This version of the article has been accepted for publication, after peer review, and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/s40430-024-04714-3

Adaptive-Optimal MIMO Nonsingular Terminal Sliding Mode Control of Twin-Rotor Helicopter System; Meta-heuristics and Super-twisting based Control Approach.

Amar Rezoug[1*], Ayoub Messah[1], Walid Ahmed Messaoud[1], Khelifa Baizid[2], Jamshed Iqbal[3]

[1] Laboratoire de Technologie Innovante, Ecole Nationale Supérieure des Technologies Avancées, Diplomatic City, Dergana-Bordj El Kiffan, Algiers, Algeria.
{amar.rezoug,a_messah,w_ahmedmessaoud}@enst.dz
[2]Department of Advanced Robotics, Istituto Italiano di Technologia, Genoa, Italy.
baizid.khelifa@gmail.com
[3]School of Computer Science, Faculty of Science and Engineering, University of Hull, United Kingdom. j.iqbal@hull.ac.uk

Abstract This research proposes a novel hybrid control technique based on Nonsingular Terminal Sliding Mode (NTSM) Control, meta-heuristic optimization algorithms and adaptive super-twisting based on Lyapunov stability analysis for controlling Quanser aero simulator helicopter. Firstly, the NTSM and NTSM Super Twisting (NTSTW) controllers are designed followed by optimization of the designed controllers using Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Salp Swarm Algorithm (SSA) and Ant Lion Optimizer (ALO). Qualitative and quantitative comparisons among the results obtained from optimized controllers are then thoroughly investigated. Simulation results confirmed that the Optimized NTSTW (O-NTSTW) based on GWO demonstrated overperformance compared with other controllers. In order to further improve its performance while ensuring robustness, this controller is hybridized with an adaptive Lyapunov super-twisting algorithm resulting in a novel controller named as Adaptive O-NTSTW (AO-NTSTW). Visualization using ROS-Gazebo is done while validating AO-NTSTW and O-NTSTW. Simulation results demonstrated the effectiveness and the superiority of AO-NTSTW compared to O-NTSTW.

List of abbreviations

Abbreviation	*Description*
ACO	Ant Colony Bee
AO-NTSTW	Adaptive Optimized Nonsingular Terminal Sliding Mode Super Twisting
BA	Bat Algorithm
CS	Cuckoo Research
DC	Direct Current
DOF	Degree Of Freedom
GWO	Grey Wolf Optimizer
LQR	Linear Quadratic Controller
MIMO	Multi-Inputs Multi-outputs
NTSM	Nonsingular Terminal Sliding Mode
NTSTW	Nonsingular Terminal Sliding Mode Super Twisting
O-NTSTW	Optimized Nonsingular Terminal Sliding Mode Super Twisting
PSO	Particle Swarm Optimization
SMC	Sliding Mode Control
SSA	Salp Swarm Algorithm
UAVs	Unmanned Aerial Vehicles
WOA	Whale Optimization Algorithm

1. Introduction

Today, Unmanned Aerial Vehicles (UAVs) have widespread applications both in civilian and military domains such as, surveillance, monitoring, firefighting, cinematography, search and rescue etc. [1-2]. In order to facilitate their use, various configurations of these systems have been considered such as birotor, quadrotor, hexarotor, fixed wing, and so on. Among the most popular configuration, the traditional Sikorsky helicopter has occupied privilege position. This type exhibits a prominent profile due to some interesting features such as, having a miniature tail and consuming moderate power consumption due to the two rotors structure. UAV simulators are laboratory devices allowing the safe validation of control laws under investigation. Indeed, the use of these systems makes it possible to closely mimic the real UAV dynamics and at the same time minimize the cost of experiments by avoiding damage to real systems and reducing the risks associated with flight tests [3]. Several helicopter simulators can be found in the literature such as 2-DOF helicopter TRMS [4] or Aero Quanser [5-6], 3-DOF hover [7], 3-DOF helicopter [8], etc.

Quanser aero is one of the newest helicopter simulators [9-10]. This device is a Multi-Inputs Multi-Outputs (MIMO) system with two inputs and two outputs, characterized by several nonlinearities such as, cross coupling, unstable dynamics in open loop, unmolded dynamics, etc. Dynamic modelling of the system is a highly challenging task due to the presence of high interactions among the various process variables and the non-accessibility of certain states. These properties allow researchers to validate the designed control approaches [11]. Technically, this system can move freely in horizontal as well as in vertical plane. Quanser aero has been used in several research works involving controllers such as testing based on trajectory tracking [9], using surface stabilization approach [12] for improving performance, realizing intelligent active force control laws [10] and so on. Control of a 2-DOF helicopter simulator is generally the subject of several approaches like, MIMO-PID (Proportional Integral Derivative), neural network, integral back stepping, higher order Sliding Mode Control (SMC) etc. In [6], a MIMO PID based control law has been proposed for a Quanser aero system i to simulate and experimentally demonstrate a Linear Quadratic (LQ) controller. Since the controller is linear, it can only tackle a small region of the system dynamics. In [13], an adaptive neural network controller is proposed to deal with backlash-like hysteresis and output constraints. The neural network was used to estimate nonlinear system dynamics. A discrete higher order SMC combined with LQR controller is proposed in [14]. A multi-objective optimization was adopted in this work. The robustness in controller due to SMC deals with the uncertainties while the Linear Quadratic Regulator (LQR) was used to guarantee the control performance. Back stepping with augmented dynamic model is proposed for a Two Rotor Aero-dynamical System (TRAS) in [15]. The actuators dynamic was considered in this work. The main problem of the aforementioned controllers is their design complexity.

Classical SMC cannot ensure finite time convergence to equilibrium states point [16]. This control approach can only guarantee asymptotic stability [17]. Terminal SMC (TSMC) is based on nonlinear sliding surface and may ensure finite time convergence. However, this approach suffers from a singularity drawback. In order to deal with this limitation, Nonsingular Terminal SMC (NTSMC) is proposed in [18]. Recently, Nonsingular TSMC (NTSMC) has drawn a significant attention in the control community [19]. NTSMC design is based on the same principle as that of the classical SMC i.e., forcing the dynamics of a given nonlinear system to follow a particular nonlinear surface in state space. Once the sliding surface is reached, the control maintains the system dynamics close to the sliding surface. Therefore, the control design is realized in two steps [20]. The first step involves choosing a sliding surface such that the desired specifications defined by the sliding surface parameters are satisfied. The second step involves the design of a control law to achieve the required performance while simultaneously guaranteeing the closed-loop stability. For the studied class of systems, SMC and its variants have been considered in some works. SMC has been considered in [5] to control a 2-DOF

(Degree Of Freedom) Quanser helicopter aero system. In [21], TSMC has been proposed, the controller is applied after linearization of the system dynamics. An adaptive Radial Basis Function Neural Network (RBFNN) global Fast-NTSMC for a twin rotor MIMO system is proposed in [22]. The objective was to mitigate the wind effects. Experimental results were reported though the work does not involve optimization.

The bio-inspired algorithms have become very promising and attractive in the engineering and allied domains. These algorithms are based on simple behaviors inherited from nature. In addition, they are easy to implement and permit finding an optimum solution for any objective or multi-objective functions. These algorithms mimic biological life or physical phenomena to solve optimization problems. Several methods have been proposed, among the most used optimization methods, we can find, Genetic Algorithms (GA) [23], Ant Colony Optimization (ACO) [24], Particle Swarm Optimization (PSO) [25], Ant Colony Bee (ACO) [26,27], Bat Algorithm (BA) and Cuckoo Research (CS) [28], Grey Wolf Optimizer (GWO)[29, 30], Ant Lion optimizer (ALO) [31], Whale Optimization Algorithm (WOA) [32], Salp Swarm Algorithm (SSA) [33] , JAYA algorithm [34] and Grasshopper optimization [35], etc.

The parameter determination of many control methods is based on the trial and error approach and thus finding the best parameter combination is difficult. In this fact, the application of bio-inspired algorithms for optimization of robust control parameters is a recent research domain. These algorithms offer a useful tool because of their ability to optimize nonlinear problems, which are difficult to deal using analytical methods. The optimization of NTSMC is a recent approach that allows to; (i) Find optimized controller parameters automatically through a stochastic research procedure [23,26,36], (ii) Resolve the optimal control problem of nonlinear systems by defining one or several fitness function(s) without involving complicated mathematics [28,37] and (iii) Provide the possibility to compare between several vectors of optimized parameters obtained from the same algorithm or from different optimization algorithms and/or one or more fitness functions.

Many meta-heuristic algorithms can address this problem; in the present research, the GWO, WOA, SAA and ALO are selected because of their global optimization ability, ease in implementation, low information requirements (low parameters to run them) and probabilistic nature. All these algorithms are proposed by Mirjalili et *al.* in [29,31,32,33]. WOA and GWO are inspired from the search and the hunt behaviors of the whales and wolves respectively. SAA is based on leader-followers principle of salps in the sea. ALO mimics the ingenious hunting procedure of ant lion insect. These algorithms have been used to find solutions of many real-world optimization problems like the optimization of a piezoactuated micropuncture mechanism [9], quadrotor systems [28], robotic systems [23,30,36], etc.

Quanser aero system is a MIMO inherently unstable system with highly coupled dynamics that leads to having cross-coupling terms in the designed controllers. The controller parts are skillfully related to all the dynamic terms of the system. This can cause intersection problem related to the choice of the control parameters. In addition, the motors actuating the system are subject to physical limitations i.e. these motors cannot surpass input voltages of [-24,24] volts, which imposes constraints in control parameters selection particularly through the objective function. The optimization procedure may be interrupted after some time due to premature stagnation of the algorithm. This stagnation is caused by random initialization of the control parameters, or by the situations involving inappropriate parameters in the research process. The aerodynamic system is subject to several internal and external disturbances such as, wind disturbances and parametric variations. These issues are considered in the control design in the present study.

To the best of authors' knowledge, the hybrid, optimized and adaptive Lyapunov controllers for the Sikorsky helicopter simulator have not been studied. To obtain the proposed controller, three steps are needed: (i) analytical design the MIMO controller (ii) finding the parameters through an optimization procedure (iii) finding the adaptive control form using the adaptive Lyapunov theorem. In this work, these algorithms are used to optimize Nonlinear Terminal Super Twisting (NTSTW) control approach to solve trajectory tracking problem of an aero Quanser simulator. In addition, an adaptive optimized controller is designed. The key contributions of the present research are listed below:

I. Proposing an optimized MIMO NTSTW (O-NTSTW) control based on GWO, WOA, ALO and SSA for trajectory tracking of Quanser Aero simulator.
II. Qualitative and quantitative comparison of GWO, WOA, ALO and SSA algorithms in case of NTSTW parameters optimization.
III. Formulation of new adaptive-optimized controller called AO-NTSTW for the Quanser aero simulator with stability guaranteed based on Lyapunov theory.
IV. Visualization using ROS-Gazebo is presented where AO-NTSTW and O-NTSW are also implemented.

The control law is obtained in two phases; first, optimization based on the aforementioned algorithms and then adaptation of the gains using the Lyapunov theory has been carried out.

The remaining of the paper is outlined as follows; Section 2 presents background to the research study Section 3 details GWO, WOA, ALO and SSA algorithms. Controller design is presented in Section 4 while simulation results are discussed in Section 5. Finally, Section 6 concludes the paper.

2. Background

In this section, firstly, a literature review of the recent algorithms and their variants applied for optimizing NTSMC is presented. This is followed by developing the dynamic model of the Quanser aero system.

2.1. *Literature review*

Table 1 lists a collection of recent and prominent works about NTSMC based on some bio-inspired optimization algorithms such as, GWO, WOA, ALO and SSA and their variants.

Table 1. State-of-the-art of the recent NTSMC based meta-heuristics works

Reporting year	Used meta-heuristic algorithm	Plant	Description	Reference
2022	JAYA algorithm	Planar cable-driven parallel robots	Jaya optimizer is utilized to select membership functions of the fuzzy SMC.	[34]
2022	Grasshopper optimization	Planar cable-driven parallel robots	Determination of an optimal fuzzy membership functions in order to have hybrid fuzzy SMC.	[35]
2022	GA	Medical parallel robotic system	Optimization of the TSMC parameters.	[23]
2022	JAYA algorithm & ABC	Tank system	Global optimization of second-order sliding mode controller parameters using a new sliding surface.	[26]
2021	ABC optimization algorithm	Micro-grid system	Optimization of terminal SMC.	[27]
2021	WOA	Lower limb rehabilitation robot	Optimization of Integral SMC.	[36]
2021	WOA	Unstable processes	Optimal values of unknown parameters of Variable Structure Control (VSC) are obtained using WOA.	[37]
2021	WOA	Delta wing aircrafts	Fast terminal super twisting SMC.	[28]
2021	Grey wolf and Weighted whale algorithm	Nonlinear dynamic system (validation to robot manipulator)	Fractional-order sliding mode backstepping controller and the fuzzy logic system parameters optimized via a grey wolf and weighted WOA.	[39]
2021	PSO	Ego vehicle	Optimization of Fast terminal SMC parameters.	[40]
2021	GWO	Quadrotor UAV system	TSMC sliding surface parameters and certain control parameters have been tuned by using GWO.	[41]

2020	Improved PSO	Train traction braking system	A NFTSM control method based on the improved multi-strategy particle swarm optimization (IMPSO) algorithm and the Radial Basis Function Neural Network (RBFNN) is proposed for the key nonlinear network control system.	[42]

2.2. *System description and modeling*

Quanser aero system (Fig. 1) [43] used in this study consists of two rotors; a main rotor and a tail rotor. The rotors reside at the ends of the beam and are driven by a DC motor [6] with the voltage in ± 24 V range. The system can perform two angular movements such as pitch and yaw. The physical parameters of the system are summarized in Table 2.

Table 2. Quanser AERO parameters

Description	Symbol	Value	Unit
Mass	m	1.075	kg
Pitch directional viscous damping	D_p	-7.59	N/V
Yaw directional viscous damping	D_y	15.8	N/V
Pitch inertia	J_p	2.15×10^{-2}	kgm^2
Yaw inertia	J_y	2.37×10^{-2}	kgm^2
Drag/air resistance coefficient	k_d	1×10^{-5}	Nm
Acceleration due to gravity	g	9.81	ms^{-2}
Distance between center of mass and origin of B (see fig.2)	l_c	0.002	mm

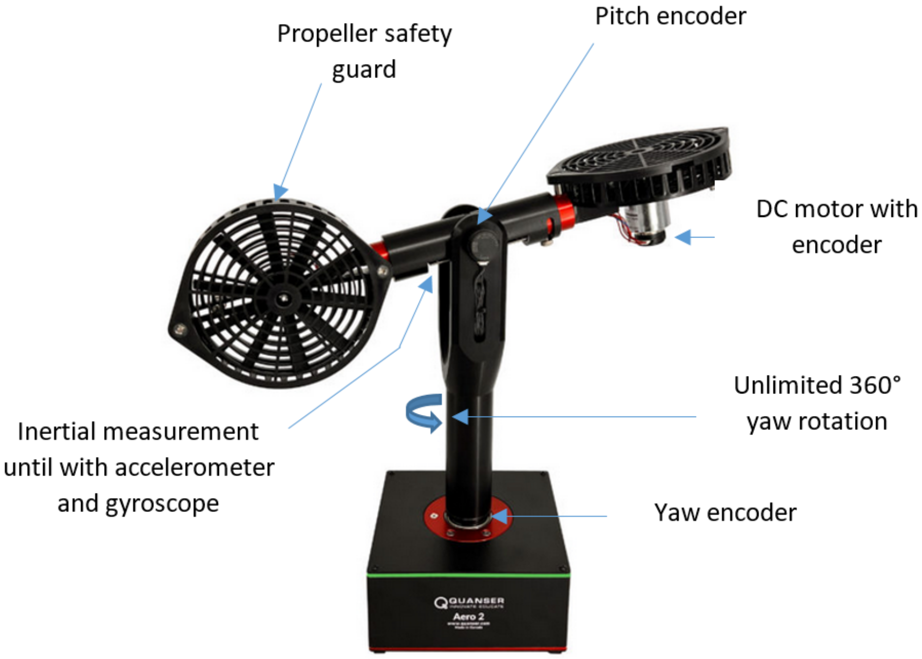


Figure 1. Quanser Aero apparatus [43]

The forces expressed in the fixed frame of the system are shown in Fig 2. The voltage V_p is applied to the pitch motor, and its rotational speed is transformed into a force F_p that acts normal to the body at a distance r_p from the pitch axis. Similarly, the yaw motor causes a force F_y which acts on the body at the distance r_y from the yaw axis as well as a torque around the pitch axis. The propeller rotation generates torque around the pitch rotor motor that governs motion around the yaw axis. Thus, the rotation of the pitch propeller causes not only a movement around the pitch axis but also around the yaw axis.

