mathematics

MDPI

Article

New Fusion Algorithm-Reinforced Pilot Control for an Agricultural Tricopter UAV

Huu Khoa Tran [1], Juing-Shian Chiou [2,*] and Viet-Hung Dang [3]

[1] Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; khoa.tran@mail.ntust.edu.tw
[2] Department of Electrical Engineering, Southern Taiwan University of Science and Technology, Tainan 71005, Taiwan
[3] Faculty of Information Technology, Duy Tan University, Danang 50000, Vietnam; dangviethungha@gmail.com
* Correspondence: jschiou@stust.edu.tw

Received: 5 August 2020; Accepted: 1 September 2020; Published: 4 September 2020

Abstract: Currently, fuzzy proportional integral derivative (PID) controller schemes, which include simplified fuzzy reasoning decision methodologies and PID parameters, are broadly and efficaciously practiced in various fields from industrial applications, military service, to rescue operations, civilian information and also horticultural observation and agricultural surveillance. A fusion particle swarm optimization (PSO)–evolutionary programming (EP) algorithm, which is an improved version of the stochastic optimization strategy PSO, was presented for designing and optimizing controller gains in this study. The mathematical calculations of this study include the reproduction of EP with PSO. By minimizing the integral of the absolute error (IAE) criterion that is used for estimating the system response as a fitness function, the obtained integrated design of the fusion PSO–EP algorithm generated and updated the new elite parameters for proposed controller schemes. This progression was used for the complicated non-linear systems of the attitude-control pilot models of a tricopter unmanned aerial vehicle (UAV) to demonstrate an improvement on the performance in terms of rapid response, precision, reliability, and stability.

Keywords: particle swarm optimization (PSO); evolutionary programming (EP); fuzzy control; proportional–integral–derivative controller; integral of absolute error (IAE) criterion; attitude control; tricopter unmanned aerial vehicle (UAV)

1. Introduction

Evolutionary programming (EP) is one of the four primary evolutionary algorithm paradigms and was first proposed by Lawrence J. Fogel in the United States in 1966 [1]. The development of EP, which is parallel to global search techniques, was continued by his son D. B. Fogel. In contrast to the genetic algorithm (GA), the EP algorithm does not include a crossover operator but only contains a mutation operator and a specific selection strategy; however, its parameters are allowed to evolve. In EP progress, the prior knowledge of dynamics system or environmental information is not necessary. Those optimization methodologies are alike in terms of their natural evolution theory and differ mostly in terms of their parameter representation approach.

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart [2] to solve optimization problems, and belongs to the class of swarm intelligence methods. This algorithm is based on stochastic and population-based searches and tries to imitate the behavior of a flock of birds and shoal of fish. The original PSO method only has a few adjusted parameters and has higher computation speed, higher accuracy, and smaller memory size compared with the methods such as neural networks,

genetic computation, and machine learning. However, a method for determining the proper values of PSO parameters at the optimal level is yet to be found. This challenging problem has received considerable attention in [3–5]. Hence, at present, in order to solve the local optima trap and improve the optimization process, many different methodologies of hybrid PSO algorithms have been enhanced to overcome this downside. For example, Juang proposed a hybrid PSO algorithm known as the hybrid version of a genetic algorithm (GA) with PSO [6], which incorporates the elite strategy for evolutionary operations of a GA. In this strategy, the upper half of optimal-performance individuals in a population is regarded as elites. Bergh and Engelbrecht [7] introduced a cooperation of PSO. Huang and Mohan [8] proposed a hybrid condition to obtain PSO robustness. To estimate the parameters of non-linear systems, Li et al. [9] presented the combination of PSO and simulated annealing for the jumping property. Next, Araújo and Uturbey [10] successfully applied a hybrid PSO algorithm to the generation and demand of the dispatch problem. Later, Ali and Tawhid solved many engineering optimization problems by using a hybrid PSO–differential evolution method in [11]. Recently, hybrid PSO algorithms have been exploited to solve unmanned aerial vehicle (UAV) path planning in [12,13]. The combination of PSO and support vector regression (SVR) for tourist arrivals forecasting is utilized by [14]. Hybrid PSO and the direct search method (DSM) for optimization of large-scale economic dispatch with valve-point effects is exploited in [15]. The hybrid swarm intelligence approach for blog success prediction is developed in [16]. All the aforementioned techniques can improve the parameters evolution speed. The use of fusion and hybrid algorithms has been increasing in recent decades because these methods combine the advantages of different algorithms and also eliminate the drawbacks of single optimization algorithms. Therefore, this study employs the fusion PSO–EP algorithm to reinforce the pilot control of an agricultural tricopter UAV.

Due to the effortless control formation, ease of design, and low cost of traditional proportional integral derivative (PID) controllers, they have been currently employed in real-world systems. Moreover, a novel hybrid control approach involving a fuzzy-PID controller was successfully used for an autonomous mini helicopter [17]. The model of fuzzy logic rule is useful when the physical progression of obtaining information is ambiguous, and limited data is available. The use of a fuzzy logic control (FLC) is proven as one of the most efficient and systematic approaches for solving non-linear practical systems by emulating human logic rather than using an accurate mathematical model.

The membership functions of fuzzy controller design are independent of extracted rules. Thus, a high system performance cannot be achieved, especially when there are many input variables in a complex system. Nevertheless, in most of circumstances, the fuzzification and de-fuzzification and also the membership functions could be adjusted to improve the performance of systems [18–21].

UAVs, which are able to autonomously operate, are considerably complicated non-linear systems with uncertainties. The operating conditions of UAVs are also unpredictable, for example, changes in direction and speed of wind, results in variations in the flight attitude, direction, speed and precision. Therefore, the control approaches of an aerial robot, especially in attitude controller design issues [22–33] find many difficulties to provide high performance, absolute precision and robustness. Thus, a tricopter, which can be successfully applied in horticultural observation and agricultural surveillance [34–38] is selected as a best model to present the efficiency of proposed controller design.

In general, this study proposed the combination of a fuzzy controller that design by the triangular membership functions and a conventional PID controller to compose the fuzzy-PID controller scheme. Then, the proposed controllers utilize the advantage of the fusion evolutionary PSO algorithm by using a fitness function associated with a performance index of the system—the integral of absolute error (IAE) [39,40]—to optimize the controller parameters. The designed controllers demonstrate the benefits of the improving performance of Tricopter pilot control models that could be fruitfully operated on agriculture fields [41–44].

2. Fusion Particle Swarm Optimization–Evolutionary Programming (PSO–EP) Algorithm-Reinforced Precision Control Design

In this section, we proposed the optimization algorithm PSO-EP for tuning the five parameters of the fuzzy-PID controller that apply to the tricopter pilot control. This simultaneous tuning task is complicated since the final trade-off in multi-objective optimization [10] must find out as fast as possible, the fusion PSO and EP algorithm is employed to save design efforts, operation time, and also obtain better performance.

2.1. Fusion PSO–EP Algorithm

The fundamental PSO technique includes a number of particles moving in a multi-dimensional searching space. The PSO algorithm is initiated on a random population and searches the optimal point by continually updating this population. The PSO algorithm improvement are as follows: (1) the evolution operators—crossover and mutation—are involved and (2) the adjustment of various free parameters is not required. Each potential solution, represented by the set of particles, is assigned a random velocity to be "flown" in the problem area. The current data can be shared with related particles at the optimal fitness time. The position and velocity vectors of each particle are remembered and updated at the moment at which the vectors reached the highest fitness level through iterations. The particle-birds that fly through the searching swarm space are being attracted toward the optimal solution.

