



An Aerial Robot for Rice Farm Quality Inspection With Type-2 Fuzzy Neural Networks Tuned by Particle Swarm Optimization-Sliding Mode Control Hybrid Algorithm

Camci, Efe; Kripalan, Devesh Raju ; Ma, Linlu; Kayacan, Erdal; Ahmadiéh Khanesar, Mojtaba

Published in:
Swarm and Evolutionary Computation

Link to article, DOI:
[10.1016/j.swevo.2017.10.003](https://doi.org/10.1016/j.swevo.2017.10.003)

Publication date:
2017

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Camci, E., Kripalan, D. R., Ma, L., Kayacan, E., & Ahmadiéh Khanesar, M. (2017). An Aerial Robot for Rice Farm Quality Inspection With Type-2 Fuzzy Neural Networks Tuned by Particle Swarm Optimization-Sliding Mode Control Hybrid Algorithm. *Swarm and Evolutionary Computation*.
<https://doi.org/10.1016/j.swevo.2017.10.003>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

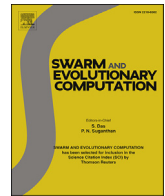
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm

Efe Camci^a, Devesh Raju Kripalani^a, Linlu Ma^a, Erdal Kayacan^{a,*},
Mojtaba Ahmadih Khanesar^b

^a School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore

^b Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Keywords:

Type-2 fuzzy neural networks
Particle swarm optimization
Sliding mode control
Unmanned aerial vehicle
System identification
Control

ABSTRACT

Agricultural robots, or agrobots, have been increasingly adopted in every aspect of farming from surveillance to fruit harvesting in order to improve the overall productivity over the last few decades. Motivated by the compelling growth of the agricultural robots in modern farms, in this work, an autonomous quality inspection over rice farms is proposed by employing quadcopters. Real-time control of these vehicles, however, is still challenging as they exhibit a highly nonlinear behavior especially for agile maneuvers. What is more, these vehicles have to operate under uncertain working conditions such as wind and gust disturbances as well as positioning errors caused by inertial measurement units and global positioning system. To handle these difficulties, as a model-free and learning control algorithm, type-2 fuzzy neural networks (T2-FNNs) are designed for the control of a quadcopter. The novel particle swarm optimization-sliding mode control (PSO-SMC) theory-based hybrid algorithm is proposed for the training of the T2-FNNs. In particular, the continuous version of PSO is adopted for the identification of the antecedent part of the T2-FNNs while the SMC-based update rules are utilized for the online learning of the consequent part during control. In the virtual environment, the quadcopter is expected to perform an autonomous flight including agile maneuvers such as steep turning and sudden altitude changes over a rice terrace farm in Longsheng, China. The simulation results for the T2-FNNs are compared with the outcome of conventional proportional-derivative (PD) controllers for different case studies. The results show that our method decreases the trajectory tracking integral squared error by %26 over PD controllers in the ideal case, while this ratio goes up to %95 under uncertain working conditions.

1. Introduction

Modern methods in agriculture have evolved tremendously over the last few decades with the advent of automation to increase crop yields in shorter time while utilizing less labour costs and resources. Motivated by the compelling growth of the agricultural robots in modern farms, unmanned aerial vehicles (UAVs) such as quadcopters are becoming increasingly popular to assist farmers for a possible increase in crop yields while decreasing the crop damage [1]. There are already successful applications to perform essential functions such as conducting air quality check, monitoring the health of crops and soil through visual imaging, or spraying fertilizers in a timely and controlled manner [2]. Today, the greater role of quadcopters in agriculture presents itself as a

challenging, yet interesting control problem.

Quadcopters are highly nonlinear, multi-input-multi-output, strongly coupled, open-loop unstable, and under-actuated systems. Therefore, their control is a fundamentally intricate problem. Moreover, in most cases, they are required to operate in subtle environments which may possibly include strong winds or unexpected gusts. Furthermore, the position and orientation information of a quadcopter is generally provided by global positioning system (GPS) and inertial measurement units (IMUs) which may yield significant noise. In this manner, the conventional, model-based methods such as proportional-integral-derivative (PID) controller [3,4], linear quadratic regulator (LQR) [5], or model predictive control (MPC) [6,7] suffer from poor performance in time-varying working conditions and noisy measure-

* Corresponding author.

E-mail addresses: efe001@e.ntu.edu.sg (E. Camci), devesh0004@e.ntu.edu.sg (D.R. Kripalani), lma004@e.ntu.edu.sg (L. Ma), erdal@ntu.edu.sg (E. Kayacan), makh@dtu.dk (M.A. Khanesar).

<https://doi.org/10.1016/j.swevo.2017.10.003>

Received 27 March 2017; Received in revised form 25 July 2017; Accepted 7 October 2017

Available online XXX

2210-6502/© 2017 Elsevier B.V. All rights reserved.

ments. Therefore, quadcopters need a robust and reliable controller which is capable of handling noise as well as uncertain working conditions while achieving a high accuracy in trajectory tracking [8–10].

As one of the model-free approaches, fuzzy logic has caught a respectable attention since it was proposed by Lotfi A. Zadeh in 1965 [11]. It makes use of linguistic variables to mimic an expert/skilled human operator's way of thinking. As opposed to Boolean logic which operates on the discrete values of either absolute 0 (false) or 1 (true), fuzzy logic is based on the intermediate degrees of the truth in a continuous range [0, 1]. Over the years, fuzzy logic has been combined together with artificial neural networks, and this combination is called as *fuzzy neural networks (FNNs)*. FNNs have been proven to be a superior method for system identification [12–14] and nonlinear control [15–19]. In all these works, FNNs serve as a preferred choice over the conventional, model-based methods as they are competent on dealing with uncertainties such as instrumentation inaccuracies or changing environmental conditions.

In the nature of FNNs, a suitable parameter update algorithm is a necessity for learning from input-output data and adapting for changing working conditions. On one hand, the optimization techniques such as genetic algorithm [20], artificial bee colony (ABC) [21,22], ant colony optimization (ACO) [23–25], and particle swarm optimization (PSO) [26–28] are widely recognized, biologically-inspired, metaheuristic parameter update methods. On the other hand, sliding mode control (SMC) [29,30], gradient descent (GD), and extended Kalman filter (EKF) are the parameter update methods which are based upon a well-established mathematical background for training FNNs [31]. While the former group assists on simplifying the optimization process via metaheuristic search, the latter provides a mathematical basis and stability analysis for the training. In this work, we propose to exploit the beneficial features of both groups. Among the first group, we focus on PSO as it is the simplest in the context of our application considering the number of parameters to be set for its operation and the information sharing mechanism between the particles. In addition, since PSO has shown a substantial promise as a robust method for solving continuous nonlinear optimization problems with a more stable convergence as compared to the other stochastic techniques [31], it is a perfect fit for our application. Likewise, among the second group, we capitalize on SMC since it has a robust and relatively simpler structure as compared to the GD-based methods. Moreover, since it has shown a respectable success together with fuzzy logic for a number of control applications with the presence of bounded disturbances or uncertainties [32,33], it is a qualified candidate for our application.

