State-of-the-Art Intelligent Flight Control Systems in Unmanned Aerial Vehicles

Fendy Santoso, Matthew A. Garratt, and Sreenatha G. Anavatti

***Abstract*— We discuss state-of-the-art intelligent robotic aircraft with the special focus on evolutionary autopilots for small unmanned aerial vehicles (UAVs). Under the umbrella of adaptive autopilots, we highlight the pros and cons of the most widely implemented intelligent algorithms against the navigational and maneuvering capabilities of small UAVs. We present several cutting-edge applications of bioinspired flight control systems that have the capability of self-learning. We also highlight several research opportunities and challenges associated with each technique.**

***Note to Practitioners*—Soft computing methods have been widely implemented in numerous engineering applications. Recent advancements in computational technology have enabled the implementations of intelligent autopilots in real time. This paper aims to discuss many aspects of the developments and implementations of soft computing techniques in aerial robotics with the main focus on its flight control systems.**

***Index Terms*— Adaptive autopilots, evolutionary algorithms, intelligent robotic aircraft, self-learning.**

I. Introduction

ROBOTIC aircraft, also formally known as "unmanned aerial vehicles (UAVs)," or colloquially as "drones," have been extensively employed in both civilian and military domains to improve safety, efficiency, and productivity (e.g., convoy protection [1], surveillance [2], law enforcement [3], agriculture and animal tracking [4], forest monitoring [5], as well as building and industrial inspections [6]). Considering the dynamics of small autonomous aircraft, which are inherently nonlinear coupled with the uncertain nature of the flight environments, designing high-performance flight control systems is a challenging task [7], [8].

While many state-of-the-art control systems still heavily rely on the availability of accurate mathematical models (e.g., linear quadratic Gaussian [9]–[11], model predictive control (MPC) [12], backstepping [13], and gain scheduling [14]), this paper looks at more flexible and intelligent approaches by leveraging on the benefits of *evolutionary algorithms* to address the current limitations of model-based control systems.

Manuscript received November 14, 2016; accepted January 1, 2017. Date of publication February 2, 2017; date of current version April 5, 2018. This paper was recommended for publication by Associate Editor K. Fregene and Editor D. Tilbury upon evaluation of the reviewers comments. This work was supported by the University of New South Wales, Canberra, Australia.

The authors are with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2612, Australia (e-mail: f.santoso@adfa.edu.au; m.garratt@adfa.edu.au; s.anavatti@adfa.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2017.2651109

There are some important considerations in designing a reliable flight control system. The first challenge is related to the *robustness* of the closed-loop control in the face of uncertainties [10], such as unpredictable external air flows (e.g., severe wind gusts) and modeling errors. The motion of a small UAV can be highly vulnerable to the adverse impacts of wind gusts that can force the system to depart from its desired trajectories. This phenomenon can also lead to significant overshoots and considerable tracking offsets, which are indeed undesirable in light of safety and efficiency issues.

Cluttered environments (e.g., city street, forest, and even inside building), where the presence of obstacles are inevitable, will introduce significant challenges for the developments of high-performance trajectory tracking systems. It is therefore critical for UAVs to possess reliable autopilot systems to fly along its desired trajectories. However, small UAVs with limited control and processing bandwidth may find them difficult to precisely maneuver around random obstacles, especially when they are moving (e.g., cars, people, or other drones). In addition, the payload limitation of small UAVs will restrict both mass and physical dimensions of the sensors that can be carried on-board, limiting the sensing capabilities of the systems.

Significant variations in plant dynamics (e.g., due to payload changes) can seriously deteriorate the performance of fixed-gain control systems [15]. To overcome this problem, one requires achieving a high degree of flight autonomy through the availability of robust and adaptive flight control systems. However, current state-of-the-art robust and adaptive control systems (e.g., multiple-model adaptive control [16]) are still not sufficiently flexible to deal with various types of uncertainties and hence their performance can be seriously limited to a certain class of uncertainty models. Also, in addition to errors in modeling, there are many unpredictable sources of uncertainties such as the adverse impacts of component failures and structural damage that should be taken into account to ensure safe and reliable flight operation.

One major problem with model-based control systems is their dependency on the accuracy of the assumed mathematical model of the system. In practice, there is no perfect mathematical model to capture the whole dynamics of any systems, even for the simplest ones. Although Petersen *et al.* [17] have developed cutting-edge model-based robust controls, the performance of linear time-invariant robust controls (e.g., H_∞ [10], [18] and μ-synthesis [19], [20]) can deteriorate in the face of *large* uncertainties, namely, the failures or substantial degradations of servos, control surfaces, and sensors.

