Control System of a Quadrotor UAV with an Optimized Backstepping Controller

Hamid Hassani
Laboratory of Intelligent Systems, Geo-resources and Renewable Energies
Faculty of sciences and technology,
Sidi Mohamed Ben Abdellah
University
Hamid.hassani@usmba.ac.ma

Anass Mansouri
Laboratory of Intelligent Systems, Geo-resources and Renewable Energies
School of Applied Sciences,
Sidi Mohamed Ben Abdellah
University

Ali Ahaitouf
Laboratory of Intelligent Systems, Geo-resources and Renewable Energies
Faculty of sciences and technology,
Sidi Mohamed Ben Abdellah
University

***Abstract*— Quadrotor is an inherently nonlinear, coupled and unstable system which require an efficient controller to perform accurate tracking without losing the target trajectory. This paper concerns the development of an optimal backstepping controller (OBS) for the quadrotor attitude. A mathematical modeling for the quadrotor dynamics was derived by taking into account its nonlinear dynamics. Ant colony optimization algorithm (ACO) was used to select the optimal parameters of the BS controller. However, the obsession behind the use of this optimization algorithm is the development of a reliable flight controller ensuring optimal performances. In order to highlight the accuracy of the proposed approach, we carried out numerical experiments using two trajectory scenarios.**

***Keywords*—** *Quadrotor, optimal backstepping controller, Ant colony optimization.*

I. Introduction

Through the scientific progress made in the unmanned aerial vehicle area and the enormous technological revolution especially in sensors technology, optimization algorithms and control approaches, quadrotor becomes one of the most attractive research topics. Moreover, the use of quadrotors in the civilian and military applications grows strongly day by day. In the civilian domain, quadrotors can perform several autonomus missions involving infrastructure inspection and rescue … In the other hand quadrotors are used especially in the aerial surveillance and the spy missions for the military applications.

Quadrotor belongs to the range of underactuated rotary wing vehicles which can perform six motions with only four motors arranged in the corners of a cross configuration as shown in fig. 1. These motions are obtained directly by varying the motors speed without any pitch blade, however, the attitude motions (Roll, Pitch and Yaw) are obtained when the balance between the torques produced by the four rotors is affected. Altitude motion is obtained by varying the total lifting force generated by the four propellers. Lateral motions along x and y axis are achieved through Roll and Pitch rotations.

We start our work by deriving the nonlinear dynamic model of the quadrotor using the famous Newton-Euler formalism [1]. However, designing a reliable mathematical model close to the real physical model is the first step for synthesizing robust flight controller that will guarantee high performance in term of trajectory tracking.

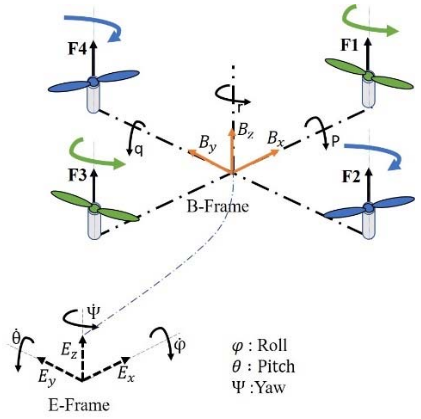


Figure. 1: Quadrotor configuration.

A great number of research works have focused on the quadrotor control design using linear[2]–[6] and nonlinear strategies [7]–[10]. Each control approach has internal parameters that determine the quality of the controller, such as PID controller which depends on the right choose of the PID gains[11], the response of the sliding mode controller depends also on the choose of the sliding surface[12], likewise backstepping controller which needs a good configuration of the backstepping gains [8]. Generally, a hard effort should be invested to identify the correct parameters manually. with a little luck, we can find an internal configuration of the controller that gives sufficiently accurate results, but we cannot know if it is the best configuration. Optimization algorithms came to remove the fog covering the internal control parameters, and to reduce the effort introduced during the configuration phase of the controller.

Going through relevant works cited in literature, one can see that several optimization algorithms have been adopted to confront problems related to the configuration of the quadrotor flight controller. Among them, Genetic Algorithm (GA) has been used in [13] for the optimization of the PID controller parameters, this work adopt an “Integral of Time multiplied by Absolut Error” (ITAE) as a cost function, the results of the GA tuned PID controller were more accurate comparing to the conventional PID. To improve the performances of the quadrotor equipped with a sliding-mode controller in path tracking missions, an interesting comparative study was proposed in [14], between three optimization algorithms, this study prove the accuracy of the Firefly Algorithm (FA) compared to Ant Colony Optimization (ACO) and Invasive Weed Optimization

978-1-7281-4813-7/19/$31.00©2019IEEE

(IWO). Another swarm intelligence based algorithm has been introduced in [15] to deal with the selecting challenge of the controller parameters, this paper uses a nonlinear backstepping strategy as a controller for an unmanned quadrotor, and a Particle Swarm Optimization (PSO) algorithm was presented to choose the optimal configuration of the backstepping controller.

