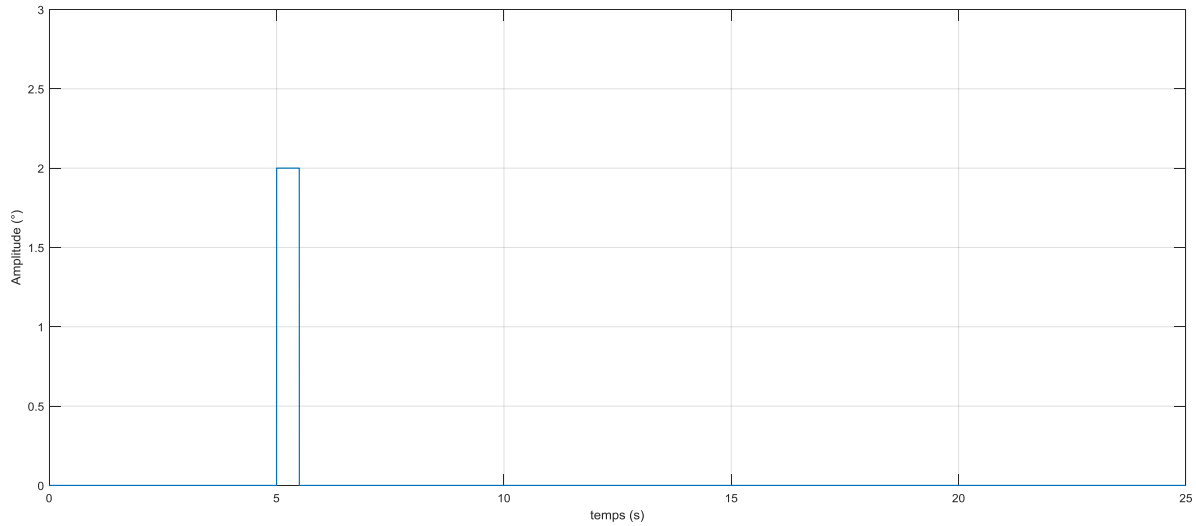
Symbole	Description	Valeur
Moteur à courant continue		
R_m	Résistance terminale	$8.4\ \Omega$
K_t	Constante de couple	$0.042\ \mathrm{N.m/A}$
K_m	Constante f.e.m moteur	$0.042\ \mathrm{V/(rad/s)}$
J_m	Moment d'inertie du moteur	$4.04\ 10^{-6}\ \mathrm{Kg.m^2}$
L_m	Inductance du moteur	$1.16\ \mathrm{mH}$
Hélice		
K_d	Coefficient de traînée	$10^{-5}\ \mathrm{V/(rad/s)}$
J_h	Inertie du moyeu d'hélice	$3.04\ 10^{-9}\ \mathrm{Kg.m^2}$
J_p	Inertie de l'hélice	$7.2\ 10^{-6}\ \mathrm{Kg.m^2}$

Tableau Paramétrées de l'hélice et le moteur a courant continue de l'Aero Quanser [1]