Assume that the searching space has a dimension D and m particles. The ith particle settle at position $X_i(t) = [x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t)]$, and the "flying" particle velocity is presented by $V_i(t) = [v_{i1}(t), v_{i2}(t), \ldots, v_{iD}(t)]$. Each particle adjusts its velocity vector based on the best tracking position $P_i(t) = [p_{i1}(t), p_{i2}(t), \ldots, p_{iD}(t)]$, and based on its neighbor $P_g(t)$ that is found as the best update particle of time t, where $P_g(t) = [p_{g1}(t), p_{g2}(t), \ldots, p_{gD}(t)]$. Then, the essential PSO algorithm can be established in Equations (1) and (2):

$$v_{iD}(t+1) = \omega v_{iD}(t) + c_1 rand_1(p_{iD}(t) - x_{iD}(t)) + c_2 rand_2\Big(p_{gD}(t) - x_{iD}(t)\Big) \quad (1)$$

$$x_{iD}(t+1) = x_{iD}(t) + v_{iD}(t+1) \quad (2)$$

where ω is the inertia weighting factor, c_1 and c_2 as learning rates. The variables $rand_1$ and $rand_2$ are random distribution values ϵ [0–1].

The PSO algorithm has the advantage of a rapid convergence in 3D searching space. However, its weakness is it is easily trapped in a local minimum. Consequently, the combination of PSO and EP will cover this issue and furthermore create the new elite algorithm.

EP is one of the four primary evolutionary paradigms and does not include the crossover operator. However, in contrast to the GA, EP employs the mutation and the strategy of individual selection. The EP population members consider each of the species, thus a new offspring is rapidly created [1]. Hence, this study proposed the combination of PSO and EP to create the elite fusion PSO–EP algorithm.

In general, many problems in different scientific disciplines and real-time technical applications can be expressed as a constrained optimization problem and this is presented in Equation (3):

$$\begin{cases} \quad minimum\ f(E) \\ subject\ to \quad e_j\, \epsilon\, \Omega,\ \forall\, \{j = 1, 2, \ldots pop\} \end{cases} \quad (3)$$

where Ω is the constraint set, $f(E)$ is utilized as the benchmark function and $E_j = [e_1, e_2, \ldots, e_{pop}]$, $j = \{1, 2, \ldots\ pop\}$ represents the variable vectors.

Initially, an individual population is generated randomly. Each individual is represented by a set of PID parameter gains $[k_P, k_I, k_D]$ and the fuzzy control gains $[k_e, k_{de}]$. The membership function of fuzzy μ is verified by using: $\mu_{\mathrm{e(t)}} : U_{e(t)} \rightarrow \left[-k_{e(t)}, k_{e(t)}\right]$, $\mu_{\mathrm{de(t)}} : U_{de(t)} \rightarrow \left[-k_{de(t)}, k_{de(t)}\right]$.

The optimization problem is related to find the minimal of the performance index $f(E)$ with five parameters: $\left[K_p,\ K_I,\ K_D,\ K_e,\ K_{de}\right] \in E$. The progression is conducted as follows:

Step 1. Velocity strategy updated. The fitness function value of each individual in the population is computed. The $x_i(t)$ value of each particle is compared with the resulting value of $x_i(t)$ from the contemporary evaluation. If the next value of $x_i(t)$ is superior than the former value, it will be set as the next location. Equations (1) and (2) are used to update the particle velocity.

Step 2. Position strategy updated. The Equation (4) is used to update particle's position and the mutation rate value *mr* is set. The positions $p_i(t)$ and $p_g(t)$ are updated using Equations (1) and (2).

$$\begin{array}{l} \text{If } rand < mr, \\ x_{iD}(t+1) = x_{iD}(t) + [v_{iD}(t+1)/2N(0,1)] \\ \text{else } x_{iD}(t+1) = x_{iD}(t) + [v_{iD}(t+1)/\sigma] \end{array} \tag{4}$$

where $N(0,1)$ is a Gaussian distribution. In $1 \le \sigma < 2$, $\sigma \in R$, σ is the deceleration factor of a particle.

Step 3. Mutation EP strategy. Each of $x_i(t)$ is mutated and then generated $X_j(t) = [x_{j1}(t), x_{j2}(t), \ldots, x_{jD}(t)]$ using Equation (5):

$$x_{jD}(t+1) = x_{jD}(t) + N\left(0, \sigma^2\right) \tag{5}$$

where x_{jD} represents the jth individual error and $N(0, \sigma^2)$ denotes a normal distribution
Step 4. Reproduction of EP. Apply roulette wheel selection technique to every particle j.
Step 5. Repeat until the optimized fitness function $f(E)$ is achieved.

2.2. Optimal Control Design

To improve the dynamic response and also reduce/eliminate the steady state error of a system, the PID controllers are introduced. Moreover, in order to improve the quality performance of the controlled systems, the fuzzy controller is usually combined to the PID control. In fuzzy controller design, the tracking error *e(t)* and the differential tracking error *de(t)* are used as fuzzy inference system inputs, as displayed in Figure 1. The conventional triangular membership functions of *e(t)* and *de(t)* are used as inputs, and the output is e_{Fuzzy} *(t).*

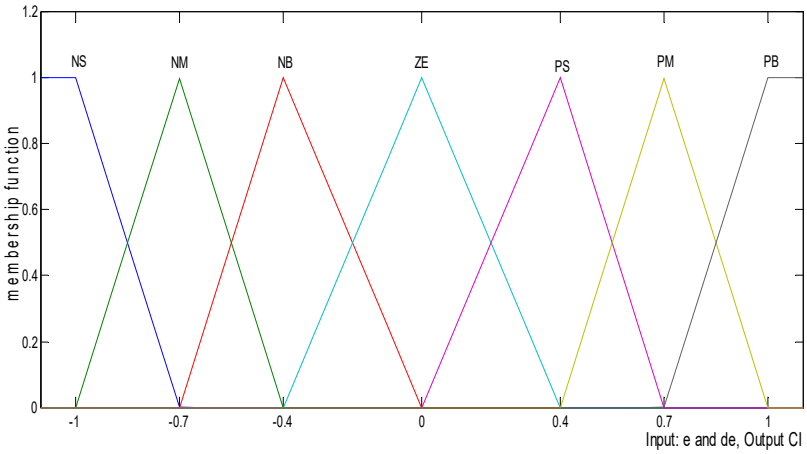


Figure 1. Triangle membership function of the fuzzification inference.

By using the set of linguistic fuzzy rules, its mechanism can produce the appropriate action for fuzzy control. The mechanism comprises one comparable control module and one input/output component: two inputs signal *e(t)* and *de(t)* and one output signal $e_{Fuzzy}(t)$. To reduce the system error, $k_{e(t)}$ *and* $k_{de(t)}$ were employed. The fuzzification method: the center of the area is utilized to minimum values of the Fuzzy Mandani type, hence, 7×7 fuzzy rules are expressed in Table 1. Because this combination yields basic execution factors of the fuzzy control rules, seven partitions were formed:

negative large (NL), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), and positive large (PL).

Table 1. The fuzzy rule table.