To deal with the attitude subsystem control, we adopt in this work the backstepping control technique tuned using ACO algorithm. The optimization of the BS controller setting has been successfully achieved, and results of numerical experiments as well as the comparative analysis prove the accuracy of the adopted control strategy in both stabilization and trajectory tracking missions.

The rest of this work is organized as follows, section II presents the mathematical formulation of the quadrotor nonlinear dynamic model, while the backstepping control design for the quadrotor attitude subsystem is described in section III, then the optimization of the control parameters will be formulated in section Ⅳ, the simulation results are presented in the penultimate section, and the last section conclude the work.

II. Quadrotor modeling

The dynamic modeling of the quadrotor takes in consideration the following assumptions:

- Two referential are needed to model the quadrotor, a fixed Earth frame and a fixed body frame.
- Quadrotor is a rigid body in a symmetrical structure.
- Propellers are characterized by a rigid structure.
- The center of mass (CM) coincides with the center of gravity (CG).
- The trust force is proportional to the square of the rotor speed $F_i = b.\Omega_i{}^2$, the same for the drag torque it is proportional to the square of the speed $M_i = d.\Omega_i{}^2$.
- Air friction is neglected.

Consider the following vectors representing the linear velocity V and the angular velocity W . For more simplification the linear velocity must be expressed in the Earth-frame, while the angular velocity should be expressed in the Body-frame.

$$V = [\dot{x}, \dot{y}, \dot{z}]^T \tag{1}$$

$$w = [\dot{\varphi}, \dot{\theta}, \dot{\Psi}]^T \tag{2}$$

Thanks to the symmetrical structure of the quadrotor which seems to be a boon giving the modeling and the control design an advantage that reduce the complexity of the calculation. Herein the linear position is defined by the cartesian variables x, y and z, while the angular position is described by Euler angles φ, θ and Ψ known more with their aeronautical names respectively Roll, Pitch and Yaw angles.

Quadrotor dynamic model is obtained using Newton-Euler formalism [1] described by the following equations:

$$m.\dot{V} = \sum F \tag{3}$$

$$I.\dot{w} + w \wedge (Iw) = \sum M \tag{4}$$

Where m is the quadrotor total mass, I is inertia matrix, $\sum F$ are the external forces and $\sum M$ are the moments acting on the quadrotor system.

1. *Translation dynamic*

Forces acting on the quadrotor are the gravitational force and the thrust force expressed by the following equation.

$$m.\dot{V} = -mg.\begin{bmatrix}0\\0\\1\end{bmatrix} + R.\begin{bmatrix}0\\0\\b.(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2)\end{bmatrix} \tag{5}$$

By expanding (5), the quadrotor translational dynamic can be writing in the following form:

$$\dot{V} = \begin{pmatrix}\ddot{x}\\\ddot{y}\\\ddot{z}\end{pmatrix} = \begin{pmatrix}\frac{U_1}{m}(\cos\varphi \sin\theta \cos\Psi + \sin\varphi \sin\Psi)\\ \frac{U_1}{m}(\cos\varphi \sin\theta \sin\Psi - \sin\varphi \cos\Psi)\\ \frac{U_1}{m}(\cos\varphi \cos\theta) - g\end{pmatrix} \tag{6}$$

In equations (5) and (6), R represents the rotation matrix mapped in (7). U_1 is the total thrust force represented by (8) and b is the thrust factor.

$$\text{R}=\begin{pmatrix}C_\theta C_\Psi & S_\varphi S_\theta C_\Psi - C_\varphi S_\Psi & C_\varphi S_\theta C_\Psi + S_\varphi S_\Psi\\ C_\theta S_\Psi & S_\varphi S_\theta S_\Psi + C_\varphi C_\Psi & C_\varphi S_\theta S_\Psi - S_\varphi C_\Psi\\ -S_\theta & S_\varphi C_\theta & C_\varphi C_\theta\end{pmatrix} \tag{7}$$

$$U_1 = b.(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \tag{8}$$

2. *Rotational dynamic*

The Newton's second law represented by (4) gives:

$$I.\dot{w} + w\ x\ (Iw) = \sum M = M_{pro} - M_{Gyro}. \tag{9}$$

Torques acting in the quadrotor system are the torque produced by the spinning of the propellers M_{pro}, and the gyroscopic torque M_{Gyro}.

$$M_{pro} = \begin{bmatrix}\text{Roll}_\text{Torque}\\ \text{Picth}_\text{Torque}\\ \text{Yaw}_\text{Torque}\end{bmatrix} = \begin{bmatrix}b.l(\Omega_4{}^2 - \Omega_2{}^2)\\ b.l(\Omega_3{}^2 - \Omega_1{}^2)\\ d(\Omega_3{}^2 + \Omega_1{}^2 - \Omega_4{}^2 - \Omega_2{}^2)\end{bmatrix} \tag{10}$$