Overall, in this work, we propose to train FNNs effectively using a novel PSO-SMC hybrid learning algorithm. We optimize their performance to achieve a desirable translational and rotational control of the quadcopter for agricultural purposes. Instead of using the conventional type-1 fuzzy neural network (T1-FNN) [34], we employ its type-2 counterpart; as the latter is claimed to give superior responses especially when the level of uncertainty is high [12,15–18]. In the proposed method, the premise part of the T2-FNN is identified by PSO, whereas, the consequent part is tuned by SMC theory-based parameter update rules. In the end, an extensive comparison on the overall performance of the proposed algorithm and the other available methods is presented in order to demonstrate the pros and cons of our novel algorithm. For the identification part, two other swarm-based methods, namely, ABC and ACO, are chosen as competitors; while for the control part, conventional proportional-derivative (PD) controllers and T1-FNNs are selected as opponents.

To the best of our knowledge, this is the first time T2-FNNs:

- are trained by the novel PSO-SMC hybrid algorithm while PSO for system identification and SMC for control.
- are applied on the control of both translational and rotational dynamics of quadcopter.

The organization of this work is as follows: The specific problem is formulated in Section 2. The details of the quadcopter model are presented in Section 3. The design procedure of the T2-FNNs and the parameter update rules are introduced in Sections 4 and 5, respectively. The simulation results and discussions on the performance of the proposed method are presented in Section 6. Lastly, the concluding remarks and future works are given in Section 7.

2. Problem statement

According to our scenario, a quadcopter is employed for the quality inspection over the rice terrace farms in Longsheng, China. Quadcopters are fairly helpful for this purpose as they may provide autonomous, fast flights in a large area which may possibly be challenging for farm labourers with all the hills, narrow passages, and muddy parts included; thus, it saves both labour resources and time. However, the quality check over rice farms with a quadcopter requires a highly accurate trajectory tracking in order to collect useful samples from different parts of the area. Moreover, as the usual trajectories over rice farms may include agile maneuvers, steep turning, or sudden altitude changes due to the geometrical and topological features of these farms, it becomes an intriguing control problem for the quadcopter.

3. Quadcopter model

Quadcopter is an aerial vehicle which has 6 degrees of freedom (6-DoF) comprising of kinematics and dynamics in 3 axes. The former denotes the translational movement in 3 axes, X_E , Y_E , Z_E ; while the latter stands for the rotational motion along 3 axes, X_B , Y_B , Z_B . The mathematical model represents these two categories of motions while abiding by the assumption that the quadcopter has a rigid and symmetrical structure.

3.1. Kinematics

Two individual coordinate systems, namely *body-fixed* and *earth-inertial* frames, are set to represent the kinematics of the quadcopter in 3D space. The body-fixed frame is the coordinate frame whose origin is located at the centre of gravity of the quadcopter. The x axis (X_B) is pointing towards the rotor 1, while the z axis (Z_B) is pointing in the direction of the thrust force created by propellers as in Fig. 1. Consequently, the y axis (Y_B) follows the orientation defined by the right-hand rule. As regards to the earth-inertial frame, its origin is located on the ground while the z axis (Z_E) pointing towards the opposite direction of the center of Earth. Using the aforementioned two coordinate frames, the location of the quadcopter can be represented as x , y , and z pertaining to the earth-inertial frame. The relationship between the two coordinate frames is given by the transformation matrix as follows (c:cos, s:sin):

$$R_{EB} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (1)$$

where Euler angles, roll (ϕ), pitch (θ), and yaw (ψ) are defined as the tilting angles of the body-fixed frame with regard to the earth-inertial frame in x , y , and z directions.

3.2. Dynamics

Newton-Euler equation is exploited for representing the dynamics of the quadcopter since the deformation of the quadcopter is neglected. The forces and moments generated by the four actuators can be approx-

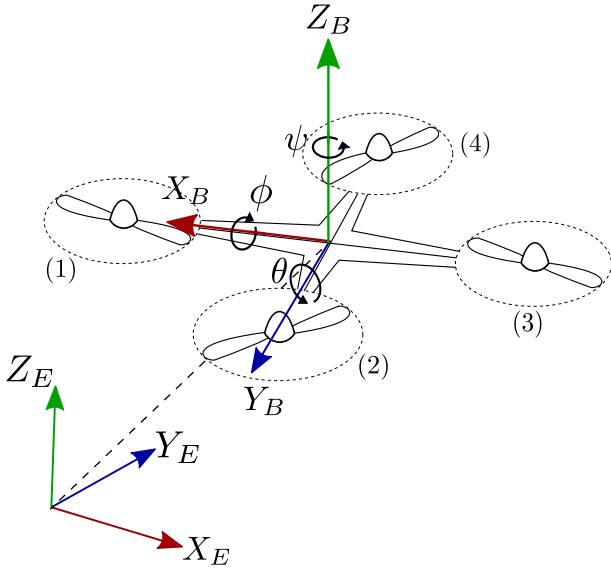


Fig. 1. Quadcopter schematic diagram.

imated as follows:

$$\begin{aligned} F_i &= k_t \Omega_i^2 \\ \tau_i &= k_m \Omega_i^2 \end{aligned} \quad (2)$$

where F_i and τ_i stand for the state-status thrust force and the reaction torque for each rotor. The parameter Ω_i denotes the speed of each rotor. The symbols k_t and k_m are the constants to illustrate the thrust and drag factors. The overall actuation forces and torques in three directions can be acquired by using F_i and τ_i . They are mapped into the input matrix accordingly with l indicating the quadcopter's arm length. The control of the system can therefore be achieved by altering the actuators' speed computed by the inverse input mapping matrix which is given as follows:

$$\begin{bmatrix} F_{total} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} k_t & k_t & k_t & k_t \\ 0 & k_t l & 0 & -k_t l \\ -k_t l & 0 & k_t l & 0 \\ -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (3)$$

where τ_x , τ_y , and τ_z are the moments around axes X_B , Y_B , Z_B . The total thrust force in Z_B direction is denoted as:

$$F_{total} = \sum_{i=1}^4 F_i$$

After relating the forces and moments with the rotor speeds, the equations of motion are presented as the simplified version of the equations in Ref. [35]:

$$\begin{aligned} \ddot{\phi} &= \dot{\theta} \dot{\psi} \frac{(I_{yy} - I_{zz})}{I_{xx}} + \frac{\tau_x}{I_{xx}} \\ \ddot{\theta} &= \dot{\phi} \dot{\psi} \frac{(I_{zz} - I_{xx})}{I_{yy}} + \frac{\tau_y}{I_{yy}} \\ \ddot{\psi} &= \dot{\theta} \dot{\phi} \frac{(I_{xx} - I_{yy})}{I_{zz}} + \frac{\tau_z}{I_{zz}} \end{aligned} \quad (4)$$

$$\begin{aligned} \ddot{x} &= (\cos(\phi) \sin(\theta) \cos(\psi) + \sin(\theta) \sin(\psi)) \frac{F_{total}}{m} \\ \ddot{y} &= (\cos(\phi) \sin(\theta) \cos(\psi) - \sin(\theta) \sin(\psi)) \frac{F_{total}}{m} \\ \ddot{z} &= -g + (\cos(\phi) \cos(\theta)) \frac{F_{total}}{m} \end{aligned} \quad (5)$$

 Table 1
Quadcopter specifications.

Symbol	Description	Value
m	Mass (kg)	1.4
I_{xx}	Moment of inertia in x (kgm ²)	0.0075
I_{yy}	Moment of inertia in y (kgm ²)	0.0075
I_{zz}	Moment of inertia in z (kgm ²)	0.0130
k_t	Thrust coefficient (Ns ²)	3.1×10^{-5}
k_m	Moment coefficient (Nms ²)	7.5×10^{-7}
l	Moment arm length (m)	0.23

where the terms I_{xx} , I_{yy} , and I_{zz} stand for the moments of inertia in 3 axes. The term m indicates the total mass of the quadcopter while the term g represents the gravitational acceleration, which is fixed as 9.81 m/s^2 in this work.

In the equations of both translational and rotational accelerations, the terms related to the aerodynamic and gyroscopic effects are ignored because they require a substantial effort to be identified precisely by a number of experiments and simulations. Moreover, they are negligible at the operating conditions of interest. The other parameters of the quadcopter are determined as in Table 1 [36].

4. Design of type-2 fuzzy neural network structure

The control structure of the T2-FNN typically consists of the fuzzifier, inference, rule base, type reducer, and defuzzifier as its key components [31]. The r th rule in the rule base can be described as follows:

$$\text{Rule}_r : \text{IF } x_1 = \tilde{A}_1^r \text{ and } x_2 = \tilde{A}_2^r \text{ THEN } f_r = \sum_{i=1}^2 a_{ri} x_i + b_r \quad (6)$$

where two inputs (x_1 , x_2) correspond to e (error) and \dot{e} (derivative of error), \tilde{A}_1^r and \tilde{A}_2^r are type-2 fuzzy sets. The parameters in the consequent part of the rules are the rule weight, a_{ri} , and the rule bias, b_r . For each input, we have two membership functions resulting in 4 rules in total. The firing strength of the r th rule due to the lower and upper membership functions are given by:

$$\begin{aligned} \underline{w}_r &= \underline{\mu}_{\tilde{A}_1^r}(x_1) * \underline{\mu}_{\tilde{A}_2^r}(x_2) * \dots * \underline{\mu}_{\tilde{A}_I^r}(x_I) \\ \overline{w}_r &= \overline{\mu}_{\tilde{A}_1^r}(x_1) * \overline{\mu}_{\tilde{A}_2^r}(x_2) * \dots * \overline{\mu}_{\tilde{A}_I^r}(x_I) \end{aligned} \quad (7)$$

where $*$ represents the t-norm (prod) operator.

Finally, the output of the controller can be calculated using the normalized values of the firing strength, \tilde{w}_r and \overline{w}_r , weighted by the sharing parameter, q . The output of the network, y , is approximated as follows [36]:

$$y = q \sum_{r=1}^N f_r \tilde{w}_r + (1 - q) \sum_{r=1}^N f_r \overline{w}_r \quad (8)$$

where the normalized values of the lower and upper output signals from the second hidden layer of the network as follows:

$$\tilde{w}_r = \frac{\underline{w}_r}{\sum_{r=1}^N \underline{w}_r}, \quad \overline{w}_r = \frac{\overline{w}_r}{\sum_{r=1}^N \overline{w}_r} \quad (9)$$

The rule-base plays a key role in the decision-making process of the T2-FNNs which would guide the quadcopter to its desired translational and rotational configuration. In total, six T2-FNNs are required to achieve control in the translational (X , Y , Z) and rotational (Roll, Pitch, Yaw) control blocks respectively, as shown in Fig. 2. For each channel, e and \dot{e} are fed into the T2-FNNs as the two input variables, x_1 and x_2 respectively. Using Gaussian membership functions, two type-2 fuzzy sets with an uncertain standard deviation are defined for each input. The consequent parts of the fuzzy sets are initialized with random numbers as they are learned online afterwards.

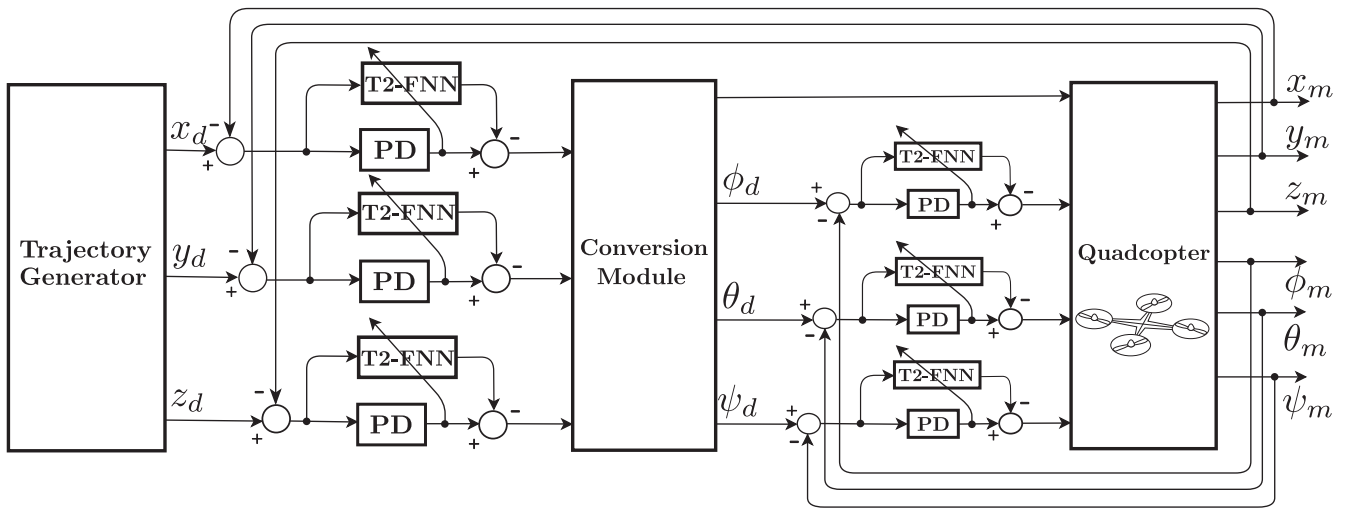


Fig. 2. Control architecture of the quadcopter.