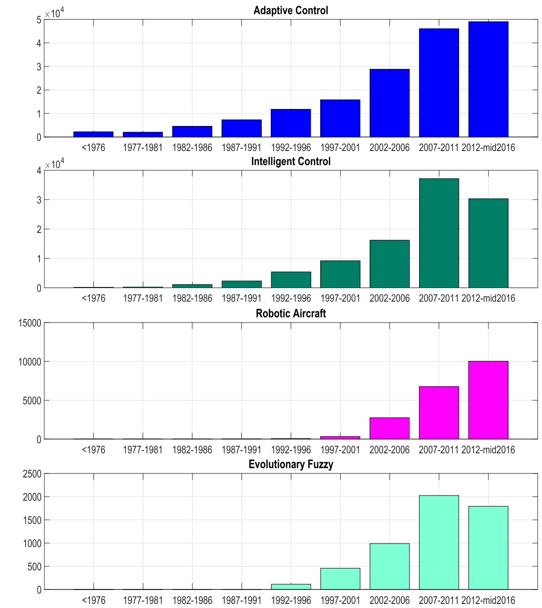

Fig. 1. Research progress in intelligent and adaptive control systems according to Scopus. The number of published papers are represented in the vertical axis, sampled every five year interval as given in the horizontal axis, indicating a substantial growth. The titles of each graph represent the search topic.

Addressing these issues, recent developments of autopilot systems have been shifted toward more flexible approaches through developments of adaptive control systems that can achieve better capability to accommodate variations in both signals and system models (i.e., plant characteristics and disturbance models) regardless of the class of the uncertainty models employed. The systems broadly deal with how to design flexible controllers to automatically learn the characteristics of the plants, track its parameter variations, and fine-tune them whenever required.

While many state-of-the-art adaptive control systems still heavily rely on mathematical models of the systems (e.g., gain scheduling [14] and feedback linearization techniques [21]), these approaches might be too complex or impractical in some situations. For instance, although gain scheduling control has been considered as one of the historically successful adaptive control techniques, this method suffers from several technical drawbacks [22]. The system is highly mathematical and quite tedious since it heavily relies on the linearization process of the aircraft dynamics through several points in the flight envelope, in addition to some joint interpolation schemes. It may also produce a system that falls short of global property. In addition, feedback linearization may be impractical in the absence of comprehensive mathematical models.

Adaptive control has become a very topical research area in the past few decades as the number of published research papers in the subject matter, according to Scopus, has grown exponentially (see Fig. 1). Under the umbrella of it, intelligent control and evolutionary fuzzy systems have also demonstrated a similar fashion, following progress in computer science. According to Moore's law (named after Gordon Moore), the speed of computers would double nearly every 18 months as a result of the doubling capacity of transistors packed in integrated circuits every year.

The implications of these revolutionary technological developments have been quite enormous, considering the role of computers as the main processing instruments and decision makers in automatic control systems. Thus, it is not surprising that this development has also stimulated the progress of intelligent learning systems (e.g., evolutionary fuzzy) that was once not possible due to the high computational demand. Current expansions of the microelectromechanical systems (MEMS) have enabled the developments of small and lightweight micromachines, including aerial robots (e.g., microaerial vehicles), which have received considerable attention. Thus, given the multidisciplinary nature of research in aerial robotics, there have been urgent demands for the development of autonomous systems, which are not only adaptive and intelligent but also computationally efficient to be able to perform online learning and to make decisions in real time.

Accordingly, current research trends in adaptive control have been progressively shifted toward the development of intelligent systems to accommodate *rules-based* or *knowledge-based* and even *learning-based* approaches such as fuzzy logic systems [22], [23], neural networks (NNs) [24], genetic algorithms (GAs) [25], [26], and swarm intelligence (SI) [27]. These systems are suitable to deal with highly complex dynamics, where it is very difficult, if not impractical, to acquire accurate mathematical models [23] (e.g., the phenomena of the "phantom yaw" in high angle-of-attack flight, caused by nose vertex shedding [22]). The other benefit is limited not only to their adaptive nature but their ability to perform self-learning.

Under the umbrella of intelligent control systems, this paper focuses on the technical aspects of the evolutionary autopilots from both the theoretical and practical standpoints. We believe that this review is the *first* in the literature, specifically addressing the topic of evolutionary aerial robotics. The review also provides a comprehensive discussion relating to some cutting-edge theoretical developments of the evolutionary algorithms. Subsequently, this review also discusses the practical implementation of those techniques, including their pros and cons in flight control systems. Finally, we also discuss several research opportunities and challenges associated with intelligent aerial robotics.

This paper is organized as follows. While Section II discusses state-of-the-art intelligent algorithms, Section III depicts the practical applications of the intelligent algorithms applied in aerial robotics. Section IV presents a comprehensive discussion regarding research opportunities and challenges, while Section V concludes this paper.

II. Intelligent Systems

Heuristic approaches provide a good alternative for solving complex dynamic systems, particularly when exact methods fail. Many biological processes can be regarded as a process of constrained optimization. Thus, under the subset of adaptive and intelligent systems, we define evolutionary algorithms as a metaheuristic optimization technique based on the generic population approaches inspired by biological evolution, such as reproduction, mutation, recombination, and selection.

This section discusses the five major approaches to evolutionary algorithms in aerial robotics, which are fuzzy inference system (FIS), artificial NNs (ANNs), GAs, SI, and hybrid.