Resulting torque of the gyroscopic effect is given by:

$$M_{Gyro} = \begin{pmatrix}\dot{\varphi}\\\dot{\theta}\\\dot{\Psi}\end{pmatrix} \wedge J_r \begin{pmatrix}0\\0\\\sum_{i=1}^{4}(-1)^{i+1}\Omega_i\end{pmatrix} = J_r\,\Omega \begin{pmatrix}\dot{\theta}\\-\dot{\varphi}\\0\end{pmatrix}. \tag{11}$$

With $\Omega = -\Omega_1 - \Omega_2 + \Omega_3 + \Omega_4$

The centripetal force is calculated as follow:

$$w \wedge I\,w = \begin{pmatrix}\dot{\varphi}\\\dot{\theta}\\\dot{\Psi}\end{pmatrix} \wedge \begin{pmatrix}I_x & 0 & 0\\ 0 & I_y & 0\\ 0 & 0 & I_z\end{pmatrix}\begin{pmatrix}\dot{\varphi}\\\dot{\theta}\\\dot{\Psi}\end{pmatrix} = \begin{pmatrix}\dot{\theta}\dot{\Psi}\,(I_z - I_y)\\ \dot{\Psi}\dot{\varphi}\,(I_x - I_z)\\ \dot{\varphi}\dot{\theta}\,(I_y - I_x)\end{pmatrix} \tag{12}$$

Replacing (10), (11) and (12) in (9), the rotational dynamic can be expressed as follows:

$$\begin{pmatrix}\ddot{\varphi}\\ \ddot{\theta}\\ \ddot{\Psi}\end{pmatrix} = \begin{pmatrix}\dot{\theta}\dot{\Psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_r}{I_x}\dot{\theta}\Omega + \frac{U_2}{I_x}\\ \dot{\varphi}\dot{\Psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_r}{I_y}\dot{\varphi}\Omega + \frac{U_3}{I_y}\\ \dot{\varphi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{U_4}{I_z}\end{pmatrix} \qquad (13)$$

With U_2, U_3 and U_4 are respectively the Roll, Pitch and Yaw moments.

By gathering the translational dynamic (6) and the rotational dynamic (13), the quadrotor full dynamic model can be writing according to the following system of equations:

$$\begin{pmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\\ \ddot{\varphi}\\ \ddot{\theta}\\ \ddot{\Psi}\end{pmatrix} = \begin{pmatrix}\frac{U_1}{m}\left(\cos\varphi\sin\theta\cos\Psi + \sin\varphi\sin\Psi\right)\\ \frac{U_1}{m}\left(\cos\varphi\sin\theta\sin\Psi - \sin\varphi\cos\Psi\right)\\ \frac{U_1}{m}\left(\cos\varphi\cos\theta\right) - g\\ \dot{\theta}\dot{\Psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_r}{I_x}\dot{\theta}\Omega + \frac{U_2}{I_x}\\ \dot{\varphi}\dot{\Psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_r}{I_y}\dot{\varphi}\Omega + \frac{U_3}{I_y}\\ \dot{\varphi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{U_4}{I_z}\end{pmatrix} \qquad \mathbf{(14)}$$

The quadrotor configuration used in this work is shown in table 1.

Table 1. Quadrotor configuration.

Parameters		***Value***	***Unit***
Mass	***m***	0.65	Kg
inertia	***I***	diag(7.5 7.5 13) 10^{-3}	kg.m^2
Rotor inertia	***Jr***	$6 * 10^{-5}$	kg.m^2
Gravity	***g***	9.81	m/s^2
Thrust coefficient	***b***	$3.13 * 10^{-5}$	--
Drag coefficient	***d***	$7.5 * 10^{-7}$	--

III. Quadrotor Control

In this section we present the control design of the quadrotor attitude using the recursive backstepping controller. The BS control design will be presented in detail for the Roll motion, while Pitch and Yaw control can be calculated by the same method.

1. Quadrotor model in the state space:

The state variable of the attitude subsystem can be written as follows:

$$X^T = \left[\varphi, \dot{\varphi}, \theta, \dot{\theta}, \Psi, \dot{\Psi}\right]^T = [x_1, x_2, x_3, x_4, x_5, x_6]^T. \quad (15)$$

Replacing the state variable in (13), then the attitude subsystem can be rewriting in the state space according to the following form:

$$\dot{X} = \begin{pmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\end{pmatrix} = \begin{pmatrix}x_2\\ x_4x_6a_1 - c_1\Omega x_4 + b_1U_2.\\ x_4.\\ x_2x_6a_2 + c_2\Omega x_2 + b_2U_3.\\ x_6.\\ x_2x_4a_3 + b_3U_4.\end{pmatrix} \qquad (16)$$

With

$$a_1 = \frac{I_y - I_z}{I_x} \qquad b_1 = \frac{1}{I_x} \qquad c_1 = \frac{J_r}{I_x}$$
$$a_2 = \frac{I_z - I_x}{I_y} \qquad b_2 = \frac{1}{I_y} \qquad c_2 = \frac{J_r}{I_y}$$
$$a_3 = \frac{I_x - I_y}{I_z} \qquad b_3 = \frac{1}{I_z} \qquad c_3 = \frac{J_r}{I_z}$$

