

Optimal Stabilization of A Quadrotor UAV by a Constrained Fuzzy Control and PSO

Hamid Boubertakh^{1,*}

University MSBY of Jijel, LAJ-Laboratoire d'Automatique de Jijel, B.P. 98 Ouled Aissa 180000 Jijel, Alegria

Abstract. This work aims to design an optimal fuzzy PD (FPD) control for the attitude and altitude stabilization of a quadrotor. The control design is done by mean of the particle swarm optimization (PSO) under the constraints of the controller interpretability and the saturation of the actuators. Concretely, a decentralized control structure is adopted where four FPD controllers are used to stabilize the quadrotor angles (roll, pitch and yaw) and height. A PSO-based algorithm is used to simultaneously tune the four constrained controllers regarding a cost function quantifying the whole system performances. The simulation results are presented to show the efficiency of the proposed approach.

1 Introduction

In the recent last years, the quadrotor unmanned aerial vehicles (UAVs) are become very familiar. This famousness is due to their success trough many applications in various domains.

The raisons behind this attraction by quadrotors return largely to their simple and useful configuration with six DOF. The movement of the system is controlled via the relative speeds of four rotors.

The quadrotor stabilization is primordial for a safe and reliable tasks achievement. Therefore, many control techniques have been applied to this end [1–6]. Among them, the classical PID control has been largely used despite its poor performances when it is badly tuned and/or used with nonlinear systems.

Fuzzy PID control [7] is presented in literature as an alternative to address the difficulties with classical PID control. It combines the advantages these controllers namely the simplicity, the interpretability of the control structure and more they can deal with non linear systems.

Last years, the population-based optimization methods have been gained a special interest to solve the fuzzy controllers design problem; Genetic Algorithm (GA) [8], Particle swarm optimization(PSO) [9], Ant Colony Optimization (ACO) [10], Bees Algorithm [11], and Gravitational Search Algorithm (GSA) [12]. Among these methods, PSO is distinguished by the simplicity and the efficiency of the algorithm trough a continuous search space.

In this paper, one suggests to apply the PSO for the fuzzy control optimal design for attitude and altitude stabilization of a quadrotor, under the system inputs saturation physical constraint and the maintain of the logical interpretability in the optimal controllers' structures.

The remainder of this paper is organized as follows. Section II gives a general description and presents a dynamical model of a quadrotor. Section III gives an overview the PSO and presents the FPD controller design method using PSO. In section IV, first the FPD decentralized control structure for the stabilization of the quadrotor is presented then PSO design method is used and simulation results are presented. Section V concludes the paper.

2 The quadrotor model

The quadrotor helicopter is a small UAV with four propellers actuated by DC motors mounted on the end of two perpendicular arms. A basic diagram is shown in Fig. 1. Each rotors pair of the same arm rotates in the same direction; one pair rotates clockwise, while the other rotates counter clockwise. The quadrotor moves by adjusting the angular velocity of each rotor.

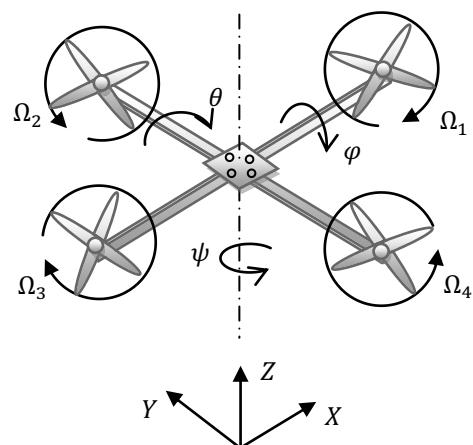


Fig. 1. The quadrotor configuration

* Corresponding author: boubert_hamid@yahoo.com

The quadrotor dynamical model is given by [1]:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_4 x_6 a_1 - x_4 a_2 \Omega_r + b_1 u_2 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= x_2 x_6 a_3 + x_2 a_4 \Omega_r + b_2 u_3 \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= x_2 x_4 a_5 + b_3 u_4 \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= \sin(x_3) \cos(x_1) * b_4 u_1 \\ \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= -\sin(x_1) b_4 u_1 \\ \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= -g + \cos(x_3) \cos(x_1) b_4 u_1\end{aligned}$$