A. Fuzzy Inference Systems

Considering the imprecise nature of abstract concepts, fuzzy systems rely on the approximate reasoning that derives a conclusion based on a set of expert fuzzy IF–THEN rules, reflected in the form of a continuous value between 0 and 1. The system consists of three fundamental constituents, namely, *fuzzification*, *rule-based inference*, and *defuzzification*, and fuzzy mapping can be regarded as a mapping from *crisp inputs* into *fuzzy sets*, while the *defuzzification* process is intended to reconvert the fuzzy output values from the crisp values. Given the linguistic control rules, fuzzy logic imitates human reasoning and decision making with natural language.

Leveraging on a set of linguistic control rules derived based on expert knowledge, it can convert the linguistic control strategy into an automatic system. Although the FIS advocates the capability to accommodate inherent uncertainties of the human knowledge in its linguistic variables and simple interaction as well as easy interpretation of the results, the system is totally reliant on the expert knowledge, and it has no capability of generalization, nor is it robust in response to the topological changes of the system [28].

Motivated by the necessity to accommodate the uncertainty in a better way, Zadeh [29] introduced *type-2* fuzzy logic, as an extension of its type-1 counterpart. Introducing a new dimension called *the footprint of uncertainty*, the system employs the same rule structure, except the presence of type-2 antecedent and consequent. Although the type-2 fuzzy system can make the system more noise tolerant, it is indeed more expensive in terms of the computational burden. Zadeh [29] later extended his work into type-*n* fuzzy systems. However, this work is still considered in its infancy. The stability analysis of fuzzy logic controller is also not trivial.

One potential challenge in implementing fuzzy logic systems is its *linguistic problems* [i.e., the generation of the fuzzy rules and the membership functions (MFs) for each fuzzy set], especially for systems with many variables [23]. Conventionally, an expert in a particular area manually defines the rules, especially for control systems with only a few input variables. However, for a large-scale system, manual setup by an expert to achieve a good control performance may not be possible, or at least the process will become very tedious, complicated, and inefficient. Thus, the development of an *automatic fuzzy tuning system* is preferable.

One straightforward approach to address this problem is by employing clustering algorithms (e.g., *c*-means clustering) to subdivide the space into many partitions before mapping the center of each cluster into a rule in line with the definition of fuzzy variables [23]. Nonetheless, this approach does not guarantee the performance of the systems, especially for large-scale systems with many input variables due to the independent nature of the MFs with respect to the extracted rules [23]. Although clustering the experimental data can decrease the number of rules, it may suppress the interpretability of the obtained classes [30]. In many cases, therefore, further tuning of the MF, including the selection of appropriate fuzzification and defuzzification methods, is still required.

Considering the nature of the fuzzy logic systems, which can be regarded as a search algorithm in high-dimensional space, it is very convenient to employ *evolutionary computations* (e.g., GAs) to determine an optimal solution over a hypersurface. This optimization problem is not trivial due to the nature of the *hypersurface* itself, which is not only unbounded and nondifferentiable but also very complex and noisy as well as being deceptive, since the performance of a similar fuzzy set can be completely different [23]. Adopting the concept of the evolutionary systems, fuzzy logic systems can undergo both structural and parametric changes as the systems evolve.

B. Artificial Neural Networks

Emulating biological neural systems, an NN can be regarded as an adaptive nonlinear system comprising interconnected neurons that exchange messages with each other. The connection is represented by numerical weights that can be tuned based on experience, making it suitable for highly parallel and intelligent learning. This approach is often used to tackle challenging optimization problems that are not suitable for conventional linear programming. Unlike fuzzy logic, the main advantages of ANNs are their learning and generalization capacity in addition to robustness to disturbances. However, determining the number of layers and neurons is not an easy task [28]. Owing to its black box nature, one should have additional knowledge relating to the characteristics of the system to interpret its functionality [28].

Although there are various architectures of NNs, such as feedforward [31], recurrent [32], self-organizing [33], Hopfield NN [34], and feedforward radial basis function (RBF) [35], feedforward and recurrent NNs are considered the most widely implemented methods in robotics. Compared with the feedforward system, connecting all layers from input and middle (hidden), all the way down, to the output layer, the recurrent ANN configuration incorporates feedback connections. Also, the concept of *simulated annealing* is often used to train the system and to perturb the ANN weights by a random value to avoid local minima scenario. Meanwhile, its learning strategies can be categorized into three main parts.

1) Supervised learning [36] is when the network is trained by input/output pairs provided by external components or by the network itself.
2) Unsupervised learning [33] occurs when the output unit is trained to respond to a certain pattern in the absence of output examples.
3) Reinforcement learning [37], which can be considered as a combination of supervised and unsupervised learning. Adopting evolutionary concept, the NN systems can undergo both structural and parametric changes as the systems evolve.

C. Genetic Algorithms

Mimicking Darwinian evolution, GAs can be regarded as a randomized heuristic optimization strategy based on the idea of natural selection. This concept was first introduced by Holland [38]. In this scenario, the population can be regarded as candidates of solutions. Hence, by evolving the population, the strongest candidate can emerge via mutation and crossover. The algorithm works well when the landscape of the fitness is continuous. Genetic information is encoded in the chromosomes. A chromosome contains hundreds to thousands of genes that can be regarded as a segment of DNA containing the code used to synthesize a protein. For instance, human cells contain 23 pairs of chromosomes making up a total of 46 chromosomes. Some examples of GA autopilot can be found in [39] and [40], while the application of GA for scheduling can be found in [41].