2. Backstepping control design for Roll motion:

The Roll motion is mapped by the following subsystem.

$$\dot{x}_1 = x_2 \qquad (17)$$
$$\dot{x}_2 = x_4x_6a_1 - c_1x_4\Omega + b_1U_2. \qquad (18)$$

The control with the backstepping method is conceptualized in two steps:

In the ***first Step:*** we define the tracking-error related to x_1 and its time derivative.

$$\begin{cases} e_1 = x_{1d} - x_1. & (18)\\ \dot{e_1} = \dot{x}_{1d} - x_2. & (19)\end{cases}$$

The lyapunov function and its time derivative are described as follow:

$$\begin{cases} V_1(e_1) = \frac{1}{2}e_1^2 & (20)\\ \dot{V}_1(e_1) = e_1\dot{e_1} = e_1(\dot{x}_{1d} - x_2). & (21)\end{cases}$$

in order to satisfy the lyapunov stabilization condition ($\dot{V}_1(e_1) < 0$) we introduce a novel virtual control x_2 according to (22), its time derivative is defined in (23).

$$x_2 = \dot{x}_{1d} + k_1e_1. \qquad (22)$$
$$e_2 = x_2 - x_{2d} = x_2 - \dot{x}_{1d} - k_1e_1. \qquad (23)$$

k_1 is a positive constant.
the derivative of the lyapunov function is then:

$$\dot{V}_1(e_1) = -k_1e_1^2. \qquad (24)$$

In the ***second step:*** we consider the novel lyapunov function described by:

$$V_2(e_1, e_2) = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \qquad (25)$$

The time derivative calculation of the augmented lyapunov function is presented below:

$$\dot{V}_2(e_1, e_2) = e_1\dot{e_1} + e_2\dot{e_2}. \qquad (26)$$

$$\dot{V}_2(e_1, e_2) = e_1(-e_2 - k_1e_1) + e_2(\dot{x}_2 - \ddot{x}_{1d} - k_1\dot{e}_1) = -e_1e_2 - k_1e_1^2 + e_2\dot{x}_2 - e_2(\ddot{x}_{1d} - k_1(e_2 + k_1e_1)) \qquad (27)$$

The stabilization of the whole system, Roll subsystem, is obtained by introducing a second positive constant k_2

$$U_2 = \frac{1}{b_1}[e_1 - a_1x_4x_6 + c_1\Omega x_4 + \ddot{x}_{1d} - k_1(e_2 + k_1e_1) - k_2e_2] \quad (28)$$

Equation (27) is then: $\dot{V}_2(e_1, e_2) = -k_1e_1^2 - k_2e_2^2$ (29)

By following the same steps for Pitch and Yaw subsystem, we obtain the full attitude backstepping control expressed below:

$$U_2 = \frac{1}{b_1}[e_1 - x_4x_6a_1 + c_1x_4\Omega + \ddot{x}_{1d} - k_1(e_2 + k_1e_1) - k_2e_2] \quad (30)$$

$$U_3 = \frac{1}{b_2}[e_3 - x_2x_6a_2 - c_2x_2\Omega + \ddot{x}_{3d} - k_3(e_4 + k_3e_3) - k_4e_4] \quad (31)$$

$$U_4 = \frac{1}{b_3}[e_5 - x_2x_4a_3 + \ddot{x}_{5d} - k_5(e_6 + k_5e_5) - k_6e_6]. \quad (32)$$

With $e_i = x_{id} - x_i$. (33)

$e_{i+1} = x_{i+1} - x_{(i+1)d} = x_{(i+1)} - \dot{x}_{id} - k_ie_i$. (34)

e_i, i=1,2,3 and K_i, i=1,2,3,4,5,6 are respectively the tracking-error and the lyapunov positive constant.

IV. Optimization

The control response quality depends on the internal parameters of the BS approach. The only imposed condition when choosing these parameters is that they are positives. A keen effort should be invested to identify the correct BS parameters manually. However, an improper choose of these parameters can degrade the performances of the whole system (response time, overshoot as well as the steady state error) as shown in fig. 2 and fig. 3. Sometimes, the effect of the improper choose can push away the quadrotor from its stability domain, and leads to undesirable behaviors as shown in fig.4. An optimization algorithm can help to reduce all these drawbacks by selecting the optimal settings of the adopted control technique.

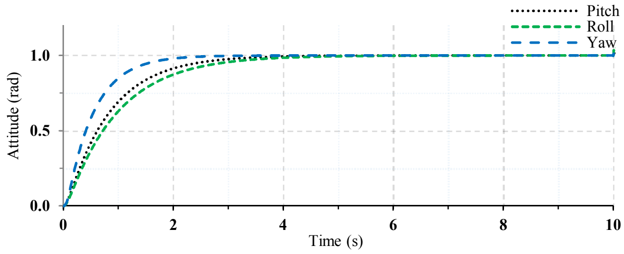


Figure 2: Step response with improper backstepping parameters.