Where

- $x_1 = \varphi$ and $x_2 = \dot{\varphi}$ are respectively the roll angle and corresponding angular velocity;
- $x_3 = \theta$ and $x_4 = \dot{\theta}$ are respectively the pitch angle and corresponding angular velocity;
- $x_5 = \psi$ and $x_6 = \dot{\psi}$ are respectively the yaw angle and corresponding angular velocity;
- $x_7 = X$, $x_9 = Y$, and $x_{11} = Z$ are the Cartesian position coordinates.
- $x_8 = \dot{X}$, $x_{10} = \dot{Y}$, and $x_{12} = \dot{Z}$ are translation velocities.

$$\begin{aligned}- \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} &= \begin{bmatrix} 0 & b & 0 & -b \\ -b & 0 & b & 0 \\ d & -d & d & -d \\ b & b & b & b \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \\ - \Omega_r &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \text{ is the sum of rotor angular velocities.} \\ - a_1 &= \frac{Iy - Iz}{Ix}; a_2 = \frac{Jr}{Ix}; a_3 = \frac{Iz - Ix}{Iy}; a_4 = \frac{Jr}{Iy}; \\ - a_5 &= \frac{Ix - Iy}{Iz}; b_1 = \frac{l}{Ix}; b_2 = \frac{l}{Iy}; b_3 = \frac{1}{Iz}; b_4 = \frac{1}{m}.\end{aligned}$$

The values of the parameters in the state space model are given by Table 1.

Table 1. The quadrotor model parameters

Parameter	Value
l	0.232 m
d	$7.5 \times 10^{-7} \text{ N.m.s}^2$
b	$3.13 \times 10^{-5} \text{ N.s}^2$
m	0.52 Kg
g	9.806 m/s^2
I_x	$6.228 \times 10^{-3} \text{ Kg.m}^2$
I_y	$6.228 \times 10^{-3} \text{ Kg.m}^2$
I_z	$1.121 \times 10^{-2} \text{ Kg.m}^2$
J_R	$6.01 \times 10^{-5} \text{ Kg.m}^2$

3 The fuzzy PD control

The fuzzy proportional derivative (FPD) control law is given by

$$u_{FPD}(t) = f(e(t), \Delta e(t)) \quad (1)$$

With ; $e(t) = y_r(t) - y(t)$ and $\Delta e(t) = \frac{de(t)}{dt}$.

Where f is the function of the fuzzy inference system, u_{FPD} is the control signal, y_r is the reference signal and y is the system's controlled output.

According to the typical structural design of FPD controllers in [13], one uses FPD controllers with 3 membership functions for input variables and 5 singletons for the output variable as shown on Fig. 2.

The rule base of the controllers is donated by Table 2.

The crisp control action is determined by the average sum (2).

$$u_{FPD}(t) = \frac{\sum_{i=1}^9 V_i C_i}{\sum_{i=1}^9 V_i}. \quad (2)$$

Where C_i is the conclusion of rule i and V_i is its truth value calculated by the algebraic product method given by: $V_i = \mu_{e_i}(e(t)) \cdot \mu_{\Delta e_i}(\Delta e)$. $\mu_{y_i}(y)$ is the membership grade of the input variable y evaluated in rule i by the corresponding membership function.

Table 2. FPD controller rule base

		Δe	
e	N	Z	P
N	$C_1 = \text{NB}$	$C_2 = \text{NS}$	$C_3 = \text{ZR}$
Z	$C_4 = \text{NS}$	$C_5 = \text{ZR}$	$C_6 = \text{PS}$
P	$C_7 = \text{ZR}$	$C_8 = \text{PS}$	$C_9 = \text{PB}$

The labels NB, NS, ZR, PS, and PB refer respectively to the linguistic terms; Negative Big, Negative Small, around Zero, Positive Small and Positive Big.