Each time an element is chosen to be mutated, it will be iterated by one-step random integer inside its range. Meanwhile, *crossover* and *mutation* are meant to facilitate an efficient search and to guide the process into a new region. While crossover explores the search space, the purpose of mutation is to facilitate genetic diversity from one generation of chromosomes to the subsequent one. Although it is possible to employ a fixed probability of crossover and mutation, this will not generally lead to an optimum outcome. Thus, the common practice is to employ a variable rate, that is, by starting out with a relatively higher value of crossover and lower value of mutation, and then decreasing the crossover value and increasing the mutation rate toward the end of the process. However, although normally linear changes are adopted, there is no hard-and-fast rule to adjust the parameters during the run.

While the GA can be a simpler search solution and may require lower memory requirement compared with an explicit analytical search method, it can get stuck on local optima (although this drawback can be mitigated through crossover). Its intrinsically parallel and distributed nature can also offer less computational time. The process begins with a population of randomly generated individuals. The fitness of every individual is evaluated in each generation and a new population is formed by the selection of multiple individuals from the current population based on their fitness. The process will be terminated after a maximum number of generations have been produced or a certain fitness level has been reached for the population. While a high fitness value suggests low crossover rate and high mutation rate, to avoid local minimum, one needs to decrease the crossover rate and increase the mutation rate. The same principle is also applicable for the variance of the fitness of the population, that is, low variance indicates mutation and high variance denotes crossover. This suggests high crossover rate ($> 80\%$) and low mutation rate ($< 1\%$) [42].

There are several ways to expedite the processing time [43], such as by employing parallel [44], incremental [45], or micro-GAs [46]. While the incremental GAs have the advantage of reduced generation cycle time, since the system produces only one or two offspring at each generation, the systems may not produce satisfactory individuals. Micro-GAs employ a very small population and thus have the potential to produce more offspring, although the systems suffer from a lack of genetic diversity [43]. Accordingly, it is not impossible for GAs to produce successive control laws that may introduce a stability issue.

D. Swarm Intelligence

SI is the collective property of a system comprising many homogenous individuals, where their social behaviors result in sophisticated global response to their environment [47]. The overall behavior of the system is derived from the interaction of individuals with each other on local information and with their environment by simple behavioral rules [47]. A number of swarm-based optimization techniques have been published in the literature: 1) the particle swarm optimization (PSO) [48], [49] [27]; 2) the ant colony optimization (ACO) [50]; 3) the artificial bee colony (ABC) optimization [51]; and 4) the firefly algorithm [52]. While this concept creates robustness to the failure of each individual, the behavior of individual agents may not necessarily represent the collective behavior of the whole system. Since there is almost no analytical mechanism to this approach, designing a swarm-based system is not an easy task.

Known as a population-based evolutionary algorithm, PSO is similar to other population-based evolutionary algorithms. However, it should be highlighted that PSO is mainly driven by the property of the social behavior, instead of the survival of the fittest as in the GA [48]. Being inspired by bird flocking or fish schooling, PSO employs a set of candidates of the solution known as the particles. In each iteration, each particle updates its position and velocity as follows: $x_{k+1}^i = x_k^i + v_{k+1}^i$. The system works based on the communication between particles and each particle has a velocity that corresponds to rate of changes of how fast it flies within the search space. The particle velocity is adjustable based on the corresponding experience of the particle as follows: $v_{k+1}^i = v_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^i - x_k^i)$, where p_k^i denotes best remembered position, whereas $c_{1,2}$ represents cognitive and social parameters and $r_{1,2}$ indicates random numbers between 0 and 1. Knowledge is represented in the previous local/global best position of all particles. It is therefore possible to explore distributed problem solving in the absence of centralized control or the provision of the global model [47]. This way, PSO has fewer parameters to adjust and can offer simplicity and a more efficient memory usage compared with GA. Unlike GA that evolves the population around a subset of the best individuals, PSO is better at maintaining diversity of the swarm since all particles use information corresponding to the most successful particle [53].

Inspired by the behavior of ant colonies in searching for the nearest route from their nests to the food sources by creating *pheromone trails* on the route they have passed, ACO provides another heuristic solution for optimization problems. Pheromones will stimulate a natural response to the other ant groups as the subsequent ants are attracted to paths with the highest pheromone concentration [47]. Thus, the shorter the path, the higher the concentration of the pheromone, since it would become the most frequently visited path. Although convergence is guaranteed, the time required is uncertain [47].

ABC is another population-based heuristic algorithm inspired by the collective behaviors of honeybees in finding food sources around the hive [54]. Conceptually, there are three groups of bees: 1) *employed bees* whose task is to bring flower nectar inside the hive to a spot called "the dance floor;" 2) *onlooker bees* that watch the dance and make a decision to choose the best flower patches; and 3) *scout bees* that carry a random search of a food around the hive. After the food source has been exhausted by the employed bees, the onlooker bees become scouts.