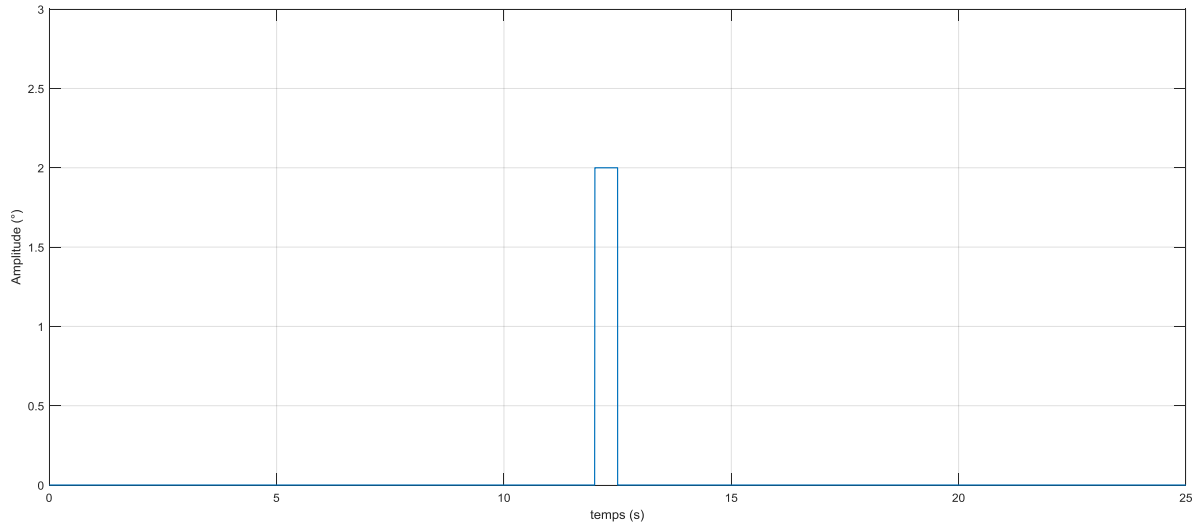
Symbole	description	valeur
J_p	Moment d'inertie autour de l'axe de tangage	0.0215 Kg.m^2
J_y	Moment d'inertie autour de l'axe de lacet	0.0237 Kg.m^2
D_p	constante de frottement de tangage	0.0071 N/V
D_y	constante de frottement de lacet	0.0220 N/V
K_{pp}	Gain de poussée de couple agissant sur l'axe de tangage à partir de l'hélice de tangage	0.0011 N.m/V
K_{py}	Gain de poussée de couple agissant sur l'axe de tangage de l'hélice à lacet	0.0021 N.m/V
K_{yp}	Gain de poussée de couple agissant sur l'axe de lacet à partir de l'hélice de tangage	-0.0027 N.m/V
K_{yy}	Gain de poussée de couple agissant sur l'axe de lacet à partir de l'hélice de lacet	0.0022 N.m/V
l_c	Distance du centre de masse au corps fixe origine du cadre	0.002 m
m	masse	1.075 Kg
g	Accélération de la gravitation	9.8065 m/s^2

Tableau Paramétrées de l'Aero Quanser [42]

2- Les signaux des perturbations utilisés dans la simulation en mode de régulation



(a)



(b)