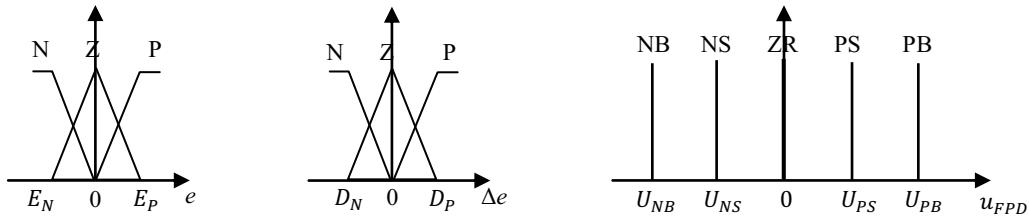
Note here that, if one considers the typical structure design of the FPD controller with

- The input membership functions and the output singletons are evenly distributed on their corresponding universes of discourse, i.e., $E_P = E_N$, $D_P = D_N$ and $U_{NB} = -U_{PB} = 2U_{NS} = -2U_{PS} = U$.
- The universes of discourse are large enough such that all possible inputs are always within their limits.

Then, the crisp control signal calculated by (2) is given by [13]:

$$u_{FPD}(t) = \frac{U}{2E} e(t) + \frac{U}{2D} \frac{de(t)}{dt}. \quad (3)$$

Equation (6) shows clearly that the typical FPD controllers are identical to the classical PD controllers with the following proportional and integral gains: $K_P = \frac{U}{2E}$ and $K_D = \frac{U}{2D}$.

**Fig 2.** Membership functions of the FPD controllers

A nonlinear FPD controller can be reached by moving the input membership functions and/or the output singletons from their modal positions. That is, the FPD controllers combine the advantages of classical PD controllers and fuzzy controllers; the simplicity, the structure interpretability, and the nonlinearity.

4 Quadrotor optimal stabilization using PSO design-based fuzzy control

4.1 Particle swarm optimization

Particle swarm optimization (PSO), initially introduced by Kennedy and Eberhart [14], is a heuristic optimization method inspired by the movement of individuals within a swarm. PSO technique conducts search in parallel using a population of particles, corresponding to individuals. Each particle represents a candidate solution to the problem at hand. In the PSO system, a particle makes use of the best position encountered by itself and that of its neighbors to steer toward an optimal solution. The performance of each particle is measured using a predefined fitness function, which quantifies the performance of the optimization problem.

The mathematical expressions for velocity and position updates are given by

$$v_{ij}^{k+1} = w v_{ij}^k + c_1 r_1 (Pbest_{ij} - x_{ij}^k) + c_2 r_2 (Gbest_j - x_{ij}^k) \quad (4)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (5)$$

Where: $1 \leq i \leq N$, $1 \leq j \leq D$, $1 \leq k \leq K$.

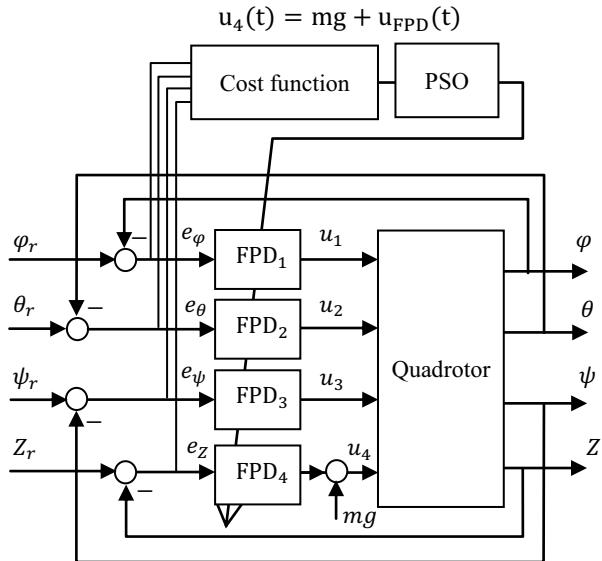
N is the number of particles in the swarm, D is the dimension of particle, and K is the maximum iterations. x_{ij}^k is the position of particle i in the dimension j at iteration k , v_{ij}^k is the velocity of particle i in the dimension j at iteration k , $pbest_{ij}$ is a personal best of particle i in the dimension j , $gbest_j$ is a global best of all particles in the dimension j . w is the inertia weight factor, c_1 and c_2 are the acceleration constants, r_1 and r_2 are random numbers in interval $[0, 1]$.