$e_{\mathbf{Fuzzy}}(t)$		$e(t)$						
		NL	**NM**	**NS**	**ZE**	**PS**	**PM**	**PL**
de(t)	**PL**	**ZE**	PS	PM	PL	PL	PL	PL
	PM	NS	**ZE**	PS	PM	PL	PL	PL
	PS	NM	NS	**ZE**	PS	PM	PL	PL
	ZE	NL	NM	NS	**ZE**	PS	PM	PL
	NS	NL	NL	NM	NS	**ZE**	PS	PM
	NM	NL	NL	NL	NM	NS	**ZE**	PS
	NL	NL	NL	NL	NL	NM	NS	**ZE**

The satisfactory performance requirement obtained by using fuzzy-PID parameter controllers has been presented in many empirical studies [16–18,29]. The formal structure of the fuzzy-PID controller can be expressed as in Equation (6):

$$u = e_{Fuzzy} + u_{PID} = e_{Fuzzy} + k_P \times e + k_I \int e.dt + k_D \times \frac{de}{dt} \tag{6}$$

The gain coefficients used in the PSO–EP algorithm for discovering the P, I, and D values are evaluated as follows: k_P, k_I, k_D in range [0, 50]. Based on the PID output range, the range of reference input values of *e(t)* and *de(t)* of fitness functions *f(E)* that are generated using Equation (7) as follows:

$$f(E) \in [f_{\min}(E), f_{\max}(E)] \Rightarrow f(E) \in [0, 1000] \tag{7}$$

In this study, the fitness function included five components: the rising time, the settling time, the peak of system response, the maximum overshoot, and the minimum fitness function criterion (α_5, α_6, α_7), as described in Equation (8):

$$\begin{aligned} f(E) = {} & \alpha_1 \times RiseTime + \alpha_2 \times SettlingTime + \alpha_3 \times PeakTime + \alpha_4 \times (\left|r - Overshoot\right|) \\ & + \alpha_5 \sum_{iter=0}^{iter=10} \left|e(iter)\right| + \alpha_6 \sum_{iter=11}^{iter=50} \left|e(iter)\right| + \alpha_7 \sum_{iter=51}^{iter=end} \left|e(iter)\right| \end{aligned} \tag{8}$$

where α_i represents weighting factors. The output performance was evaluated by using the fitness function. Then, the values of these parameters were recorded. Based on the high output performance of the controller design, the information of response including: rising time, settling time, peak time and overshoot (or undershoot) make the decision about the best performance. Fastest response means that the fastest rising time is also fastest settling time, related to α_1 and α_2, the next is without overshoot or peak, relative to α_3 and α_4 and the final is the steady state error that is minimum, relative to α_5, α_6 and α_7. We assigned appropriate the weights [α_1, α_2, α_3, α_4] = [30, 25, 10, 5] and the corresponding relative weights [α_5, α_6, α_7] = [7, 5, 3] were also computed. The error in the output performance is highlighted. Therefore, if the fitness function *f(E)* is rising then the hybrid PSO–EP algorithm will make the decision that this particle must be ignored in the group of particles. Thus, the particle groups that cover an enormous concealed error can be eliminated, and the system convergence speed can be increased. For evaluating the control design quality, the definite perform index is exploited. The fusion PSO–EP algorithm reinforced using fuzzy-PID controllers was used to assign optimal control gains for a tricopter to achieve a trade-off between flight pilot convenience and system stability, and this

algorithm is explained in Figure 2. The system performance index uses the IAE criterion, that is [39,40] as follows: $\text{IAE} = \int_0^\infty |e(t)|.dt$.

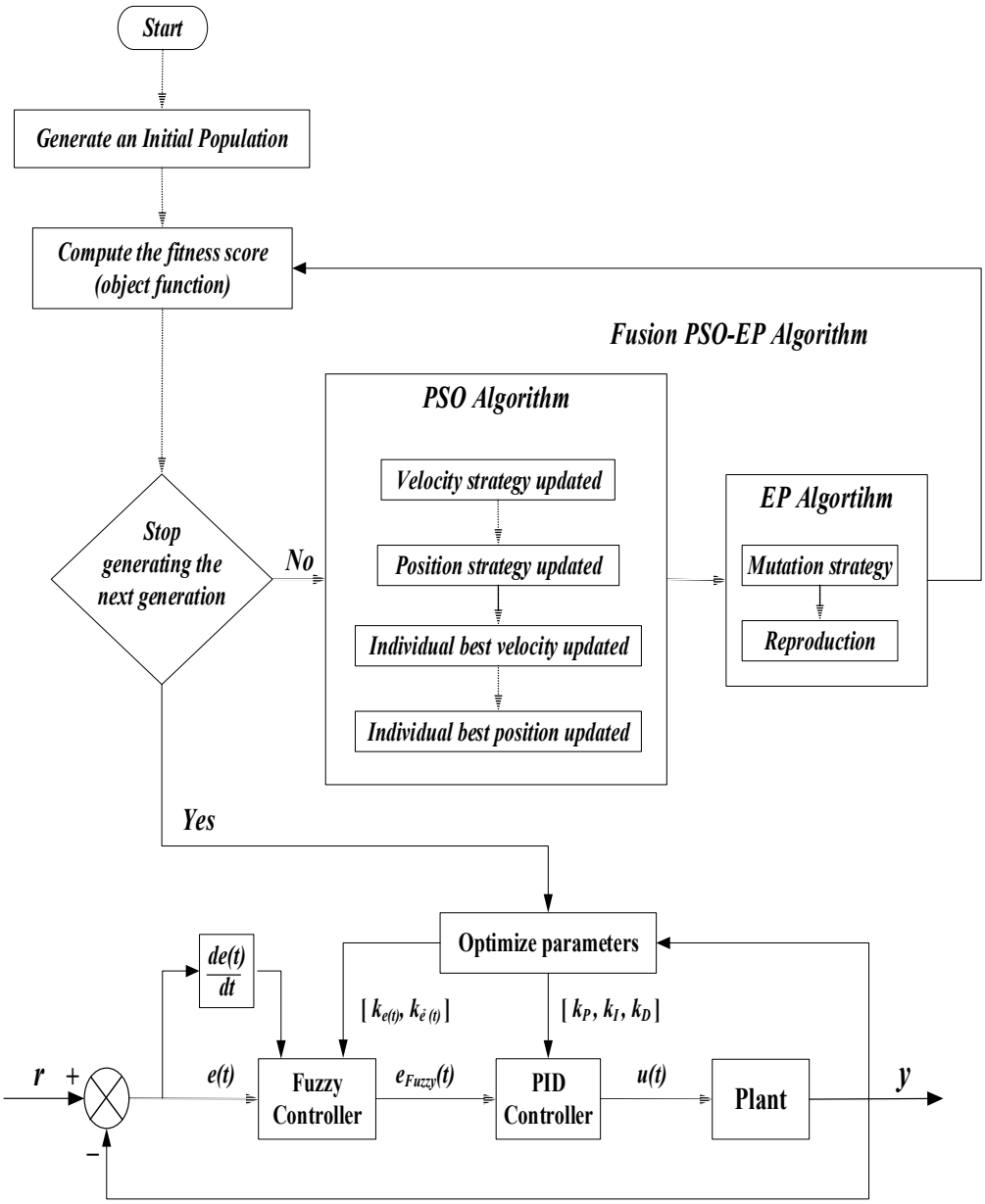


Figure 2. Fusion particle swarm optimization–evolutionary programming (PSO–EP) algorithm and reinforced fuzzy-PID (proportional integral derivative) controllers.

3. An Agricultural Tricopter Unmanned Aerial Vehicle (UAV) Model

3.1. Flight Hovering Mode

The UAVs possessing the property of vertical take off and landing (VTOL-UAVs) will have the formation like a helicopter or drone/quadrotor, hexa-copter or octo-copter. Nevertheless, more rotors lead to a more risky operation and higher power consumption. Hence, this paper presents a novel tricopter UAV that may be applied successfully to agriculture activities at any environment. Agricultural Tricopter model, based on the Newton–Euler formulation was developed and obtained from previous studies [22–24,31,41,42], as shown in Figure 3. The UAV was described using two left-hand axes—one

is the axes of the Earth coordinate system, and the other is a body frame. The positive x axis is aligned with the two front rotors (rotor 1 and 2) and the positive y axis is with the right (rotor 2). However, the positive z axis is directed downward. The roll φ, pitch θ, and yaw ψ triple pilot angles were with the positive x, y, and z axes, respectively. The tilt angle μ was determined by using the y–z coordinate axis.

