PID controllers design for a quadrotor system using teaching learning based optimization

NAIMA BOUHABZA[a*], KAMEL KARA[a], MOHAMED LAID HADJILI[b]
[a]Labortoire des systèmes électriques et télécommande, Faculté de Technologie, Université Blida, Blida, ALGERIA
[b]Haute Ecole Bruxelles-Ecole Supèrieure d'Informatique B.P 270, Route de Soumaa , 67 Rue Royale, Bruxelles, BELGIUM

Abstract: Optimized Proportional Integral Derivative controllers are designed to control the translational and rotational motions of a quadrotor system with six degrees of freedom. The teaching learning based optimization algorithm is used to obtain the proportional, integral and derivative gains of six PID controllers so that the integral time absolute error criterion is minimized. The control objective, is to enforce the horizontal position, altitude and yaw angle of the quadrotor to track their desired reference trajectories while stabilizing its roll and pitch angles. The efficiency and the control performance of the proposed scheme are demonstrated through numerical simulation and compared with those of the PID controllers designed using genetic algorithm, the sliding mode control and other control techniques proposed in the literature. The simulation study shows the good performance of the proposed control scheme in terms of transient response characteristics, tracking accuracy and disturbance rejection.

Key-Words: Quadrotor,TLBO, PID Control, Optimization, Unmanned aerial vehicles

Received: February 27, 2021. Revised: August 5, 2021. Accepted: August 17, 2021. Published: August 30, 2021.

1 Introduction

A quadrotor is a kind of Unmanned Aerial Vehicles (UAV) and it belongs to the class of Vertical Take Off and Landing (VTOL), with four rotors usually placed at the ends of a cross, two rotors rotate in clockwise and the others rotate in anticlockwise. The quadrotor is an under-actuated system with six degrees of freedom, three translational and three rotational motions, and only four inputs. The quadrotor dynamic is highly non-linear and strongly coupled. For this reason; the task of control it is a fundamentally difficult and interesting problem. The quadrotor application range is wide and covers several fields: military, civil, commercial, agriculture and research. It can replace human in: dangerous missions, surveillance, and rescue. This kind of vehicles is characterized by its: small size, simple construction, high maneuverability and low maintenance costs. The only disadvantage is the high consumption of electrical energy due to the four motors.

Known as an adequate platform to design and test the control laws, several control approaches have been developed to control the quadrotor system, including classical techniques such as : PID controller [1-4], Linear Quadratic Regulator (LQR) [1],[5], and sliding mode control [6-10], intelligent control methodologies like: neural network control [11-13], fuzzy logic control [14-16] , and predictive control [17-20] , and hybrid control systems such as: fuzzy sliding mode control using backstepping approach [21], nonlinear adaptive controller using backstepping technique mixed with neural networks[22], adaptive sliding mode controller based on neural networks and backstepping [23] and PID controller based on artificial neural networks [24].

The PID controller is a very old technique that is still used in many industrial applications. The success of the PID controller is due to its simple structure and ease of its implementation. In[1], the authors have presented a comparison between PID controller and linear quadratic regulator technique, where a linearized model of the quadrotor system was used to design the PID and LQ controllers. A PD control technique has been proposed in [25] which include a vision system to estimate the pose

of the helicopter. It has been demonstrated that a quadrotor fitted with a camera can detect a pattern on the ground and navigate around it at a desired altitude. A comparison study, of the attitude control of the quadrotor, between PD, inverse dynamic, Backstepping and sliding mode controllers has been presented in [26]. The PID controller performance of the quadrotor has been experimentally analyzed in [3], and it has been shown that the PID controllers can robustly stabilize the quadrotor. The authors of the work presented in [4] have used three different structures of the PID controller to control the orientation of the quadrotor, and have experimentally confirmed the effectiveness of the PID control scheme. In another study, a fuzzy-PID controller has been suggested to stabilize the quadrotor and compared to PD and fuzzy logic controllers [27].

Although the PID control technique has been widely used to control the quadrotor system, the most difficult side for a PID control scheme is the attainment of its best parameters values. In the literature, there are several techniques for obtaining the parameters of a PID controller: the classical techniques as Ziegler Nichols, heuristic optimization techniques like Genetic algorithm (GA) and Particle Swarm optimization (PSO) and the artificial intelligence techniques such as neural networks and fuzzy logic. In order to enhance the transient and the steady state responses of the quadrotor system, the genetic algorithm was used to find the optimal gains of the used PID controllers [28]. Genetic algorithm has been also used to obtain the PID parameters to stabilize the H-shaped racing quadcopter and to attain the desired altitude and orientation [29]. The PID control technique, the sliding mode control approach and the genetic algorithm have been combined to control the quadrotor system [30]. The genetic algorithm was used to optimize the control parameters to achieve a good trade-off between faster time response and consumed energy. In [31], a nonlinear PID control structure to control the translational and rotational motions of a quadrotor system has been designed using genetic algorithm. Particle swarm optimization method, as well as other evolutionary algorithms, has been also used, in several works dealing with the quadrotor control problem, to tune the parameters of a PID controller[32,33].

In this paper, a PID control scheme is developed to stabilize the translational and rotational motions of the quadrotor. The PID controllers' gains are obtained by minimizing the Integrated Time Absolute Error (ITAE) criterion, using the Teaching Learning Based Optimization (TLBO) algorithm. The transient response characteristics and the robustness propriety, with respect to measurement noise and external disturbances, of the proposed control scheme are investigated and compared with those of the PID controller designed using genetic algorithm [31], the sliding mode control [21], the conventional PID and LQR techniques proposed in[37].

The remainder of the paper is organized as follows: Section 2 presents the mathematical model of the quadrotor where the motors dynamic is considered. Section 3 describes the Teaching Learning Based Optimization algorithm. Section 4 explains the controller structure. Simulation results are presented in section 5. Some conclusions are driven in section 6.

2 Quadrotor dynamic model:

Fig.1 presents a quadrotor configuration, that consists of an inertial frame G, a body fixed frame B, and two pairs of motors (1,3) and (2,4), each of them can generate independent thrusts $T_i, i = 1,2,3,4.$

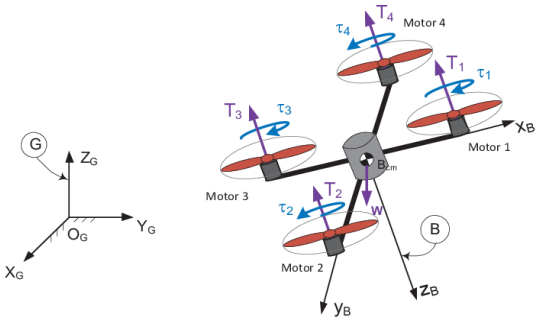


Fig.1: Quadrotor aircraft scheme

The vertical movement of the quadrotor is achieved by increasing or decreasing the speed of the four motors. The displacement along the X axis is obtained by increasing or decreasing the speeds of the motors (1) and (3) (pitch motion). The displacement along the Y axis is obtained by increasing or decreasing the speeds of the motors (2) and (4) (roll motion). The difference in counter-torque between each pair of motors causes the yaw motion.