Figure Les signaux des perturbations injectés en (a) tangage (b) lacet

3- Les schémas Simulink de la simulation

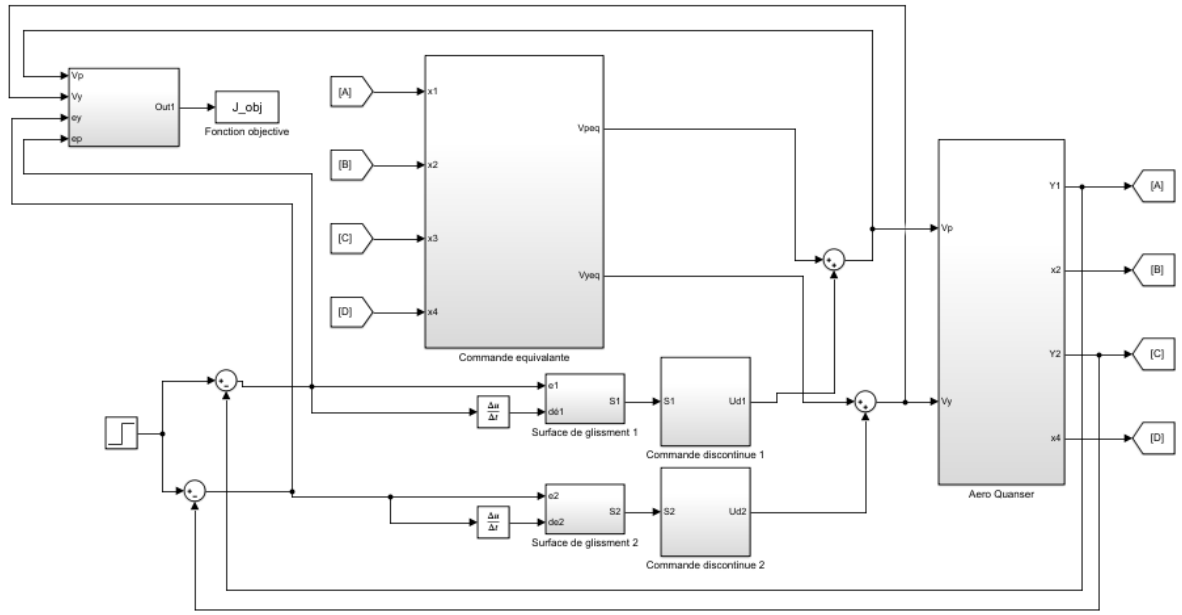


Figure Schéma Simulink de l'optimisation des commandes robuste

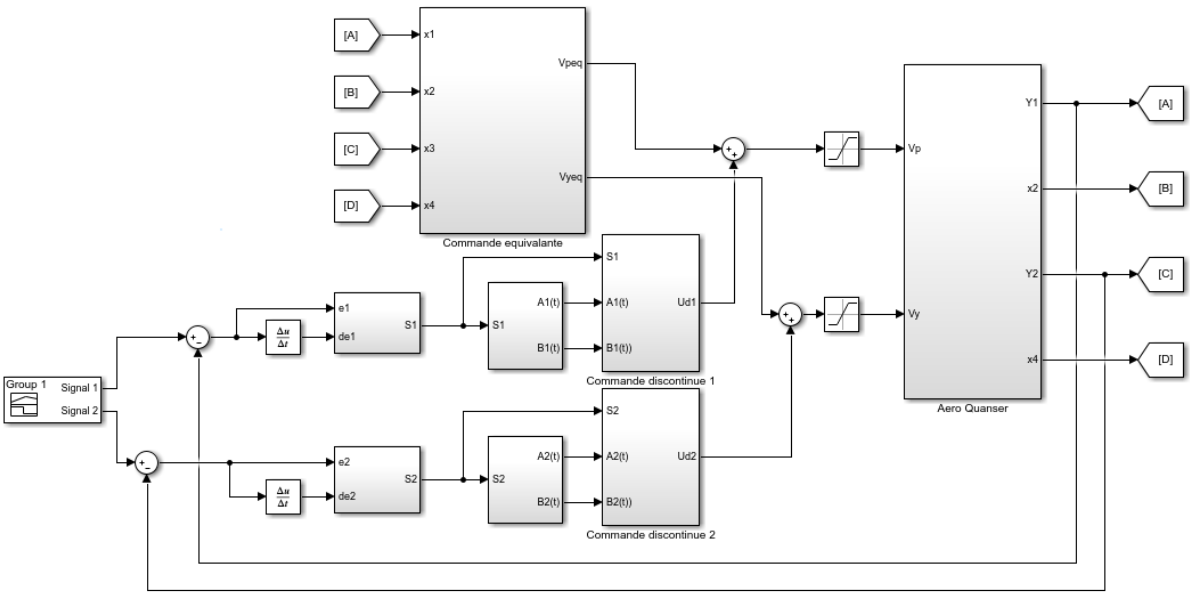


Figure Schéma Simulink de la commande ANTSTWO

4- Evolution des paramètres adaptatifs en fonction de temps (après 6000 s)

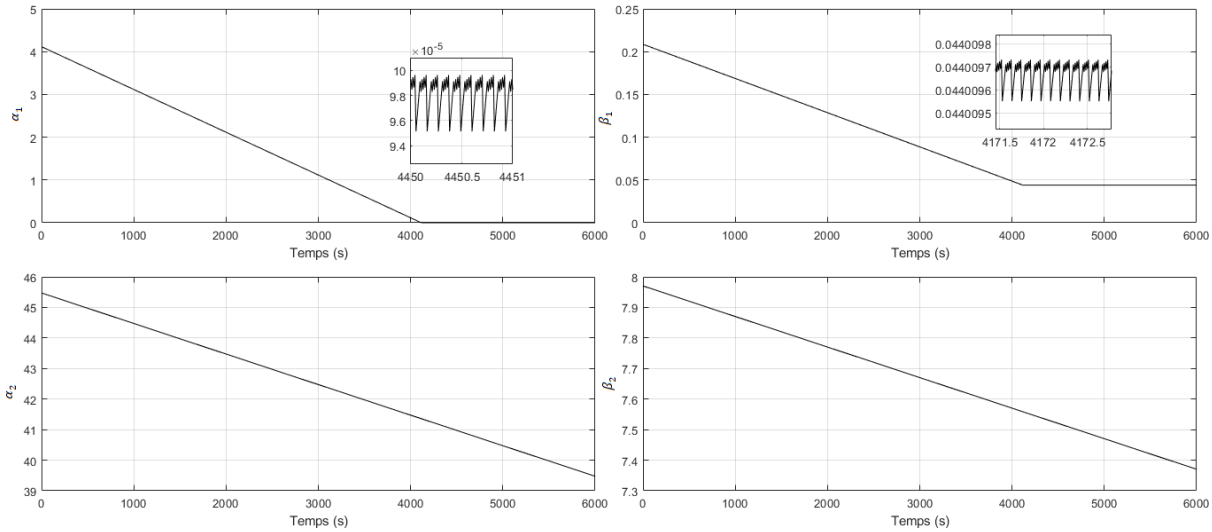


Figure L'évolution des paramètres adaptatifs en fonction de temps (après 6000 s)