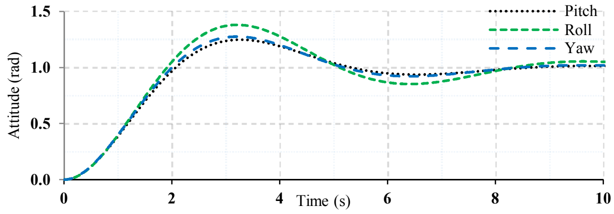


Figure 3: Step response with improper backstepping parameters.

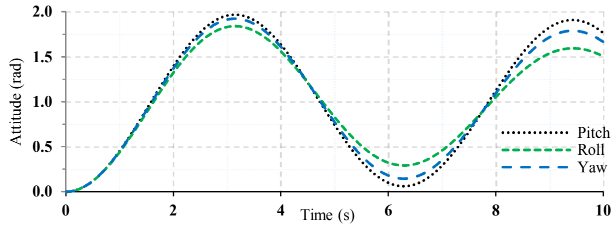


Figure 4: Step response with improper backstepping parameters.

The present study concerns the optimization of the BS parameters using an intelligent Ant Colony Optimization algorithm.

A. *Ant Colony Optimization (ACO):*

1. *State of art*

In nature, foraging is an optimization tasks, where animals try to get maximum of food by consuming the minimum of energy, this can be achieved in ant colony by finding the shortest path between the nest and the food sources. ACO algorithm represents the mathematical formulation of ants behavior. Developed for the first time by Marco Dorigo in 1992 [16], Inspired by the impressive cooperative contribution of ants when choosing the shortest way in food research missions.

ACO technique is based on a graph representation consisting of N nodes connected by M arcs. By its nature, ants prefer to follow the shortest arcs rich with pheromone trails, this is described by a probability that depends on the concentration of the pheromone trails $\tau_{i,j}$, which decreases by the effect of evaporation, and the visibility value $\eta_{i,j}$ as shown in the following equation.

$$p_{i,j}^k = \begin{cases} \frac{(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}{\sum_{f \epsilon N_i^k}(\tau_{i,f})^{\alpha}(\eta_{i,f})^{\beta}} & j \epsilon N_i^k \\ 0 & Otherwise \end{cases} \quad (35)$$

Where α and β are positives constant controlling the influence of the pheromone trails concentration and the visibility value on the probability of ant decision, N_i^k is the neighborhood of ant k when existing in node i.

The visibility value of node j by ant k is described by:

$$\eta_{i,j} = {}^{1}/_{d_{i,j}} \quad (36)$$

Where $d_{i,j}$ is the distance between node i and j.

In the perfect case, when an arc (i, j) is traversed by ants, the amount of pheromone changes according to the following equation:

$$\tau_{i,j} = \tau_{i,j}^{prev} + \sum_{k=1}^{n} \Delta\tau_{i,j}^k. \quad (37)$$

Where $\tau_{i,j}^{prev}$ is the amount of pheromone trails previously existing in arc (i, j), n is the total number of ants and $\Delta\tau_{i,j}^k$ is the amount of pheromone deposit on arc(i, j) by ant k defined as follow:

$$\Delta\tau_{i,j}^k = \begin{cases} \frac{Q}{L^k} & if\ ant\ k\ uses\ arc\ (i,j)\ in\ its\ tour \\ 0 & Otherwise \end{cases} \quad (38)$$

Otherwise, the amount of pheromone does not keep, but evaporate after the movement of ants from the current node to the next node. Then the pheromone concentration became:

$$\tau_{i,j} = (1-\sigma).\tau_{i,j}^{prev} + \textstyle\sum_{k=1}^{n} \Delta\tau_{i,j}^{k}. \qquad (39)$$

Where σ is the evaporation rate.

The accuracy of the adopted algorithm was tested using several test functions before moving to the principal goal of this study which is optimizing the controller parameters, the cost function used for this purpose, is cited in the following section.

2. Cost function

The commonly used cost functions based on the integral error are cited below:

$\boldsymbol{IAE} = \int_0^\infty |e(t)|dt.$ Integral Absolut Error

$\boldsymbol{ITAE} = \int_0^\infty t|e(t)|dt.$ Integral Time multiply Absolut Error.

$\boldsymbol{ISE} = \int_0^\infty e^2(t)dt.$ Integral Square Error.

$\boldsymbol{ITSE} = \int_0^\infty te^2(t)dt.$ Integral Time multiply Square Error.

Generally, the selection of the optimal control parameters refers to the minimization of the cost function. So, the adjusted parameters are those corresponding to the minimum value of the cost function. In our case, the cost function is designed to achieve two main objectives: the minimization of the ISE, to ensure fast response, and the minimization of the overshoot, to obtain smoot response. Then, the cost function is chosen to be as follow:

$$\text{Cost function} = \alpha.ISE + \beta.overshoot. \qquad (40)$$

The total cost function for the quadrotor attitude is defined as follows:

$$\textbf{Total Cost function} = \alpha_\varphi\, ISE_\varphi + \beta_\varphi.\text{overshoot}_\varphi + \alpha_\theta\, ISE_\theta + \beta_\theta.\text{overshoot}_\theta + \alpha_\Psi\, ISE_\Psi + \beta_\Psi.\text{overshoot}_\Psi \quad (41)$$

Where α_x and β_x are respectively the weighting constants for the Integral Square Error (ISE) and the overshoot, in this case α_x and β_x which define the priority of each objective function are equal for the three motions (Pitch, Roll and Yaw)

V. Simulation and Results

To test the accuracy of the proposed controller, ACO tuned BS, a Simulink model was developed in MATLAB interface. And The block diagram showing the core of the offered method is illustrated in fig. 5.