Taking into account frictions due to the aerodynamic torques, drag forces, gyroscopic effects and the rotors dynamic, the translational and the rotational dynamics of the quadrotor, using the Newton_Euler formalism, are given by [21]:

$$\begin{cases} \ddot{x} = \frac{1}{m}\{(U_x)U_1 - K_{ftx}\dot{x}\} \\ \ddot{y} = \frac{1}{m}\{(U_y)U_1 - K_{fty}\dot{y}\} \\ \ddot{z} = \frac{1}{m}\{(cos\varphi cos\theta)U_1 - K_{fty}\dot{x}\} - g \\ \ddot{\varphi} = \frac{1}{I_x}\{\dot{\theta}\dot{\psi}(I_y - I_z) - K_{fax}\dot{\varphi}^2 - J_r\bar{\Omega}\dot{\theta} + dU_2\} \\ \ddot{\theta} = \frac{1}{I_y}\{\dot{\varphi}\dot{\psi}(I_z - I_y) - K_{fay}\dot{\theta}^2 + J_r\bar{\Omega}\dot{\varphi} + dU_3\} \\ \ddot{\psi} = \frac{1}{I_z}\{\dot{\theta}\dot{\varphi}(I_x - I_y) - K_{faz}\dot{\psi}^2 + bU_4\} \end{cases} \quad (1)$$

$$\begin{cases} U_x = cos\varphi sin\theta cos\psi + sin\varphi sin\psi \\ U_y = cos\varphi sin\theta sin\psi - sin\varphi cos\psi \end{cases} \quad (2)$$

$U_i, i = 1,2,3,4$: are the control inputs of the system and are proportional to the angular velocities of motors.

$$\begin{cases} U_1 = b\omega_1^2 + b\omega_2^2 + b\omega_3^2 + b\omega_4^2 \\ U_2 = b\omega_4^2 - b\omega_2^2 \\ U_3 = b\omega_3^2 - b\omega_1^2 \\ U_4 = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \\ \bar{\Omega} = \omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 \end{cases} \quad (3)$$

From the quadrotor dynamic equations (1), the nonholonomic constraints expressions are given by:

$$\begin{cases} \varphi_{de} = arcsin(U_x sin\psi_{de} - U_y cos\psi_{de}) \\ \theta_{de} = arcsin\left(\frac{U_x}{cos\varphi cos\psi_{de}} - \frac{sin\varphi sin\psi}{cos\varphi cos\psi_{de}}\right) \end{cases} \quad (4)$$

The motors dynamic is given by the following equations [21]:

$$\begin{cases} V = ri + L\frac{di}{dt} + k_e\,\omega \\ k_m = J_r + C_s + k_r\omega^2 \end{cases} \quad (5)$$

Then the model chosen for the rotors is as follows [21]:

$$\begin{cases} \dot{\omega}_i = b_m V_i - \beta_0 - \beta_1\omega_i - \beta_2\omega_i^2, \\ i = [1,2,3,4] \\ \beta_0 = \frac{C_s}{J_r}, \beta_1 = \frac{k_e k_m}{rJ_r}, \beta_2 = \frac{k_r}{J_r}, \beta_0 = \frac{k_m}{rJ_r} \end{cases} \quad (6)$$

The description of the parameters model is given in Table 1.

Table 1: Description of the model parameters

Parameters	Description
$[x\ y\ z]$	linear position of the quadrotor
$[\varphi\ \theta\ \psi]$	the angular position of the quadrotor
$[\omega_1, \omega_2, \omega_3, \omega_4]$	The motors speeds vector.
U_1	Vertical thrust
U_2	Pitching moment
U_3	Rolling moment
U_4	Yawing moment
$\bar{\Omega}$	The overall residual propeller angular speed.
m	Total mass
g	Gravitational force
$[I_x, I_y, I_z]$	Moment of inertia vector
$[K_{fax}, K_{fay}, K_{faz}]$	Aerodynamic friction coefficient
$[K_{ftx}, K_{fty}, K_{ftz}]$	Drag force coefficient
b	Thrust coefficient
d	Drag coefficient
V	Motor input
k_e	Electrical torque constant
k_m	mechanical torque constant
k_r	load constant torque
r	motor internal resistance and
C_s	Solid friction
J_r	Motor inertia

3 Teaching learning based optimization algorithm

The TLBO algorithm is based on the teaching-learning process that evaluates the teacher's influence on the performance of learners in the class [34-36]. The algorithm defines two basic modes of learning: (i) learning from a teacher (known as the teacher phase) and (ii) learning from other learners (known as learner phase). The various subjects provided to the population of learners are considered as the different design variables in the optimization problem. The learner's result is the fitness value of the optimization problem. The teacher is the best solution in the entire population; it corresponds to the best value of the objective function. This algorithm requires only two control parameters: the population size and the maximum

number of iterations, which hugely simplify its implementation.
The first step is the initialization of the algorithm parameters: population size (n), number of design variables (m) and their upper and lower limits (u,l), the maximum number of iterations (K_max). The initial population is randomly generated according to the following equation:

$$X_{ij} = rand * (u_j - l_j) + l_j \quad (7)$$

$$i = 1, \dots, n, \qquad j = 1, \dots, m$$

Where, rand is a random number between 0 and 1.
For each iteration k (k=1,…..,K_max), the teacher and learner phases, are performed as follows:

Teacher phase:

The teacher tries to improve the mean result of the optimization variables and he is considered as a highly learned person.
The following formula is used to calculate the mean result of all students in a given subject:

$$M_j(k) = \sum_{i=1}^{n} \frac{X_{ij}(k)}{n}, \qquad j = 1, \dots, m \quad (8)$$

Then, the variable that gives the best value of the fitness function is considered as a teacher X_bestj (k) in a subject j. The difference between the existing mean result for each subject and the teacher's corresponding result for each subject is calculated as follows:

$$diff_j(k) = rand * \left(X_{bestj}(k) - T_f M_j(k)\right) \quad (9)$$

$$j = 1, \dots, m$$

where T_f is the teaching factor, which can be either 1 or 2. It is given by:

$$T_f = round(rand + 1) \quad (10)$$

After, new solutions $X_{ij}^{new}(k)$ must be generated according to the equation below:

$$X_{ij}^{new}(k) = X_{ij}(k) + diff_j(k) \quad (11)$$

$$i = 1, \dots, n, \qquad j = 1, \dots m$$

If the new solution $X_{ij}^{new}(k)$ gives the best value of the fitness function it is accepted and becomes the current solution $X_{ij}(k)$, otherwise the current solution must be kept unchanged.

Learner phase:

During this phase, the learners enhance their knowledge by interacting with each other. Two learners $X_{Pj}(k)$ and $X_{Qj}(k)$ are randomly selected to interact among themselves.

If $X_{Pj}(k)$ gives better value of the fitness function than $X_{Qj}(k)$, a new solution $X_{Pj}^{new}(k)$ is generated according to the following equation :

$$X_{Pj}^{new}(k) = X_{Pj}(k) + rand * (X_{Pj}(k) - X_{Qj}(k)) \quad (12)$$

Otherwise:

$$X_{Pj}^{new}(k) = X_{Pj}(k) + rand * (X_{Qj}(k) - X_{Pj}(k)) \quad (13)$$

If the new solution $X_{Pj}^{new}(k)$ gives the best value of the fitness function, it is accepted and replaces the current solution. Otherwise the current solution is kept unchanged.

4 Controllers design

4.1 Control structure

The quadrotor's translational and rotational motions are controlled by six PID controllers. Figure 2 shows the control structure, which is composed of two control loops: the inner loop control and the outer loop control.

The equation of the conventional PID controller is given as follows:

$$U_{PID} = k_p e + k_d \dot{e} + k_i \int e dt \quad (14)$$

where e is the error signal, $\dot{e}$ is the error derivative and k_p,k_i, k_d are the proportional, integral and derivative gains of the PID controller, respectively.

Inner Loop Control:

In this part, the input control is available, where the control signal for the altitude, Roll, Pitch and Yaw angles are given as follows:

$$U_k = {k_p}^h e_k + {k_d}^h \dot{e_k} + {k_i}^h \int e_k dt \quad (15)$$

$$k = 1,2,3,4 \ and \ h = z, \varphi, \theta, \psi$$

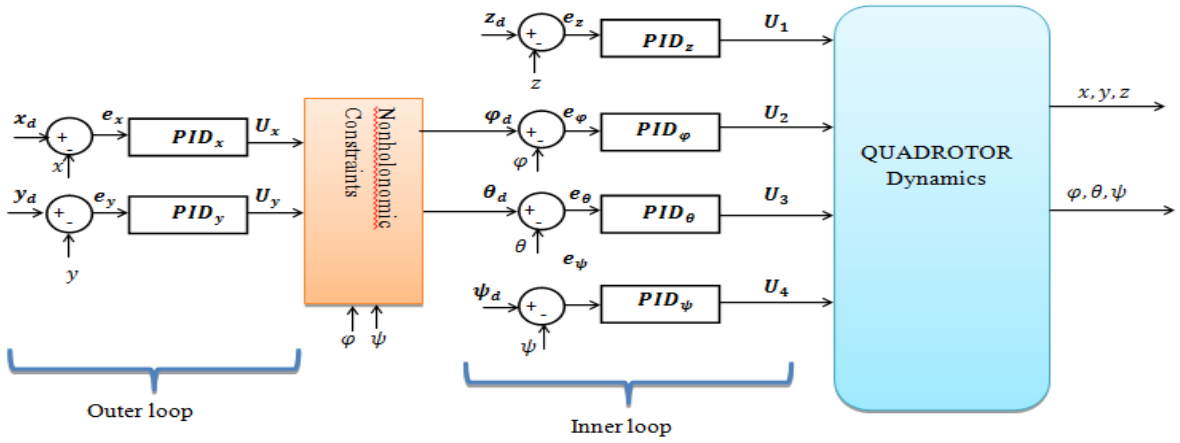


Fig. 2: Control structure of the quadrotor

where, k_p^z, k_d^z, k_i^z are the PID controller parameters of the altitude z, $k_p^\varphi, k_d^\varphi, k_i^\varphi$ are the PID controller parameters of the roll angle φ, $k_p^\theta, k_d^\theta, k_i^\theta$ are the PID controller parameters of the pitch angle θ and $k_p^\psi, k_d^\psi, k_i^\psi$ are the PID controller parameters of the yaw angle ψ. The error signals are given by:

$$\begin{cases} e_1 = z_d - z_m \\ e_2 = \varphi_d - \varphi_m \\ e_3 = \theta_d - \theta_m \\ e_4 = \psi_d - \psi_m \end{cases} \quad (16)$$

z_d, φ_d, θ_d and ψ_d are the desired reference trajectories of the altitude, the roll, the pitch and the yaw angles respectively, and z_m, φ_m, θ_m and ψ_m are the corresponding measured signals.

From the control equations (3) the desired rotational speeds of the motors are calculated according to:

$$\begin{cases} \omega_{1ref} = \sqrt{U_1/4b - U_3/2b - U_4/4d} \\ \omega_{2ref} = \sqrt{U_1/4b - U_2/2b + U_4/4d} \\ \omega_{3ref} = \sqrt{U_1/4b + U_3/2b - U_4/4d} \\ \omega_{4ref} = \sqrt{U_1/4b + U_2/2b + U_4/4d} \end{cases} \quad (17)$$

PID controllers are used to control the rotational speeds of the DC motors according to the bloc diagram of figure 3, where V_i $(i = 1,2,3,4)$ are the voltages applied on motors and ω_{iref} are the desired rotational speeds obtained using equation (17).

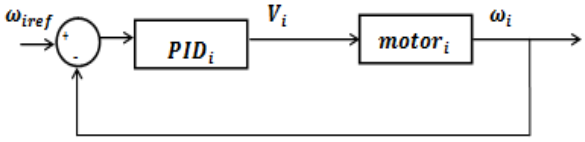


Fig. 3: Control structure of motors

The PID controllers' gains for the motors are given in Table 2.

Table 2: PID parameters values for the DC motors control

	k_p	k_i	k_d
PID_i	1	1	0

Outer Loop Control:

The control system has no real control inputs for the motion in (x, y) plane, so x and y cannot be directly controlled, for this reason the following virtual controllers are designed.

The control signals for the position x and y are given by:

$$U_h = {k_p}^h e_j + {k_d}^h \dot{e}_j + {k_i}^h \int e_j dt \quad (18)$$

$$h = x, y, \quad j = 5,6$$

Where, k_p^x, k_d^x, k_i^x are the PID controller parameters of the position x, and k_p^y, k_d^y, k_i^y are the PID controller parameters of the position y.

The corresponding error signals are given by:

$$\begin{cases} e_5 = x_{de} - x_m \\ e_6 = y_{de} - y_m \end{cases} \quad (19)$$

where, x_d and y_d are the desired reference trajectories and x_m and y_m are the corresponding measured signals.

4.2. Control algorithm

The parameters values of the PID controllers are determined through minimizing the integrated time absolute error criterion using the TLBO algorithm. The ITAE criterion is defined as follows:

$$J = \sum_{i=1}^{6} ITAE_i$$
$$ITAE_i = \int_0^{\infty} t * |e_i(t)| dt \quad (20)$$

In the following $s = 1, \dots, n$, $j = 1, \dots, m$, $k = 1, \dots, K_{max}$, where n is the population size, m is the number of parameters to be optimized, in this case there are 18 parameters that are the gains of the six PID controllers, and K_{max} is the maximum number of iterations of the optimization algorithm.

The fitness function is the criterion J and the used population is given by:

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_n \end{bmatrix} \quad (21)$$

where each element of the matrix $\boldsymbol{X}$ is defined as follows:

$$X_s = [k_{p_s}^x \; k_{d_s}^x \; k_{i_s}^x k_{p_s}^y k_{d_s}^y k_{i_s}^y k_{p_s}^z \; k_{d_s}^z \; k_{i_s}^z \dots \dots$$
$$\dots \dots . k_{p_s}^{\varphi} \; k_{d_s}^{\varphi} k_{i_s}^{\varphi} \; k_{p_s}^{\theta} \; k_{d_s}^{\theta} \; k_{i_s}^{\theta} \; k_{p_s}^{\psi} \; k_{d_s}^{\psi} \; k_{i_s}^{\psi}] \quad (22)$$
$$s = 1, \dots, n$$

To each element X_s of the population $\boldsymbol{X}$ is associated a fitness function J_s defined by equation (20). The different steps of the optimization algorithm are summarized as follows:

Step 0: Select the size of the population (n) and the maximum number of iterations K_{max}, set the number of the optimization parameters ($m = 18$). Specify the upper and lower values of each optimization parameter ($u_j, l_j, \quad j = 1, \dots, m$) and the different reference trajectories.

Step 1: Set the iterations index $k = 1$ and generate a random population $\boldsymbol{X(k)}$ using equation (7).

Step 2: Compute the corresponding control laws $U_i(k), i = 1, \dots, 4$, $U_x(k)$ and $U_y(k)$ using the current population $\boldsymbol{X(k)}$ and equation (14).

Step 3: Determine the system responses to the computed control signals and compute the corresponding errors.

Step 4: For each element $X_s(k)$ of the population $\boldsymbol{X(k)}$, **c**alculate the corresponding value of the fitness function $J_s(X_s(k))$ using equation (20). The element $X_h(k)$ with the minimum value $J_h(X_h(k)), h \in \{1,2,3, \dots, n\}$ of the fitness function is taken as a teacher.

$X_{best}(k) = X_h(k)$ such that:

$$J_h(X_h(k)) \leq J_s(X_s(k)),$$
$$h \in \{1,2, \dots, n\}, s = 1, \dots, n$$

Step 5: Compute the mean result of the learners $M_s(X_s(k))$ using equation (8), the difference values $diff_s(k)$ using equation (9) and the teaching factor T_f using equation (10).

Step 6: Generate the new population $\boldsymbol{X}^{new}(k)$ according to equation (11).

Step 7: Compute the corresponding control laws $U_i(k), i = 1, \dots, 4$, $U_x(k)$ and $U_y(k)$ using the new population $\boldsymbol{X}^{new}(k)$ and equation (14) . Then, determine the system responses to the computed control signals and compute the corresponding errors.

Step 8: Evaluate the fitness $J_s(X_s^{new}(k))$ of each element of $\boldsymbol{X}^{new}(k)$.

If a new solution $X_s^{new}(k),\ s = 1, \dots, n$, gives the small value of the fitness function than the other solutions $X_i(k),\ i = 1, \dots, n$, it is accepted and becomes the current solution, otherwise the current solution is kept unchanged.

Step 9: Randomly choose two solutions $X_P(k)$ and $X_Q(k),\ P, Q \in \{1,2, \dots, n\}, P \neq Q$, and generate new solution $X_P^{new}(k)$ according to equations (12) and (13).

Step 10: Compute the corresponding control laws $U_i(k), i = 1, \dots, 4$, $U_x(k)$ and $U_y(k)$ using the new solution $X_P^{new}(k)$ and equation (14) . Then, determine the system responses to the computed control signals and compute the corresponding errors.

Step 11: Evaluate the fitness $J_P(X_P^{new}(k))$.

If $J_P(X_P^{new}(k)) < J_Q(X_Q(k))$, the new solution $X_P^{new}(k)$ replaces the current solution. Otherwise the current solution is kept unchanged.