4.2 Application to the stabilization of the quadrotor by fuzzy control

4.2.1. The control structure

The control goal is to stabilize angles and height of the quadrotor. That is, one uses four decentralized FPD controllers as presented by the bloc diagram on the

Fig. 3. The controllers FPD_i ($i = 1, 2, 3$) have the same structure of section 3, however, the FPD_4 is modified as:

**Fig. 3** Decentralized FPD control

4.2.2. The parameters' design vector

The design parameters' vector is constructed by the positions of the input membership functions and the positions of the output singletons of the four FPD controllers. However, one takes symmetrical fuzzy controllers with respect to zero, i.e., $E_{Pi} = -E_{Ni} = E_i$, $D_P = -D_N = D_i$, $U_{PSi} = -U_{NBi} = U_{1i}$ and $U_{PBi} = -U_{NBi} = U_{2i}$ where the index i refer to controller FPD_i . In addition, because of the quadrotor body symmetry, one takes identical controllers for the roll and pitch angles (i.e. $FPD_1 = FPD_2$).

That is, the parameters' vector is then

$$P = [E_1 \ D_1 \ U_{11} \ U_{12} \ E_3 \ D_3 \ U_{31} \ U_{32} \ E_4 \ D_4 \ U_{41} \ U_{42}]^T$$

4.2.3. The cost function

The effectiveness of the FPD control scheme is evaluated at the end of the whole control system response to desired inputs by a unique cost function evaluating simultaneously the four system responses. Therefore, in order to overcome the non homogeneity of the outputs, in the cost function instead of using the errors signals directly, one uses the relative errors defined as : $e_1(t) = \frac{e_\phi(t)}{e_\phi(t_0)}$, $e_2(t) = \frac{e_\theta(t)}{e_\theta(t_0)}$, $e_3(t) = \frac{e_\psi(t)}{e_\psi(t_0)}$ and $e_4(t) = \frac{e_Z(t)}{e_Z(t_0)}$. Without loss of generality, one

considers in this work the integral sum of relative errors (ISRE) cost function given by

$$ISRE = \int_{t_0}^{t_f} (e_1^2(t) + e_2(t)^2 + e_3^2(t) + e_4^2(t)) dt$$

4.2.4. The optimization problem constraints

In this work one consider two types of constraints.

- *The control structure interpretability*

One has to preserve the semantic interpretability of the fuzzy rule bases of the controllers such that all parameters values correspond concretely to their linguistic terms (e.g.: a minus signed value correspond to the term ‘Negative’, a positive signed value correspond to the term ‘Positive’, a ‘Small’ value is less than a ‘Medium’ value which is itself less than a ‘Big’ value,...).

- *The system input constraints*

In practice, the system inputs are subject to physical constraints often caused by the actuators limitations. That can degrade the closed system performance and may leads to the instability, if one does not take care of the problem in the control design. In this work, one considers the actuators’ amplitudes saturation such as a control inputs u_i do not come over certain bounds, i.e. $u_{min_i} \leq u_i \leq u_{max_i}$.

$$u_i = \begin{cases} u_{max_i} & \text{if } u_i > u_{max_i} \\ u_{min_i} & \text{if } u_i < u_{min_i} \\ u_i & \text{otherwise} \end{cases}$$

In this work one takes:

$$-1 \leq u_{FPD_i} \leq +1, (i = 1, \dots, 3)$$

$$0 \leq u_{FPD_4} \leq +15$$

4.2.5. The algorithm

In the algorithm of the FPD control design using PSO, a particle is one value of the FPD controllers’ parameters vector. A particle swarm is a group of parameters vector values which are manually or randomly initialized in search space defined by limits of the components of the tuning parameters vector. So, a particle position x is the controllers’ parameter vector value, $Pbest$ is the best parameters values vector regarding the cost function, $Gbest$ is the global best parameters values vector of all particles. The algorithm steps are summarized in table 3.