In order to develop the Quanser aero nonlinear model, the following assumptions are taken into account : (i) Main and tail rotors are of the same dimensions and are equidistant from the center of rotation, (ii) Pitch propeller is parallel to the ground when the pitch angle is zero, (iii) Pitch angle increases positively when the front rotor is moved up, the body rotates counterclockwise around the y-axis, and the front rotor voltage is positive, and (iv) Yaw angle increases positively when the body rotates counterclockwise around the z-axis. In this case, the tail rotor voltage is positive.

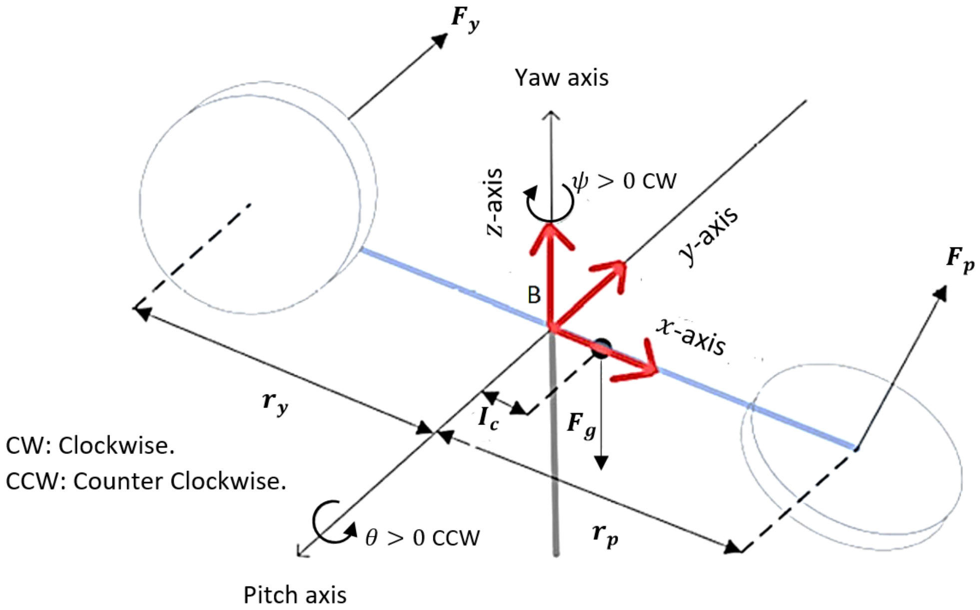


Figure 2. Schematic diagram showing the forces applied to the Quanser Aero

The fixed body center of mass is represented in Cartesian coordinates by

$$\begin{cases} x_c = l_c \cos\psi \cos\theta \\ y_c = l_c \sin\psi \cos\theta \\ z_c = l_c \sin\theta \end{cases} \tag{1}$$

So, the total potential energy (E_p) of the system due to the gravitational effect is,

$$E_p = mgl_c \sin\theta \tag{2}$$

The total kinetic energy (E_c) of the system is the sum of rotational kinetic energies on the pitch (E_{c_1}) and yaw axes (E_{c_2}) and the kinetic energy generated by the translational movement of the center of mass (E_{c_3}). Thus,

$$E_c = E_{c_1} + E_{c_2} + E_{c_3} \tag{3}$$

where

$$\begin{cases} E_{c_1} = \frac{1}{2} J_p \dot{\theta}^2 \\ E_{c_2} = \frac{1}{2} J_y \dot{\psi}^2 \\ E_{c_3} = \frac{1}{2} m{l_c}^2 \begin{bmatrix} (-\sin(\psi)\cos(\theta)\,\dot{\psi} - \cos(\psi)\sin(\theta)\,\dot{\theta})^2 \\ +(\sin(\psi)\sin(\theta)\,\dot{\theta} - \cos(\psi)\cos(\theta)\dot{\psi}) + (\cos(\theta)\,\dot{\theta})^2 \end{bmatrix} \end{cases}$$

Using these values of kinematic energies, (3) can be rewritten as,

$$E_c = \frac{1}{2}(J_p\dot{\theta}^2 + J_y\dot{\psi}^2 + m{l_c}^2\dot{\psi}^2 cos^2\theta + m{l_c}^2\dot{\theta}^2) \tag{4}$$

Lagrange's equation is used to find the equations of motion for the pitch and yaw propellers as,

$$\left.\begin{aligned}\frac{\partial}{\partial t}\left(\frac{dL}{d\dot{\theta}}\right)-\frac{dL}{d\theta}=Q_1\\ \frac{\partial}{\partial t}\left(\frac{dL}{d\dot{\psi}}\right)-\frac{dL}{d\psi}=Q_2\end{aligned}\right\} \tag{5}$$

where Q_1 and Q_2 are the forces acting on the pitch axis and the yaw axis respectively. L is the Lagrangian operator.

$$\left.\begin{aligned}Q_1=\tau_p-D_p\dot{\theta}\\ Q_2=\tau_y-D_y\dot{\psi}\end{aligned}\right\} \tag{6}$$

where

$$\left.\begin{aligned}\tau_p=K_{pp}V_p+K_{py}V_y\\ \tau_y=K_{yp}V_p+K_{yy}V_y\end{aligned}\right\} \tag{7}$$

where τ_p and τ_y are the torques generated by the pitch and yaw motors respectively. K_{pp}, K_{py}, K_{yp} and K_{yy} are respectively torque thrust gains from the pitch rotor, cross-torque thrust gain acting on the pitch from the yaw rotor, cross-torque thrust gain acting on the yaw from the pitch rotor and torque thrust gain from the yaw rotor. Based on the Lagrangian formulation (5), the system dynamics can be derived as given in (8).

$$\begin{cases}(J_p+m{l_c}^2)\ddot{\theta}=\left(K_{pp}V_p+K_{py}V_y-D_p\dot{\theta}-mgl_c cos(\theta)-m{l_c}^2\dot{\psi}^2 sin(\theta)cos(\theta)\right)\\ \left(J_y+m{l_c}^2cos^2(\theta)\right)\ddot{\psi}=\left(K_{yp}V_p+K_{yy}V_y-D_y\dot{\psi}+2m{l_c}^2\dot{\theta}\dot{\psi}sin(\theta)cos(\theta)\right)\end{cases} \tag{8}$$

Choosing $[\theta \quad \dot{\theta} \quad \psi \quad \dot{\psi}]=[x_1 \quad x_2 \quad x_3 \quad x_4]$ as state space variables, (8) can be rewritten as (9).

$$\begin{cases}\dot{x}_1=x_2\\ \dot{x}_2=f_1+g_{11}V_p+g_{12}V_y\\ \dot{x}_3=x_4\\ \dot{x}_4=f_2+g_{21}V_p+g_{22}V_y\end{cases} \quad and \quad \begin{cases}y_1=x_1\\ y_2=x_3\end{cases} \tag{9}$$

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

$$f_1=\frac{-D_px_2-mgl_c cos(x_2)-ml^2{x_4}^2 sin(x_1)cos(x_1)}{(J_p+m{l_c}^2)} \ ; f_2=\frac{-D_yx_4+2m{l_c}^2x_2x_4 sin(x_1)cos(x_1)}{\left(J_y+m{l_c}^2cos^2(\theta)\right)}$$

3. GWO, WOA, ALO and SSA algorithms

In this section, the mathematical formulation of GWO, WOA, ALO and SSA algorithms are given in detail. Explanation of the optimization mechanism for each algorithm is presented.

3.1. *Grey Wolf Optimizer (GWO)*

GWO has been proposed in 2014 by Mirjalili *et al.* [29]. GWO simulates the process of gray wolves hunting. The wolves are classified into alpha (α), beta (β), delta (δ), and omega (ω). Wolves go through a search or exploration stage followed by an exploitation or hunting stage. Their hunting strategy can be summarized in three stages: pursue, encircle, and attack. After following the prey, they circle him/her until the prey is constrained to move and finally, they attack the prey.

The optimal solution is represented by the prey, and each wolf corresponds to a potential solution. As the wolves shift their positions, they aim to move closer to the optimal solution for convergence . In order to mimic the encircling behavior of wolves, (10) and (11) are proposed.

$$\vec{D} = \left|\vec{C} \times \overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{10}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \times \vec{D} \tag{11}$$

where $\vec{X}$ is the position of the gray wolf, $\overrightarrow{X_p}$ is the position of the prey, t is the current iteration and *t+1* is the next iteration, $\vec{D}$ is the distance between $\vec{X}$ and $\overrightarrow{Xp}$, $\vec{A}$ and $\vec{C}$ are the coefficients given in (12) and (13) respectively.

$$\vec{A} = 2 \times a \times \overrightarrow{r_1} - a \tag{12}$$

where $\vec{a} = 2(1 - \frac{actual\_iteration}{\max\_iteration})$

$$\vec{C} = 2 \times \overrightarrow{r_2} \tag{13}$$

where r_1 and r_2 are random values in the range [0, 1]. Since there is no a-priori knowledge about the optimal point (prey) in the solution space, the hunting behavior is estimated with the three best solutions found among the wolves. Thus, the proposed hunting behavior is given in (14).

$$\begin{gathered}
\overrightarrow{D_\alpha} = \left|\vec{C} \times \overrightarrow{X_\alpha}(t) - \vec{X}(t)\right| \qquad \overrightarrow{D_\beta} = \left|\vec{C} \times \overrightarrow{X_\beta}(t) - \vec{X}(t)\right| \\
\overrightarrow{D_\delta} = \left|\vec{C} \times \overrightarrow{X_\delta}(t) - \vec{X}(t)\right| \qquad \overrightarrow{X_1}(t+1) = \overrightarrow{X_\alpha} - \overrightarrow{A_1} \times \overrightarrow{D_\alpha} \\
\overrightarrow{X_2}(t+1) = \overrightarrow{X_\beta} - \overrightarrow{A_2} \times \overrightarrow{D_\beta} \qquad \overrightarrow{X_3}(t+1) = \overrightarrow{X_\gamma} - \overrightarrow{A_3} \times \overrightarrow{D_\gamma} \\
\vec{X}(t+1) = \frac{\overrightarrow{X_1}(t+1) + \overrightarrow{X_2}(t+1) + \overrightarrow{X_3}\,(t+1)}{3}
\end{gathered} \tag{14}$$

Given that the position of the prey is unknown, the initial position is randomly determined. For this reason, $\overrightarrow{Xp}(t)$ vector is multiplied with random value $\vec{C}$ indicated by (13). $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$ and $\overrightarrow{D_\gamma}$ respectively represent the distances between $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, and $\overrightarrow{X_\gamma}$ with reference to rest of the prey (the ω wolves)denoted as $\overrightarrow{X(t)}$. It is also important to indicate that the designation of $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\gamma}$ are obtained from different values of the objective function, when $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\gamma}$ correspond to the best three values of the objective function.

The algorithm divides the search process into two main parts: exploration and exploitation:

- *Exploration* is the search for prey. In this phase, search process is based on the position of alpha (α), beta (β) and delta (δ) wolves. The last kind of wolves diverge to search and converge to attack the prey. In GWO, the magnitude of $\vec{A}$ is used to model this phenomenon. If $|A|$>1, the wolf diverges from the prey (search for another prey) to find a solution. In order to avoid stagnation at a local optimum, another random parameter $\vec{C}$ is introduced.

- *Exploitation* consists of prey attacking. This task is modeled by decreasing the value of a in (12), which restricts the variation interval of $\vec{A}$ in $[-a, a]$. When the random values of $|A|$<1, the wolf is forced to attack towards the prey.

3.2. ***Whale optimization algorithm (WOA)***

Whales are mammals that exhibit emotional behavior and communicate with one another. Whales hunting process is quite interesting; they hunt their prey near the water surface creating bubbles. The prey of whales is small herds of fish. WOA mimics the whale hunting behavior as highlighted by Mirjalili and Andrew in [32]. Although the formulation of WOA is quite similar to that of GWO, the use of a spiral to simulate the attacking step is the main specialty of WOA. Their hunting plan consists of three steps; encircling prey, spiral bubble net feeding maneuvering, and searching for prey. They recognize their prey, create spiral bubbles and surround them like the wolves do. In WOA, the global optimum is not known a-priori, the prey is indicated by the best accepted optimal solution as in GWO. The population positions are updated to the best optimal solution as given in (15) and (16).

$$\vec{D} = \left|\vec{C} \times \overrightarrow{X_*}(t) - \vec{X}(t)\right| \tag{15}$$

$$\vec{X}(t+1) = \overrightarrow{X_*}(t) - \vec{A} \times \vec{D} \tag{16}$$

where $\vec{D}$ is absolute distance between the prey and the whale. t and $t+1$ are the current and the next iterations respectively. $\vec{X}$ is the position vector of the whales, and $\overrightarrow{X_*}$ is the position of the best solution reached until a particular iteration and updated at each iteration. $\vec{A}$ and $\vec{C}$ are coefficients vectors calculated by (12) and (13) respectively.

The whales' bubble strategy is also simulated in the algorithm in the present research as given in (17) and (18). In this algorithm, both strategies are used with 50% chance, the whales choose between encircling or spiraling to update their positions as given in (19).

$$\overrightarrow{D'} = \left|\overrightarrow{X_*(t)} - \overrightarrow{X(t)}\right| \tag{17}$$

$$X(t+1) = D' e^{bl} \cos(2\pi l) + X_*(\mathrm{t}) \tag{18}$$

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X_*}(t) - \vec{A}.\vec{D} & if\ p < 0.5 \\ \overrightarrow{D'} e^{bl} \cos(2\pi l) + \overrightarrow{X_*}(\mathrm{t}) & if\ p > 0.5 \end{cases} \tag{19}$$

where p is a random number in the range [0,1]. l is a random number in [−1,1], D' indicates the distance of i^{th} whale by rapport of the actual prey position. b is a real-numbered constant.

The shrinking of the encirclement and the continuous influx of the spiraling bubbles are both intended for exploitation. Some random moves are also accepted to have an exploration behavior as given in (20) and (21).

$$\vec{D} = \left|\vec{C} \times \overrightarrow{X_{rand}}(t) - \vec{X}(t)\right| \tag{20}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}}(t) - \vec{A}.\vec{D} \tag{21}$$

where $\overrightarrow{X_{rand}}$ is a random position vector (a random whale) which is chosen from the current population.

3.3. *Salp Swarm Algorithm (SSA)*

Salps usually create chains and move cooperatively. SSA is one of the recent nature-inspired optimization algorithms developed by Mirjalili *et al.* [33]. SSA mimics the foraging behaviors of salps. In SSA, salps are classified into two groups: leader and followers. SSA has only one parameter to adapt and is easy to implement. The followers follow the position of the salp leader. The position of the leading salp is updated using (22).

$$x_j^1 = \begin{cases} F_j + c_1\{(ub_j - lb_j)c_2 + lb_j\}; & if\ c_3 \geq 0 \\ F_j - c_1\{(ub_j - lb_j)c_2 + lb_j\}; & if\ c_3 < 0 \end{cases} \tag{22}$$

where x_j^1 is the position of the leading salp, F_j is the position of the food source, both of these positions are in the j^{th} dimension. In addition, ub_j and lb_j represent respectively the upper bound and the lower bound of the j^{th} dimension respectively. c_1 is calculated using (23). c_2 and c_3 are random numbers. In addition, the followers salps update their position using (24).

$$c_1 = 2 * e^{-(\frac{4t}{T})^2} \tag{23}$$

$$x_j^i = \frac{1}{2}at^2 + v_0 t \tag{24}$$

where t is the current iteration, T is the maximum of iteration. v_0 is the initial velocity, $a = \frac{\delta v}{\delta t} = v_{final} - v_0$ is the acceleration (with $\delta t = 1$) and $v_{final} = x - x_0$. Replacing these terms in (24), the positions of the follower salps can be calculated by (25).

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{25}$$

where x_j^i is the position of i^{th} follower salp in j^{th} dimension and $i \geq 2$.

3.4. *Ant Lion Optimizer (ALO)*

Ant Lion Optimizer (ALO) is inspired by the hunting behavior of antlion insect. ALO is another nature-inspired optimization algorithm developed by Mirjalili especially for the solution of continuous optimization problems [31]. ALO mimics the interaction between antlions and ants. Thus, the artificial ants move in the search space and the antlions are allowed to hunt them. Since ants move stochastically in nature for searching food, a random move (walk) is chosen to model their movement. The ants' random movements (walks) are updated with (26).

$$X_i^t = \frac{\left(X_i^t - a_i\right) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \tag{26}$$

where X_i^t is the normalized position of the ant, a_i and d_i are respectively the minimum and the maximum of the random walk in the i^{th} variable and $c_i^t$and d_i^t are respectively the minimum and the maximum of the random walk of the i^{th} variable for iteration t.

$$c_i^t = f_j^t + c^t \tag{27}$$

$$d_i^t = f_j^t + d^t \tag{28}$$

where c^t and d^t are respectively the minimum of all variables and the vector representing the maximum of all variables, c_j^t is the minimum of all variables for j[th] ant and d_j^t is the maximum of all variables for j^{th} ant, f_j^t represents the position of the selected j^{th} antlion, all of these parameters are obtained at t^{th} iteration.