Step 12: The best solution is $X_{best}(k) = X_h(k)$ such that:

$$J_h\big(X_h(k)\big) \leq J_s\big(X_s(k)\big),$$

$$h \in \{1,2, \dots, n\}, s = 1, \dots, n$$

Step 13: Increment the iteration index $k = k + 1$. if $k < K_{max}$ evaluate the mean result of the learners $M_s(X_s(k))$ using equation (8), the difference values $diff_s(k)$ using equation (9), the teaching factor T_f using equation (10) and go to step 6. Otherwise, the best solution corresponding to the current sampling time is that of step 12.

For the next sampling time the current population is taken as an initial population and the algorithm is started from step 1.

5 Simulation results

5.1 Simulation parameters

The quadrotor model and the control algorithm, described above, are implemented in MatLab/SIMULINK environment. Table 3 gives the parameters values of the quadrotor model used in the simulation.

Table 3: Parameters of the Quadrotor model

Parameter	Value
m	$0.486\ kg$
g	$9.8\ m/sec^2$
$[I_x, I_y, I_z]$	$(3.8278, 3.8288, 7.6566) * 10^{-3}\ N.m\ /\ rad\ /\ s^2$
$[K_{fax}, K_{fay}, K_{faz}]$	$(5.5670, 5.5670, 6.3540) * 10^{-4}\ N.m\ /\ rad\ /\ s$
$[K_{ftx}, K_{fty}, K_{ftz}]$	$(5.5670, 5.5670, 6.3540) * 10^{-4}\ N.m\ /\ rad\ /\ s$
b	$2.9842 * 10^{-5}\ N.m/\ rad\ /\ s$
d	$2.2320\ * 10^{-7}\ N.m/\ rad\ /\ s$
J_r	$2.8385 * 10^{-5}\ N.m\ /\ rad\ /\ s^2$
β_0	189.63
β_1	6.0612
β_2	0.0122
b_m	280.19

The parameters values of the TLBO algorithm are given in Table 4:

Table 4: Parameters values of the TLBO algorithm

parameter	value
n	10
m	15
$[l_{x,y}\ \ u_{x,y}]$	[0 10]
$[l_z\ \ u_z]$	[0 50]
$[l_{\varphi,\theta,\psi}\ \ u_{\varphi,\theta,\psi}]$	[0 1]
K_{max}	50

$l_{x,y}$, $u_{x,y}$ are the lower and the upper limits of the PID controllers gains for the horizontal position (x, y), l_z , u_z are the lower and the upper limits of the PID controller gains for the altitude z. and $l_{\varphi,\theta,\psi}$, $u_{\varphi,\theta,\psi}$ are the lower and the upper limits of the PID controllers gains for the angles φ, θ and ψ.

The obtained gains of the six PID controllers are given in Table 5.

Table 5: Gains values of the PID controllers using the TLBO algorithm

	k_p	k_d	k_i
x	4,50	2,129	3.011×10^{-5}
y	7.416	5.730	9.272×10^{-5}
z	33.340	19.040	3.807×10^{-5}
φ	0.608	0.111	7.4×10^{-3}
θ	0.234	0.081	8×10^{-3}
ψ	0.741	0.604	2.559×10^{-5}

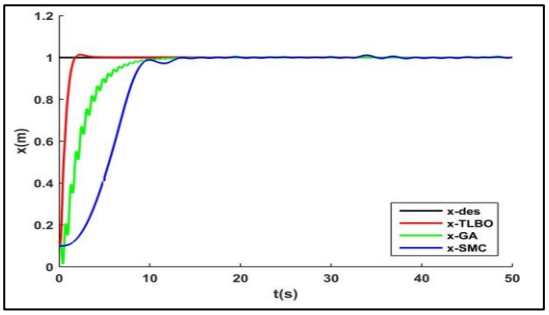
The performance of the proposed control scheme is compared with that of the optimized PID controllers using GA [31], and the sliding mode control strategy proposed in **[**21**].** The parameters values of the optimized PID controllers using GA are taken from [31] and are listed in Table 6.

Table 6: Gains values of the PID controllers using the GA algorithm

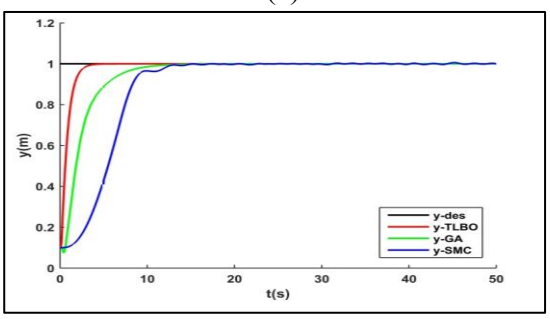
	k_p	k_d	k_i
x	0.28	0.63	2.73×10^{-6}
y	0.36	0.88	1.56×10^{-5}
z	184.02	0.25	103.73
φ	0.88	0.3	0.9
θ	0.62	0.05	0.81
ψ	0.99	0.56	0.49

5.2. Case of step reference trajectories

In this part, a unit step is used as a reference trajectory for the position (x, y), the altitude (z) and the yaw angle ψ of the quadrotor. Figs. 4-12 show the obtained results using the aforementioned control schemes. From these figures, it is clear that the best control performance, namely the tracking accuracy, the speed of response, the overshoot and the smoothness of the control signals, is obtained in the case of the proposed control scheme. It should be noted that the presence of oscillations (chattering phenomenon) on the control signals, in the case of the SMC strategy, leads to poor control performance.

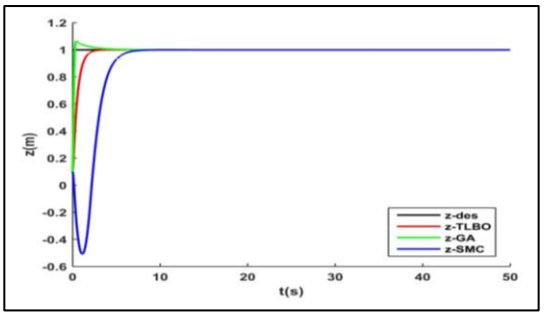


(a)

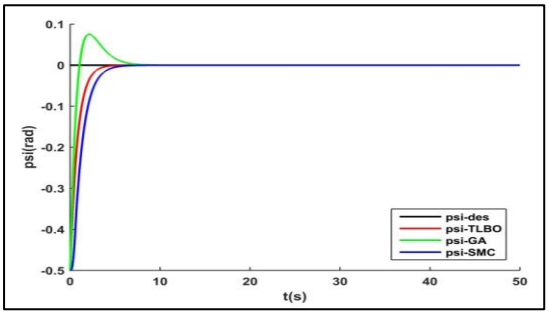


(b)

Fig.4: Horizontal position of the quadrotor ((a) x-postion, (b) y-position)

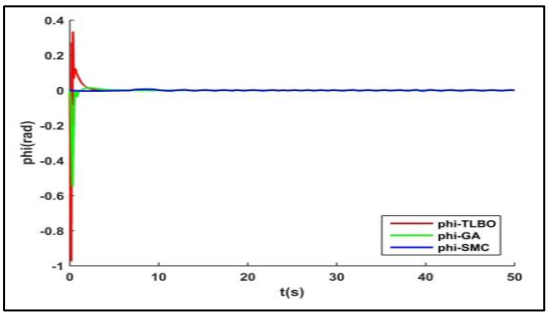


(a)

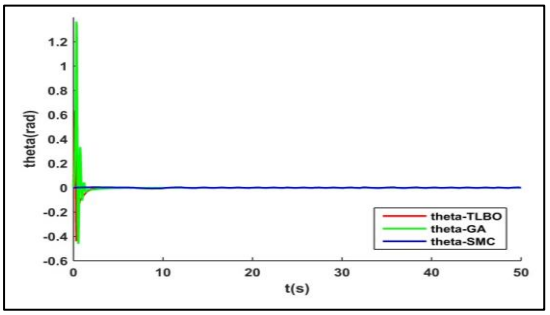


(b)