5. Hybrid training algorithm

The parameter tuning of FNNs can be accomplished by a variety of the aforementioned training methods in Section 1. Nevertheless, the selection of the most suitable method has a significant impact on the control performance. In this work, the parameter tuning is achieved by two different methods, namely PSO-SMC, in a hybrid manner. The rationale behind this selection can be explained in two ends. On one hand, since the relation between the FNN output and the antecedent parameters is highly nonlinear, the calculation of the gradient of the cost function during optimization is fairly difficult. Therefore, the use of a gradient-free training method, namely *PSO*, is more promising for the first end. On the other hand, the computation-based methods can be well adopted for the training of consequent parameters because the relation between the FNN output and them is rather linear. Hence, it is possible to benefit from a computation-based method with a well-established mathematical stability analysis, namely *SMC*, for tuning the consequent parameters.

The hybrid training procedure can be divided into two key phases: system identification and control. During the identification phase, the antecedent parameters of the T2-FNNs are identified by using only PD controllers. The inputs of the identifier are e and \dot{e} , while the target values of the identifier are the PD controller outputs. Once the system identification is completed, the control phase is initialized with the identified parameters of the T2-FNNs. During this phase, the SMC parameter update rules are in effect to provide continuous online training during trajectory tracking.

5.1. PSO algorithm for system identification

System identification is a fundamental step for controller development in order to get a good grasp on quadcopter dynamics before directly flying the vehicle. It provides stable initial conditions for the quadcopter while avoiding unexpected, extreme responses for certain inputs that may arise during flight. As the quadcopter has fairly complex dynamics, we select various types of intelligently-chosen inputs in order to identify each parameter precisely. The inputs are designed based on the system identification inputs for flight vehicles in Ref. [37].

From the PSO point of view, we utilize single swarm optimization without higher order terms in dynamics of particles. This type of PSO might be one of the simplest versions of the original PSO available in literature. Current state-of-art presents a great amount of research in order to improve the performance of the original PSO. Utilizing higher-order PSO (HPSO) via higher-order terms in update equations to search

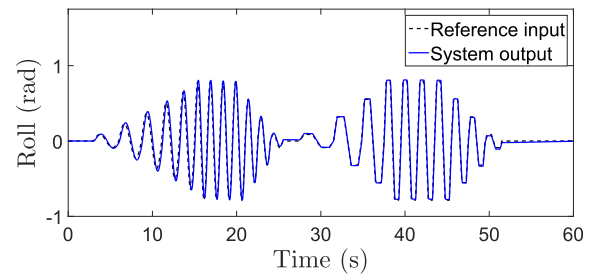


Fig. 3. System identification of the T2-FNN in Roll.

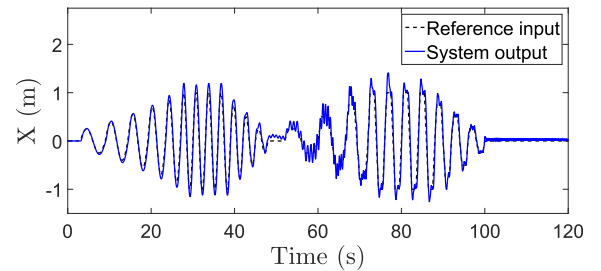


Fig. 4. System identification of the T2-FNN in X axis.

for a wider area and wider range of movements of particles [31], alleviating the premature convergence by perturbing the swarm movement [38] or using multiple swarms simultaneously [39], and hybridization of PSO with the other optimizers [40] are famous examples of promising research directions. Nevertheless, in order to account for the simplicity in the context of our application, we avoid the complex variants of PSO which might be demanding in the regard of computation effort. Specifically, we choose the continuous version of PSO (CPSO) [31].

The system identification procedure of the quadcopter essentially consists of two stages. In the first stage, the three T2-FNNs in the inner control loop for rotational dynamics are identified. In the second stage, the three T2-FNNs in the outer control loop for translational dynamics are subsequently identified in a similar manner. For the T2-FNNs in the inner loop, up to 45 degrees of sinusoidal and doublet-type inputs are used with the frequency range of 0 Hz–0.7 Hz. As for the T2-FNNs in the outer loop, up to 1 m of the same input types are utilized with the frequency range of 0 Hz–0.3 Hz. Figs. 3 and 4 show the respective

system output for the T2-FNN controllers in Roll and X channels during the system identification phase. The identification process is replicated ten times in order to ensure convergence and obtain generalized results.

During identification, a 72-dimensional search-space, which corresponds to 12 parameters (8 standard deviation and 4 center values) for each of the six T2-FNNs, is initially populated with a swarm of 8 random, potential solutions or *particles*. An iterative search is then conducted whereby each particle travels through the search-space with a dynamically adjusted velocity based on the experience of its own motion as well as the experience of other particles within the swarm as illustrated in Fig. 5. In this study, the search-space for the antecedent parameters are specified within the ranges $[-1, 1]$ and $[0, 1]$ for the centres and standard deviations, respectively. The dynamics of the l th dimension of the s th particle is formulated according to the following expression [31]:

$$\begin{aligned} \dot{x}_{sl}(t) &= v_{sl}(t) \\ \dot{v}_{sl}(t) &= (\beta - 1)v_{sl}(t) + \gamma_1(lbest_{sl}(t) - x_{sl}(t)) \\ &\quad + \gamma_2(gbest_{sl}(t) - x_{sl}(t)) \end{aligned} \quad (10)$$

$$x_{sl}(0) = x_{sl0}$$

$$v_{sl}(0) = v_{sl0}$$

where $s = 1, 2, \dots, N$ and $l = 1, 2, \dots, n$.