4.2.6. Simulations

In the simulations, one takes:

- The initial states: $\varphi(0) = 0.5 \text{ rad}$, $\theta(0) = -0.5 \text{ rad}$, $\psi(0) = 0.5 \text{ rad}$, and $Z(0) = 0 \text{ m}$.
- The desired angles and height : $\varphi_d = 0 \text{ rad}$, $\theta_d = 0 \text{ rad}$, $\psi_d = 0 \text{ rad}$, and $Z_d = 10 \text{ m}$,
- The cost function spans from $t_0 = 0$ sto $t_f = 20 \text{ s}$.
- The PSO algorithm parameters: $\omega = 0.9$, $c_1 = 2$, $c_2 = 2$, $N = 10$, $K = 100$.
- The initial parameters values are chosen randomly between the following bounds: $0 \leq E_i \leq 10$,

$$0 \leq D_i \leq 10, \quad 0 \leq U_{i1} \leq 15, \quad 15 \leq U_{i2} \leq 30. \quad (i=1, \dots, 4).$$

The Fig. 4 shows the evolution of the cost function with iterations. Table 4 gives the optimal parameters’ values. Fig. 5 show the angles (roll, pitch and yaw), Fig. 6 show the control signals u_1 , u_2 and u_3 . Fig 7 show the height and Fig. 8 show the control signal u_4 .

Table 3. The constrained PSO algorithm

```

Begin
k=0
Initialize all particles positions and speeds
Repeat form k=1:K
  For each particle
    – Compute the controllers actions  $u_{FPD_i}$  (i=1,...4)
    – Apply the control actions to de system.
    – Evaluate the cost function of the particle.
    If the constraints are not all satisfied
      – Attribue the worst value to the corresponding cost function.
  End
  – Evaluate Pbest
End
  – Evaluate Gbest
  For each particle
    – Update the particles’ speeds and positions by (4) and (5) respectively.
  End
  k←k+1
End
The optimal parameters’ vector is Gbest.
End

```

Table 4. The optimal paramers’ values

Parameter	Optimal value
$E_1 = E_2$	2.5077
$D_1 = D_2$	5.3927
$U_{11} = U_{21}$	4.5098
$U_{12} = U_{22}$	41.9965
E_3	2.6048
D_3	21.9211
U_{31}	3.5153
U_{32}	15.531
E_4	3.6861
D_4	5.8867
U_{41}	8.5665
U_{42}	19.0788