Fig. 5: Altitude position (a) and yaw angle (b) of the quadrotor

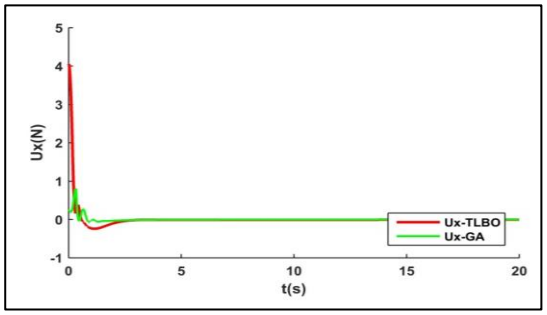


(a)

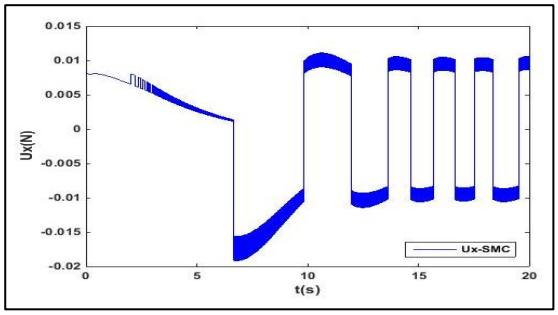


(b)

Fig. 6: Phi angle (a) and theta angle (b) of the quadrotor

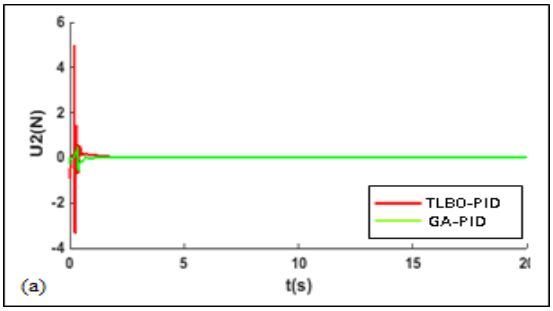


(a)

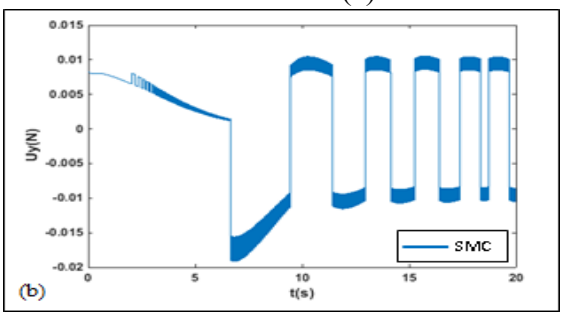


(b)

Fig. 7: Control signals of the x-position ((a) using TLBO and GA, (b) using SMC)

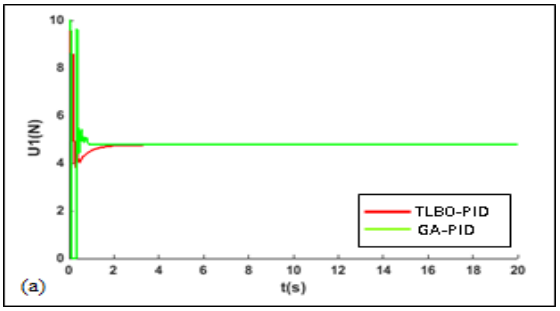


(a)

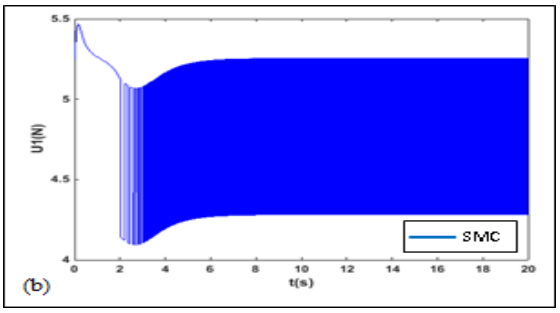


(b)

Fig. 8: Control signals of the y-position ((a) using TLBO and GA, (b) using SMC)

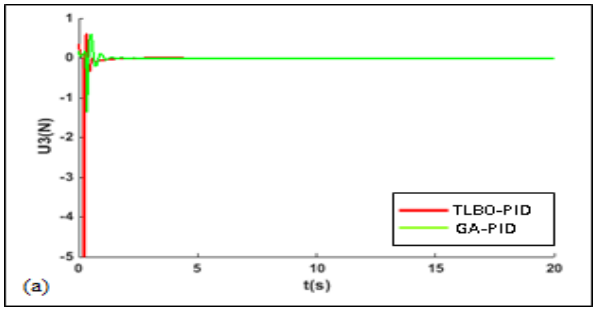


(a)

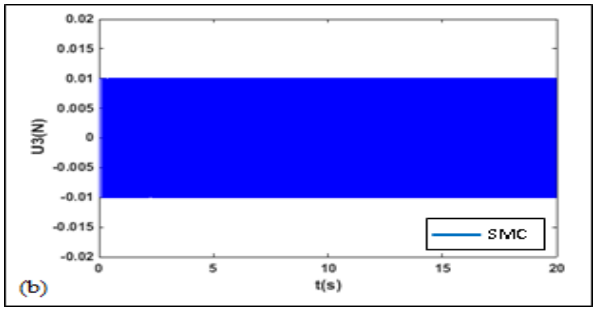


(b)

Fig. 9: Altitude control signals ((a) using TLBO and GA, (b) using SMC)

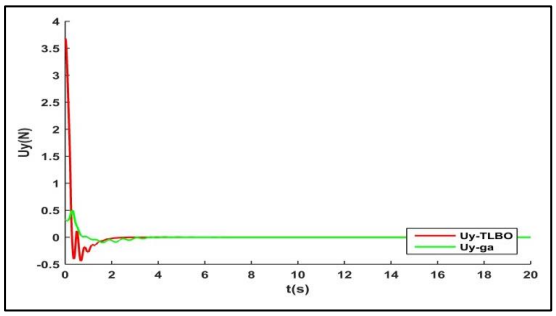


(a)

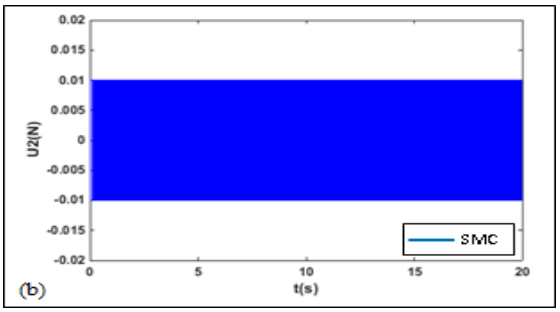


(b)

Fig. 11: Control signals of the θ-angle ((a) using TLBO and GA, (b) using SMC)

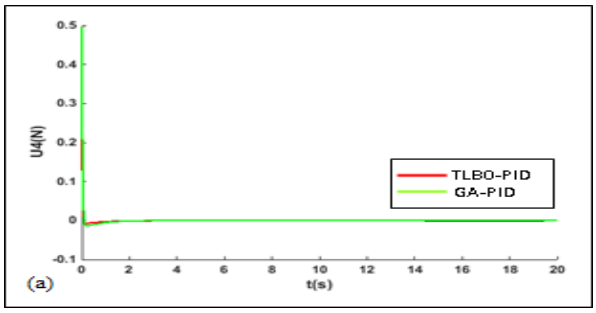


(a)

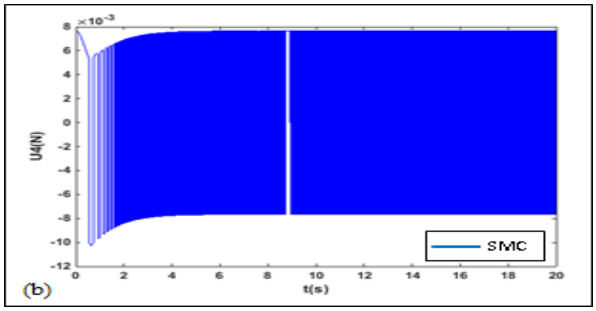


(b)

Fig. 10: Control signals of the φ-angle ((a) using TLBO and GA, (b) using SMC)



(a)



(b)

Fig. 12: Control signals of the ψ-angle ((a) using TLBO and GA, (b) using SMC)

Table 7 gives the ITAE criterion values for the three controllers. Except for the altitude z, the smallest values of the ITAE are obtained in the case of the based TLBO PID controllers.