4. Control Design

In this section, we present NTSMC, O-NTSMC, and AO-NTSTW in detail. For optimization purposes, an objective fitness function based on the addition of ITAE (Integral of Time multiplied by Absolute Error) and ISCO (Integral of Squared Control) also will be described.

4.1. NSTSMC

The terminal sliding surfaces associated with each subsystem is given as,

$$\left.\begin{aligned} S_{x_1} &= |x_2|^{\gamma_1} sign(x_2) + \lambda_1 x_1 \\ S_{x_2} &= |x_4|^{\gamma_2} sign(x_4) + \lambda_2 x_3 \end{aligned}\right\} \tag{29}$$

where λ_1, γ_1, λ_2 and γ_2 are positive real values.

The time derivative of (29) gives,

$$\begin{bmatrix} \lambda_1 \dot{x}_1 + \gamma_1 |x_2|^{\gamma_1 - 1} \dot{x}_2 sign(x_2) \\ \lambda_2 \dot{x}_3 + \gamma_2 |x_4|^{\gamma_2 - 1} \dot{x}_4 sign(x_4) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{30}$$

Replacing (9) into (30) yields

$$\begin{bmatrix} \lambda_1 x_2 + \gamma_1 |x_2|^{\gamma_1 - 1} \left(f_1 + g_{11} V_{peq} + g_{12} V_{yeq}\right) sign(x_2) \\ \lambda_2 x_4 + \gamma_2 |x_4|^{\gamma_2 - 1} \left(f_2 + g_{21} V_{peq} + g_{22} V_{yeq}\right) sign(x_4) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{31}$$

This implies,

$$\begin{bmatrix} \lambda_1 x_2 + \gamma_1 |x_2|^{\gamma_1 - 1} f_1 sign(x_2) + \left(g_{11} V_{peq} + g_{12} V_{yeq}\right) \gamma_1 |x_2|^{\gamma_1 - 1} sign(x_2) \\ \lambda_2 x_4 + \gamma_2 |x_4|^{\gamma_2 - 1} f_2 sign(x_4) + \left(g_{21} V_{peq} + g_{22} V_{yeq}\right) \gamma_2 |x_4|^{\gamma_2 - 1} sign(x_4) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{32}$$

Some terms of (32) can be transferred to the second member as,

$$\begin{bmatrix} \left(g_{11} V_{peq} + g_{12} V_{yeq}\right) \gamma_1 |x_2|^{\gamma_1 - 1} sign(x_2) \\ \left(g_{21} V_{peq} + g_{22} V_{yeq}\right) \gamma_2 |x_4|^{\gamma_2 - 1} sign(x_4) \end{bmatrix} = - \begin{bmatrix} \lambda_1 x_2 + \gamma_1 |x_2|^{\gamma_1 - 1} f_1 sign(x_2) \\ \lambda_2 x_4 + \gamma_2 |x_4|^{\gamma_2 - 1} f_2 sign(x_4) \end{bmatrix}$$

Terms $\gamma_1|x_4|^{\gamma_1-1}sign(x_2)$ and $\gamma_2|x_4|^{\gamma_2-1}sign(x_4)$ can be transformed to the second part of the equation which yields,

$$\begin{bmatrix}(g_{11}V_{peq}+g_{12}V_{yeq})\\(g_{21}V_{peq}+g_{22}V_{yeq})\end{bmatrix}=-\begin{bmatrix}(f_1+\lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2))\\(f_2+\lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4))\end{bmatrix} \tag{33}$$

The first term of (33) can be written in matrix form as

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

This implies,

$$\begin{bmatrix}V_{peq}\\V_{yeq}\end{bmatrix}=-\begin{bmatrix}g_{11}&g_{12}\\g_{21}&g_{22}\end{bmatrix}^{-1}\begin{bmatrix}(f_1+\lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2))\\(f_2+\lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4))\end{bmatrix} \tag{34}$$

It is now easy to calculate the inverse of $\begin{bmatrix}g_{11}&g_{12}\\g_{21}&g_{22}\end{bmatrix}$ as,

$$\begin{bmatrix}g_{11}&g_{12}\\g_{21}&g_{22}\end{bmatrix}^{-1}=\frac{1}{g_{12}g_{21}-g_{11}g_{22}}\begin{bmatrix}g_{22}&-g_{12}\\-g_{21}&g_{11}\end{bmatrix} \tag{35}$$

By replacing (35) in (34), we obtain the equivalent controller,

$$\begin{cases}V_{peq}=\dfrac{1}{g_{12}g_{21}-g_{11}g_{22}}\begin{pmatrix}g_{22}(f_1+\lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2))\\-g_{12}(f_2+\lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4))\end{pmatrix}\\V_{yeq}=\dfrac{1}{g_{12}g_{21}-g_{11}g_{22}}\begin{pmatrix}-g_{21}(f_1+\lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2))\\+g_{11}(f_2+\lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4))\end{pmatrix}\end{cases} \tag{36}$$

4.2. O-NTSTW

NTSTW is the algebraic sum of the equivalent control and the discontinuous control. In this research, the discontinuous control part is replaced by a super twisting controller. The final control law is given as,

$$\begin{cases}V_p=\left\{\dfrac{1}{g_{12}g_{21}-g_{11}g_{22}}\begin{bmatrix}g_{22}(f_1+\lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2))\\-g_{12}(f_2+\lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4))\end{bmatrix}\right\}+\Delta V_p\\V_y=\left\{\dfrac{1}{g_{12}g_{21}-g_{11}g_{22}}\begin{bmatrix}-g_{21}(f_1+\lambda_1{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2))\\+g_{11}(f_2+\lambda_2{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4))\end{bmatrix}\right\}+\Delta V_y\end{cases} \tag{37}$$

where β_1, β_2,α_1 and α_2 are positives reals values. The discontinuous control part is a STW controller such that,

$$\begin{cases}\Delta V_p=\dfrac{-\alpha_1}{2}|S_1|^{\frac{1}{2}}sign(S_\theta)-\dfrac{\beta_1}{2}\displaystyle\int_0^{t_f}sign(S_{x_1})\,dt\\\Delta V_y=\dfrac{-\alpha_2}{2}|S_2|^{\frac{1}{2}}sign(S_\psi)-\dfrac{\beta_2}{2}\displaystyle\int_0^{t_f}sign(S_{x_2})\,dt\end{cases} \tag{38}$$

Remark 1: We can remark that the sliding surfaces (29) have a *sign* function, this term conserves the real sign of the state variables x_2 and x_4 while avoiding the singularity of the sliding variables if one of x_2 or x_4 value is negative. Moreover, the *sign* function may cause the chattering phenomenon. This chattering is attenuated at the same time by $|x_2|^{\gamma_1}$ and $|x_4|^{\gamma_2}$ respectively. Same explanation is given for the control (36), where signals generated by $sign(x_2)$ and $sign(x_4)$ will be attenuated respectively by$|x_2|^{2-\gamma_1}$ and $|x_4|^{2-\gamma_2}$.

Objective function

The objective function considered in this work is a sum of two performance indices, which are ITAE and ISCO. The ITAE based objective function have the following advantages; (i) Minimizing the stabilization time since the time is included in the criterion, and (ii) Minimizing the maximum overshoot because of the multiplication of the time by the absolute error. ISCO is used in order to avoid a large control effort, which can damage the system actuators. In addition, ISCO permits the optimization of the energy consumption.

$$J(e,u) = \int_0^t (t^T Q|e| + u^T R u)dt \tag{39}$$

The matrix $Q \in R_+^{n\times n}$ is chosen to guarantee the precision, rapidity in a minimum stability time. The matrix $R \in R_+^{n\times n}$ ensures that control inputs do not exceed their maximum values. Thus, R can safeguard the system actuators.

Remark 2: O-NTSTW controller parameters $\lambda_1, \lambda_2, \alpha_1, \beta_1, \alpha_2$ and β_2 in (29) and (38) are replaced respectively by λ_1^*, λ_2^*,α_1^*,β_1^*,α_2^* and β_2^*. These new parameters are optimized by GOW, WOA, ALO or SSA.

4.3. AO-NTSTW

Control of uncertain and perturbed nonlinear systems is a challenging task [44]. In order to improve performance particularly in terms of precision, and to remedy uncertainties, we propose a combination of O-NTSTW controller with an adaptive super twisting one [45]. The new approach has an optimized O-NTSTW sliding surface and its adaptive parameters are initialized with those obtained through the optimization step. The proposed controller makes it possible to attenuate chattering and guarantees rapid convergence in finite time. The gain of the new controller AO-NTSTW are given by (40)

$$\dot{\alpha} = \begin{cases} \omega_1\sqrt{\dfrac{\gamma_1}{2}} sign(|S| - \mu), if \alpha > \alpha_m \\ \eta, if \alpha \le \alpha_m \end{cases} \qquad \beta = 2\varepsilon\alpha \tag{40}$$

where $\omega_1, \eta, \varepsilon, \gamma_1, \alpha_m$ and μ are positive real constants, $|S(0)| > \mu$ and $\alpha(0) > \alpha_m$.

The design philosophy of the AO-NTSTW control is to dynamically increase the controller gains $\alpha(t)$ and $\beta(t)$ until the sliding mode is established. Then the gains start to decrease. This gains reduction must change its direction when the sliding variable or its derivative begins to deviate from the equilibrium point. This makes it possible to reduce the amplitude of the chattering since the gains are not overestimated, instead they are adjusted to handle uncertainties and disturbances. The structure of the AO-NTSTW controller is shown in Fig 3.

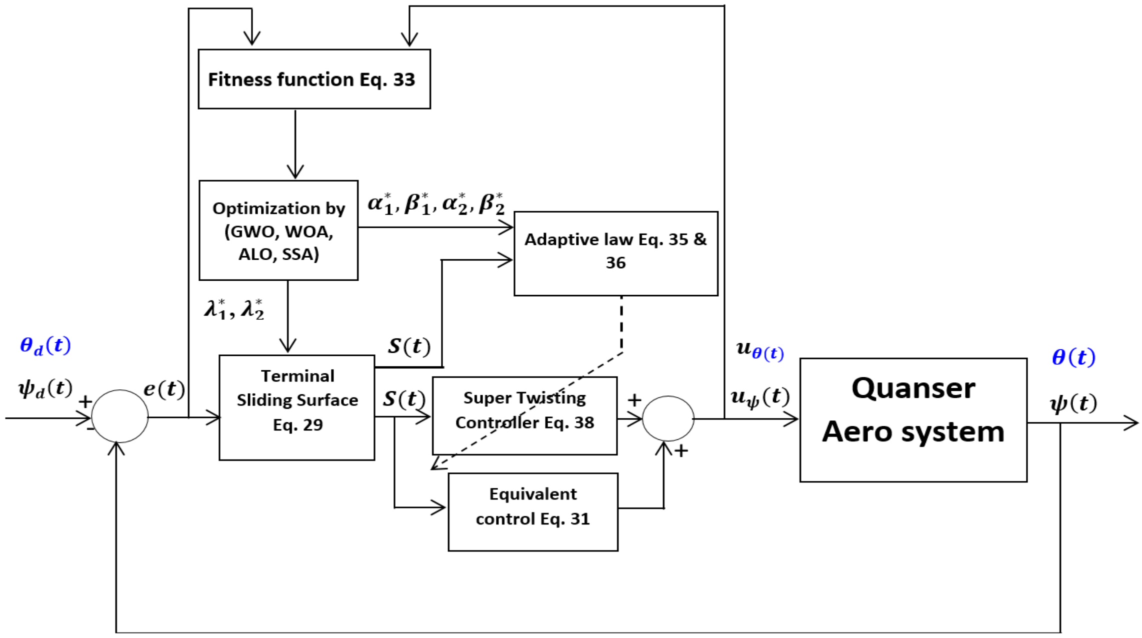


Figure 3. Structure of the AO-NTSTW controller

In order to have a controller that is both optimal and adaptive, we initialized the gains of the AO-NTSTW with the O-NTSTW optimal parameters. This avoids the problem of finding the initial parameters for the adaptive control. The optimal adaptive gains are given by

$$\begin{aligned} \alpha_1^*(t) &= \alpha_1(t) + \alpha_1^* \wedge \beta_1^*(t) = \beta_1(t) + \beta_1^* \\ \alpha_2^*(t) &= \alpha_2(t) + \alpha_2^* \wedge \beta_2^*(t) = \beta_2(t) + \beta_2^* \end{aligned} \tag{41}$$

where α_1^*,β_1^*,α_2^*,$\beta_2^*$are the O-NTSTW parameters determined in the optimization step. $\alpha_1(t)$,$\beta_1(t)$,$\alpha_2(t)$, $\beta_2(t)$ are obtained through (40). The details on system stability are given in [45]. The designed control law for AO-NTSTW is given in (42).

$$\begin{aligned} V_p &= \left\{\frac{1}{g_{12}g_{21} - g_{11}g_{22}}\left[g_{22}\left(f_1 + \lambda_1^*{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2)\right) \right.\right. \\ &\qquad \left.\left. - g_{12}\left(f_2 + \lambda_2^*{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4)\right)\right]\right\} + \Delta V_p(t) \\ V_y &= \left\{\frac{1}{g_{12}g_{21}-g_{11}g_{22}}\left[-g_{21}\left(f_1 + \lambda_1^*{\gamma_1}^{-1}|x_2|^{2-\gamma_1}sign(x_2)\right) + g_{11}\left(f_2 + \right.\right.\right. \\ &\qquad \left.\left.\left. \lambda_2^*{\gamma_2}^{-1}|x_4|^{2-\gamma_2}sign(x_4)\right)\right]\right\} + \Delta V_y(t) \end{aligned} \tag{42}$$

where,

$$\begin{cases} \Delta V_p(t) = \dfrac{-\alpha_1^*(t)}{2}|S_1|^{\frac{1}{2}}sign(S_\theta) - \dfrac{\beta_1^*(t)}{2}\displaystyle\int_0^{t_f} sign\left(S_{x_1}\right) dt \\ \Delta V_y(t) = \dfrac{-\alpha_2^*(t)}{2}|S_2|^{\frac{1}{2}}sign(S_\psi) - \dfrac{\beta_2^*(t)}{2}\displaystyle\int_0^{t_f} sign\left(S_{x_2}\right) dt \end{cases} \tag{43}$$

The new AO-NTSTW control approach can be written as Theorem 1.

Theorem 1: An AO-NTSTW control (42) exists for the system (9) and satisfies the adaptive Lyapunov stability-based law (34) with the optimization parameters (40) obtained by GOW, WOA, ALO or SSA.

Remark 2: Since α_1^*,β_1^*,α_2^*,β_2^* are used as the initial condition of (39), these will not affect the closed-loop stability of the system (9).

5. Simulation Results

This section aims to present the simulation results of the synthesized controllers. In order to make a credible comparison among the proposed optimization algorithms. First, the results of the objective function optimization to improve the convergence speed and the energy consumption are presented. Second, several different control laws were applied to the system. Third, the proposed new controller is simulated and compared with the best-performing commands resulting from the optimization.

In order to find the best algorithm to adopt for the Quanser aero system, four meta-heuristic algorithms such as: GWO, WOA, SSA, and ALO were used to provide the O-NTSMC, O-STW, and O-NTSTW parameters. Optimizations are performed in order to find the optimal control parameters that minimize the objective function. The experiment was repeated ten (10) times for each algorithm, in order to avoid local minima, premature convergence of the algorithms as well as having best results. The objective function used, in which it is given in its general form by (39), in this case of study it becomes,

$$J(e,u) = \int_0^{tf} \left(\begin{bmatrix} t & t \end{bmatrix} R \begin{bmatrix} |V_p| \\ |V_y| \end{bmatrix} + \begin{bmatrix} V_p & V_y \end{bmatrix} Q \begin{bmatrix} V_p \\ V_y \end{bmatrix} \right) dt \tag{44}$$

with:

$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $Q = \begin{bmatrix} 0.0015 & 0 \\ 0 & 0.0015 \end{bmatrix}$, $tf = 20\, Sec$, using these parameters' J can be written as:

$$J(e,u) = \int_0^{tf} t|e_\theta| + t|e_\psi| + 0.0015{V_p}^2 + 0.0015{V_y}^2 dt \tag{45}$$

The choice of Q values prevent the control inputs V_p and V_y from exceeding their maximum values of $[-24 \quad 24]$ volts. With $e_\theta = \theta - \theta_d$ and $e_\psi = \psi - \psi_d$.