Throughout the optimization process, the best position of each particle or local best, *lbest*, and the best position of the swarm, global best, *gbest*, are iteratively calculated and updated based on minimizing the pre-defined objective function which is the integral squared error (ISE) in this case and given as follows:

$$ISE = \int_0^{\infty} e^2(t)dt \quad (11)$$

The constants γ_1 and γ_2 represent the exploitation weights which accelerate each particle toward its local best and global best positions. As regards to $rd()$, it represents random numbers (0–1) which stochastically weights the swarm’s exploration capabilities. Intuitively, low values allow particles to roam far from the target regions before being pulled back, while high values result in abrupt movements toward, or past, target regions. Furthermore, the suitable selection of the exploration weight (β) provides a balance between the global and local explorations, thus requiring less iteration on average to find sufficiently optimal solutions. In this study, a slightly greater importance is given to

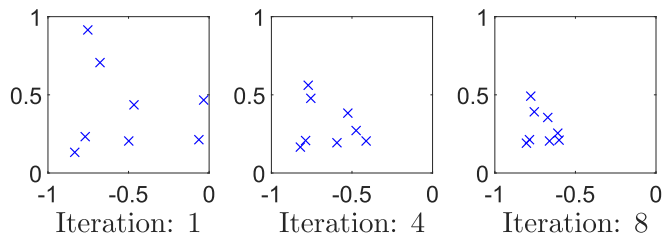


Fig. 5. Swarm movement.

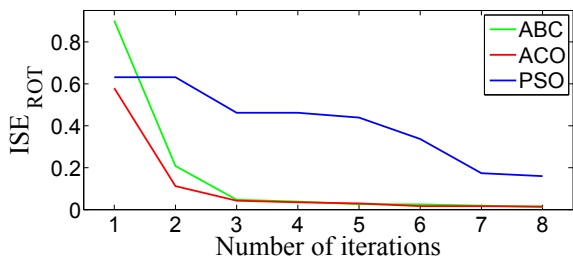


Fig. 6. ISE variation of the rotational controllers over iterations.

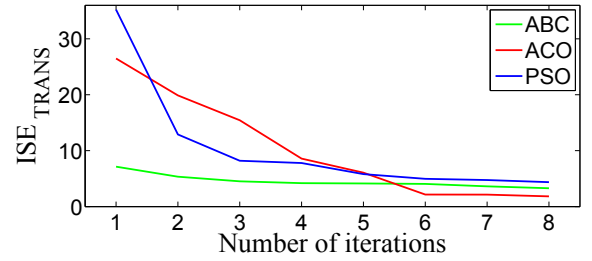


Fig. 7. ISE variation of the translational controllers over iterations.

Table 2

Average computation times over ten runs.

Method	Computation Time (s)
ABC	218.13
ACO	271.52
PSO	110.56

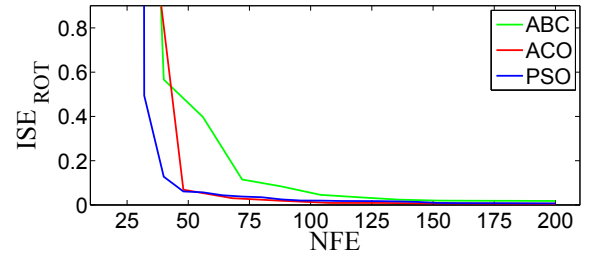


Fig. 8. ISE variation of the rotational controllers over NFE.

exploration such that PSO is performed using $\beta = 0.4$ and $\gamma_1 = \gamma_2 = 0.3$.

In addition to the PSO algorithm, we conduct the identification with two other swarm-based methods, namely, *ABC* and *ACO*. In order to make a reliable comparison among the algorithms, we utilize the same settings in these methods as our PSO, wherever applicable, i.e., swarm size, search-space. We conduct the simulations in MATLAB on a Dell Precision Tower 5810 desktop computer with a 3.2 GHz Intel® XEON 8 (E5-1680) processor and 64 GB 2400 MHz DDR4 RAM on Linux operating system.

The ISE variation throughout the iterations for the rotational and translational controllers are given in Figs. 6 and 7, respectively. As can be seen, the two advance competitors, ABC and ACO, yield less ISE as compared to PSO. Nevertheless, this superiority comes at the cost of computational complexity. Table 2 presents the total computation time required for each method. As can be seen, the computation time required for PSO is approximately half of the time required for the other two. This is mainly because ABC and ACO have a larger number of fitness evaluations (NFEs) at each iteration. In this regard, we also compare the performance of each algorithm with a fixed NFE of 200, as can be seen in Figs. 8 and 9. In this batch of simulations, PSO yields better ISE results with its ability to visit a number of different solution candidates in a comparably faster manner. In overall, considering the complexity-performance trade-off, the performance of PSO is fairly comparable with the other two while the computation time is dominantly less. Besides, since the identification phase in this work will be followed by the online learning in control phase, the over-identification seems redundant. Adding on the stability analysis made by the authors, available in Ref. [31], PSO presents itself as a desirable method among the swarm-based optimization methods.

Tables 3 and 4 summarize the antecedent parameters identified for each of the six T2-FNNs using the system identification procedure as

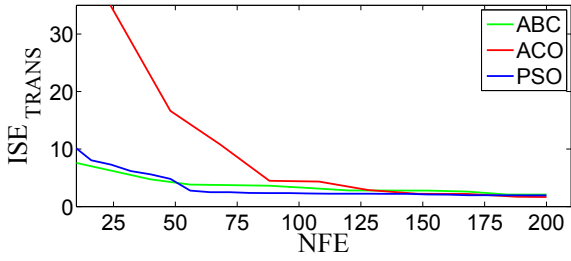


Fig. 9. ISE variation of the translational controllers over NFE.

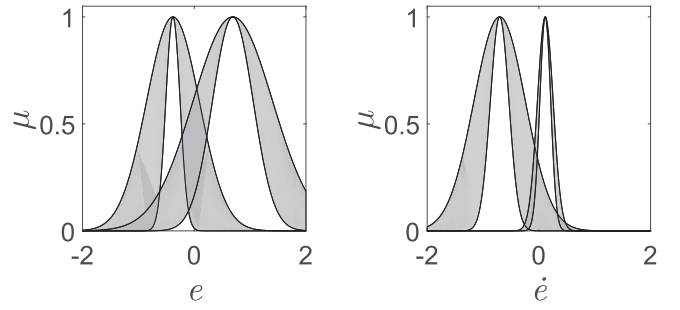


Fig. 11. Type-2 membership functions in X.

Table 3

Antecedent parameters of the rotational T2-FNNs.