Table 7: Values of the ITAE criterion

	ITAE		
	PID-TLBO	PID-GA	SMC
x	0.4916	5.2527	17.3950
y	0.5419	5.5124	18.0615
z	0.2668	0.2112	6.7524
ψ	0.3290	0.6981	0.7250

For the three considered controllers, the values of the different characteristics of the transient responses (rising time t_r, settling time t_s and overshoot peak M_P) are gathered in Table 8. The numerical values of this table bring out the good characteristics of the transient responses of the proposed control scheme.

A comparison of the transient response characteristics of the proposed control scheme with those of the nonlinear PID controller presented in [31], the LQR and the PID controllers proposed in [37] is presented in Table 9. It can be concluded, from this table, that the based TLBO PID control methodology presents the best transient responses characteristics.

5.3 Case of circular and helical reference trajectories

In the following, the capability of the proposed control scheme to track circular and helical paths is illustrated. We give in Table 10 the different trajectories that correspond to both circular and helical paths. The obtained results are shown in Fig.13 for the circular path and in Fig.14 for the helical one. It can be observed that a good tracking accuracy of the desired trajectories is obtained for the three controllers.

Table 10: Specification of the circular and the helical reference trajectories

	Circular path	Helical path
x_d	$\cos(0.1\pi t)$	$\cos(0.1\pi t)$
y_d	$\sin(0.1\pi t)$	$\sin(0.1\pi t)$
z_d	u(t)	0.2t
ψ_d	0	0

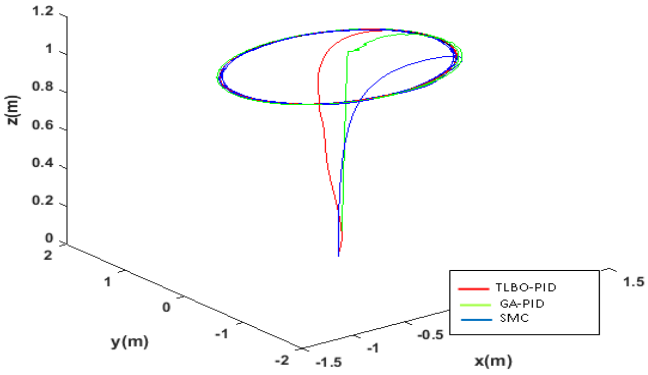


Fig.13: Tracking of a circular path

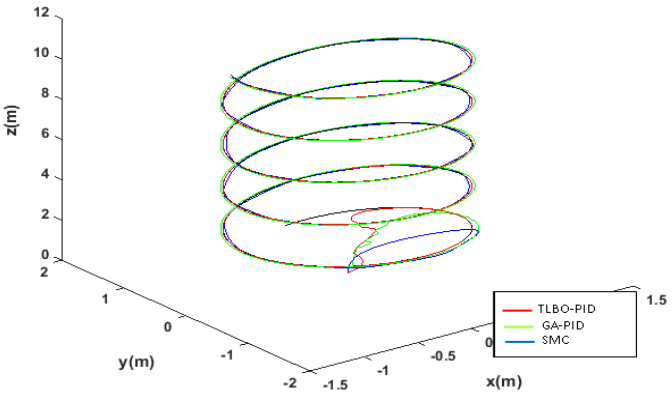


Fig. 14: Tracking of a helical path

Table 8: Transient responses characteristics for (x, y, z, ψ)

	TLBO-PID			GA-PID			SMC		
	$t_r(s)$	$t_s(s)$	$M_P(\%)$	$t_r(s)$	$t_s(s)$	$M_P(\%)$	$t_r(s)$	$t_s(s)$	$M_P(\%)$
x	0.9626	1.5671	1.39	4.3418	8.5126	0	5.4954	12.3609	0
y	1.5004	2.7912	0	4.6539	9.3559	0	5.5966	12.0889	0
z	1.1393	2.0686	0	0.1720	2.3683	6.57	2.3045	5.8493	0
ψ	1.7380	3.1174	0	0.7640	5.6174	22.68	2.5446	4.8405	0

Table 9: Comparison of the transient responses characteristics for several control schemes

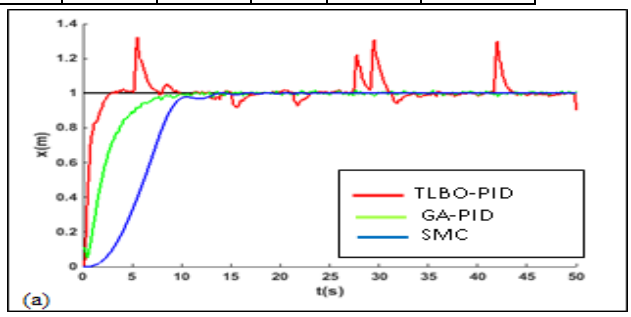
	PID-TLBO			GA-PID [31]			PID [37]			LQR [37]		
	$t_r(s)$	$t_s(s)$	$M_P(\%)$	$t_r(s)$	$t_s(s)$	$M_P(\%)$	$t_r(s)$	$t_s(s)$	$M_P(\%)$	$t_r(s)$	$t_s(s)$	$M_P(\%)$
x	0.962	1.567	1.39	1.152	4.657	0.505	4.407	24.854	0	2.753	4.453	0.439
y	1.500	2.791	0	1.572	3.023	0.195	4.062	24.680	0.114	2.753	4.453	0.439
z	1.139	2.068	0	0.677	1.283	0.505	0.223	10.704	0	2.734	4.365	0.433
ψ	1.738	3.117	0	0.252	3.840	8.152	0.127	1.213	4.819	2.174	3.881	0

5.4. Robustness study

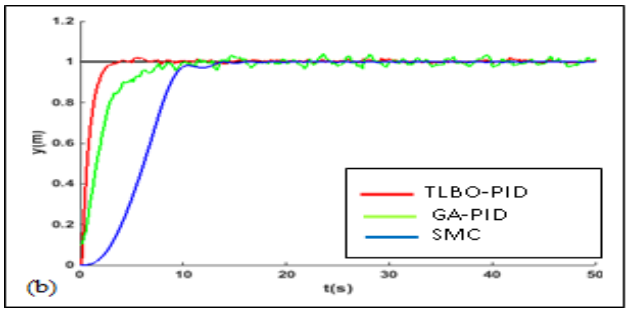
The robustness of the proposed control scheme with respect to measurement noise and external disturbances is investigated. We use in this study unit step reference trajectories for the horizontal position (x, y), the altitude (z) and the yaw angle(ψ).

Measurement noise:

It is important to study the effect of measurement noise on the performance of a control system. To simulate the sensors noise, a white Gaussian noise with zero mean and 10^{-8} variance is added to the measured variables $(x, y, z, \varphi, \theta, \psi)$. The simulation results given in Figs.15-17 illustrate that the SMC strategy has better robustness against the measurement noise than the optimized PID controllers using TLBO and GA.