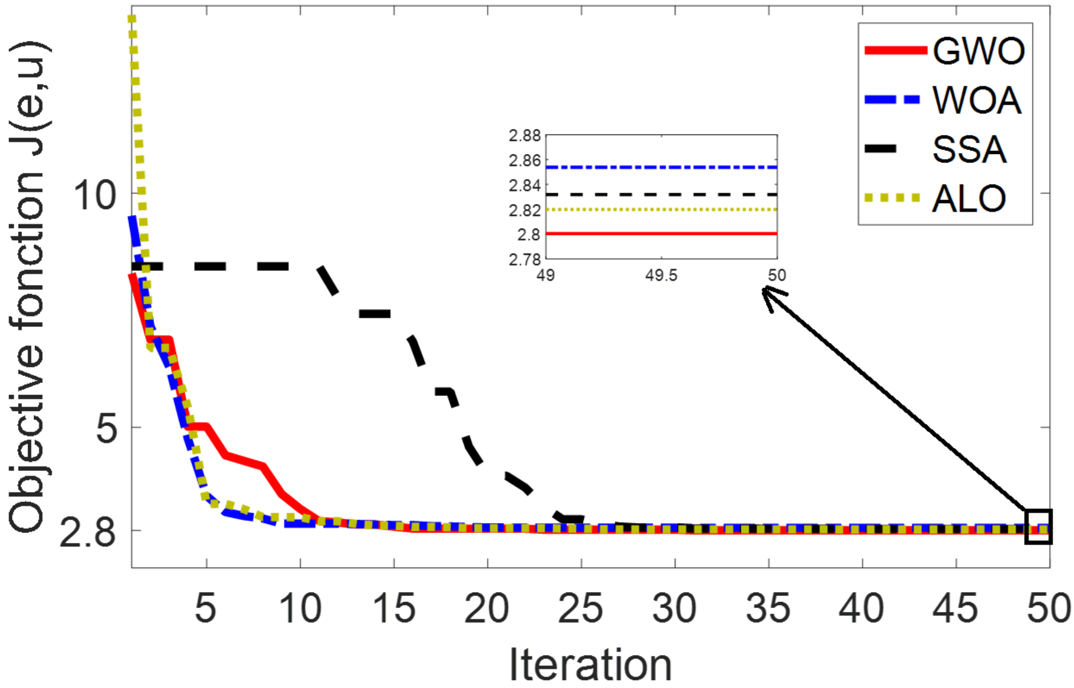
Each algorithm is tested 10 times under the following conditions: (1) Number of research agents: 30, (2) Number of iterations: 50, (3) Same objective function J. Used PC (Personal Computer) technical characteristics are: Processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 8G of RAM. The sampling time of the control system in simulation results is 0.01 Sec. The results of the objective function optimization are presented in the tables 3, 4 and 5. Three values of the objective function were presented in these tables such as: the best value and the corresponding experiment number where it was obtained, the worst value, as well as the average of the objective functions over 10 trials. These results values helped to determine the best algorithm, in an effective way.

Table 3. Objective function values in the case of O-NTSMC

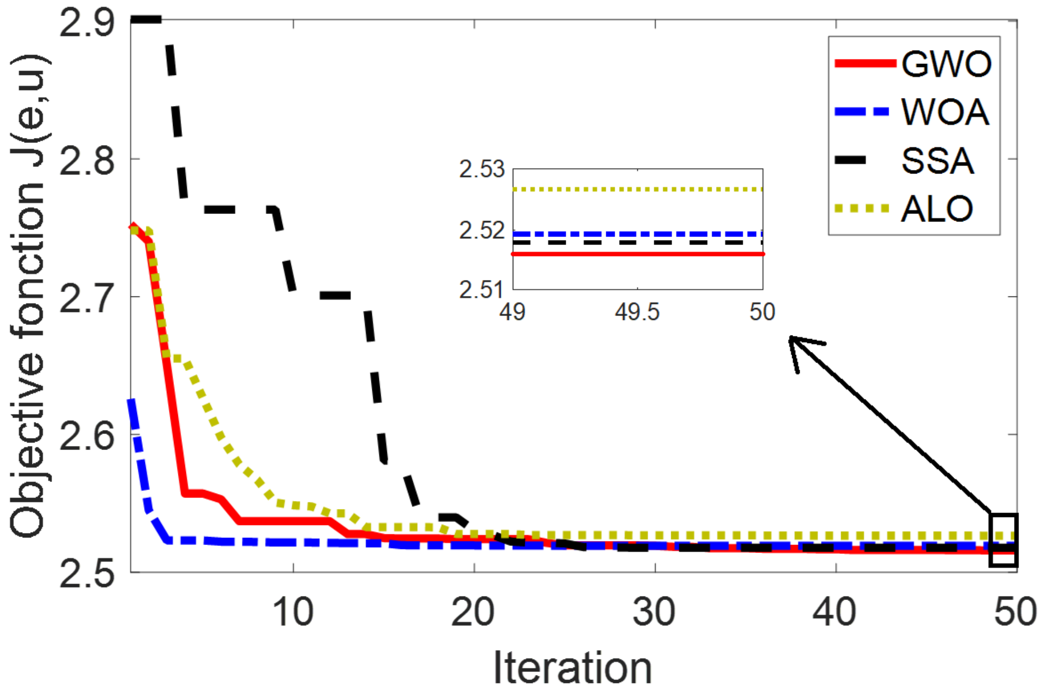
Algorithm	Best value of $J(e,u)$	Worst value of $J(e,u)$	Average value of J (over 10 trials)	experiment number where the best value of J was obtained
GWO	**2,80034044**	**2,85683905**	**2,81685419**	4
WOA	2,85374485	2,95269486	2,87405766	1
SSA	2,8317953	7,03750848	3,60071992	1
ALO	2,81980906	5,63474808	3,54016723	8

Table 4. Objective function values in the case of O-STW

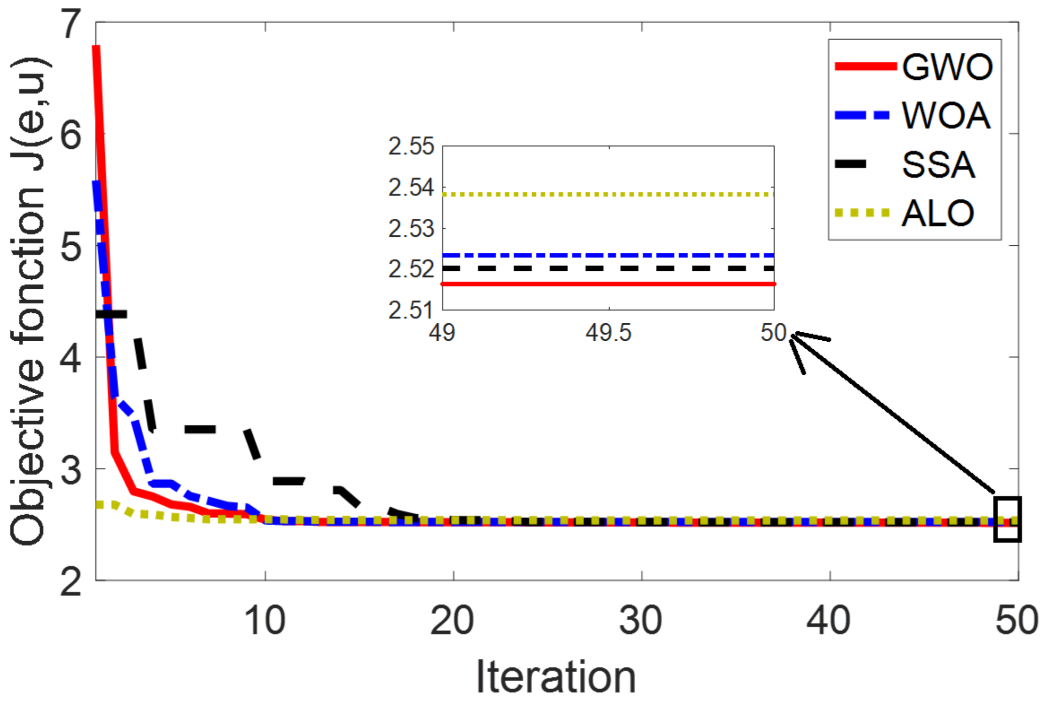
Algorithm	Best value of $J(e,u)$	Worst value of $J(e,u)$	Average value of $J(e,u)$ (over 10 trials)	experiment number where the best value of J was obtained
GWO	**2,51585141**	**2,52280305**	**2,51784165**	1
WOA	2,51922217	2,59324303	2,53927148	7
SSA	2,51776391	2,69928311	2,55970367	8
ALO	2,52657595	2,64169054	2,55694725	8



(a)



(b)



(c)

Figure 4. Objective function J (a) O-NTSMC (b) O-STW and (c) O-NTSTW

Table 5. Objective function values in the case of O-NTSTW

Algorithm	Best value of $J(e,u)$	Worst value of $J(e,u)$	Average value of $J(e,u)$ (over 10 trials)	experiment number where the best value of J was obtained
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

The optimization algorithms are classified in table 6 according to the values of the objective function of each controller. Table 7 gives a ranking of the means of J for the different algorithms and for the different applied controllers.

Table 6. Ranks of objective functions

Controllers	GWO	WOA	ALO	SSA
O-NTSMC	**1**	3	2	2
O-STW	**1**	3	4	2
O-NTSTW	**1**	3	4	2

Table 7. Rank of objective function Means of 10 tests

Controllers	GWO	WOA	ALO	SSA
O-NTSMC	**1**	2	3	4
O-STW	**1**	2	3	4
O-NTSTW	**1**	2	4	3

According to Table 6 and Table 7, the worst results in terms of minimum value of the objective function are obtained by the WOA and ALO algorithms, while GWO has achieved minimum values of the objective function in most cases. In addition, the GWO algorithm performed better in term of the best average over 10 trials. The WOA algorithm obtains a reasonable value of the average objective function over 10 trials. On the other hand, the highest average values were given by the SSA and ALO algorithms. Based on this, we can confirm that the GWO algorithm showed superior performance compared to the WOA, SSA and ALO algorithms.

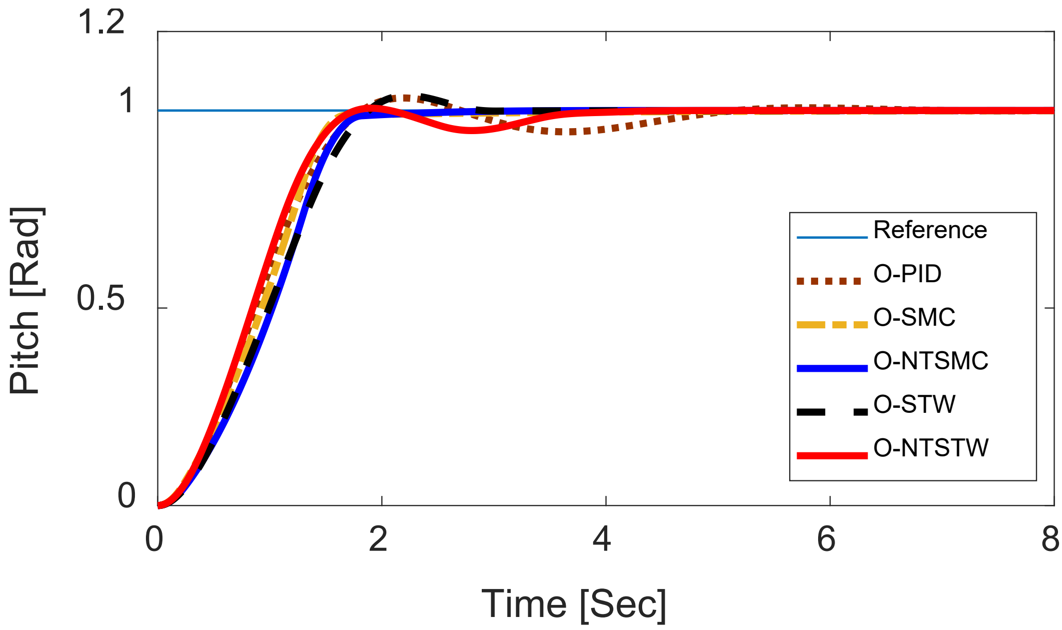
A peculiarity is noticed in these results concerning the SSA algorithm, this algorithm begins to converge after few number of iterations. The most visible case is the SMC controller SSA based optimization, where the algorithm starts to converge after the 18 iterations, this represents 36% of the total number of iterations, which is not negligible.

Fig. 4 shows the best values of the objective function per iteration, for the four optimization algorithms. In the three controllers, it is noticed that SSA-based optimization is the slowest compared to the other optimization algorithms (for example, in the case of O-NTSTW, it reaches its minimum, after 26 iterations, this represents more than 50% of the numbers of iterations). We notice that WOA and ALO converge quickly. GWO algorithm presents a good convergence speed to reach the minimum of the objective function (for example in the case of SMC and NTSMC it reaches its minimum respectively just after 16 and 10 iterations, this represents 32% and 20% from all iterations, respectively).

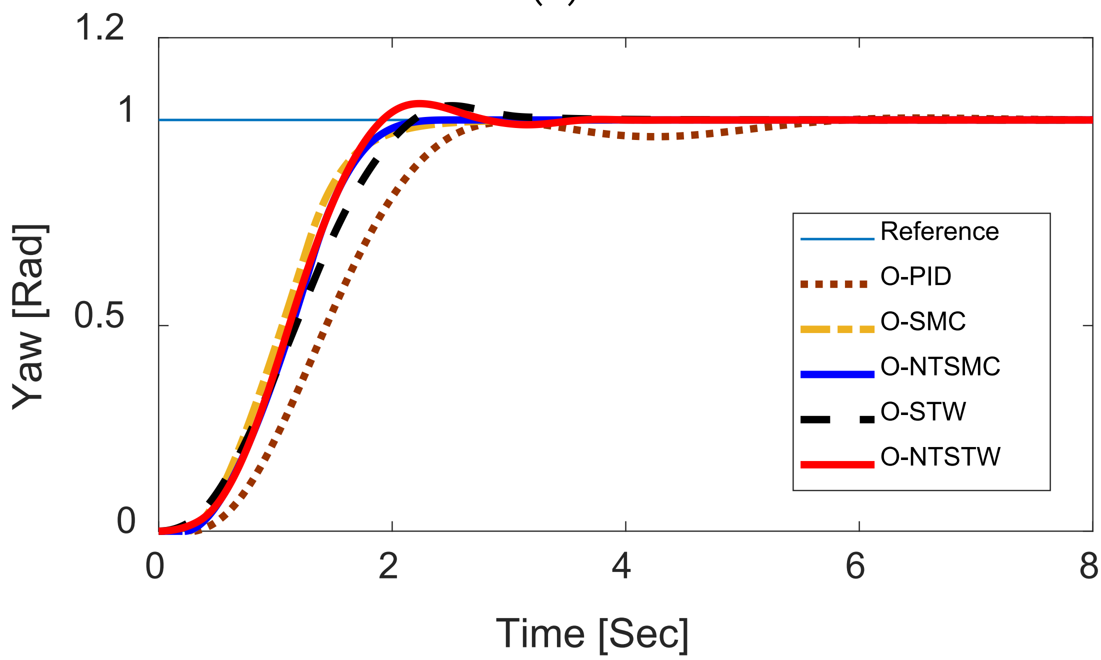
The use of 50 iterations for all algorithms helps to obtain a high degree of effectiveness of the results. Were, it is noticed that the objective functions for all results reach their minimum values before the 50 iterations for the 10 trials.

Application of the best controllers

In order to compare the best optimized controllers (i.e., best parameters for each controller according to the minimum value of the objective function), the angular responses, sliding surfaces and the control voltages for the pitch and yaw outputs are presented in Fig. 5, Fig. 6 and Fig. 7 respectively. Results of classical optimized PID and optimized SMC with linear sliding surfaces, also, are included in this section.