There are two types of dance in the hive [47]. The *round dance* is performed when the region visited by an employed bee is near the hive and it contains information about the nectar quality of the visited flower as a guidance for other bees, and *the waggle dance* is performed by the bees to exchange information about the flower patch, including the direction, distance from hive, and its quality [47], [53]. In this scenario, the position of the food source indicates the possible solution to the optimization problem, while the amount of nectar in the food source denotes the quality (fitness) of the corresponding solution. The higher the amount of nectar, the greater the probability of the food source being chosen by the onlooker. On arriving at a certain area, the employed bees choose a new food source in the neighborhood of one saved in memory. If the new food has better nectar than the old source, then the memory will be updated.

Another heuristic approach is inspired by flashing behavior of fireflies in [52] and [55]. Initially, the algorithm generates a random population of feasible solutions. All fireflies within the population are to explore solution search space, with a shared knowledge among fireflies to guide the search to the best location. The firefly moves in the multidimensional search space, with a continuous update of the attractiveness based on the knowledge of the firefly and its neighbor [47]. As described in [52], the flashing characteristics of the fireflies satisfy the following condition. First, all fireflies are unisex and attractiveness is proportional to the level of the brightness. This will tend to cause the less bright firefly to move closer to the brighter one. The attractiveness will decrease as the distance grows, as described by the following equations: $L_i = L_0 e^{(-\mu d)^2}$, where L_i indicates the intensity of light at ith iteration, L_0 is the initial intensity, μ denotes light absorption coefficient, and d is the Euclidean distance between two fireflies. While the initial value of the flashing light is set constant, the modification of light absorption parameter is the key of convergence of the algorithm.

Inspired by birds and insects, another intelligent behavior is known as *aerial flocking*, which allows several robots to move together in unity, to enhance the capability of the whole systems. The main advantage of this strategy is its robustness and scalability since the failure of one or several agents only does not cause the mission to fail [56]. This way, one can share the load of the systems (e.g., communication, computational, and sensing) across multiple agents. Therefore, the ability to keep a predefined distance among the neighboring nodes along with the ability to avoid obstacles is of critical importance. One straightforward implementation of flocking is satellite positioning systems for earth monitoring. Reynolds' rules [57] consist of three main stages, alignment, cohesion, and separation, and is the pioneering research in aerial flocking.

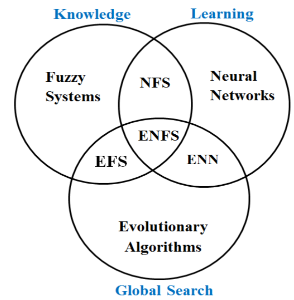

Fig. 2. Concept of hybrid intelligent systems, namely, evolutionary neuro-fuzzy systems. While the letter "F" denotes fuzzy systems, the letter "E" stands for evolutionary algorithms, and the letter "N" represents the NN systems [59].

Hauert *et al.* [58] developed swarms of fixed wing robots addressing limited communication range issues that are often faced by small, lightweight, and low-cost flying robots. Small drones are often faced by another challenge, as they need to maintain their forward velocity to avoid stalling and to avoid them from creating a sharp turn. Hauert *et al.* [58] employed Reynolds' rules [57], encompassing three main steps (e.g., alignment, cohesion, and separation), to achieve robot flocking. They conducted extensive computer simulations to highlight the challenges in maintaining coherent flocks, particularly when the communication range is very limited with respect to the communication range. They investigated swarm behaviors when flocking and when parameters prevented flocking. Considering the fact that flocking performance can be enhanced by increasing the maximum turn rate of the robots and their communication range, their research suggested the maximum turn rate of 0.7 rad/s and a communication range of 150 m is an optimum setting.

E. Hybrid Evolutionary Systems

Developing a hybrid system may be required to achieve the most cost-effective solutions [59]. Fig. 2 depicts the concept of hybrid evolutionary systems, comprising three major streams (e.g., fuzzy systems, NNs, and evolutionary algorithms). In what follows, we shall compare and contrast several hybrid learning systems widely implemented in robotics.

1) Neurofuzzy Systems: Neurofuzzy systems are widely known for their ability to model complex dynamic systems. Fusion of the ANN and the FIS has attracted a great deal of attention in many engineering disciplines because of their complementary natures. While fuzzy systems do not have an automatic learning mechanism, such as NNs, and hence require expert knowledge to be useful, fuzzy systems can mimic human linguistic reasoning process in their linguistic form. Their suitability to depict the behavior of complex, imprecise, or ill-defined information to be amenable for model-based control systems has been established beyond doubt.

NNs are purposefully designed to facilitate automatic learning, given the availability of the training data, as the systems can *learn from scratch* by adjusting the interconnection among layers [24]. The learning process of NNs is somewhat faster than the evolutionary algorithms.

There are two major hybrid neurofuzzy approaches, namely, *cooperative* and *concurrent* systems [24]. In a cooperative model, the FIS MF can be determined from the ANN learning mechanism or from the training data, while the rule-based system can be determined by fuzzy clustering algorithms. In a concurrent model, the FIS system is continuously assisted by the ANN to compute the required parameters, especially for the case of unmeasurable input variables. The fused system will share data structures and knowledge representations. Nonetheless, the gradient descent-based conventional learning method cannot be directly implemented given the nondifferentiable nature of the functions used in the inference process. Some major hybrid neurofuzzy architectures are FALCON [60], ANFIS [61], GARIC [62], NEFCON [63], FINEST [64], and FUN [65].