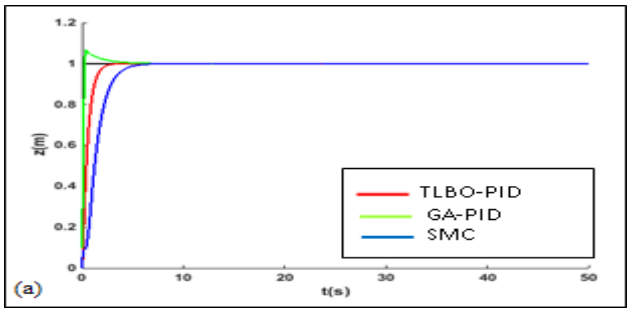


(a)

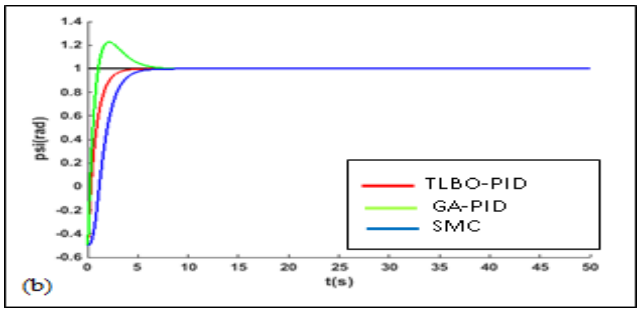


(b)

Fig.15: Tracking results with measurement noise for the horizontal position ((a) x-position, (b) y-position)

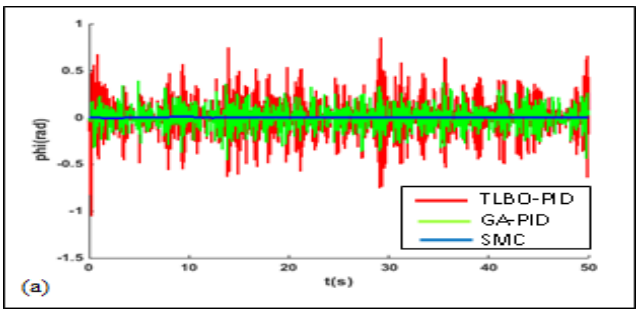


(a)

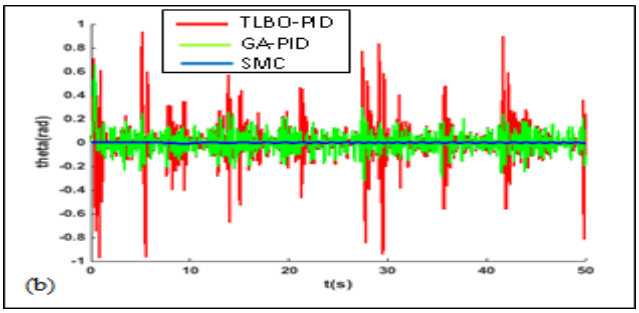


(b)

Fig.16: Tracking results with measurement noise for the altitude (a) and the yaw angle (b)



(a)

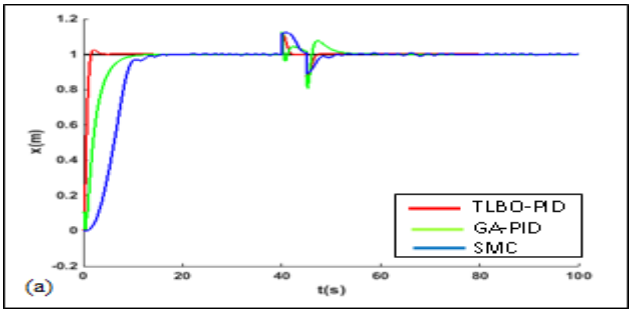


(b)

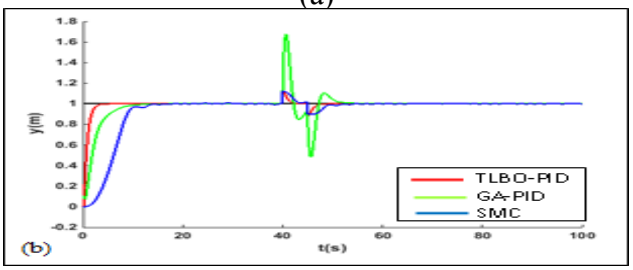
Fig.17: Tracking results with measurement noise for the pitch (a) and roll angles (b)

External disturbances:

To evaluate the robustness of the proposed control scheme against external disturbances, it is assumed that the horizontal position $(x,\ y)$ and the altitude (z) of the quadrotor are subject, during the time interval [40s 45s], to additive disturbances of amplitude 0.12 m. The obtained results using unit steps as reference trajectories for (x, y, z, ψ) are illustrated in Figs.18-20. The proposed control scheme, as well as the SMC methodology, has the capability to compensate the disturbance effect in a short time and with a small overshoot peak value. A large overshoot peak value is observed in the case of the based GA PID control strategy.

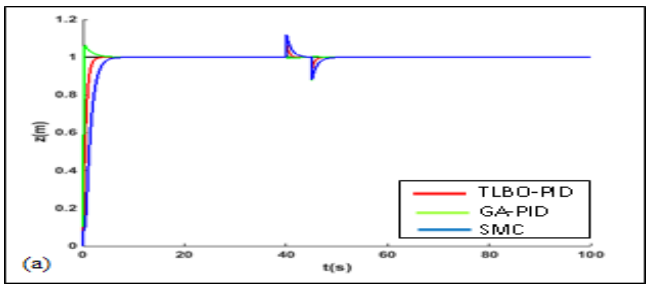


(a)

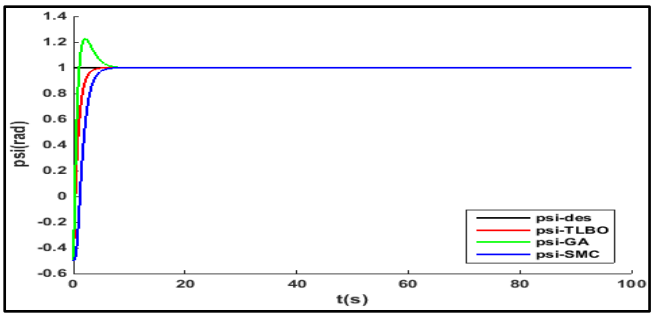


(b)

Fig.18: Tracking results in presence of external disturbances for the horizontal position ((a) x-position, (b) y-position)

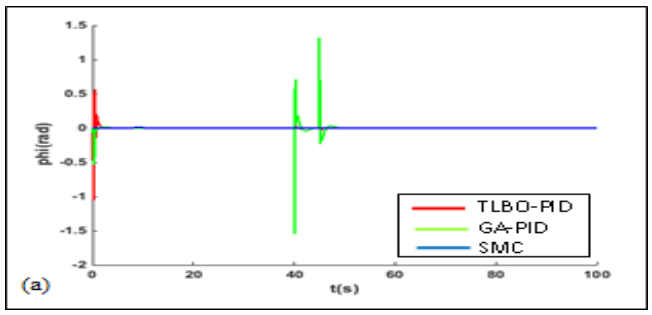


(a)

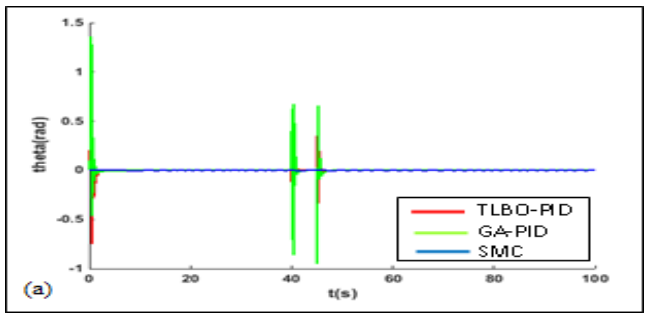


(b)

Fig.19: Tracking results in presence of external disturbances for the altitude (a) and the yaw angle (b)



(a)



(b)

Fig. 20: Tracking results in presence of external disturbances for the pitch (a) and roll angles (b)

6 Conclusion

In this paper, proportional integral derivative controllers was proposed to control the translational and the rotational motions of the quadrotor. Obtaining the optimal gains values of the PID controllers was formulated as a minimization problem of the ITAE criterion, then the teaching learning based optimization algorithm was used to solve this problem. The complete nonlinear model of the quadrotor, including the motors dynamic was used to design and evaluate the performance of the controllers. Unlike many other meta heuristic algorithms and except for the population size and the maximum number of iterations, the TLBO algorithm presents the major advantage for not requiring any control parameter. Therefore, the TLBO algorithm is simple, effective and do not require significant computing time. Furthermore, the efficiency of the TLBO algorithm has been proved in many applications. Hence, combining the PID controller and the TLBO algorithm leads to a simple, powerful and efficient control scheme. Indeed, it was shown through the achieved study that, despite the complexity of the quadrotor model, the proposed control scheme gives satisfactory control performance. In fact, it was illustrated that the proposed control method is able to make the quadrotor track the desired horizontal position, altitude and yaw angle while stabilizing the pitch and the roll angles. The control performance of the proposed control, namely the characteristics of the transient response, the tracking accuracy, and the robustness propriety, was analyzed through a simulation study and compared to that of other control methodologies such as SMC and GA based PID control method. The achieved comparative study has effectively demonstrated the effectiveness of the proposed control scheme.