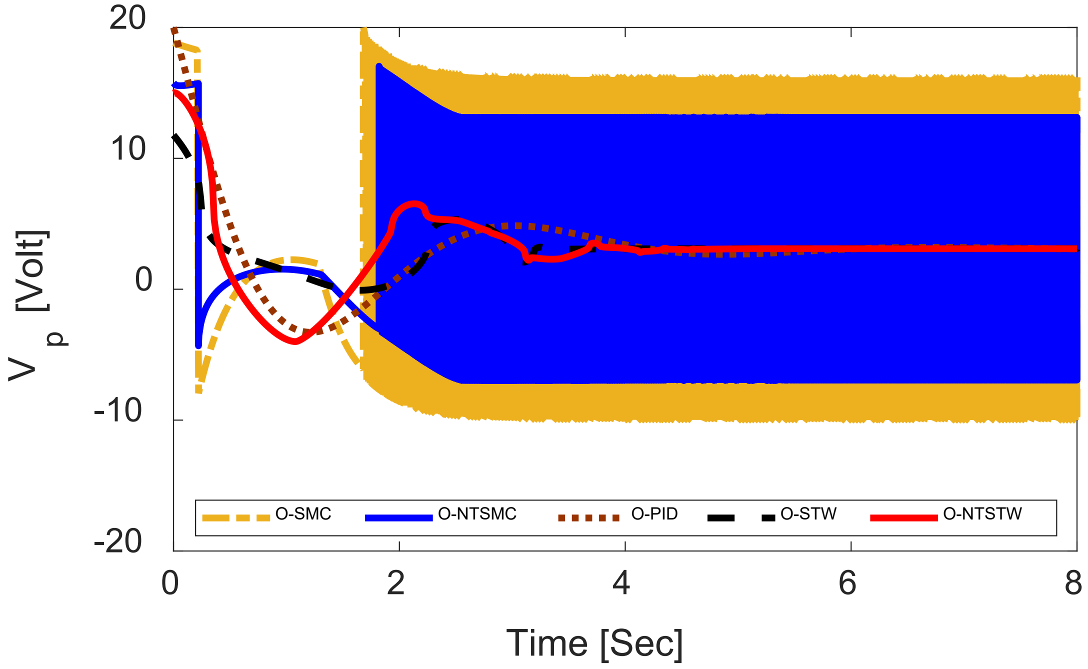


(a)

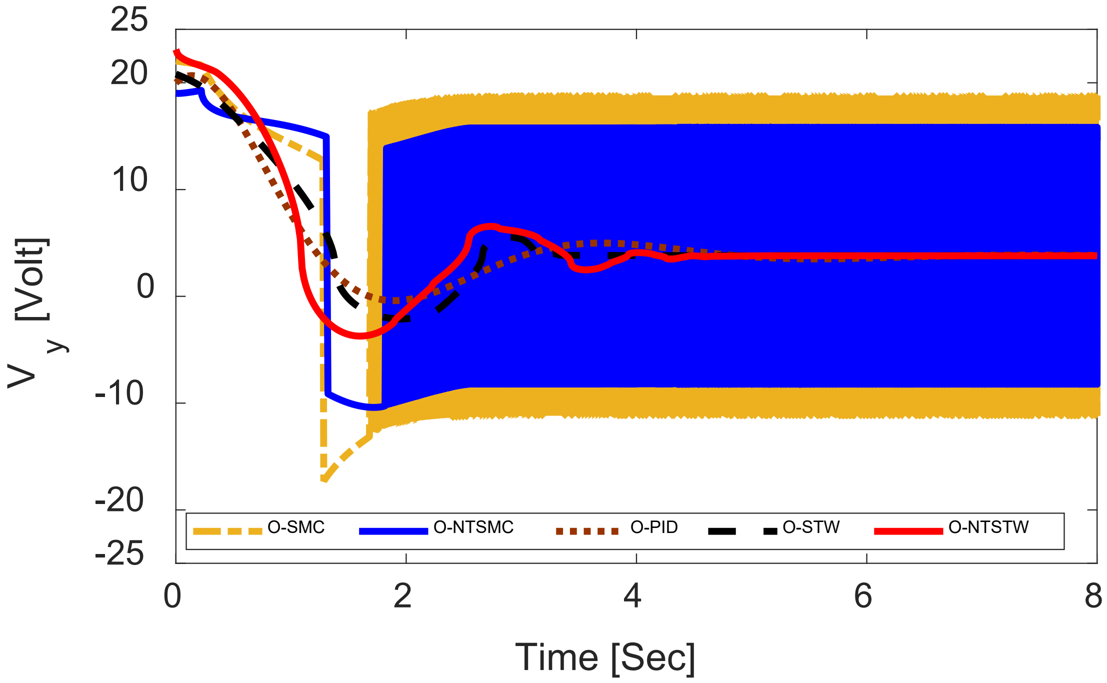


(b)

Figure 5. Responses of (a) pitch angle and (b) yaw angle of optimized O-PID, O-SMC, O-NTSMC, O-STW, O-NTSTW controllers

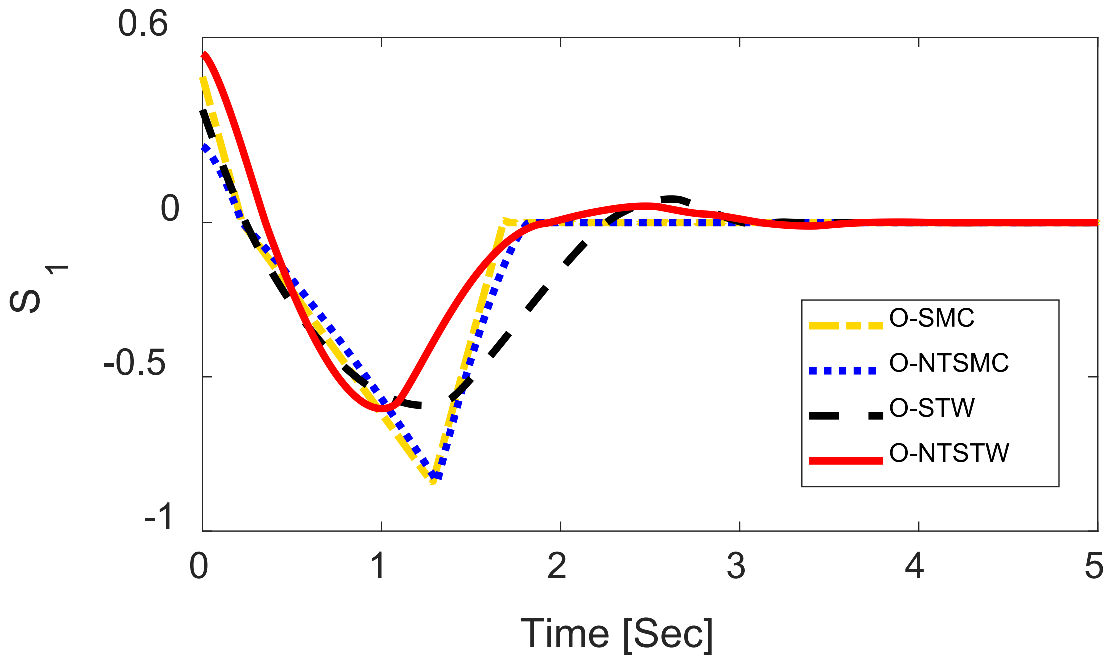


(a)

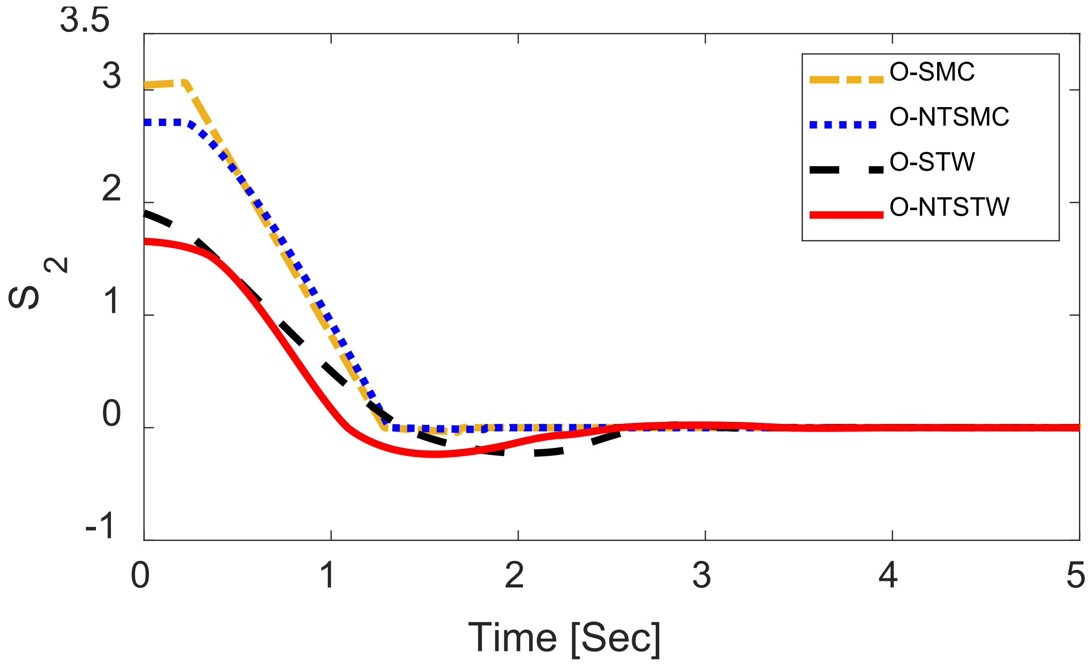


(b)

Figure 6. Control signals (a) pitch and (b) yaw for the optimized O-PID, O-SMC, O-NTSMC, O-STW, O-NTSTW controllers



(a)



(b)

Figure 7. Sliding surfaces (a) pitch (b) yaw of O-PID, O-SMC, O-NTSMC, O-STW, O-NTSTW controllers

Table 8. Comparison between the different control laws

	Controller	RT (s)	ST (s)	OS (%)	State Error (°)	ISCO
Pitch	O-NTSMC	1.1970	1.8914	**0**	3.766-E06	645.318
	O-STW	0.7010	1.0720	0.5920	**4.067 E-12**	**631.540**
	O-NTSTW	**0.6646**	**1.0132**	0.3835	7.552 E-12	631.878
Yaw	O-NTSMC	1.3726	2.3906	**8 e-06**	9.311 E-12	1051.912
	O-STW	**0.7299**	**1.1279**	0.3241	**5.214 E-12**	**1002.800**
	O-NTSTW	0.7374	1.1457	0.2388	1.469 E-09	1003.799

Table 8 summarizes the obtained performances using the four optimization algorithms. This table helps to compare these controllers with more punctuality. It is noted that in terms of rapidity of convergence, the two algorithms O-STW and O-NTSTW reached minimum values of RT and TS with a small advantage for the O-NTSTW in pitch, whose values are 0.6646 *sec* and 1.0132 *sec*, respectively. On the other hand, in yaw the advantage is for the O-STW with the values of RT=0.7299 sec and ST= 1.127 sec, which has a significant improvement in these performances in the case of these controllers compared to the O-NTSMC. However, it is noticed that O-NTSMC has a zero overshoot, which is logical, because this controller does not contain the integrator function. Moreover, the overshoots found are very small, which do not affect the behaviour of the system. The steady-state error (Table 8) has been improved by a significant value especially in pitch, it is around 100% of improvement in the case of O-STW and O-NTSTW. The energy consumption is almost the same in all cases and the differences are negligible. On the other hand, in the case of O-NTSMC the control signals have some chattering effect. The O-STW controller contains small fluctuations, which might affect the actuators of the robot in long time. While, the O-NTSTW controller is the best in this context, as it does not present any visible chattering. This is interpreted by the influence of the equivalent O-NTSMC-based control and the super-twisting part.

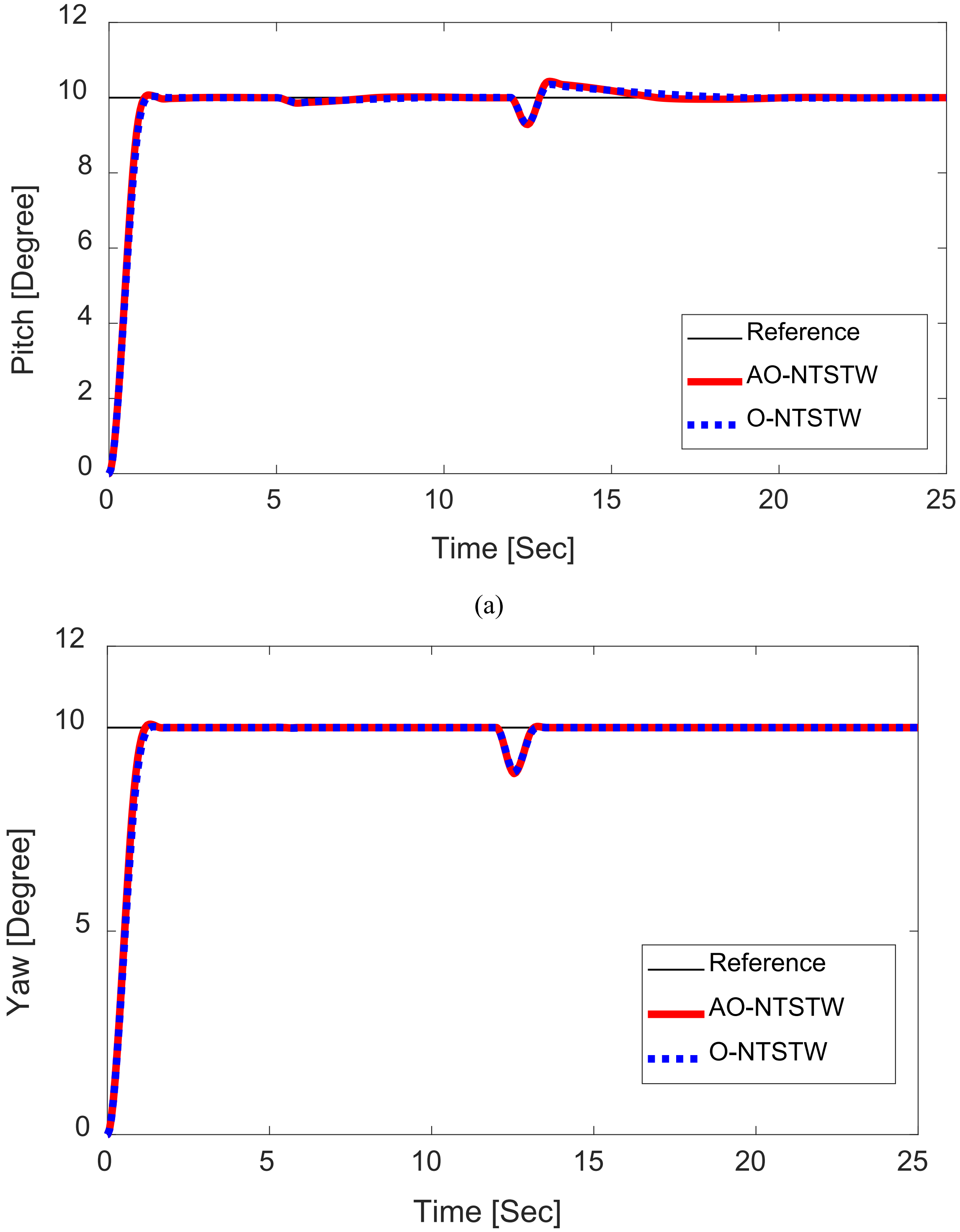
Comparison of O-NTSTW and AO-NTSTW

In this section, a comparison between O-NTSTW and AO-NTSTW is addressed. The chosen adaptive control parameters are given in table 9. Uncertainties and disturbances are a principal issue in the control of such system. Since sliding mode approaches are robust against these issues. It is specified that disturbance must has an upper bound. In the simulation phase the assessed controllers (O-NTSTW and AO-NTSTW) are examined against perturbations and uncertainty.

Table 9. Parameters of the *AO-NTSTW*

	$\boldsymbol{\omega_1}$	$\boldsymbol{\eta}$	$\boldsymbol{\varepsilon}$	$\boldsymbol{\gamma_1}$	$\boldsymbol{\mu}$	$\boldsymbol{\alpha_m}$
Pitch	0.001	0.0001	0.02	2	0.02	0.0001
Yaw	0.001	0.0001	0.05	2	0.06	0.0001

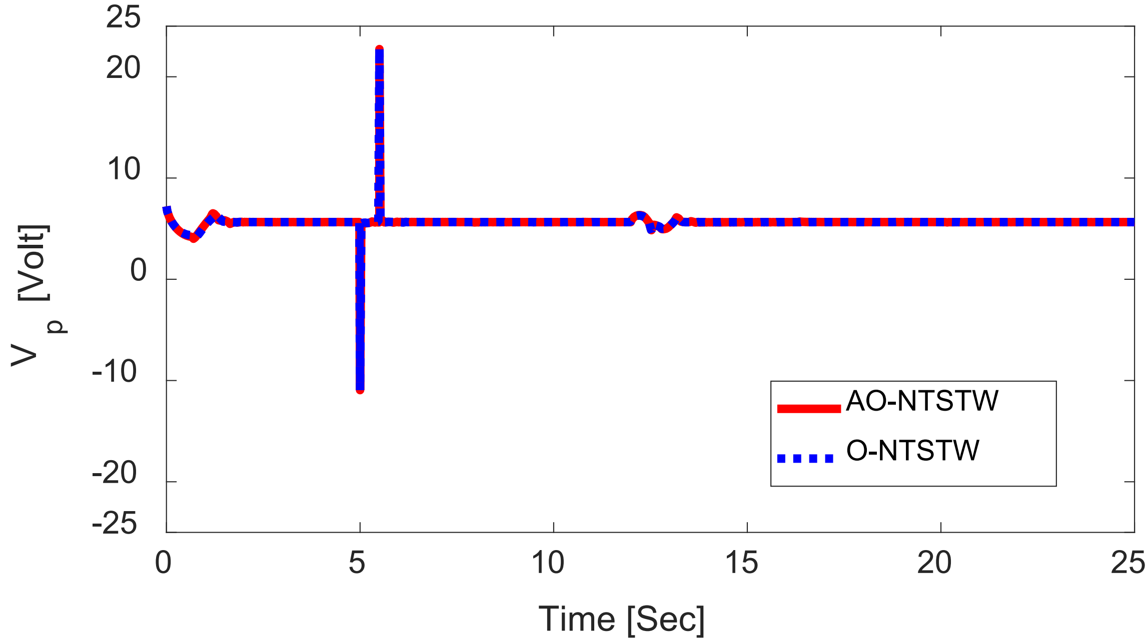
In this experiment, the system is examined in regulation mode. The system is disturbed with two pulses for a duration equal to 0.5 *sec*. The first disturbance is applied to the pitch angle at t=5 *sec* and the second is applied to the yaw angle at t=12 *sec*. This allows to examine the controller robustness from the cross-coupling effect and the external disturbance. Obtained results are presented in Fig. 8, 9 and 10.