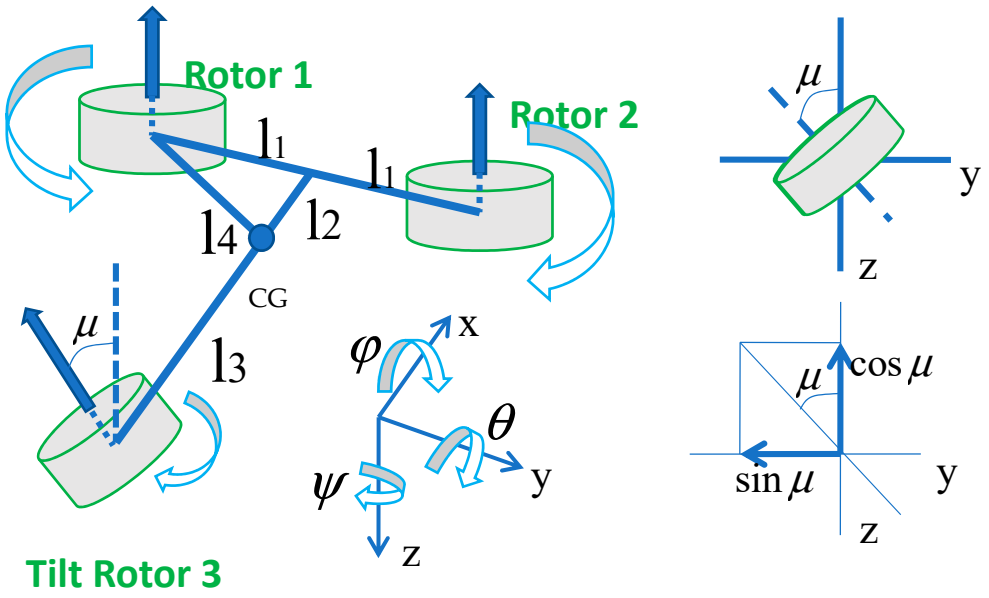


Figure 3. An agricultural tricopter model.

The dynamic of the agricultural tricopter has six degrees of freedom. One tilt angle and three rotor speeds are the four inputs to pilot control. The UAVs have commands similar to those of a traditional helicopter: collective, lateral, longitudinal, and yaw or pedal [22–24,31,41,42]. The commands are expressed as δcol (collection), δlat (roll control), δlon (pitch control), and δped (yaw control). The roll (φ) control was created because the two front rotors were maneuvered at different speeds. For example, when the speed of rotor 1 is low and the speed of rotor 2 is high, the UAV moves toward the left and vice versa. When the third tail rotor changes its velocity, pitch (θ) control is realized. Yaw (ψ) control is conducted by varying the tilt angle μ. The model notations are described in Table 2. The dynamics of the designed model of a UAV's rigid body were derived using Newton's laws and could be easily translated and rotated in a three-dimensional space. The motion of non-linear equations is expressed in Equations (9)–(11):

Force equations:

$$\begin{cases} F_x = m(\dot{u} - rv + qw) + mg\sin\theta \\ F_y = m(\dot{v} + ru + pw) - mg\sin\phi\cos\theta \\ F_z = m(\dot{w} + pv - qr) - mg\cos\phi\cos\theta \end{cases} \tag{9}$$

Moment equations:

$$\begin{cases} L = I_{xx}\dot{p} - qr(I_{yy} - I_{zz}) - I_{xz}r - I_{xz}pq \\ M = I_{yy}\dot{q} - pr(I_{zz} - I_{xx}) - I_{xz}(r^2 - p^2) \\ N = I_{zz}\dot{r} - pq(I_{xx} - I_{yy}) - I_{xz}qr \end{cases} \tag{10}$$

Rotation equations:

$$\begin{cases} \ddot{\phi} = \dot{p} = qr(\frac{I_{YY} - I_{ZZ}}{I_{XX}}) + \frac{L}{I_{XX}}u_2 \\ \ddot{\theta} = \dot{q} = pr(\frac{I_{ZZ} - I_{XX}}{I_{YY}}) + \frac{L}{I_{YY}}u_3 \\ \ddot{\psi} = \dot{r} = pq(\frac{I_{XX} - I_{YY}}{I_{ZZ}}) + \frac{L}{I_{ZZ}}u_4 \end{cases} \tag{11}$$

Table 2. The agricultural tricopter notation.

	The Agricultural Tricopter Notation
F_X, F_Y, F_Z	The External Forces
L, M, N	The external moments
(u, v, w)	The translational velocities
(p, q, r)	The rotational velocities (angular velocities)
(Φ, θ, Ψ)	The rotational angles (Roll, Pitch, Yaw)
(I_{xx}, I_{YY}, I_{ZZ})	The rotational inertias
u_2, u_3, u_4	The channel inputs (Roll, Pitch, Yaw)

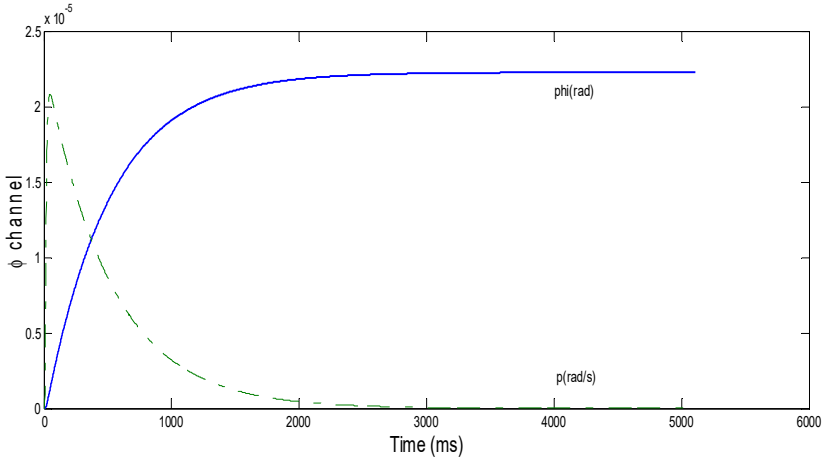
The hovering mode is the flying mode where the object hangs in the air. This phenomenon occurs when a physical force against the attraction of the gravity is applied; thus the object is stable. The hovering state equilibrium point, named as trimpoint, must be found out because of the tricopter stability issue mentioned previously [41–44]. The Newton–Raphson method [45] that possesses the vital benefit is rapid quadratic convergence, is the superior selection to calculate the trimpoint of the tricopter non-linearization model. This model has four equations as demonstrated in Equation (12):

$$\begin{cases} U_X = l_1 b_1 (\Omega_1{}^2 - \Omega_2{}^2) \\ U_T = b_1 (\Omega_1{}^2 + \Omega_2{}^2) + b_2 \Omega_3{}^2 \cos\mu \\ U_Y = l_2 b_1 (\Omega_1{}^2 + \Omega_2{}^2) - l_3 b_2 \Omega_3{}^2 \cos\mu \\ U_Z = -l_4 (d_1 \Omega_1{}^2 - d_2 \Omega_2{}^2) + l_3 (d_3 \Omega_3{}^2 + b_3 \Omega_3{}^2 \sin\mu) \end{cases} \tag{12}$$

where b is thrust factor, d is drag factor. $\Omega_1, \Omega_2, \Omega_3$ are the rotor speeds of three rotorcrafts, respectively and μ is the tail tilt angle. Assume that the thrust $T_i = b\Omega_i{}^2$ and torque $Q_i = d\Omega_i{}^2$ is constant mean that b and d factor are fixed. U_T is the total thrust of UAV in the z-axis. U_X, U_Y, U_Z are roll, pitch and yaw moments, respectively. Consequently, the system is asymptotically stable. Therefore, the linear quadratic regulator (LQR) control, which is an appropriate state feedback controller, is exploited to warrant and to keep the system in stable state.