2) Fuzzy Genetic Algorithm: While the first task in the fuzzy system is to specify each fuzzy set, comprising the rule and MFs, the main challenge relating to the fuzzy-GA system is related to the complete encoding process of the fuzzy system into the integer-based chromosome, instead of binary one [23]. Each element in a chromosome has a certain range of integers in line with the represented parameter in the original fuzzy system. Given an unknown system, however, one can guess the maximum number of acceptable rules considering the maximum number of constraints. The main advantage of GA is its ability to search in several directions, lessening the problem of getting trapped in *local minima*.

The choice of the fitness function is of important consideration. While the individual's collection of genes encodes the problem into a string, the fitness function indicates the performance of the system. Hence, finding a good fitness measurement for a system is of critical importance. Unlike the gradient-based methods, GA can be implemented to evolve systems with different types of fitness functions, including nondifferentiable and discontinuous ones. This leads to questions on how to define good fitness measurement. However, the most commonly used function is the mean square error or the absolute difference error.

A well-known approach to optimize the parametric fuzzy *if–then rule* is the concept of fuzzy entropy maximization [66], rather than based on the reduction of the training error. One major limitation of GAs is due to its lengthy processing time and the possibility to get trapped in a local minima. These issues can be addressed by leveraging on the advantages of the DE algorithm. For instance, Lampinen and Zelinka [67] developed a hybrid generation scheme to accelerate the convergence of the original DE algorithm, while Hassanein *et al.* [68], [69] proposed a hybrid fuzzy learning system using entropy and DE for the identification of dynamic systems.

3) Evolutionary Neurofuzzy Systems: Integrating evolutionary concepts into neurofuzzy systems can address some limitations of simpler form of hybridization [70]. For instance, the steepest descent optimization technique used in neurofuzzy system does not guarantee the convergence of the system as it may be trapped in local minimum. Nor is the successful tuning of the fuzzy MF guaranteed [59]. To overcome this challenge, evolutionary approaches can be embedded to perform structure and parameter optimization of the fuzzy system in FuGeNeSys [70]. In addition, evolving neurofuzzy [71] can adjust the MFs and the structure of the rules evolve. However, such systems can be computationally intensive and hence require a careful consideration so as not to limit their practical usefulness, especially for small UAVs with limited payload.

4) Genetic Learning–PSO: GAs and PSO have their own merits and demerits discussed as follows. While the PSO algorithm represents the evolution process in the form of particle updates toward their best positions in the absence of a selection operator, GA employs three basic reproductive operators (i.e., selection, crossover, and mutation). Also, while GA employs an omnidirectional reproduction process, the search in canonical PSO is more directional, that is, to drive particles by their previous best position and the global best position [26]. Accordingly, one can expect GA to achieve better explorability than PSO, whereas PSO can facilitate faster convergence. PSO has poor robustness to multimodal functions, while GA has superior global search ability to locate the global optimum of a multimodal function, albeit with much slower convergence speed. Biologically speaking, GA deals with a long-term process since it modifies the genotypes of the individual, while PSO deals with short-term iterative learning activities to adjust the phenotypes of particles [26]. The research question is how to combine both merits to improve the overall performance, that is, how to improve the global search ability of PSO using GA techniques without slowing down the searching process?

Gong *et al.* [26] developed a new framework for fusing PSO with GA, known as genetic learning–PSO algorithm, for advancement in learning. A population of particles derived from the PSO concept will interact to search for a solution. The algorithm comprises two stages, namely, exemplar generalization and particle updates. While particles can learn from the generalization of exemplars by genetic operators, the historical search information can guide the evolution of the exemplar. Performing crossover, mutation, and selection will diversify the constructed exemplars and improve their quality. This will result in improved efficiency and better global search capability of the system. The research by Gong *et al.* [26] indicates that the proposed algorithm outperforms the other search algorithms (e.g., EDA [73], ABC [54], and continuous ACO [72]) on a majority of benchmarks, such as global search ability, accuracy, speed, reliability, and scalability.

III. Intelligent Unmanned Aerial Vehicles

In this section, we will discuss several cutting-edge intelligent flight control systems for small UAVs. The summary of this discussion can be found in Table I.

A. Intelligent Flight Control Systems and Modeling

Flight control systems play a vital role in determining the maneuvering capability of a UAV. In what follows, we will discuss several cutting-edge intelligent flight control systems that were originally derived from nature.