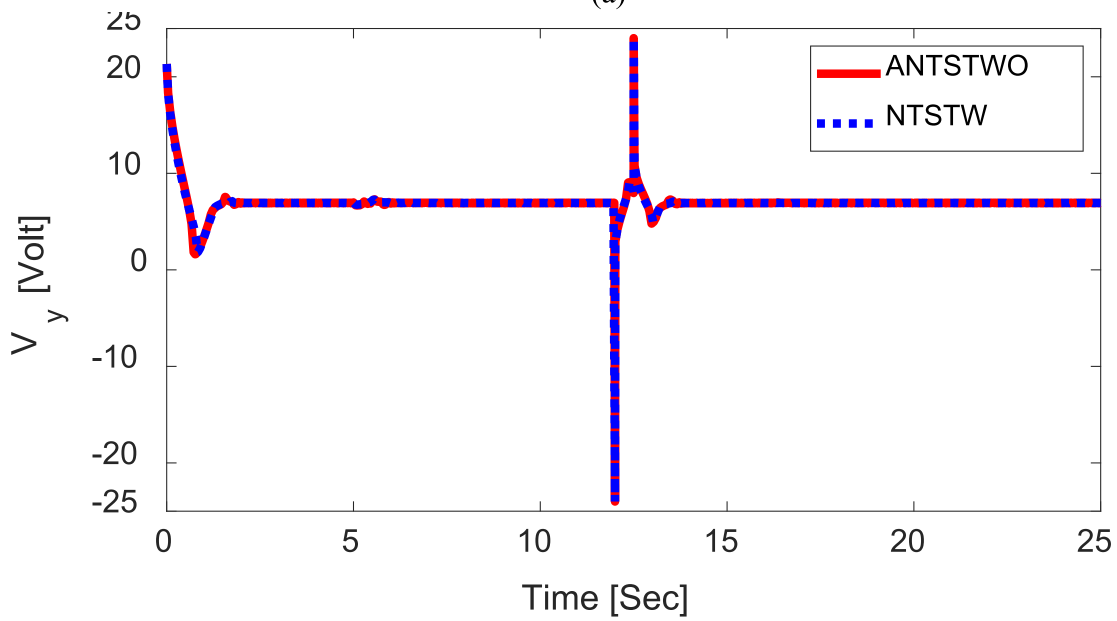
(a)

(b)

Figure 8. Step response (a) pitch (b) yaw for AO-NTSTW and O-NTSTW controllers



(a)



(b)

Figure 9. Control signals AO-NTSTW and O-NTSTW (a) Pitch (b) Yaw

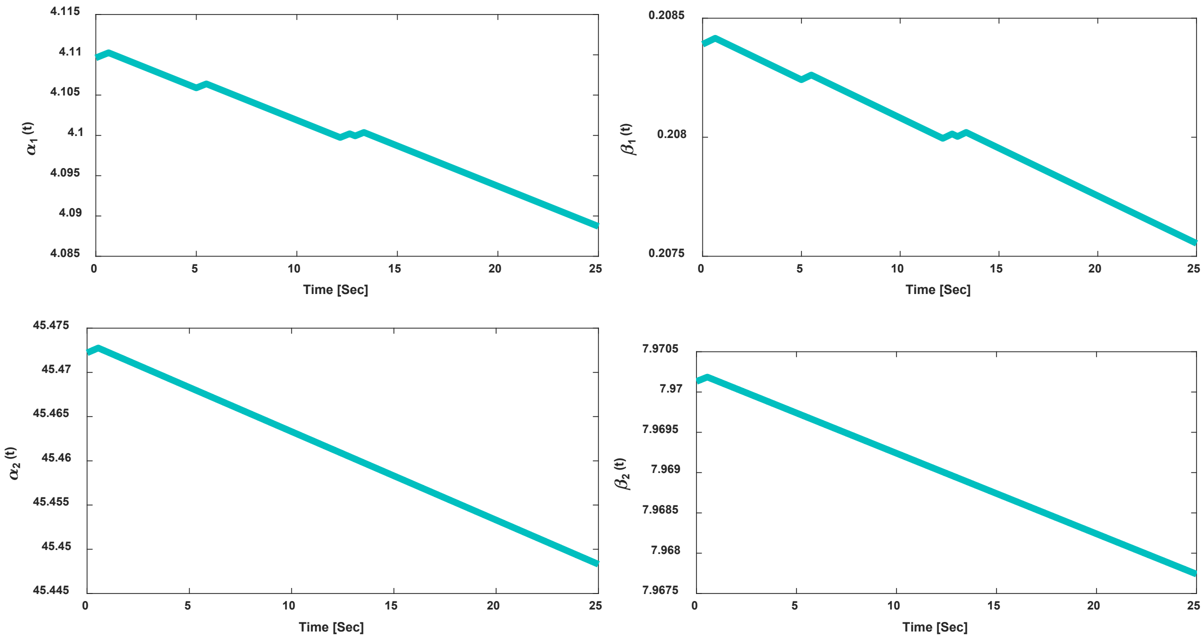
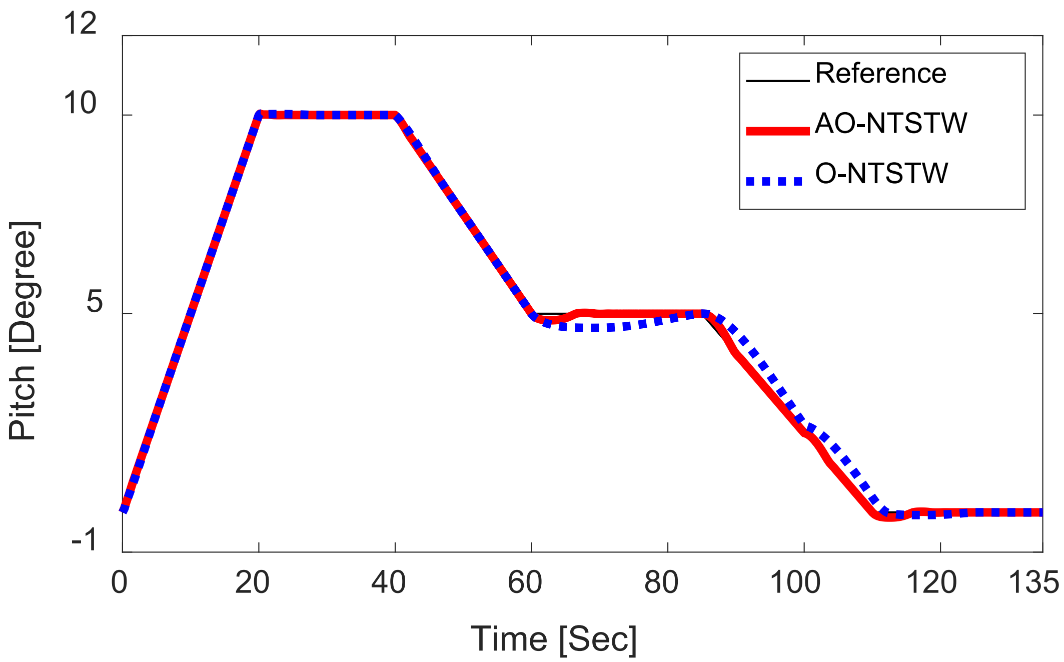
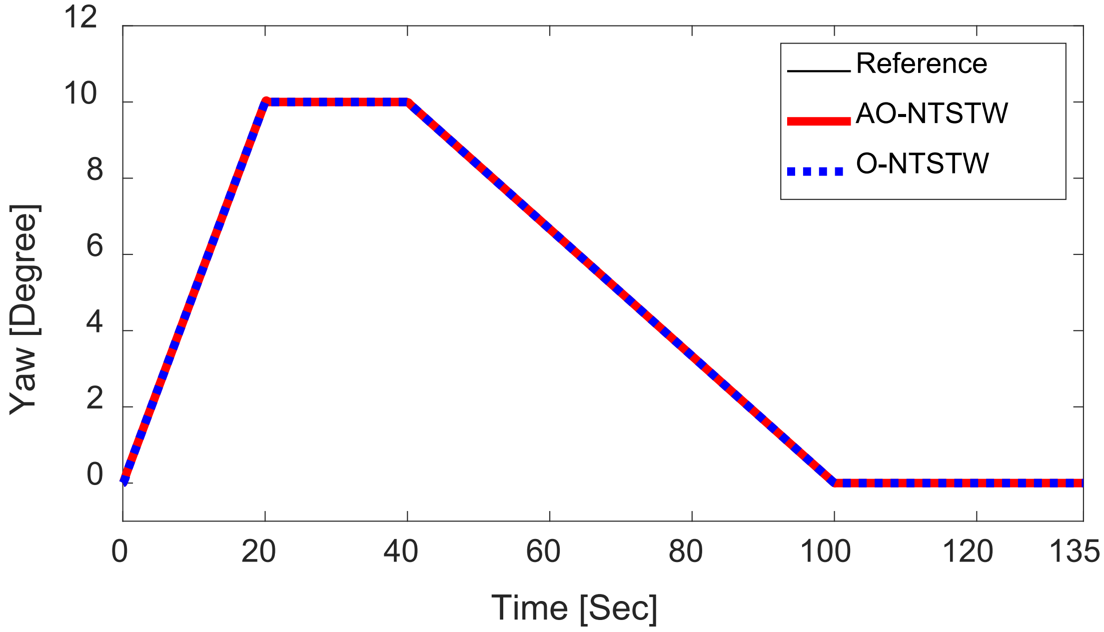


Figure 10. Adaptive gains evolution

In this part, we tested O-NTSTW and AO-NTSTW for a trajectory that contains a combination of constant, variable positions (speed profile). This scenario allows to assess the controllers' effectiveness against coupling dynamic effects.

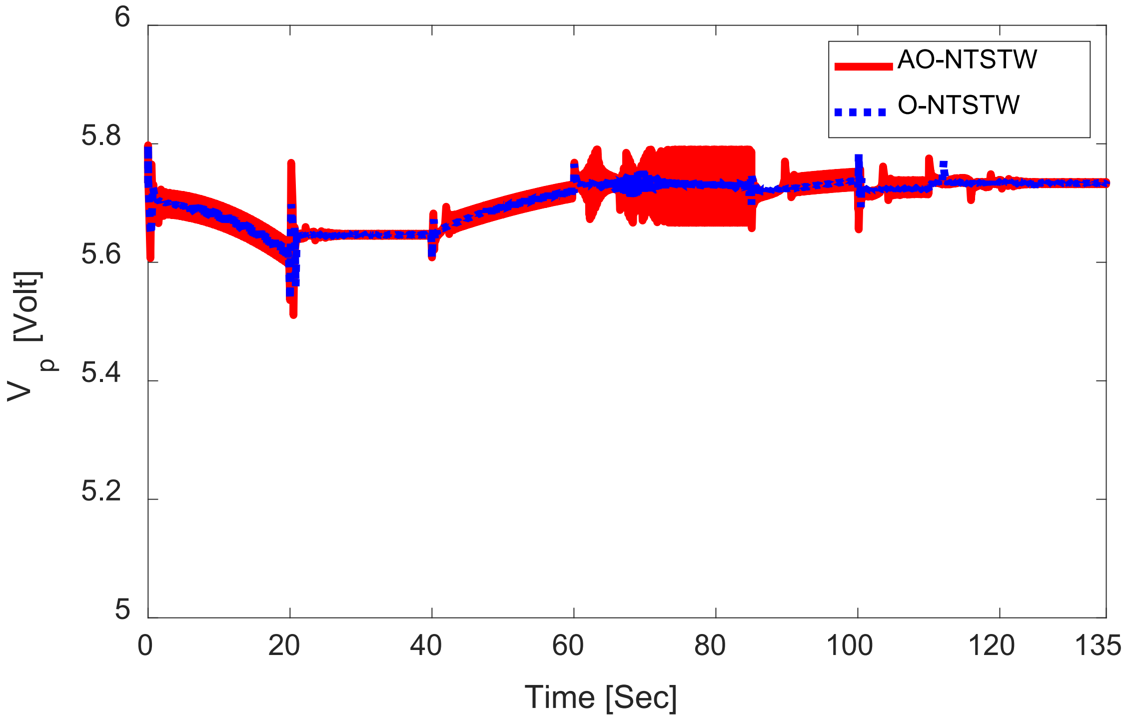


(a)

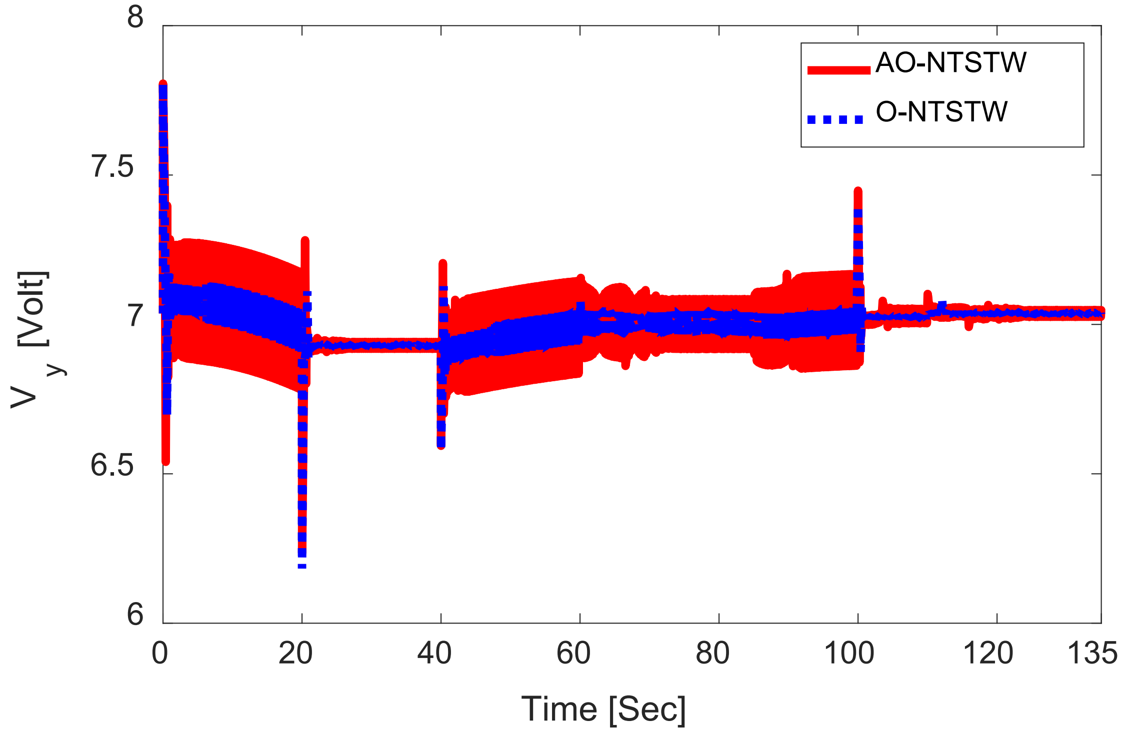


(b)

Figure 11. System responses of AO-NTSTW and O-NTSTW controllers (a) Pitch (b) Yaw



(a)



(b)