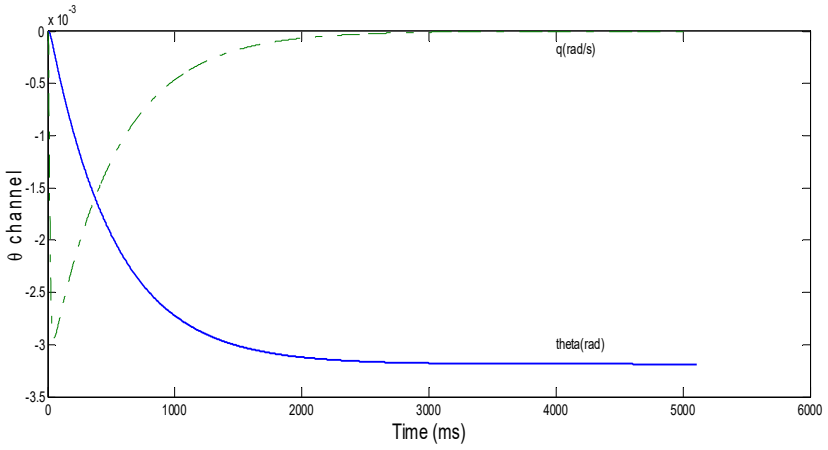
3.2. Linear Quadratic Regulator (LQR) Controller Design

The Tricopter system must be asymptotically stable to maintain the hovering mode at all the operation time. Hence, the continuous system exists a state feedback control with the input is $u(t) = -k.x(t)$. The feedback gain matrix K is solved by using the Riccati equation $A^T P + PA + Q - PBR^{-1}B^T P = 0$. If $k = -R^{-1}B^T P$, then the close loop system is asymptotically stable as shown in Figure 4.

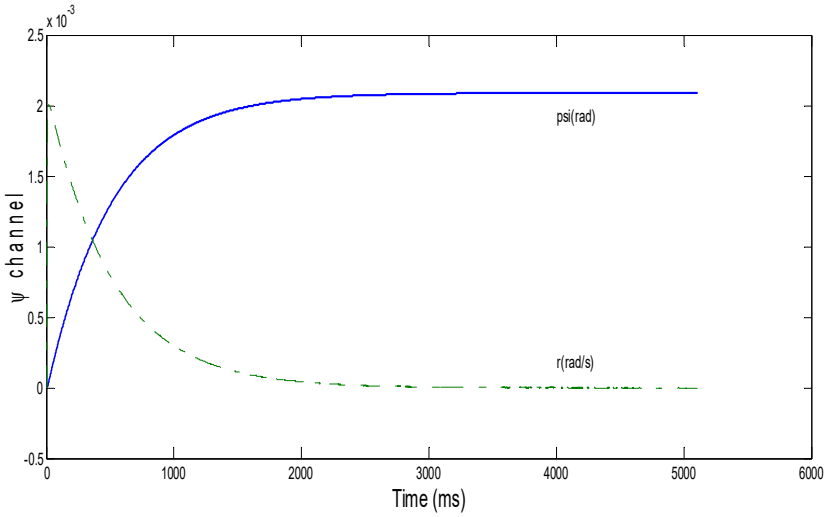


(**a**) Roll channel

Figure 4. *Cont.*



(**b**) Pitch channel



(**c**) Yaw channel

Figure 4. Tricopter hovering mode with linear quadratic regulator (LQR) control in each channel: (**a**) roll, (**b**) pitch and (**c**) yaw.

4. Numerical Simulation Results

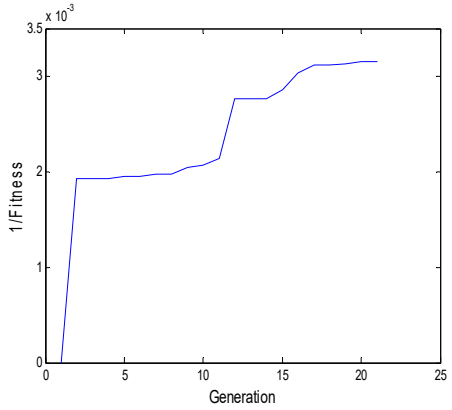
The object simulation parameters that are listed in Table 3, were derived from previous studies [21,22,29]. However, the value of the tilt angle μ should be negative in the initial condition to maintain system stability. The simulation pilot controls are demonstrated in each channel RPY (roll, pitch, and yaw). The speed of each rotorcraft that was determined using the Newton–Raphson method is as follows: Ω_1 = 8418 rpm, Ω_2 = 8418 rpm, Ω_3 = 8676 rpm. Moreover, the tilt angle μ is −0.0210 rad (−2.407°).

The parameters of the fusion PSO–EP algorithm are provided when the number of population generations for each of the roll, pitch, and yaw angle pilot controls is set as 20 generations. The EP mutation rate *mr* is 0.1 and the deceleration factor of the particle σ = 1. The weighting factor [α_1, α_2, α_3, α_4, α_5, α_6, α_7] = [30, 25, 10, 5, 7, 5, 3], and PID gains $k_{PID} \in [0, 50]$. The simulation sampling time is 0.01. The fuzzy-PID controller gains obtained after tuning through the Ziegler–Nichols method are as follows: k_P = 0.6, k_I = 0.34, k_D = 0.28, $k_{e(t)}$ = 1.23, $k_{de(t)}$ = 1.58 and the fitness perform index IAE = 0.9208.

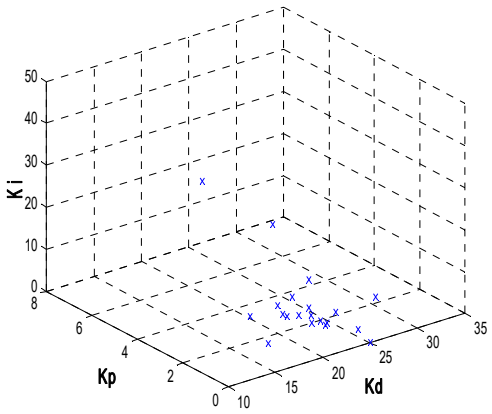
Table 3. Numerical simulation parameters.