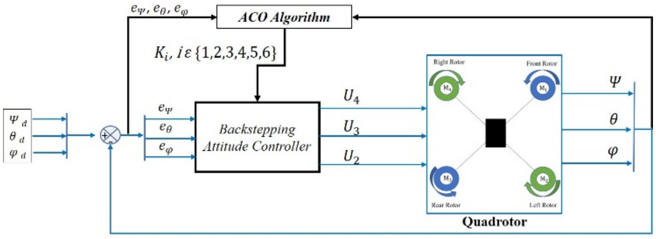


Figure 5: Simulink model.

The ACO algorithm is based on probabilistic operators, that's why results of the algorithm changes from one execution to another, to deal with that, we repeat the algorithm execution 20 time to get more accurate results and we keep the best one at the end. Six parameters have to be adjusted using the present algorithm. The setpoint angles are $[\varphi, \theta, \Psi]^T = [30, 30, 30]^T$ (deg), with an initial configuration set to zero.

The adjusted parameters of the BS controller are given below.

Table 2: Optimal backstepping parameters.

K_1	K_2	K_3	K_4	K_5	K_6
24,6795	18,0353	18,9353	24,4699	15,7370	21,5696

The convergence of the ACO algorithm is depicted in figure 6.

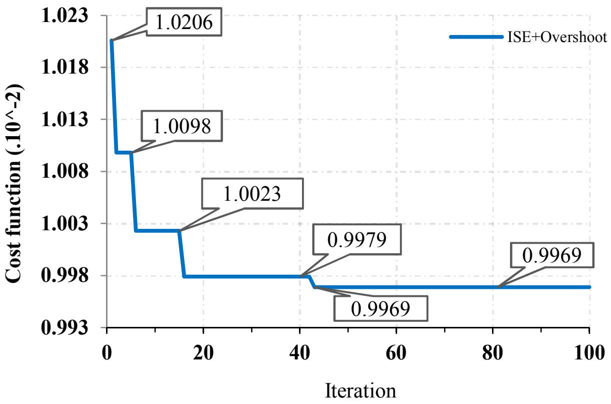


Figure 6: Cost function convergence for the quadrotor attitude.

The ACO algorithm ensures the finite time convergence and leads to a good value of the cost function. As shown in fig. 6, the best value obtained using ACO is 0.9969*10^(-2) reached in the iteration 43.

The following figure shows the simulation results for the Roll, Pitch and Yaw angles using the adjusted parameters.

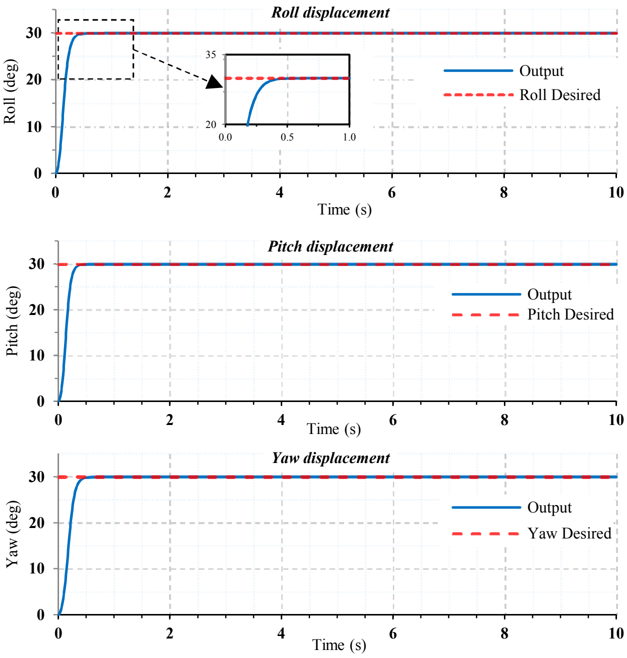


Figure 7: Roll, Pitch and Yaw displacement.

It can be clearly seen that the simulation results for the rotational subsystem using ACO tuned BS are characterized by a quickly response, does not exceed 0.5 s for the three angles, with zero overshoot.

To prove the sovereignty of the proposed method, two numerical experiments were performed. The first concerns the problem of attitude stabilization, while the second investigates the problem of path tracking. For the sake of comparison, a PID and smoothed sliding mode controller are also given.