ملخص

يركز مشروع نهاية الدراسة هذا على تطبيق عنصر تحكم قوي، قابل للتكيف ومثالي يسمى "التحكم التكيفي التوافقي الفائق الطرفي غير العيني الأمثل" على جهاز محاكاة طائرات الهليكوبتر أيرو كونسر. أولاً، قدمنا نظام أجهزة الطيران وتشغيلها بالإضافة إلى وصف وخصائص أيرو كونسر ونموذجها الديناميكي. بعد ذلك، قمنا بتجميع أنواع مختلفة من التحكم وقمنا بتحسين معلماتها باستخدام خوارزمية التحسين المستوحاة من المخلوقات الحيوية، ثم درسنا النتائج التي تم الحصول عليها من أجل المقارنة بين خوارزميات التحسين وفقًا لأدائها وكذلك استجابات النظام المتعلقة بالأوامر لتحديد ثم أضاف الأمر الأكثر قوة والأمثل تكيفًا للمكاسب باستخدام نظرية ليابونوف .وأخيرًا، قم بإنشاء نموذج ثلاثي الأبعاد لـ أيرو كونسر وتصوره من خلال تطبيق أمرنا وسلوكه في بيئة افتراضية واقعية تعتمد على .ROS-GAZEBO يسمح عنصر التحكم ANTSTWOالمتقن بالتحكم التكيفي والقوي في النظام، وقد تم تحسين كل شيء من خلال طرق الكشف عن مجريات الأمور الفوقية للحصول على أداء أفضل وتحكم مثالي. هذا النهج سيجعل من الممكن حل مشكلة اختيار معلمات التحكم، وإعداد تحكم تكيفي وقوي ومثالي وتسهيل تفسير النتائج في محاكاة ثلاثي الأبعاد للروبوت.

كلمات مفتاحية: Aero-Quanser، التحسين، الاستدلال الفوقي، التحكم القوي، التحكم التكيفي، ليابونوف، ROS-Gazeb، محاكاة ثلاثية الأبعاد.

Résumé

Ce projet de fin d'étude porte sur l'application d'une commande robuste, adaptative et optimale appelé « commande adaptive nonsingular terminal supertwisting optimale » sur le simulateur d'hélicoptère Aero Quanser. Dans un premier temps, nous avons présenté les systèmes avioniques et leurs fonctionnements ainsi qu'une description sur les caractéristiques de l'Aero Quanser et son modèle dynamique. Puis, nous avons synthétisé plusieurs lois de commande et optimisé leurs paramètres à l'aide d'algorithmes d'optimisation méta-heuristiques. Quatre algorithmes ont été étudiés selon la minimisation d'une fonction objective imposée et leurs rapidités de convergence aux différents minimums. Par la suite, les meilleures commandes parmi le PID, SMC, NTSMC et NTSTW obtenues ont été comparées dont nous avons constaté que celle appelée NTSTW basée sur le GWO est la meilleur en performances. Une nouvelle commande nommée ANTSTWO a été proposée dans ce travail, il s'agit d'une approche à la fois optimale et adaptative, où l'adaptation est étudiée par l'usage de la théorie de Lyapunov. Et enfin, nous avons créé un modèle 3D d'Aero Quanser et nous l'avons visualisé en l'application de notre commande sur un environnement virtuel réaliste basé sur ROS-GAZEBO.

Mots clé : Aero-Quanser, optimisation, Méta-heuristiques, commande robuste, commande adaptative, Lyapunov, ROS-Gazebo, Simulation 3D.

Abstract

This end-of-study project focuses on the application of a robust, adaptive and optimal control called “optimal adaptive nonsingular terminal supertwisting control” on the Aero Quanser helicopter simulator. First, we presented the avionics systems and their operation as well as a description of the characteristics of the Aero Quanser and its dynamic model. Then, we synthesized several control laws and optimized their parameters using metaheuristic optimization algorithms. Four algorithms have been studied according to the minimization of an imposed objective function and their speed of convergence at different minima. Subsequently, the best commands among the PID, SMC, NTSMC and NTSTW obtained were compared of which we found that the one called NTSTW based on the GWO is the best in performance. A new command called ANTSTWO has been proposed in this work, it is an approach that is both optimal and adaptive, where adaptation is studied by the use of Lyapunov's theory. And finally, create the 3D model of the Aero Quanser and visualize it with the application of our command its behavior on a realistic virtual environment based on ROS-GAZEBO.

Key words: Aero-Quanser, optimization, Meta-heuristics, robust control, adaptive control, Lyapunov, ROS-Gazebo, 3D Simulation.