Annotation	Parameter
I_{xx}	0.0057211 kg·m^2
I_{yy}	0.073933 kg·m^2
I_{zz}	0.012545 kg·m^2
$b_1 = b_2$	0.00013678
$d_1 = d_2$	0.000024323
l_1	19 cm
l_2	12.2 cm
l_3	23 cm
l_4	22.5 cm
Mass = m	0.84 kg

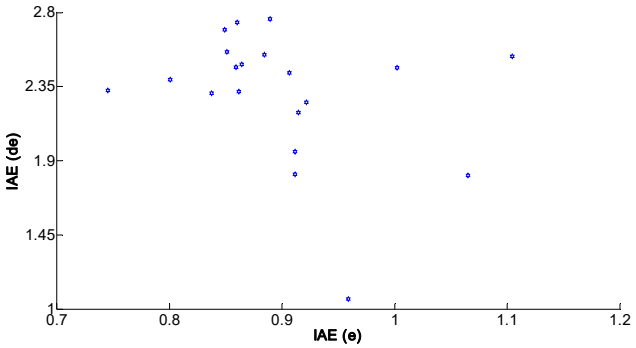
According to the pilot angle control of a UAV in the roll (Figure 5), Pitch (Figure 6), and Yaw (Figure 7) axes of each channel, we exhibited a useful system model as follows: (a) The fitness function of the proposed controller design. The fitness function is smaller (or the inverse fitness is larger) mean that the optimal convergence is better. (b: top right) PID parameter in three dimensions convergence space. The convergence of PID gains is chosen in the range [0–50]. (c: middle) The minimum of IAE(e) and IAE(de) in two dimensions. The convergences of fuzzy gains are selected and performed in 2D. (d) Response of the controllers.



(**a**) Fitness function

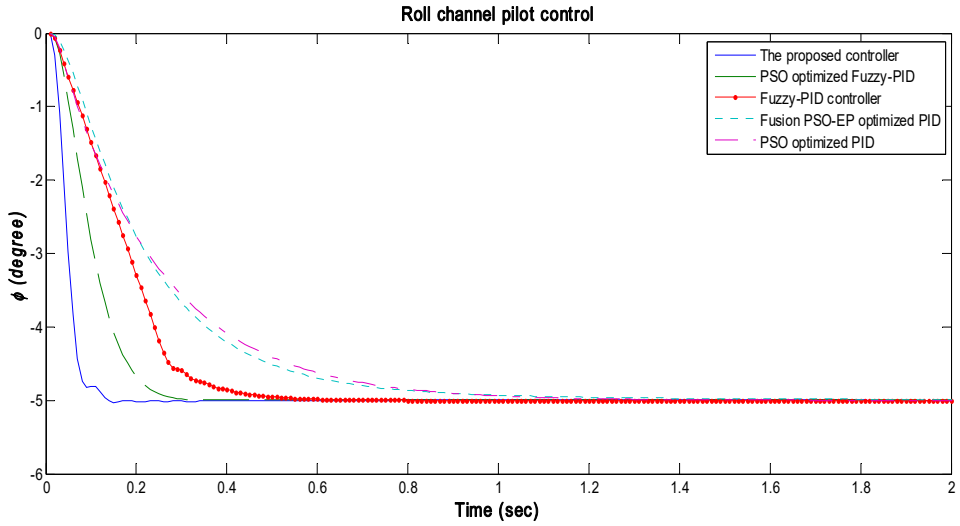


(**b**) PID parameters in space



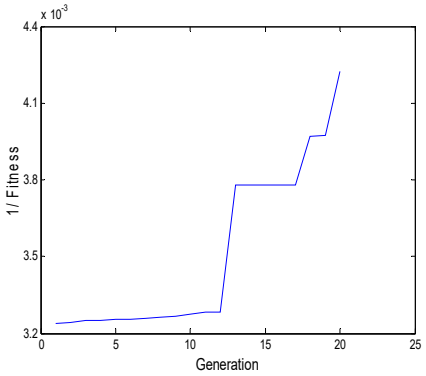
(**c**) Integral of the absolute error (IAE) of fuzzy gains

Figure 5. *Cont*.

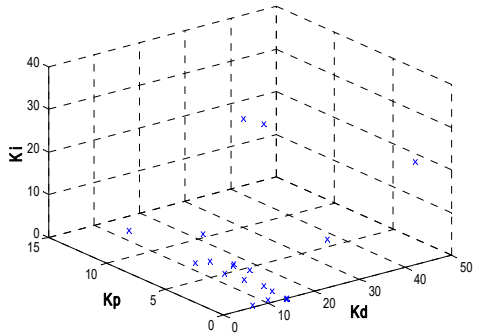


(**d**) Roll—the proposed controller design vs. the others

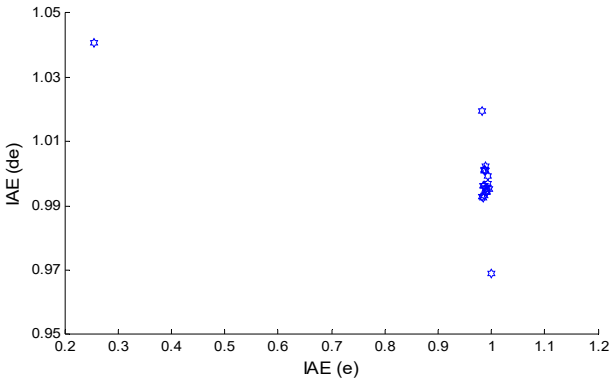
Figure 5. Roll channel pilot control. (**a**: top left) Fitness function of the proposed controller design. (**b**: top right) PID parameter in 3-Dimensions convergence space. (**c**: middle) The minimum of IAE(e) and IAE(de) in 2-Dimensions. (**d**: bottom) Controllers of the roll angle response.



(**a**) Fitness function

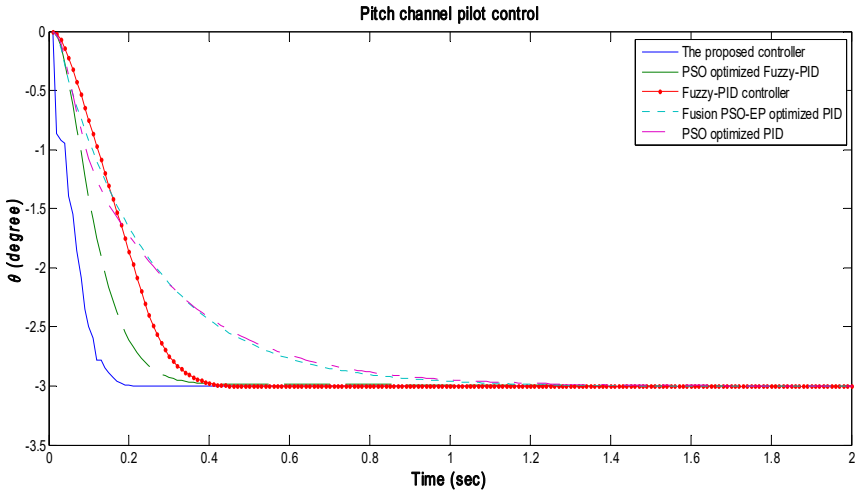


(**b**) PID parameters in space



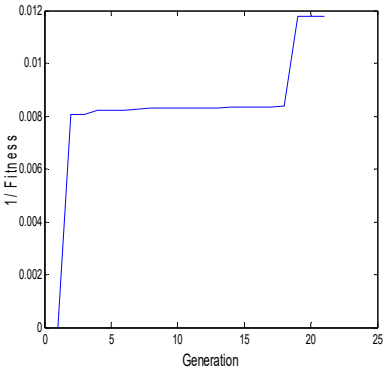
(**c**) IAE of fuzzy gains

Figure 6. *Cont*.