Johnson and Kannan [74] proposed a neural-network-based adaptive flight control system for an autonomous helicopter (Yamaha R-Max) to achieve accurate tracking of position commands. This work was the first in the literature that employs adaptation to compensate for modeling errors in all 6 DOF. They employed a nonlinearly parametrized NN to provide online adaptation. To overcome the issue of unwanted adaptation, they employed the *pseudocontrol hedging* (PCH) technique to modify the dynamics of the inner loop reference to facilitate continued adaptation. PCH is a technique in model reference adaptive control (MRAC) to avoid the system from attempting to adapt the characteristics of plants and controllers [75] by moving the reference model backwards (hedging), meaning the adaptive element is prevented from seeing the system characteristics as a model of tracking error. The same principle was implemented to prevent unwanted adaptation to inner loop dynamics and to avoid interaction between the inner and outer loops. By hedging the outer loop, it was possible to adapt to uncertainty in the translational dynamics. The combination of the PCH and gain selection (obtained by combined analysis of the two loops) can alleviate the effect of the timescale separation [22] between the inner loop attitude control and the outer loop trajectory control, which dictates much higher bandwidth requirement for the inner loop (attitude control) in most aerial vehicles. Thus, the outer loop bandwidth can be set closer to that of the inner loop leading to improved tracking performance. The NN was fully capable of adapting rapid flight condition from the baseline at hover to the maximum speed of the helicopter in the absence of an accurate mathematical model at each point, unlike model-based approaches. The autopilot was tested to the trajectory and attitude controls of an unmanned Yamaha R-Max helicopter.

Tang *et al.* [76] introduced a novel PSO approach to enhance the controllability, where a set of UAVs fly together in formation. The PSO was implemented to select the pinned nodes and to determine the control gain. This includes a global search technique, a modified simulated binary crossover, and an adaptive inertia weight scheme. Besides, Tang *et al.* [77] also introduce an importance-based pinning strategy. The algorithm turns out to be quite effective and reliable in achieving its mission. However, there is room for improvement such as addressing the issue of faster convergence speed and controllability of time delay in complex networks. Global controllability also deserves further attention as this paper focuses on local controllability.

Fu *et al.* [77] developed a new *cross-entropy (CE) optimization-based fuzzy controller* to regulate the heading of a quadcopter UAV in order to avoid obstacles. While the concept of the CE (Kullback–Leibler) distance was derived from the fundamental concept of information theory, the optimization algorithm was motivated by an adaptive concept for estimating the probability of rare events, e.g., variance minimization. The process involves a two-step iterative procedure. In the first step, a random data sample representing scaling factors given in the form of a set of MFs of fuzzy logic controllers is generated based on a specific mechanism; in the subsequent stage, the parameters of the random mechanism will be updated based on data to achieve a more accurate sample in the next iteration. The effectiveness of the proposed algorithm was investigated through a series of simulation and real flight experiments using an AR.Drone quadcopter. The research by Fu *et al.* [78] indicates that the optimized fuzzy logic autopilot led to reasonably good performance as the drone was able to precisely navigate to avoid obstacles. Furthermore, there was a 64% reduction of the initial rule base (from 125 to 45 rules) as a result of the optimization of the scaling factor and the MFs.

Motivated by the robustness issue, Babaei *et al.* [78] addressed the issue of altitude holding for a nonminimum phase rotary wing UAV by means of *fuzzy-genetic* algorithms. Given the fact that fuzzy logic autopilots are designed based on the expert's knowledge and experience, the task of designing such rules become complicated for a large and complex dynamic system. Leveraging on the benefits of the multiobjective GA, they optimized the tuning mechanism of the fuzzy logic controller to achieve better performance with respect to a certain cost function. They also compared the performance of the proposed system with a classical autopilot. In addition, they also proposed a single-loop control scheme, unlike the conventional approach that employs more control loops resulting in the demand of having more measurements. While the autopilot was designed based on a nominal linear model, the robustness of the system was investigated based on the degraded linear model and nonlinear model to accommodate uncertainties. The research by Babaei *et al.* [79] indicates an improved performance in terms of the time response characteristics for both phugoid and short-period modes, in addition to the robustness and the adaptation of the system in regard to large commands.

Ghiglino *et al.* [25] introduced a simple heuristic-based tuning procedure to achieve a near optimal PID gain for a quadcopter UAV *starting from a scratch* for controlling attitude and thrust. They leverage on the advantages of ABC. Despite its simplicity and suitability for an online tuning scenario, the algorithm turns out to be very efficient and in some cases can outperform traditional GAs. Unlike many automatic tuning techniques discussed in the literature, such as MRAC [79]–[81], this approach can tune the PID controller in the absence of a stable initial gain, eliminating the burden of an initial tuning process. In this algorithm, a colony of artificial bees, acting as agents, search for rich artificial food sources representing potential optimal solutions for a given problem. Thus, the optimization problem can be seen as a problem of finding the best parameter vector that minimizes a certain objective function. Accordingly, the artificial bees will randomly discover a set of initial solution vectors, before iteratively employing the search strategies by means of the neighbor search mechanism. Although there is a possibility of the algorithm getting trapped in the local optima for a complex model, there are some modified versions of the algorithm to overcome this issue [82], [83]. To highlight the efficacy of the proposed algorithm, Armano and Farmani [83] and Lee *et al.* [84] perform both offline and online tuning. The offline tuning version is more computationally intensive as it can always achieve a set of optimal gain with a virtually zero standard deviation, and hence it can be used as a benchmark to gauge the performance of the online tuning. Their research indicates that the performance of the online tuning is reasonably close to the offline counterpart, with 80% less computational time than offline tuning.