Figure 12. Control signals in the case of the speed profile (a) pitch (b) yaw

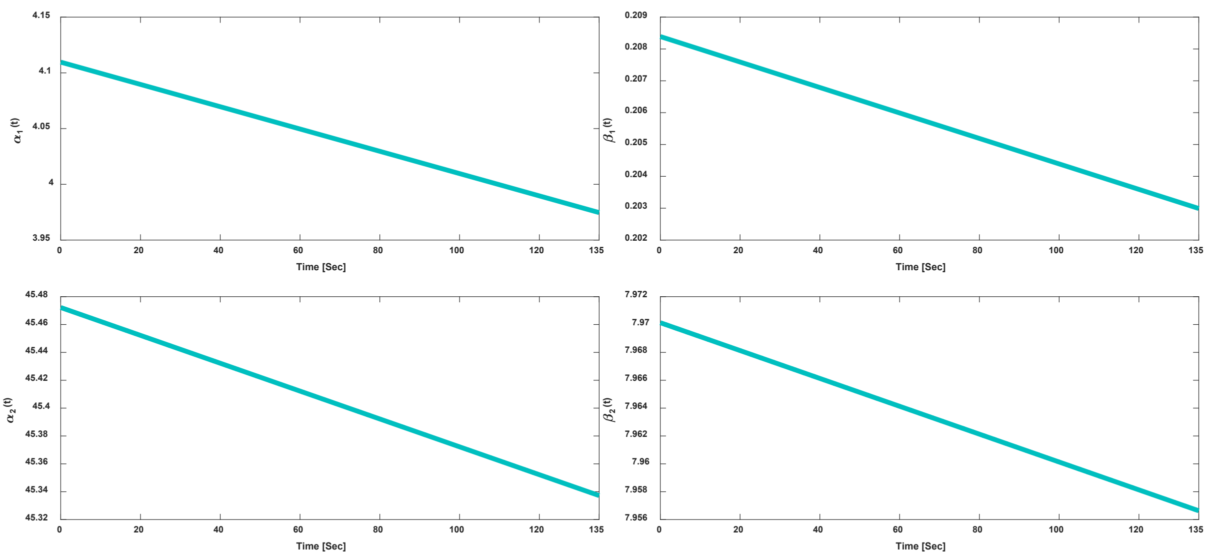


Figure 13. Evolution of the adaptive gains of the AO-NTSTW control (speed profile)

Table 10. Performances of O-NTSTW & AO-NTSTW in regulation mode

Angle	Controller	ISE	ISCO
Pitch	O-NTSTW	0.0121	**792.08**
	AO-NTSTW	**0.0118**	792.42
Yaw	O-NTSTW	0.0130	**1252.05**
	AO-NTSTW	**0.0127**	1256.41

Table 11. Performances of O-NTSTW & AO-NTSTW in trajectory tracking

Angle	Controller	ISE	ISCO
Pitch	O-NTSTW	0.0020	**4227.22**
	AO-NTSTW	**8.872 e-05**	4227.31
Yaw	O-NTSTW	4.718 e-07	**6367.30**
	AO-NTSTW	**2.272 e-07**	6368.24

In the case of regulation mode, the results illustrated in Figures 11-13 help to understand and verify the principle of the AO-NTSTW control. In addition, this simulation allows us to examine the two controllers AO-NTSTW and O-NTSW. When the system is far from the desired signal, the adaptive gains are increased between 0 *sec* and 0.7 *sec*, and when the desired signal is reached, the adaptive gains try to decrease it, this can be seen after 0.7 *sec*.

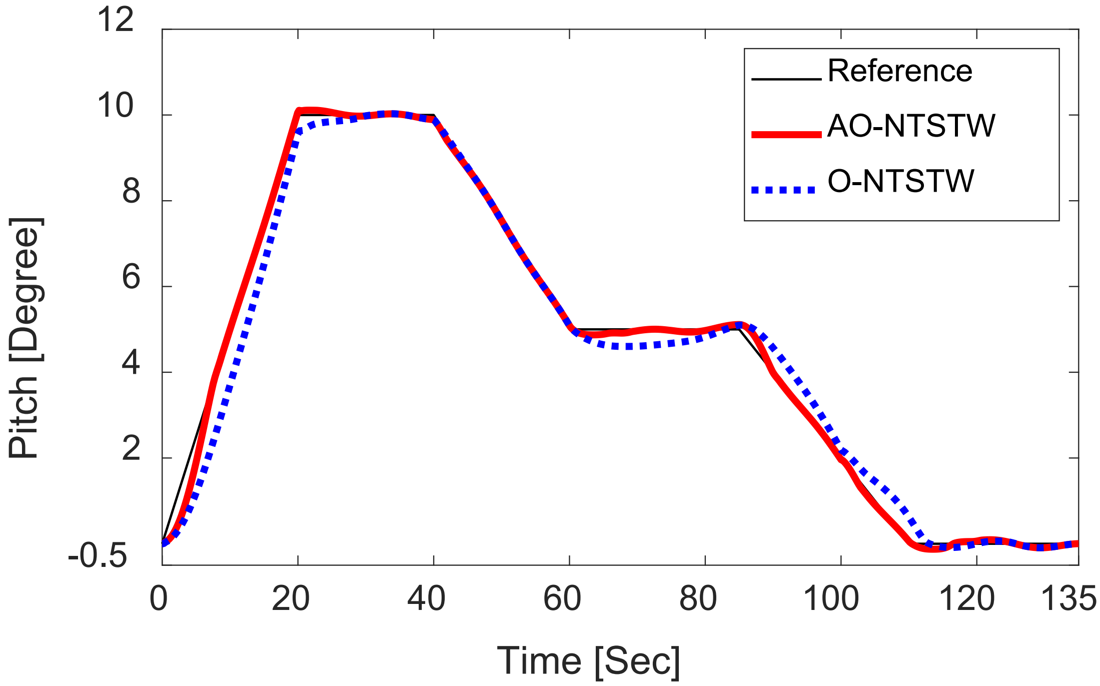
As shown in Fig. 8, the AO-NTSTW handles the effect of coupling and disturbance rejection more effectively than the O-NTSTW. The system rectifies its position and returns fast to the desired signal. Also, the gains begin to decrease until a new disturbance is added. In this simulation, the behaviour of adaptive gains and control signals can be observed between 5 sec and 12 sec, between 5 sec and 5.5 sec, and between 12 sec and 12.5 sec. From Table 10 and Table 11, the AO-NTSTW minimizes the ISE index compared to the O-NTSTW by 2.54% and 2.3% for the pitch and yaw, respectively. This can be justified by the decrease in the chattering amplitude. However, the energy consumption increased by 0.34% for yaw brought by the AO-NTSTW, this small over consumption is caused by the improvement of the precision.

Also, in trajectory tracking mode, the AO-NTSTW is more efficient than O-NTSTW. According to Fig. 11. We can see that the O-NTSTW control struggles to follow the trajectory, especially in pitch angle due to the gravitational effect and the axis coupling effect because the torque generated by the yaw motor affects the pitch axis considerably. AO-NTSTW gives an amplitude chattering, which is remarked by the increase in ISCO index. At the same time, the ISE index decreased considerably by 34.36% and 29.36% by the AO-NTSTW for the pitch (in regulation and in trajectory tracking, respectively). In the case of yaw, practically, there is no improvement of ISE and ISCO.

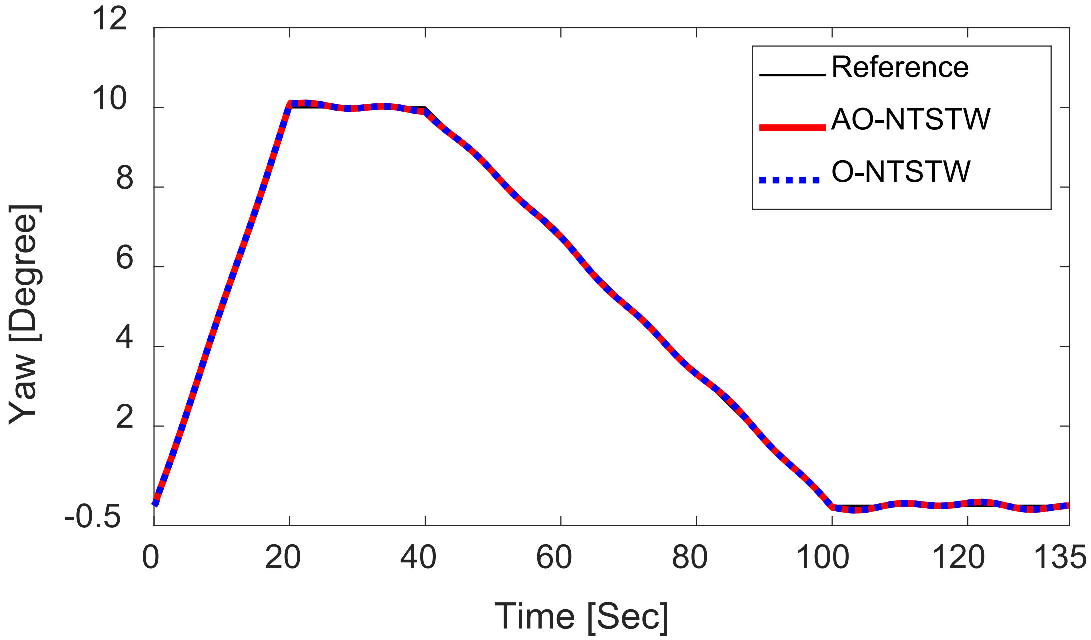
In order to test the proposed controller against disturbances and uncertainties the following experience is carried out, mass payload is added to the system which represents 10% of the total mass of the Quanser aero, hence the total mass is becomes $m_T = m_{qs} + 0.1 \times m_{qs}$, where: m_T is the total mass of the Quanser aero and the load mass, m_{qs} is the quanser aero mass. At the same time, a wind disturbance is applied to the output of the system. Mathematically this wind is modeled by the equation (46) as :

$$d = [0.01(sin(0.5t) + sin(0.2t)) \quad 0.01(sin(0.5t) + sin(0.2t))]^t \tag{46}$$

Equation (46) has two terms first term is the disturbance applied to the pitch movement and the other disturbance is applied to the yaw movement. These uncertainty and disturbance are applied to the Quanser aero in the trajectory tracing mode. The obtained results are given is the figures 14-16. Table. 12 summarizes the obtained performances.

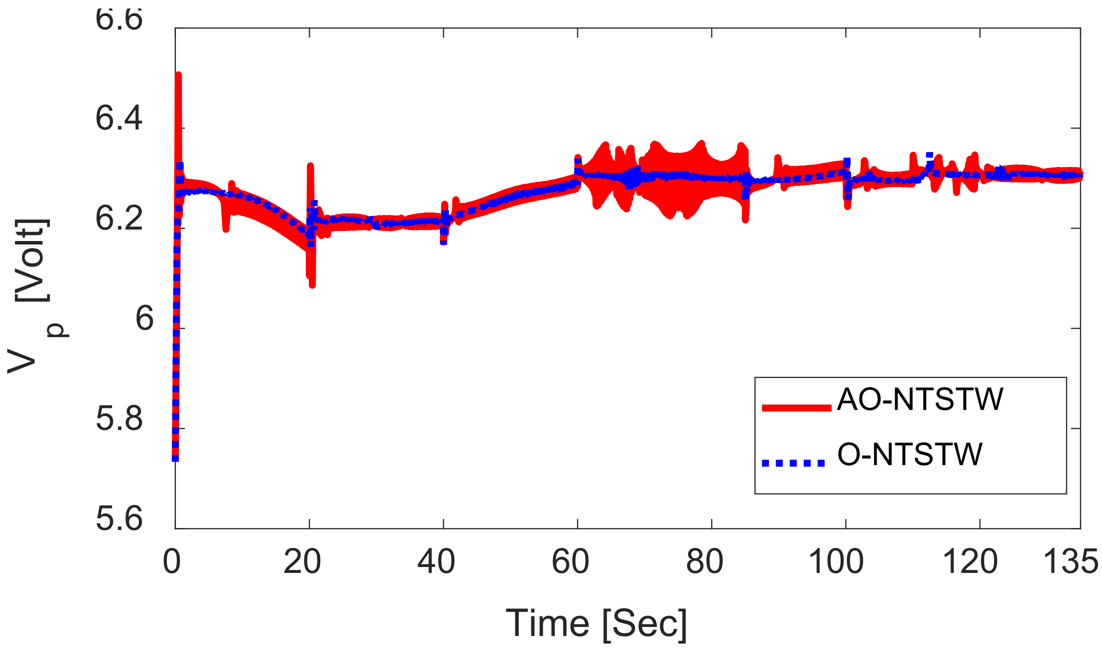


(a)

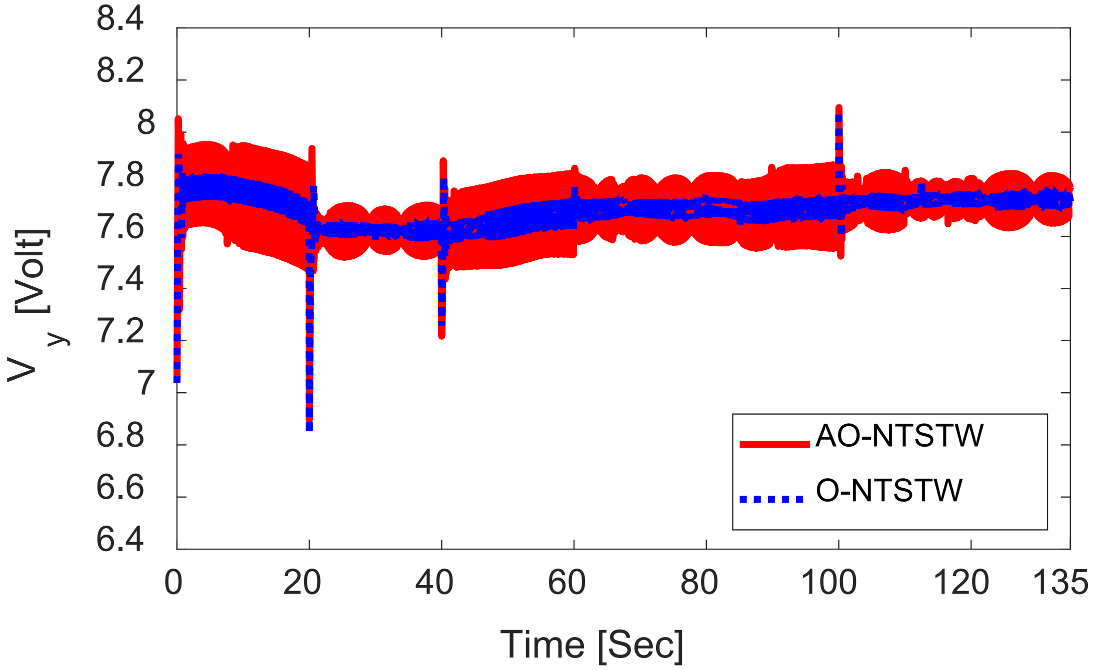


(b)

Figure 14. System responses of AO-NTSTW and O-NTSTW controllers against mass uncertainty and wind perturbation (a) Pitch (b) Yaw



(a)



(b)

Figure 15. Control signals in the presence of mass uncertainty and wind disturbance (a) pitch (b) yaw

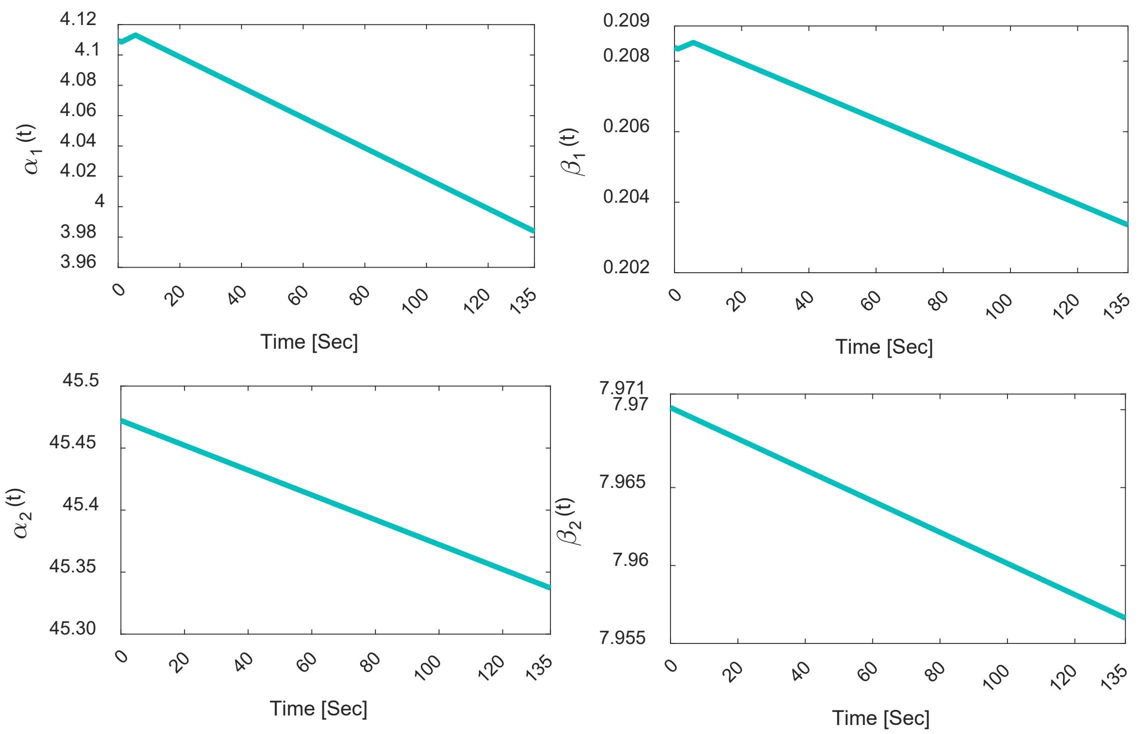


Figure 16. Evolution of the adaptive gains of the AO-NTSTW control in the presence of mass uncertainty and wind disturbance

Table 12. Performances of O-NTSTW & AO-NTSTW in the presence of mass uncertainty and wind disturbance

Angle	Controller	ISE	ISCO
Pitch	O-NTSTW	0.009	**$5.3163\ 10^{3}$**
	AO-NTSTW	**$6.1554\ 10^{-4}$**	$5.3140\ 10^{3}$
Yaw	O-NTSTW	$3.5692\ 10^{-7}$	**$8.0077\ 10^{3}$**
	AO-NTSTW	**$1.7702\ 10^{-7}$**	$8.0051\ 10^{3}$

From the accuracy pointview it is clear that the AO-NTSTW controller has given better results than the O-NTSTW, this results is remarkable from the response of the pitch angle where we can see that the ISE performance is equal to 6.1554 10-4, however for the same angle ISE is equal to 0.009 in the case of O-NTSTW. Adaptive parameters have played a central role in the adjustment of the control signals. The adaptive gains were decreased from their initial values. The control signals have an acceptable fluctuation for this class of system. In order to deal with the uncertainty and disturbances the adaptive control signals have more fluctuation then the O-NTSTW, this is justified by the adaptability effects.