A. *Hovering mode*

We investigate in this simulation the quadrotor attitude stabilization problem, this simulation is carried out during 10 s with the following initial configuration $[\varphi, \theta, \Psi]^T = [10\,,10\,,10]^T$. Here the controller has to stabilize the quadrotor at the given altitude 1m, and to set the quadrotor orientation to zero degree.

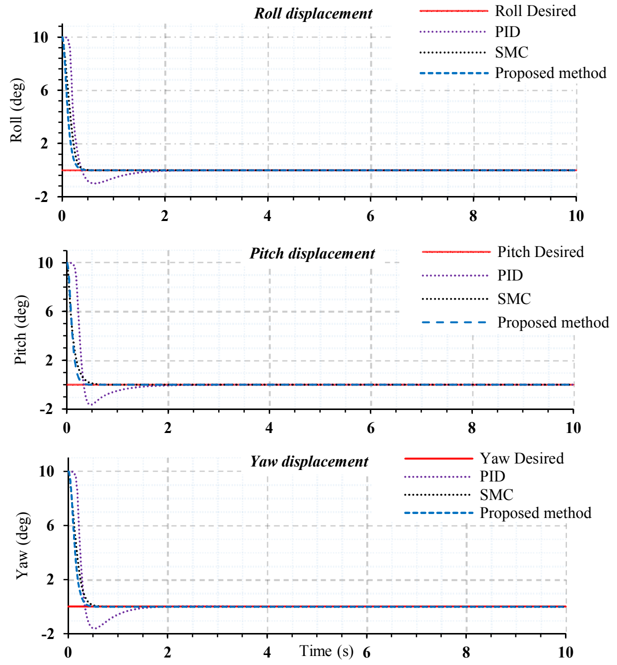


Figure 8: Roll, Pitch and Yaw movement stabilization

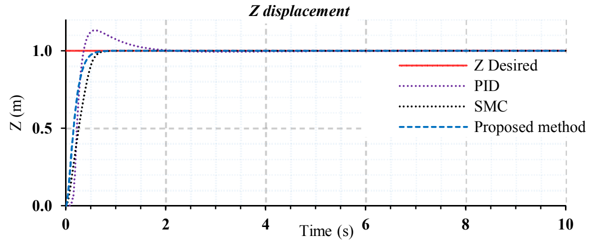


Figure 9: Altitude stabilization

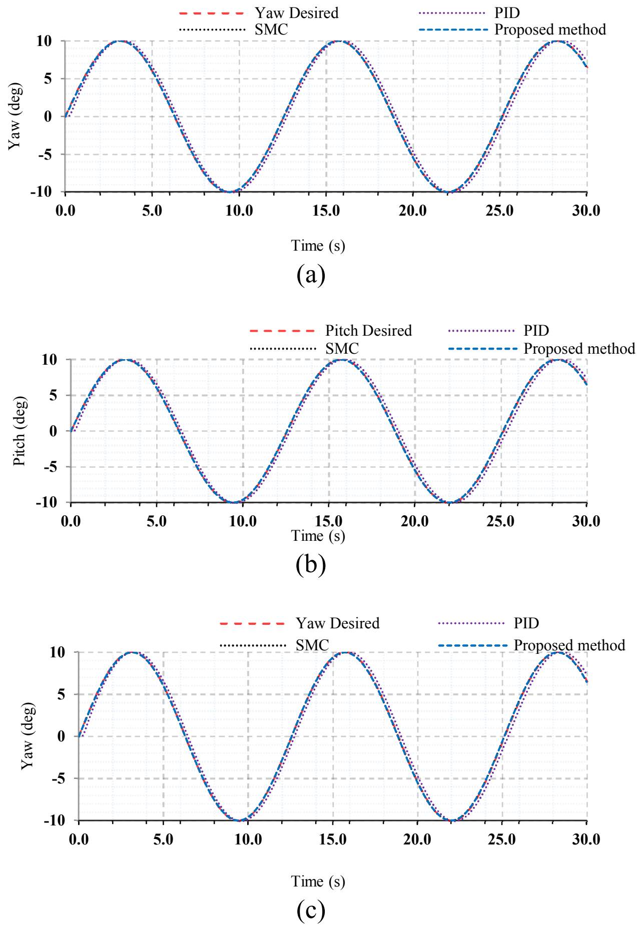
As results of this simulation, the three controllers behave well, with a superiority of the offered method which present the best response, in the other hand the sliding mode technique shows better results than the PID controller which presents a considerable overshoot. The proposed controller can stabilize the quadrotor quickly with neglectable steady state error.

Note here that the sliding mode controller used for the comparison purpose is based on a hyperbolic tangent function in order to reduce the inherently chattering phenomenon and a proportional plus constant reaching law (PPCRL) is adopted to improve the response time [17], [18].

B. *Tracking mode*

The second simulation corresponded to a path tracking problem, the predefined trajectory is a sinusoidal function with a period of T=12.5 s, it is expressed as follows:

$$\begin{bmatrix}\varphi_d\\ \theta_d\\ \Psi_d\end{bmatrix} = \begin{bmatrix}10\,.\sin\left(\frac{2\pi}{T}\right)t\\ 10\,.\sin\left(\frac{2\pi}{T}\right)t\\ 10\,.\sin\left(\frac{2\pi}{T}\right)t\end{bmatrix} \qquad (42)$$



(a)

(b)

(c)

Figure 10: Quadrotor attitude tracking using OBS, (a) roll displacement, (b) pitch displacement, (c) yaw displacement.