Rotational controllers		e			\dot{e}		
		c	σ_u	σ_l	c	σ_u	σ_l
Roll	MF1	-0.47	0.72	0.49	-0.52	0.85	0.33
	MF2	0.62	0.41	0.11	0.39	0.84	0.63
Pitch	MF1	-0.26	0.60	0.22	-0.65	0.79	0.48
	MF2	0.67	0.71	0.05	0.81	0.76	0.53
Yaw	MF1	-0.68	0.71	0.45	-0.54	0.73	0.24
	MF2	0.71	0.67	0.15	0.71	0.64	0.34

Table 4

Antecedent parameters of the translational T2-FNNs.

Translational controllers		e			\dot{e}		
		c	σ_u	σ_l	c	σ_u	σ_l
X	MF1	-0.38	0.47	0.13	-0.70	0.46	0.17
	MF2	0.69	0.71	0.36	0.11	0.14	0.10
Y	MF1	-0.36	0.33	0.19	-0.40	0.65	0.59
	MF2	0.38	0.68	0.27	0.68	0.74	0.18
Z	MF1	-0.78	0.52	0.22	-0.19	0.83	0.35
	MF2	0.21	0.25	0.07	0.82	0.45	0.27

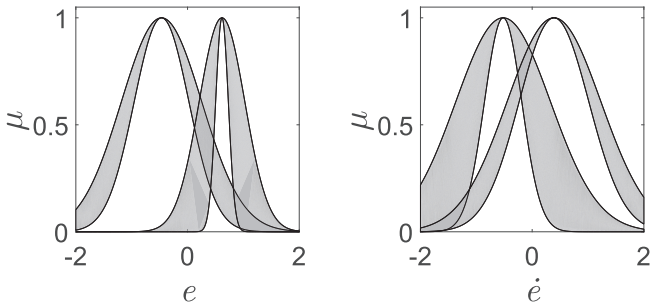


Fig. 10. Type-2 membership functions in Roll.

described before. Figs. 10 and 11 illustrate the nature of the membership functions which correspond to the antecedent parameters in Roll and X channels, respectively.

5.2. SMC training algorithm for control

SMC serves as a discontinuous control signal methodology which can stir a system with nonlinear dynamics to its equilibrium and sustain its stability. The entire control process involves two phases: *reaching phase* which refers to the state when the signal is moving towards the sliding manifold, and *sliding phase* which represents the state when the control response maintains on the sliding manifold [31]. Although, this strategy is proven to be effective in numerous case studies for con-

trol, SMC method is restricted by the capability of handling discrepancy from external disturbances and chattering due to the high frequency switching signal. To overcome this problem, SMC can be used along with a T2-FNN whereby the degree of estimation from the T2-FNN is extensively recognized to compromise the limitation of SMC specially in noisy conditions. Therefore, in this work, sliding mode fuzzy controllers are implemented as an integration of T2-FNNs for the overall control structure, and SMC for the consequent parameters update. A pair of direction-altered signal is applied onto the control system whereby e and \dot{e} converge to the sliding surface in a finite period of time. This approach effectively combines the advantages of both methods: fuzzy system guarantees the general approximation property while SMC contributes to the stability and favorable robustness. The two critical parameters a and b , as the consequent part of the T2-FNNs, are updated based on the detailed update rules below [31]:

$$\dot{a}_{ri} = -x_i \frac{q\tilde{w}_r + (1-q)\tilde{w}_r}{(q\tilde{w}_r + (1-q)\tilde{w}_r)^T(q\tilde{w}_r + (1-q)\tilde{w}_r)} \text{asgn}(u_c) \quad (12)$$

$$\dot{b}_r = \frac{q\tilde{w}_r + (1-q)\tilde{w}_r}{(q\tilde{w}_r + (1-q)\tilde{w}_r)^T(q\tilde{w}_r + (1-q)\tilde{w}_r)} \text{asgn}(u_c)$$

where the learning rate α is updated based on the following equation [31]:

$$\dot{\alpha} = \gamma(I+2)|u_c| - \nu\gamma\alpha; \quad 0 < \gamma, \nu \quad (13)$$

where u_c corresponds to the control input generated by PD controller while I is the number of inputs; it is 2 for e and \dot{e} . For all of the update rules, the stability proof by using a suitable Lyapunov function is available in Ref. [31].

6. Simulation studies

In this study, a realistic trajectory for rice farm quality inspection, as shown in Fig. 12, is selected to demonstrate the capabilities of the proposed controller. The autonomous navigation of the quadcopter begins at the shed for a crop inspection with a climb to a height of 10 m. After this phase, the vehicle hovers across the farm to a second (higher) area of interest and performs the crop inspection on a descent from the height of 60 m. Finally, the quadcopter returns to its initial position which is the shed. This trajectory is defined based on the geometrical and topological features of the terrace farm. The simulation studies for this reference trajectory are conducted with the quadcopter dynamic model in MATLAB for three different cases. In the first case, the trajectory tracking performance of the proposed T2-FNN is investigated in ideal conditions, without the effect of noise in the system. In the second case, a realistic uncertainty, measurement error in GPS is introduced. As regards to the third case, the performance is evaluated for the conditions in which GPS error is introduced together with IMU noise. During the operation, the control system ensures that the control inputs do not exceed the dynamic constraints of the quadcopter.



Fig. 12. A typical trajectory over rice farms for quality inspection.

6.1. Trajectory tracking in ideal conditions

In this case, trajectory tracking is performed under ideal conditions whereby the influence of noise is neglected. The respective performance of the PD controller and the proposed T2-FNN are presented in Fig. 13. Evidently, both controllers show relatively satisfactory performance. However, with a deeper analysis, the PD controller yields a larger steady-state error along the trajectory. On the other hand, the proposed T2-FNN is able to decrease the steady-state error of the system with the help of its online learning capability as a part of its adaptive controller structure. Fig. 14 shows the evolution of the Euclidean error for both controllers. As can be seen from Fig. 14, during the adaptation period, the error of the proposed controller is higher when compared to the PD controller. However, once it adapts itself and tunes its parameters accordingly, it yields a superior performance than that of the conventional PD controller. The evolution of the adaptive learning rate (α) over time is shown in Fig. 16. Furthermore, it can be seen that the output of the PD controller always tends to go to zero, and the T2-FNN is likely to take over the control responsibility of the system. This trend is represented in Figs. 17 and 18 which show the

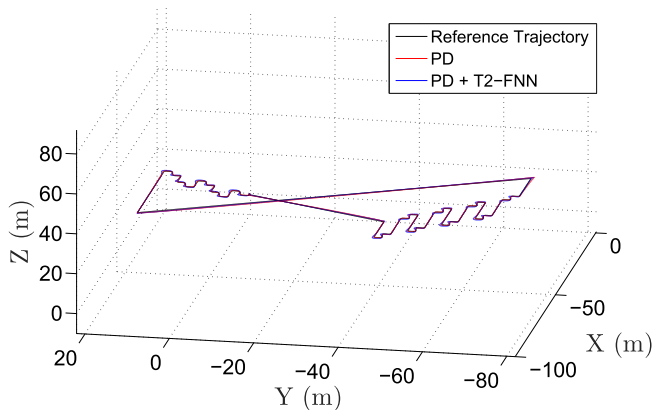


Fig. 13. PD and T2-FNN trajectory tracking response without noise.