In Fig. 6 the chattering effect has occurred which can be a destructive effect for the system. This phenomenon is caused by the discontinuous control part ($-ksign(s)$), in order to resolve this major drawback, the STW and the adaptive STW approaches (results in fig. 12) have been proposed; it is interesting here to indicate that these controllers or any other higher order sliding mode controllers can only attenuate the chattering and do not remove it totally. The fluctuation appeared in this result are coming from the nature of the imposed trajectory which is characterized by some changes of directions. From the practice perspective, the amplitude of these fluctuations is supportable by the actuator of this system.

Visualisation system

In order to have the visualization of the results, in this Subsection, a 3D model of the Aero Quanser is developed and then simulated using Gazebo environment in parallel with MATLAB/Simulink. We developed a ROS package on MATLAB/Simulink's environment to send position commands to Gazebo's controller. Figure 17 illustrates the developed software modules in order to apply the AO-NTSTW that we proposed using ROS. Fig. 18 shows the results of the simulation, the Figure 18a represents the starting position of the robot while Figure 18b represents its final position.

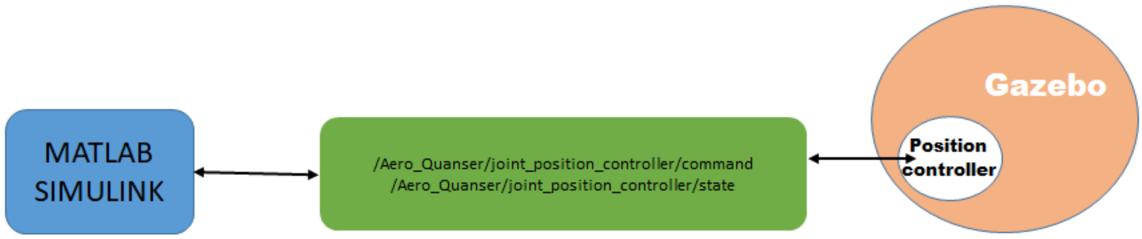
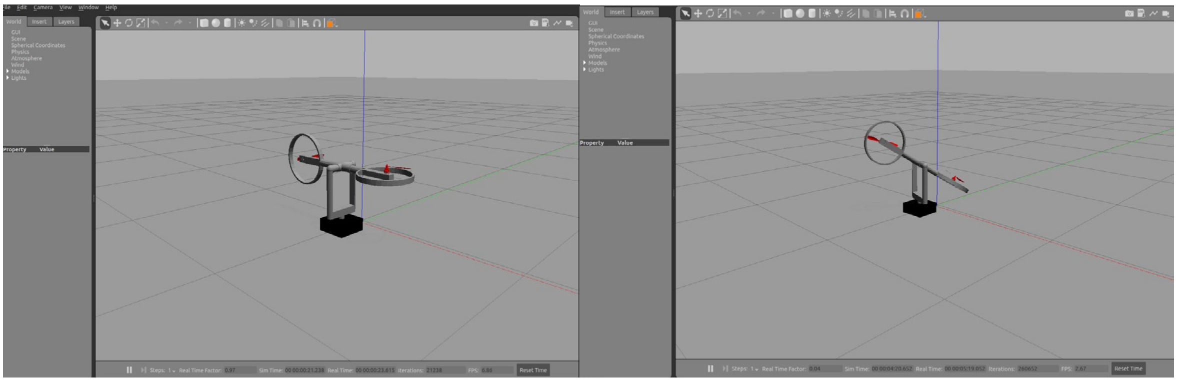


Figure 17. A schema of the developed software using Matlab/Simulink and Gazebo



(a) (b)

Figure 18. Quanser aero in Gazebo environnement (a) initial position (b) final position

6. Conclusion

In this paper, a new Adaptive Optimal Nonsingular Terminal Super Twisting (AO-NTSTW) controller was proposed for Quanser aero helicopter simulator. We have developed and compared optimisation results based on four metaheuristic algorithms. The results demonstrated that NTSTW based GWO is the most efficient algorithm compared to the WOA, SSA and ALO algorithms from the point of view of the objective function's optimal solution of the used ITAE and ISCO . The objective function was carefully chosen in order to guarantee a compromise between precision and control robustness. Where, the obtained results show that the GWO-based NTSTW control performs better than the O-NTSMC and O-STW controls in most aspects. Finally, the O-NTSTW command was compared to the AO-NTSTW controller subjected to numerous robustness assessments. The AO-NTSTW proposed controller, in this article, showed a significant superiority on all levels (i.e., performance and robustness) in both regulation mode and trajectory tracking.

References

[1] Tsouros, D. C., Bibi, S., & Sarigiannidis, P. G. (2019). A review on UAV-Based Applications for Precision Agriculture. Information, 10(11), 349.

[2] O. Mechali, L. Xu, X. Xie and J. Iqbal, “Theory and practice for autonomous formation flight of quadrotors via distributed robust sliding mode control protocol with fixed-time stability guarantee”, Control Engineering Practice, 123: 105150, 2022.

[3] S. G. Khan, S. Bendoukha, W. Naeem and J. Iqbal, “Experimental validation of an integral sliding mode-based LQG for the pitch control of a UAV-mimicking platform”, Advances in Electrical and Electronics Engineering, 17(3):275-284, 2019.

[4] Paul, P. K., & Jacob, J. (2020). H_2 Vs H_∞ control of TRMS via output error optimization augmenting sensor and control singularities. Ain Shams Engineering Journal, 11(1), 77-85.

[5] Kumar, S., & Dewan, L. (2022). A comparative analysis of lqr and smc for quanser aero. In Control and Measurement Applications for Smart Grid: Select Proceedings of SGESC 2021 (pp. 453-463). Singapore: Springer Nature Singapore.

[6] Gopmandal, F., & Ghosh, A. (2022). LQR-based MIMO PID control of a 2-DOF helicopter system with uncertain cross-coupled gain. IFAC-PapersOnLine, 55(22), 183-188.

[7] Hoffman, D., Rehan, M., MacKunis, W. et al. Quaternion-based Robust Trajectory Tracking Control of a Quadrotor Hover System. Int. J. Control Autom. Syst. 16, 2575–2584 (2018). https://doi.org/10.1007/s12555-018-0112-z

[8] Zhu, X., & Li, D. (2021). Robust fault estimation for a 3-DOF helicopter considering actuator saturation. Mechanical Systems and Signal Processing, 155, 107624.

[9] Reyhanoglu, M.; Jafari, M.; Rehan, M. Simple Learning-Based Robust Trajectory Tracking Control of a 2-DOF Helicopter System. Electronics, 2022, 11.13: 2075.

[10] Abdelmaksoud, Sherif I.; Mailah, Musa; 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, 9: 4886-4901.

[11] Saleem, O.; Abbas, F.; Iqbal, J. Complex Fractional-Order LQIR for Inverted-Pendulum-Type Robotic Mechanisms: Design and Experimental Validation. Mathematics 2023, 11, 913. doi: 10.3390/math11040913

[12] Kim, S. K., & Ahn, C. K. (2021). Performance-Boosting Attitude Control for 2-DOF Helicopter Applications via Surface Stabilization Approach. IEEE Transactions on Industrial Electronics, 69(7), 7234-7243.

[13] Zhao, Z., Zhang, J., Liu, Z., Mu, C., and Hong, K.S. (2022). Adaptive neural network control of an uncertain 2-DOF helicopter with unknown backlash-like hysteresis and output constraints. IEEE Transactions on Neural,Networks and Learning Systems.

[14] Boukadida, W., Benamor, A., Messaoud, H., & Siarry, P. (2019). Multi-objective design of optimal higher order sliding mode control for robust tracking of 2-DoF helicopter system based on metaheuristics. Aerospace Science and Technology, 91, 442-455.

[15] Haruna, A., Mohamed, Z., Efe, M. Ö., & Basri, M. A. M. (2020). Improved integral backstepping control of variable speed motion systems with application to a laboratory helicopter. ISA transactions, 97, 1-13.

[16] Shtessel, Y., Edwards, C., Fridman, L., & Levant, A. (2014). Sliding Mode Control and Observation (Vol. 10). New York: Springer New York.

[17] Ahmad, S.; Uppal, A.A.; Azam, M.R.; Iqbal, J. Chattering Free Sliding Mode Control and State Dependent Kalman Filter Design for Underground Gasification Energy Conversion Process. Electronics 2023, 12, 876. doi: 10.3390/electronics12040876

[18] 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.
[19]X. Yu, Y. Feng and Z. Man, "Terminal Sliding Mode Control – An Overview," in IEEE Open Journal of the Industrial Electronics Society, vol. 2, pp. 36-52, 2021, doi: 10.1109/OJIES.2020.3040412.
[20] Anjum, M.B.; Khan, Q.; Ullah, S.; Hafeez, G.; Fida, A.; Iqbal, J.; Albogamy, F.R. Maximum Power Extraction from a Standalone Photo Voltaic System via Neuro-Adaptive Arbitrary Order Sliding Mode Control Strategy with High Gain Differentiation. Appl. Sci. 2022, 12, 2773. doi: 10.3390/app12062773
[21] Ekbote, A. K., Srinivasan, N. S., & Mahindrakar, A. D. (2011). Terminal sliding mode control of a twin rotor multiple-input multiple-output system. IFAC Proceedings Volumes, 44(1), 10952-10957.
[22] Ghellab, M. Z., Zeghlache, S., Djerioui, A., & Benyettou, L. (2021). Experimental validation of adaptive RBFNN global fast dynamic terminal sliding mode control for twin rotor MIMO system against wind effects. Measurement, 168, 108472.
[23] Azizi, S., Soleimani, R., Ahmadi, M., Malekan, A., Abualigah, L., & Dashtiahangar, F. (2022). Performance Enhancement of an Uncertain Nonlinear Medical Robot with Optimal Nonlinear Robust Controller. Computers in Biology and Medicine, 146, 105567.
[24] Rezoug, A., Achour, Z., & Hamerlain, M. (2014, December). Ant Colony Optimization of Type-2 Fuzzy Helicopter Controller. In 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014) (pp. 1548-1553).
[25] Eberhart, R., & Kennedy, J. (1995, November). Particle Swarm Optimization. In Proceedings of the IEEE international conference on neural networks (Vol. 4, pp. 1942-1948).
[26] Laware, A. R., Navthar, R. R., Bandal, V. S., & Talange, D. B. (2022). Global Optimization of Second-Order Sliding Mode Controller Parameters Using a New Sliding Surface: An experimental verification to process control system. ISA transactions, 126, 498-512.
[27] Bagheri, A., Jabbari, A., & Mobayen, S. (2021). An Intelligent ABC-Based Terminal Sliding Mode Controller For Load-Frequency Control of Islanded Micro-Grids. Sustainable Cities and Society, 64, 102544.
[28] M. S. Zatout, A. Rezoug, A. Rezoug, K. Baizid & J. Iqbal (2022) Optimisation of Fuzzy Logic Quadrotor Attitude Controller – Particle Swarm, Cuckoo Search and BAT Algorithms, International Journal of Systems Science, 53:4, 883-908, doi: 10.1080/00207721.2021.1978012.
[29] Mirjalili, S, S. M. Mirjalili, and Andrew L. "Grey Wolf Optimizer." Advances in engineering software 69 (2014): 46-61.
[30] A. Rezoug, J. Iqbal, M. Tadjine, Extended Grey Wolf Optimization–Based Adaptive Fast Nonsingular Terminal Sliding Mode Control of a Robotic Manipulator. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 2022;236(9):1738-1754. doi:10.1177/09596518221099768.
[31] Mirjalili, S. "The Ant Lion Optimizer" Advances in engineering software 83 (2015): 80-98.
[32] Mirjalili, S., and Andrew L. "The whale optimization algorithm." *Advances in engineering software* 95 (2016): 51-67.
[33] Mirjalili, S, et al. "Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems." Advances in engineering software 114 (2017): 163-191.
[34] Aghaseyedabdollah, M., Abedi, M., & Pourgholi, M. (2022). Supervisory adaptive fuzzy sliding mode control with optimal Jaya based fuzzy PID sliding surface for a planer cable robot. Soft Computing, 26(17), 8441-8458.
[35] Aghaseyedabdollah, M., Abedi, M., & Pourgholi, M. (2022). Supervisory adaptive interval type-2 fuzzy sliding mode control for planar cable-driven parallel robots using Grasshopper optimization. Iranian Journal of Fuzzy Systems, 19(5), 111-129. doi: 10.22111/ijfs.2022.7160

[36] Sabah, N., Hameed, E., & Al-Huseiny, M. S. (2021). Optimal Sliding Mode Controller Design Based on Whale Optimization Algorithm for Lower Limb Rehabilitation Robot. Applied Computer Science, 17(3).
[37] Kumar, S., & Ajmeri, M. (2021). Optimal Variable Structure Control with Sliding Modes for Unstable Processes. *Journal of Central South University*, *28*(10), 3147-3158.
[38] Al-Qassar, A. A., Al-Obaidi, A. S. M., Hasan, A. F., Humaidi, A. J., Nasser, A. R., Alkhayyat, A., & Ibraheem, I. K. (2021). Finite-Time Control of Wing-Rock Motion for Delta Wing Aircraft Based on Whale-Optimization Algorithm. *Indonesian Journal of Science and Technology*, *6*(3), 441-456.
[39] Han, Seongik. "Grey Wolf and Weighted Whale Algorithm Optimized IT2 Fuzzy Sliding Mode Backstepping Control with Fractional-Order Command Filter for a Nonlinear Dynamic System." *Applied Sciences* 11.2 (2021): 489.
[40] El Hajjami, L., Mellouli, E. M., & Berrada, M. (2021). Robust adaptive non-singular fast terminal sliding-mode lateral control for an uncertain ego vehicle at the lane-change maneuver subjected to abrupt change. *International Journal of Dynamics and Control*, *9*(4), 1765-1782.
[41] Fessi, R., Rezk, H., Bouallegue, S.: Grey wolf optimization based tuning of terminal sliding mode controllers for a quadrotor. CMC-Computers Materials & Continua 68(2), 2265–2282 (2021)
[42] Kong, X., & Zhang, T. (2020). Non-singular Fast Terminal Sliding Mode Control of High-Speed Train Network System Based on Improved Particle Swarm Optimization Algorithm. *Symmetry*, *12*(2), 205.
[43] https://www.quanser.com/products/aero-2/
[44] O. Khan, M. Pervaiz, E. Ahmad and J. Iqbal, “On the derivation of novel model and sophisticated control of flexible joint manipulator”, Revue Roumaine des Sciences Techniques-Serie Electrotechnique et Energetique, 62(1): 103-108, 2017.
[45] Shtessel, Y., Taleb, M., & Plestan, F. (2012). A Novel Adaptive-Gain Supertwisting Sliding Mode Controller: Methodology and Application. Automatica, 48(5), 759-769.