It can be seen that the quadrotor equipped with an optimized backstepping controller can track the predefined trajectory with high accuracy, the same for the sliding mode control which give satisfactory results, also the PID controller can offer to the quadrotor the ability of path tracking, but it still limited by the appearance of steady state error.

VI. Conclusion

In this paper, a nonlinear dynamic model was presented for an unmanned aerial vehicle (represented by a quadrotor) based on the Newton Euler methodology. The mathematical formulation of the backstepping controller was described in detail for the quadrotor attitude, then, the backstepping parameters optimization was developed using the Ant Colony

Optimization algorithm. Simulation results demonstrate that the controller parameters obtained using ACO algorithm were successfully selected and provide accurate-superior results compared to the PID and SMC techniques. Future works will focus on the quadrotor control design under external disturbances. Also, the design of the quadrotor mechanical model as well as the development of a quadrotor flight simulator will be addressed.

VII. ACKNOWLEDGMENTS

The Authors would like to thank gratefully CNRST-Maroc for the financial support.

VIII. REFERNCES

[1] T. Luukkonen, "Modelling and control of quadcopter," *Indep. Res. Proj. Appl. Math. Espoo*, vol. 22, 2011.

[2] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2004, vol. 3, pp. 2451–2456 vol.3, doi: 10.1109/IROS.2004.1389776.

[3] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 2013, pp. 1–6.

[4] S. Bharat, A. Ganguly, R. Chatterjee, B. Basak, D. K. Sheet, and A. Ganguly, "A Review on Tuning Methods for PID Controller," *Asian J. Converg. Technol. AJCT*, 2019.

[5] E. Okyere, A. Bousbaine, G. T. Poyi, A. K. Joseph, and J. M. Andrade, "LQR controller design for quad-rotor helicopters," *J. Eng.*, vol. 2019, no. 17, pp. 4003–4007, 2019, doi: 10.1049/joe.2018.8126.

[6] C. S. Subudhi and D. Ezhilarasi, "Modeling and Trajectory Tracking with Cascaded PD Controller for Quadrotor," *Procedia Comput. Sci.*, vol. 133, pp. 952–959, Jan. 2018, doi: 10.1016/j.procs.2018.07.082.

[7] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2005, pp. 2247–2252.

[8] T. Madani and A. Benallegue, "Backstepping Control for a Quadrotor Helicopter," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3255–3260, doi: 10.1109/IROS.2006.282433.

[9] M. Navabi and H. Mirzaei, "Robust Optimal Adaptive Trajectory Tracking Control of Quadrotor Helicopter," *Lat. Am. J. Solids Struct.*, vol. 14, no. 6, pp. 1040–1063, Jun. 2017, doi: 10.1590/1679-78253595.

[10] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Modelling and stabilizing control laws design based on backstepping for an UAV type-quadrotor," *IFAC Proc. Vol.*, vol. 40, no. 15, pp. 245–250, 2007.

[11] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. M. Smailzadeh, "Attitude control of a quadrotor with optimized PID controller," *Intell. Control Autom.*, vol. 4, no. 03, p. 335, 2013.

[12] D. Mercado, P. Castillo, R. Castro, and R. Lozano, "2-Sliding Mode Trajectory Tracking Control and EKF Estimation for Quadrotors.," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 8849–8854, 2014.

[13] H. K. Tran and T. N. Nguyen, "Flight Motion Controller Design using Genetic Algorithm for a Quadcopter," *Meas. Control*, vol. 51, no. 3–4, pp. 59–64, Apr. 2018, doi: 10.1177/0020294018768744.

[14] İ. C. Dikmen, T. KARADAĞ, and C. YEROĞLU, "Multi-Parameter Optimization of Sliding-Mode Controller for Quadcopter Application," *Anatol. Sci.-Bilgi. Bilim. Derg.*, vol. 3, no. 1, pp. 14–28.

[15] M. Basri, M. Ariffanan, K. A. Danapalasingam, and A. R. Husain, "Design and optimization of backstepping controller for an underactuated autonomous quadrotor unmanned aerial vehicle," *Trans. FAMENA*, vol. 38, no. 3, pp. 27–44, 2014.

[16] M. Dorigo, "Optimization, learning and natural algorithms," *PhD Thesis Politec. Milano*, 1992.

[17] W. Gao and J. C. Hung, "Variable structure control of nonlinear systems: A new approach," *IEEE Trans. Ind. Electron.*, vol. 40, no. 1, pp. 45–55, 1993.

[18] B. Brahmi, M. H. Laraki, A. Brahmi, M. Saad, and M. H. Rahman, "Improvement of sliding mode controller by using a new adaptive reaching law: Theory and experiment," *ISA Trans.*, 2019.