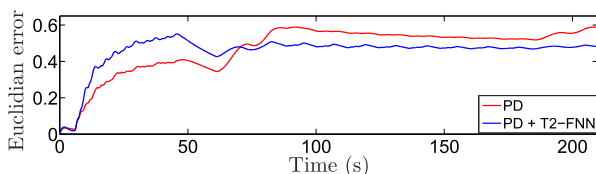


Fig. 14. Euclidean error variation over time.

respective controller output contributions over time for the X and Y controllers.

6.2. Trajectory tracking in noisy conditions

In this part, in order to investigate the effect of GPS and IMU measurement uncertainties, two cases with different levels of noise are considered. Firstly, low noise levels in GPS are examined by introducing ± 1 m error margin in x , y , and z axes using white Gaussian noise. Subsequently, higher levels of noise are introduced by considering the additional contribution of IMU measurement error with a magnitude of $\pm 10^\circ$ in roll, pitch, and yaw angles. In both cases, it is evident that the PD controllers are unable to handle the noise in the system. On the contrary, the T2-FNNs deal well with noise and are observed to be capable of tracking the trajectory fairly well. The respective trajectory tracking responses in the x and y axes can be seen in Fig. 15. The superior performance of the T2-FNN can be attributed to its adaptive structure which is also capable of handling uncertainties in the system. Furthermore, when compared to its type-1 counterpart, the T2-FNNs exhibit greater

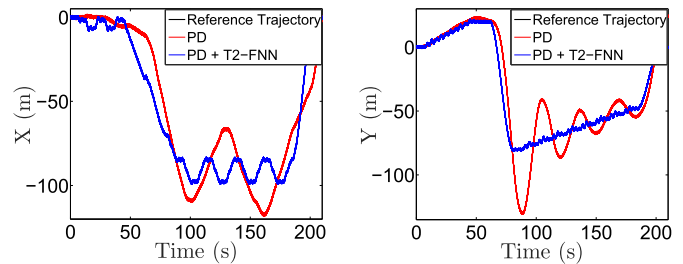


Fig. 15. PD and T2-FNN response in x and y axes.

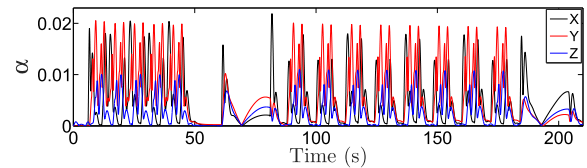


Fig. 16. Evolution of adaptive learning rate (α) over time.

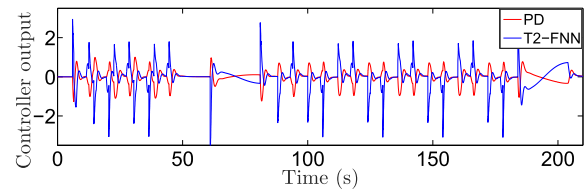


Fig. 17. Control outputs of PD and T2-FNN in x axis.

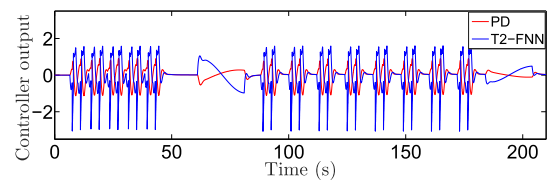


Fig. 18. Control outputs of PD and T2-FNN in y axis.

Table 5
ISE results.

Controller	ISE		
	No noise	Noise in GPS	Noise in GPS + IMU
PD	46.77	102960.00	459560.00
PD + T1-FNN	37.49	265.82	279.42
PD + T2-FNN	34.29	229.52	234.27

capability in dealing with noise as can be seen from the ISE results in Table 5.

7. Conclusion and future work

7.1. Conclusion

The proposed control algorithm, a conventional PD controller in parallel with a T2-FNN, yields better performance in handling noise and measurement uncertainties in the system when compared to its type-1 counterpart or the conventional PD controller working alone. This result is a proof that the use of the T2-FNNs is a preferable choice for the real-time applications whereby the influence of noise is inevitable.

7.2. Future work

The long-term objective of this study is to develop a T2-FNN based control structure which is capable of achieving an effective quadcopter control in natural landscapes which may constitute demanding environments and require more agile and swift maneuvers.