Acknowledgement

The authors acknowledge the support of the DGRSDT (Direction Générale de la Recherche Scientifique et du Développement Technologique), Algeria.

References:

[1] S. Bouabdallah, A. Noth and R. Siegwart: PID vs LQ control techniques applied to an indoor micro quadrotor. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sendai, Japan, (2004), 2451-2456.

[2] L. S.Atheer, M. Moghavvemi, A. F. M. Haider and S. G. Khalaf: Modelling and PID controller design for a quadrotor unmanned air vehicle. 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR). Cluj-Napoca, Romania , (2010).

[3] J. Li and Y. Li: Dynamic analysis and PID control for a quadrotor. 2011 IEEE International Conference on Mechatronics and Automation. Beijing, China , (2011), 573-578.

[4] G. Szafranski and R. Czyba: Different approaches of PID control UAV type quadrotor. Proceedings of the International Micro Air Vehicle conference and flight competition, 2011 summer edition. (2011), 70-75.

[5] I.D. Cowling, O. A. Yakimenko, J. F. Whidborne and A. K. Cooke: A prototype of an autonomous controller for a quadrotor UAV. Proceedings of the European Control Conference (ECC). Kos, Greece, (2007), 4001-4008.

[6] R. Xu and U. Ozguner: Sliding mode control of a quadrotor helicopter. Proceedings of the 45th IEEE Conference on Decision and Control. San Diego, CA, USA , (2006), 4957-4962.

[7] H. Bouadi, M. Bouchoucha and M. Tadjine: Sliding mode control based on backstepping approach for an UAV type-quadrotor. International Journal of Mechanical and Mechatronics Engineering. 26(5), (2007), 22-27.

[8] S.Bouabdallah and R. Siegwart: Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. Proceedings of the 2005 IEEE International Conference on Robotics and Automation. Barcelona, Spain, (2005), 2247-2252.

[9] K.Runcharoon and V. Srichatrapimuk: Sliding mode control of quadrotor. The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE). Konya, Turkey, (2013), 552-557.

[10] A.R. Patel, M.A. Patel and D.R. Vyas: Modeling and analysis of quadrotor using sliding mode control. Proceedings of the 44th Southeastern Symposium on System Theory (SSST). Jacksonville, FL, USA , (2012), 111-114.

[11] C. Nicol, C. Macnab and A. Ramirez-Serrano: Robust neural network control of a quadrotor helicopter. 2008 Canadian Conference on Electrical and Computer Engineering. Niagara Falls, ON, Canada, (2008), 001233-001237.

[12] J. Muliadi and B. Kusumoputro: Neural network control system of UAV altitude dynamics and its comparison with the PID control system. Hindawi Journal of Advanced Transportation. Volume 2018, (2018), 1-18.

[13] Y. A. Khan Niazi and V. Uddin: Modeling and Neural Control of Quad rotor Helicopter. Yanbu Journal of Engineering and Science. (2), (2011), 35-49.

[14] M. Santos , V. Lopez and F. Morata. Intelligent fuzzy controller of a quadrotor. 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering. Hangzhou, China, (2010), 141-145.

[15] A. Sharma and A. Barve: Controlling of Quad-rotor UAV using PID controller and Fuzzy logic controller. International Journal of Electrical, Electronics and Computer Engineering. 1(2), (2012), 38-41.

[16] A. Prayitno, V. Indrawati and G. Utomo: Trajectory tracking of AR. Drone quadrotor using fuzzy logic controller. TELKOMNIKA. 4(12), (2014), 819-828.

[17] M. Kamel, M. Burri and R. Siegwart: Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles. IFAC PapersOnLine . 50(1), (2017), 3463-3469.

[18] A. Kuyumcu and I. Bayezit: Augmented model predictive control of unmanned quadrotor vehicle. 11th Asian Control Conference. Gold Coast, QLD, Australia, (2017),1626-1631.

[19] T.T. Ribeiro, A. G. S. Conceicao, I. Sa and P. I. Corke : Nonlinear model predictive formation control for quadcopters. IFAC-PapersOnLine. 48(19), (2015), 39-44.

[20] M. Abdolhosseini, : An efficient model predictive control scheme for an unmanned quadrotor helicopter. J Intell Robot Syst. (70), (2013), 27-38.

[21] S. Zeghlache, D. Saigaa, K. Kara, A. Harrag and A. Bouguerra: Backstepping sliding mode controller improved with fuzzy logic: Application to the quadrotor helicopter. Archives of Control Sciences. 22(3), (2012), 315-342.

[22] T. Madani and A. Benallegue, Adaptive control via backstepping technique and neural networks of a quadrotor helicopter. IFAC Proceedings. 41(2), (2008), 6513-6518.

[23] M. Wan, M. Chen and K. Yan: Adaptive sliding mode tracking control for unmanned autonomous helicopters based on neural networks. Complexity (Hindawi). (2018), 1-11.

[24] J. Gómez-Avila ; C. López-Franco ; A. Y. Alanis and N. Arana-Daniel: Control of Quadrotor using a Neural Network based PID. 2018 IEEE Latin

American Conference on Computational Intelligence. Gudalajara, Mexico, (2018).

[25] B. Erginer and E. Altug: Modeling and PD control of a quadrotor VTOL vehicle. 2007 IEEE Intelligent Vehicles Symposium. Istanbul, Turkey, (2007), 894-899 .

[26] İ.C. Dikmen, A. Arisoy and H. Temeltas: Attitude control of a quadrotor. 2009 4th International Conference on Recent Advances in Space Technologies. Istanbul, Turkey, (2009), 722-727 .

[27] M. Rabah, A. Rohan, Y.J. Han and S. H. Kim: Design of Fuzzy-PID Controller for Quadcopter Trajectory-Tracking. International Journal of Fuzzy Logic and Intelligent Systems. 18(3), (2018), 204-213.

[28] K. Khuwaja, T. I. C.onstantin, N. Lighari and R.C. Tarca: PID Controller Tuning Optimization with Genetic Algorithms for a Quadcopter. Recent Innovations in Mechatronics. 1(5), (2018), 1-7.

[29] A. Alkamachi, and E. Erçelebi: Modelling and genetic algorithm based-PID control of H-shaped racing quadcopter. Arabian Journal for Science and Engineering. (42), (2017), 2777-2786.

[30] Y. Bouzid, H. Siguerdidjane and Y. Bestaoui: Sliding Modes based Nonlinear PID Controller for Quadrotor: Theory and Experiment. 14th International Conference on Informatics in Control, Automation and Robotics. Madrid, Spain , (2017), 286-294.

[31] A.A.Najm and I.K. Ibraheem and A.I.J. Technology, Nonlinear PID controller design for a 6-DOF UAV quadrotor system. Engineering Science and Technology, an International Journal. 4(22), (2019), 1087-1097.

[32] M.A Khodja, : Tuning PID attitude stabilization of a quadrotor using particle swarm optimization (experimental). International Journal for Simulation and Multidisciplinary Design Optimization . 8(A8), (2017), 1-9.

[33] M.J. Mohammed, M.T. Rashid, and A.A.J.I.J.o.C.A. Ali, Design optimal PID controller for quad rotor system. International Journal of Computer Applications. 106(3), (2014), 15-20.

[34] R.V. Rao, V.J. Savsani and D.P. Vakharia: Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Information Sciences. 183(1), (2012), 1-15.

[35] R.V. Rao and V. Patel: An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. International Journal of Industrial Engineering Computations. 3(4), (2012), 535-560.

[36] R.V.R ao, V.J. Savsani and D.P. Vakharia: Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design. 43(3), (2011), 303-315.

[37] D.Tosun, Y. Isık and H. Korul: Comparison of PID and LQR controllers on a quadrotor helicopter. International journal of systems applications, engineering & development.9, (2015), 136-143.

Contribution of individual authors to the creation of a scientific article

Naima Bouhabza carried out the simulation and the optimization, the implementation of the TLBO algorithm, Preparation of the published work.
Kara Kamel verification of results, work support.
Mohamed Laid Hadjili Contribute to supervising work.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)