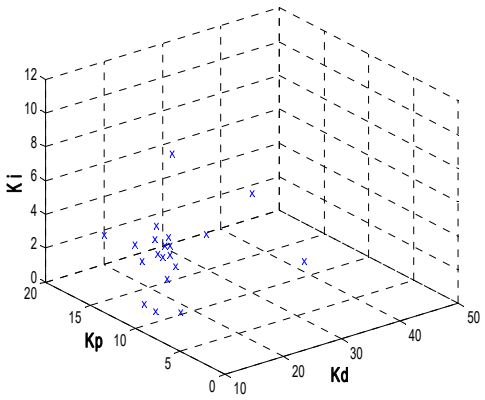


(d) Pitch—the proposed controller design vs. the others

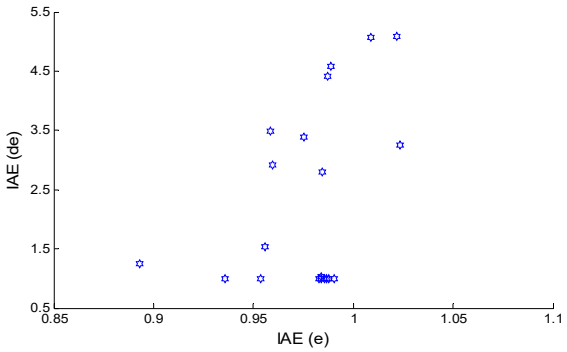
Figure 6. Pitch channel pilot control. (**a**: top left) Fitness function of the proposed controller design. (**b**: top right) PID parameter in 3-D convergence space. (**c**: middle) The minimum of IAE(e) and IAE(de) in 2-D. (**d**: bottom) Controllers of the pitch angle response.



(**a**) Fitness function

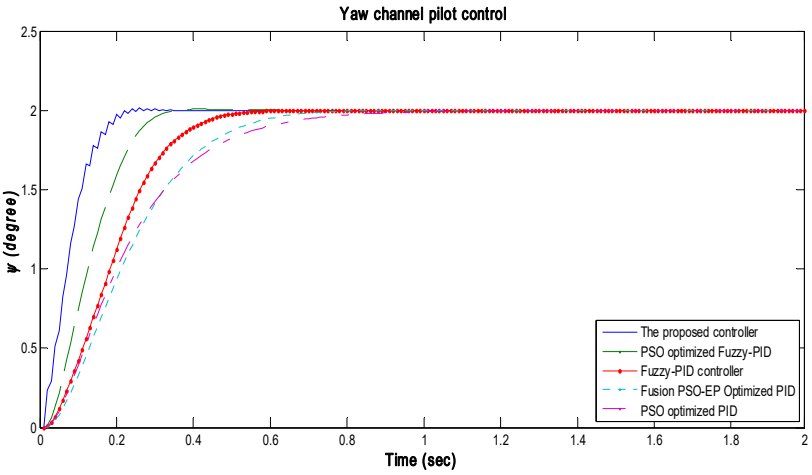


(**b**) PID parameters in space



(**c**) IAE of fuzzy gains

Figure 7. *Cont.*



(**d**) Yaw—the proposed controller design vs. the others

Figure 7. Yaw channel pilot control. (**a**: top left) Fitness function of the proposed controller design. (**b**: top right) PID parameter in 3-Dimensions convergence space. (**c**: middle) The minimum of IAE(e) and IAE(de) in 2-Dimensions. (**d**: bottom) Controllers of the yaw angle response.

5. Discussion

The comparison between the results obtained using five controller strategies integrated with the fusion EP-PSO algorithm is presented in Table 4. The five controller strategies that were implemented to pilot the tricopter attitude channels roll, pitch and yaw are: PSO-optimized PID gains, fusion PSO–EP optimized PID gains, fuzzy-PID controller, PSO-optimized fuzzy-PID gains, and the proposed methodology: fusion PSO–EP algorithm reinforced fuzzy-PID controller.

Table 4. The response performances in comparison.

	Roll (ϕ)		*Pitch (θ)*		*Yaw (ψ)*	
Controller Design (Second)	**Rising Time**	**Settling Time**	**Rising Time**	**Settling Time**	**Rising Time**	**Settling Time**
PSO optimal PID	0.412	1.238	0.433	1.189	0.386	0.782
Fusion EP-PSO optimal PID	0.381	1.192	0.433	1.191	0.384	0.652
Fuzzy-PID	0.327	0.489	0.253	0.391	0.318	0.541
PSO optimal Fuzzy-PID	0.218	0.283	0.195	0.377	0.196	0.362
The proposed method	0.122	0.183	0.118	0.194	0.156	0.201

Firstly, the PSO algorithm performs well when applied to find out the three optimal P, I and D control gains. The responses are just after 1 second. Secondly, the fusion EP-PSO is exploited to tune PID control, the responses of this method are faster than the original PSO. Next, the fuzzy-PID controller is designed by the Ziegler–Nichols method as mentioned above. This design display the advantage of the fuzzy-PID control method in faster response and stable. Later, The fuzzy-PID control with five parameters: e, de, P, I and D is successfully optimized by using the fundamental PSO algorithm. This means that the PSO algorithm works well in 5 dimensions. The system responses are faster than 5%. Finally, the proposed fusion EP-PSO algorithm is implemented well to show the fastest responses just after 0.2 s. However, the responses perform the control trade-off in jagged/sawtooth phenomenon to achieve the faster target but neither error nor overshoot. Moreover, the steady-state error is almost zero mean that the errors, between the input command and output response, of the proposed controllers design to tricopter models are tiny. Generally, all of the controller designs are achieved the high-quality performance without overshoot, high precision and high reliability.

6. Conclusions

Five controller designs were applied to a tricopter pilot system in this research work. The proposed fusion PSO–EP algorithm is reinforced to determining the parameters of the fuzzy-PID controllers, it then fruitfully implemented the tricopter attitude models in each pilot channel: roll, pitch, and yaw. The proposed controller design provides the optimal pilot performance in only after 0.2 s when compared with the other controller designs. The accomplished performance reveals that the proposed controller provides the fastest response, no overshoot, high reliability, and high stability. The efficient controller schemes for the non-linear models of an agricultural tricopter are successfully achieved. In further work, the disturbance attack to the system models would be also taken into consideration and the most imperative need is implementing the proposed algorithms to real-time systems.

Author Contributions: Methodology, H.K.T. and J.-S.C.; Supervision, H.K.T.; Visualization, H.K.T.; Resources, J.-S.C.; Validation, J.-S.C.; Writing—Review and Editing, V.-H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was financially supported by the "Center for Cyber-physical System Innovation" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan and the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-218-028 and MOST 109-2221-E-218-020-MY2. This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2016.18. The APC was funded by Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-218-028 and MOST 109-2221-E-218-020-MY2.

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*, 1st ed.; Springer: Berlin, Germany, 2003.
2. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks IV, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
3. Clerc, M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In Proceedings of the ICEC, Washington, DC, USA, 6–9 July 1999; pp. 1951–1957.
4. Eberhart, R.C.; Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; pp. 84–88.
5. Bergh, F.; Engelbrecht, A.P. A study of particle swarm optimization particle trajectories. *Inf. Sci.* **2006**, *176*, 937–971.
6. Bergh, F.; Engelbrecht, A.P. A Cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 225–239.
7. Juang, C.F. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Syst. Man Cybern. B* **2004**, *34*, 997–1006. [CrossRef] [PubMed]
8. Huang, T.; Mohan, A.S. A hybrid boundary condition for robust particle swarm optimization. *IEEE Antennas Wirel. Propag. Lett.* **2005**, *4*, 112–117. [CrossRef]
9. Li, L.L.; Wang, L.; Liu, L.L. An effective hybrid PSOSA strategy for optimization and its application to parameter estimation. *Appl. Math. Comput.* **2006**, *179*, 135–146. [CrossRef]
10. Thaís de Fátima Araújo and Wadaed Uturbey. Performance assessment of PSO, DE and hybrid PSO-DE algorithms when applied to the dispatch of generation and demand. *Int. J. Electr. Power Energy Syst.* **2013**, *7*, 205–217.
11. Ali, A.F.; Tawhid, M.A. A Hybrid PSO and DE Algorithm for Solving Engineering Optimization Problems. *Appl. Math. Inf. Sci.* **2016**, *10*, 431–449. [CrossRef]
12. Wu, X.D.; Bai, W.B.; Xie, Y.; Sun, X.Z.; Deng, C.C.; Cui, H.T. A hybrid algorithm of particle swarm optimization, metropolis criterion and RTS smoother for path planning of UAVs. *Appl. Soft Comput.* **2018**, *73*, 735–747. [CrossRef]