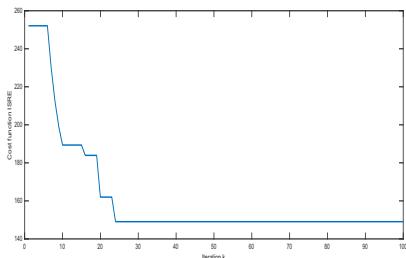


Fig 4. Cost function

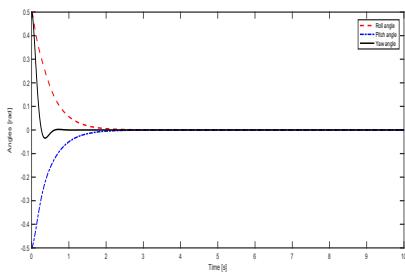


Fig. 5 Angles φ , θ and ψ

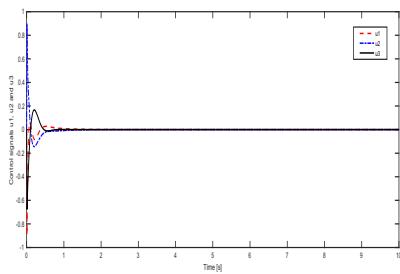


Fig. 6 Control signals u_1 , u_2 and u_3

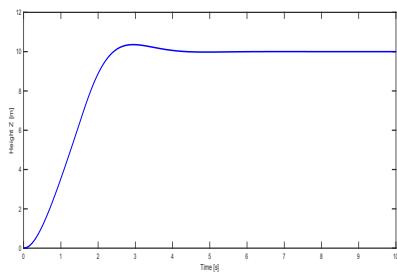


Fig 7. Height Z

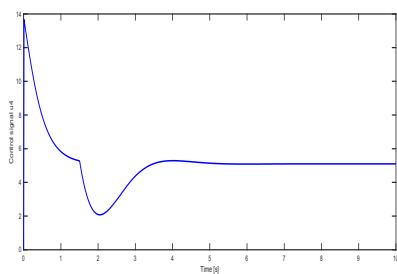


Fig 8. Control signal u_4

5 Conclusion

In this paper, one has proposed a PSO-based method to FPD control design for the stabilization of a quadrotor under the actuators saturation and interpretability of the FPD control. Four FPD controllers are used in a decentralized control structure to stabilize the quadrotor attitude and height. The constrained PSO method is used to tune the FPD controllers' parameters simultaneously in order to minimize an integral square error cost function which quantity the whole system performance. The simulation results show the efficiency of the proposed method.

References

1. S. Bouabdallah, *Design and control of quadrotors with application to autonomous flying* (Ph.D. diss. École Polytechnique Fédérale de Lausanne, 2007).
2. T. Madani, A. Benallegue, Backstepping control for a quadrotor helicopter, *IEEE International Conference on Intelligent Robots & Systems*, 3255-3260 (2006).
3. A. Tayebi and S. McGilvray, *IEEE Trans. Syst. Tech.*, **14**, 552-570 (2006)
4. P. Castillo, A. Dzul, R. Lozano, *IEEE Trans. Syst., Man, Cybern. B*, **32**, 239-248 (2002).
5. K. U. Lee, H. S. Kim and Y.-H. Choi, Hovering control for a quadrotor, *12th Conference on Control, Automation and Systems*, 162-167 (2012)
6. H. Bolandi, M. Rezaei, M. Smailzadeh, Intelligent Control and Automation, **4**, 335-342 (2013).
7. R.R. Yager, D.P. Filev, *Essential of fuzzy modeling and control* (John Wiley & Sons Inc, 1995)
8. J.M. Adams, K.S. Rattan, A genetic multi-stage fuzzy PID controller with a fuzzy switch, *Proc. IEEE Int. Conf. on Systems, Man, Cybernetics, Tucson, AZ, USA*, **4**, 2239-2244 (2001)
9. H. Boubertakh, S. Labiod and M. Tadjine, PSO to design decentralized fuzzy PI controllers: Application for a Helicopter, *19th Mediterranean Conference on Control and Automation (MED'12)*, IEEE, pp. 633-637 (2012)
10. H. Boubertakh, M. Tadjine, P.-Y. Glorennec, S. Labiod, Tuning fuzzy PID controllers using ant colony optimization, *17th Mediterranean Conference on Control and Automation (MED'09)*, IEEE, 13–18 (2009)
11. C. Craveo, F. Valdez, O. Castillo, *Appl. Soft Comp.*, **43**, 131-142 (2016)
12. C.-T. Chao, M.-T. Liu, J.-S. Chiou, Y.-J. Aided Engineering and Software, **33**, 1659-1667 (2016)
13. H. Boubertakh, M. Tadjine, P.Y. Glorennec, S. Labiod, *ISA Trans.*, **49**, 543-551 (2010)
14. J. Kennedy, R. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, pp. 1942-1948 (1995)