References

- [1] R. Sahni, Agricultural Drones Market to Reach 3.69 Billion Dollars by 2022 Says a Latest Research, 2016.
- [2] C. Anderson, Agricultural Drones, 2014.
- [3] J. Li, Y. Li, Dynamic analysis and pid control for a quadrotor, in: *Mechatronics and Automation (ICMA), 2011 International Conference on*, IEEE, 2011, pp. 573–578.
- [4] S.F. Ahmed, K. Kushsairy, M.I.A. Bakar, D. Hazry, M.K. Joyo, Attitude stabilization of quad-rotor (uav) system using fuzzy pid controller (an experimental test), in: *Computing Technology and Information Management (ICITIM), 2015 Second International Conference on*, IEEE, 2015, pp. 99–104.
- [5] F. Rinaldi, S. Chiesa, F. Quagliotti, Linear quadratic control for quadrotors uavs dynamics and formation flight, *J. Intel. Robotic Syst.* 70 (1–4) (2013) 203–220.
- [6] U. Eren, A. Prach, B.B. Koçer, S.V. Raković, E. Kayacan, B. Açıkmese, Model predictive control in aerospace systems: current state and opportunities, *J. Guid. Control, Dyn.* 40 (7) (2017) 1541–1566.
- [7] M. Mehndiratta, E. Kayacan, Receding horizon control of a 3 dof helicopter using online estimation of aerodynamic parameters, *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* (2017) 0954410017703414.
- [8] A. Sarabakha, N. Imanberdiyev, E. Kayacan, M.A. Khanesar, H. Hagsras, Novel levenberg-marquardt based learning algorithm for unmanned aerial vehicles, *Inf. Sci.* 417 (2017) 361–380.
- [9] K. Chang, Y. Xia, K. Huang, D. Ma, Obstacle avoidance and active disturbance rejection control for a quadrotor, *Neurocomputing* 190 (2016) 60–69.
- [10] T. Lee, M. Leok, N.H. McClamroch, Nonlinear robust tracking control of a quadrotor uav on se (3), *Asian J. Control* 15 (2) (2013) 391–408.
- [11] L.A. Zadeh, Fuzzy sets, *Inf. Control* 8 (3) (1965) 338–353.
- [12] E. Kayacan, E. Kayacan, M.A. Khanesar, Identification of nonlinear dynamic systems using type-2 fuzzy neural networks a novel learning algorithm and a comparative study, *Ind. Electron. IEEE Trans.* 62 (3) (2015) 1716–1724.
- [13] J. Tavooosi, A.A. Suratgar, M.B. Menhaj, Stable anfis2 for nonlinear system identification, *Neurocomputing* 182 (2016) 235–246.
- [14] X. Dong, Y. Zhao, H.R. Karimi, P. Shi, Adaptive variable structure fuzzy neural identification and control for a class of mimo nonlinear system, *J. Frankl. Inst.* 350 (5) (2013) 1221–1247.
- [15] E. Kayacan, O. Kaynak, Sliding mode control theory-based algorithm for online learning in type-2 fuzzy neural networks: application to velocity control of an electro hydraulic servo system, *Int. J. Adapt. Control Signal Process.* 26 (7) (2012) 645–659.
- [16] O. Castillo, L. Amador-Angulo, J.R. Castro, M. Garcia-Valdez, A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems, *Inf. Sci.* 354 (2016) 257–274.
- [17] E. Kayacan, E. Kayacan, H. Ramon, O. Kaynak, W. Saeys, Towards agrobots: trajectory control of an autonomous tractor using type-2 fuzzy logic controllers, *Mech., IEEE/ASME Trans.* 20 (1) (2015) 287–298.
- [18] M.A. Khanesar, E. Kayacan, M. Reyhanoglu, O. Kaynak, Feedback error learning control of magnetic satellites using type-2 fuzzy neural networks with elliptic membership functions, *Cybern. IEEE Trans.* 45 (4) (2015) 858–868.
- [19] A.J. Khakshour, M.A. Khanesar, Model reference fractional order control using type-2 fuzzy neural networks structure: implementation on a 2-dof helicopter, *Neurocomputing* 193 (2016) 268–279.
- [20] R. Martínez-Soto, O. Castillo, J.R. Castro, Genetic algorithm optimization for type-2 non-singleton fuzzy logic controllers, in: *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, Springer, 2014, pp. 3–18.
- [21] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, *J. Glob. Optim.* 39 (3) (2007) 459–471.
- [22] C. Caraveo, F. Valdez, O. Castillo, Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation, *Appl. Soft Comput.* 43 (2016) 131–142.
- [23] O. Castillo, H. Neyoy, J. Soria, P. Melin, F. Valdez, A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot, *Appl. Soft Comput.* 28 (2015) 150–159.
- [24] A. Mousa, I. El-Desoky, Stability of pareto optimal allocation of land reclamation by multistage decision-based multipheromone ant colony optimization, *Swarm Evol. Comput.* 13 (2013) 13–21.
- [25] A.A.A. Mousa, Hybrid ant optimization system for multiobjective economic emission load dispatch problem under fuzziness, *Swarm Evol. Comput.* 18 (2014) 11–21.
- [26] F. Olivas, F. Valdez, O. Castillo, P. Melin, Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic, *Soft Comput.* 20 (3) (2016) 1057–1070.
- [27] Y.-L. Chen, J. Cheng, C. Lin, X. Wu, Y. Ou, Y. Xu, Classification-based learning by particle swarm optimization for wall-following robot navigation, *Neurocomputing* 113 (2013) 27–35.
- [28] Y. Shi, R.C. Eberhart, Fuzzy adaptive particle swarm optimization, in: *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1, IEEE, 2001, pp. 101–106.
- [29] G. Xu, F. Liu, C. Xiu, L. Sun, C. Liu, Optimization of hysteretic chaotic neural network based on fuzzy sliding mode control, *Neurocomputing* 189 (2016) 72–79.
- [30] M. Vahedi, M. Hadad Zarif, A. Akbarzadeh Kalat, Speed control of induction motors using neuro-fuzzy dynamic sliding mode control, *J. Intel. Fuzzy Syst.* 29 (1) (2015) 365–376.
- [31] E. Kayacan, M.A. Khanesar, *Fuzzy Neural Networks for Real Time Control Applications: Concepts, Modeling and Algorithms for Fast Learning*, Butterworth-Heinemann, 2015.
- [32] S. Masumpoor, M.A. Khanesar, et al., Adaptive sliding-mode type-2 neuro-fuzzy control of an induction motor, *Expert Syst. Appl.* 42 (19) (2015) 6635–6647.
- [33] Q. Gao, G. Feng, Z. Xi, Y. Wang, J. Qiu, Robust control of t-s fuzzy time-delay systems via a new sliding-mode control scheme, *IEEE Trans. Fuzzy Syst.* 22 (2) (2014) 459–465.
- [34] E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm, *IEEE Trans. Cybern.* 43 (1) (2013) 170–179.
- [35] S. Bouabdallah, P. Murrieri, R. Siegwart, Design and control of an indoor micro quadrotor, in: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5, IEEE, 2004, pp. 4393–4398.
- [36] R. Maslim, H. Chaoyi, Z. Yixi, J. Linhao, B.B. Kocer, E. Kayacan, Performance evaluation of adaptive and nonadaptive fuzzy structures for 4d trajectory tracking of quadrotors: a comparative study, in: *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, IEEE, 2015, pp. 1–7.
- [37] R. Jategaonkar, *Flight Vehicle System Identification: a Time Domain Methodology*, vol. 216, AIAA, Reston, VA, USA, 2006.
- [38] Z. Xinchao, A perturbed particle swarm algorithm for numerical optimization, *Appl. Soft Comput.* 10 (1) (2010) 119–124.
- [39] N.J. Cheung, X.-M. Ding, H.-B. Shen, Optifel: a convergent heterogeneous particle swarm optimization algorithm for takagi-sugeno fuzzy modeling, *IEEE Trans. Fuzzy Syst.* 22 (4) (2014) 919–933.
- [40] T. Niknam, B. Amiri, An efficient hybrid approach based on pso, aco and k-means for cluster analysis, *Appl. Soft Comput.* 10 (1) (2010) 183–197.