13. Huang, C.; Fei, J.Y. UAV Path Planning Based on Particle Swarm Optimization with Global Best Path Competition. *Int. J. Pattern Recognit. Artif. Intell.* **2018**, *32*, 1859008. [CrossRef]
14. Liu, H.H.; Chang, L.C.; Li, C.W.; Yang, C.H. Particle Swarm Optimization-Based Support Vector Regression for Tourist Arrivals Forecasting. *Comput. Intell. Neurosci.* **2018**. [CrossRef]
15. Chen, C.L.; Lin, Y.L.; Feng, Y.C. Optimization of large-scale economic dispatch with valve-point effects using a modified hybrid PSO-DSM approach. *J. Mar. Sci. Technol.* **2018**. [CrossRef]
16. Hsu, C.I.; Wu, S.P.J.; Chiu, C.C. A Hybrid Swarm Intelligence Approach for Blog Success Prediction. *Int. J. Comput. Intell. Syst.* **2019**, *12*, 571–579. [CrossRef]
17. Sanchez, E.N.; Becerra, H.M.; Velez, C.M. Combining fuzzy, PID and regulation control for an autonomous mini-helicopter. *Inf. Sci.* **2007**, *177*, 1999–2022. [CrossRef]
18. Passino, K.M.; Yurkovich, S. *Fuzzy Control*; Addison-Wesley: Boston, MA, USA, 1998.
19. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. *Robot. Modeling and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
20. Precup, R.E.; Preitl, S. PI-Fuzzy controllers for integral plants to ensure robust stability. *Inf. Sci.* **2007**, *177*, 4410–4429. [CrossRef]
21. Juang, Y.T.; Chang, Y.T.; Huang, C.P. Design of fuzzy PID controllers using modified triangular membership functions. *Inf. Sci.* **2008**, *178*, 1325–1333. [CrossRef]
22. Yoo, D.W.; Oh, H.D.; Won, D.Y.; Tahk, M.J. Dynamic modeling and stabilization techniques for tri-rotor unmanned aerial vehicles. *Int. J. Aeronaut. Space Sci.* **2010**, *11*, 167–174. [CrossRef]
23. Yoon, S.; Lee, S.J.; Lee, B.; Kim, C.J.; Lee, Y.J.; Sung, S.K. Design and flight test of a small Tri-rotor unmanned vehicle with a LQR based onboard attitude control system. *Int. J. Innov. Comput. Inf. Control.* **2013**, *9*, 2347–2360.
24. Chiou, J.S.; Tran, H.K.; Peng, S.T. Attitude control of a single tilt tri-rotor UAV system: Dynamic modeling and each channel's nonlinear controllers design. *Math. Probl. Eng.* **2013**, 6. [CrossRef]
25. Sabatini, R.; Rodríguez, L.; Kaharkar, A.; Bartel, C.; Shaid, T.; Zammit-Mangion, D. Low-cost navigation and guidance systems for unmanned aerial vehicles—Part 2: Attitude determination and control. *Annu. Navig.* **2013**, *20*, 97–126. [CrossRef]
26. Russo, A.; Invernizzi, D.; Giurato, M.; Lovera, M. Adaptive augmentation of the attitude control system for a multirotor UAV. *Eur. Conf. Aerosp. Sci.* **2017**. [CrossRef]
27. Njinwoua, B.J.; Wouwer, A.V. Cascade attitude control of a quadcopter in presence of motor asymmetry. *IFAC-PapersOnLine* **2018**, *51*, 113–118. [CrossRef]
28. Koch, W.; Mancuso, R.; West, R.; Bestavros, A. Reinforcement Learning for UAV Attitude Control. *ACM Trans. Cyber-Phys. Syst.* **2019**. [CrossRef]
29. Burggräf, P.; Martínez, A.R.P.; Roth, H. Quadrotors in factory applications: Design and implementation of the quadrotor's P-PID cascade control system. *SN Appl. Sci.* **2019**. [CrossRef]
30. Mishra, S.; Rakstad, T.; Zhang, W. Robust Attitude Control for Quadrotors Based on Parameter Optimization of a Nonlinear Disturbance Observer. *ASME J. Dyn. Sys. Meas. Control* **2019**, *141*, 081003. [CrossRef]
31. Tran, H.K.; Chiou, J.S.; Nguyen, T.N.; Vo, T. Adaptive Fuzzy Control Method for a Single Tilt Tricopter. *IEEE Access* **2019**, *7*, 161741–161747. [CrossRef]
32. Wu, X.H.; Song, S.M. Consistency Monitoring for Spacecraft Attitude Estimators. *J. Aerosp. Eng.* **2019**, 32. [CrossRef]
33. Brescianini, D.; D'Andrea, R. Tilt-Prioritized Quadrocopter Attitude Control. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 376–387. [CrossRef]
34. Zhang, C.; Kovacs, J.M. The application of small unmanned aerial systems for precision agriculture: A review. *Precis. Agric.* **2012**, *13*, 693–712. [CrossRef]
35. Hassanalian, M.; Abdelkefi, A. Classifications, applications, and design challenges of drones: A review. *Prog. Aerosp. Sci.* **2017**, *91*, 99–131. [CrossRef]
36. Rao Mogili, U.M.; Deepak, B.B.V.L. Review on Application of Drone Systems in Precision Agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [CrossRef]
37. Shakhatreh, H.; Sawalmeh, A.; Al-Fuqaha, A.I.; Dou, Z.; Almaita, E.; Khalil, I.M.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles: A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2018**. [CrossRef]

38. Yao, H.; Qin, R.; Chen, X. Unmanned Aerial Vehicle for Remote Sensing Applications—A Review. *Remote Sens.* **2019**, *11*, 1443. [CrossRef]
39. Martins, F.G. Tuning PID Controllers using the ITAE Criterion. *Int. J. Eng. Educ.* **2005**, *21*, 867–873.
40. Tan, W.; Liu, J.; Chen, T.; Marquez, H.J. Comparison of some well-known PID tuning formulas. *Comput. Chem. Eng. J.* **2006**, *30*, 1416–1423. [CrossRef]
41. Stevens, B.L.; Lewis, F.L. *Aircraft Control. and Simulation*, 1st ed.; John Wiley: Hoboken, NY, USA, 1992.
42. Padfield, G.D. *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*, 2nd ed.; AIAA—American Institute of Aeronautics and Astronautics: Reston, USA, VA, 2007.
43. Carrillo, G.L.R.; Dzul, A.; Lozano, R. Hovering quad-rotor control: A comparison of nonlinear controllers using visual feedback. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 3159–3170. [CrossRef]
44. Lee, K.U.; Kim, H.S.; Park, J.B.; Choi, Y.H. Hovering control of a quadrotor. In Proceedings of the 12th International Conference on Control. Automation and Systems, Jeju island, Korea, 17–21 October 2012; pp. 162–167.
45. Deuflhard, P. *Newton Methods for Nonlinear Problems*, 1st ed.; Springer: Berlin, Germany, 2005.