Dierks and Jagannathan [84] developed a novel nonlinear autopilot for a quadcopter UAV using NN and *output feedback techniques*. The NN was introduced to learn the dynamics of the UAV online in the face of uncertainties, e.g., friction and blade flapping. The purpose of introducing the NN observer is to estimate the translational and rotational velocities of the UAV (e.g., dynamics and velocity vectors), since the output feedback control law is developed under an assumption that only the position and attitude of the UAV are measurable. A novel NN virtual control structure was implemented to allow the desired translational velocities to be fully controlled using the pitch and roll of the UAV. Dierks and Jagannathan [85] demonstrate the effectiveness of the proposed adaptive backstepping control algorithm in the face of uncertainties from the unknown nonlinear dynamics and disturbances. Their research indicates that the estimation of errors of each NN, observer, and virtual controller as well as the position, orientation, and velocity tracking errors satisfies semiglobally uniformly ultimately bounded condition. In addition, the NN observer can relax the separation principle, that is, by employing the same Lyapunov candidate for the NN observer errors as the UAV tracking errors. Their numerical results also confirm that the proposed controller outperforms the conventional linear controller.

Rezoug *et al.* [85] proposed an intelligent autopilot for a helicopter system comprising a *type-2 fuzzy* combined with the ACO technique as a defuzzifier. ACO is a probabilistic-based optimization method, inspired by the behavior of ants in finding paths between the nest or colony and the food source. While the manipulated inputs are the voltage of the main and the side motors, the two measured output signals are elevation and azimuth. The algorithm employs a set of paths presented as feedforward interconnected graphs to find the best path for solving the imposed problem. The benefits of the algorithm were investigated through a series of computer simulations. The main advantage of type-2 fuzzy system compared with the conventional type-1 fuzzy is its ability to handle uncertainties [86]. Type-2 fuzzy can be regarded as a generalization of type-1 fuzzy in the sense that the system defines uncertainties in its MFs and is not limited on the linguistic variables. Thus, the MF of the type-2 Fuzzy is in 3-D space to incorporate the footprint of uncertainties. The architecture of the type-2 fuzzy system contains five steps, namely, fuzzifier, rule-base, fuzzy inference engine, type-reducer, and defuzzifier [86], which can be regarded as an optimization problem with multiple constraints. To highlight the benefits of the proposed system, Castillo and Melin [87] compare the performance of the ACO with respect to the performance of the well-known PSO technique, which is also inspired by the behaviors of animals such as birds, fishes, and bees. In this regard, the particles iteratively move through the problem space until all converge to the same point. Their research indicates that hybrid ACO and type-2 fuzzy can outperform both standard type-2 fuzzy and the PSO algorithm, as indicated by near-zero static error in all cases and its best rise time in addition to its ability to conserve the control energy.

Amaral and Crisotomo [87] introduced a hybrid neurofuzzy autopilot for a Sikorsky S-61 helicopter model, comprising a Takagi–Sugeno-based fuzzy logic system and a general regression NN (GRNN) controller. The autopilot has only two outputs, namely, longitudinal and lateral cyclic, which controls the tilt of the main rotor and hence pitch and roll. While the GRNN technique facilitates one-pass learning algorithm with a highly parallel structure to allow for a smooth and rapid transition from one observed value to another, the system also employs a clustering algorithm to ease the computation amount given a training large data. The reason to employ a hybrid FLC/GRNN is their complementary nature. While fuzzy logic can combine numerical training data and expert linguistic knowledge, GRNN can facilitate rapid adaptive approach since it is noniterative process. Also, the smoothing parameter can be set large to smooth out noisy data or small to allow for a nonlinear regression surface to closely approximate the training data. However, GRNN requires the availability of large training sample to accurately represent the dynamics of the system. To overcome this drawback, Amaral and Crisotomo [88] also employed a clustering algorithm based on the Euclidean distance to acquire the representative exemplars and reject other data points. This way, training data can be reduced to achieve a faster controller. They verify the performance of the system with computer simulations.

Avanzini and Minisci [88] developed a hybrid evolutionary-H_∞ control technique to synthesize the controller gains that minimize a weighted combination of the infinite norm for sensitivity and transfer function in order to simultaneously meet both disturbance attenuation and robust stability requirements. To achieve satisfactory performance over the whole flight envelope, they also implemented the gain scheduling technique. They employed a hypothetical aircraft model and introduced two different approaches, namely, a single objective constrained optimization process and biobjective search. The effectiveness of the proposed algorithms was investigated through extensive computer simulations. Their research indicates the benefit of the evolutionary approach over traditional control design in addressing the time domain restriction during synthesis of the control law, rather than by means of a trial-and-error technique based on the *a posteriori* investigations. Also, the biobjective approach enables the determination of controllers that can perform very well outside nominal condition. However, the limitation of this approach is related to the limited portion of the flight envelope included in the analysis.

Garratt and Anavatti [89] proposed a nonlinear control technique for controlling the heave of an unmanned rotorcraft as seen in Fig. 3(a). They developed a hybrid plant model, comprising the known dynamics and a black box representation of it. The system employs a simple feedforward NN on the helicopter PC104 flight computer. Under manual control, the helicopter is set to hover before the control system is handed over to the NN autopilot. The system generates the desired trajectories to achieve a sequence of random step changes

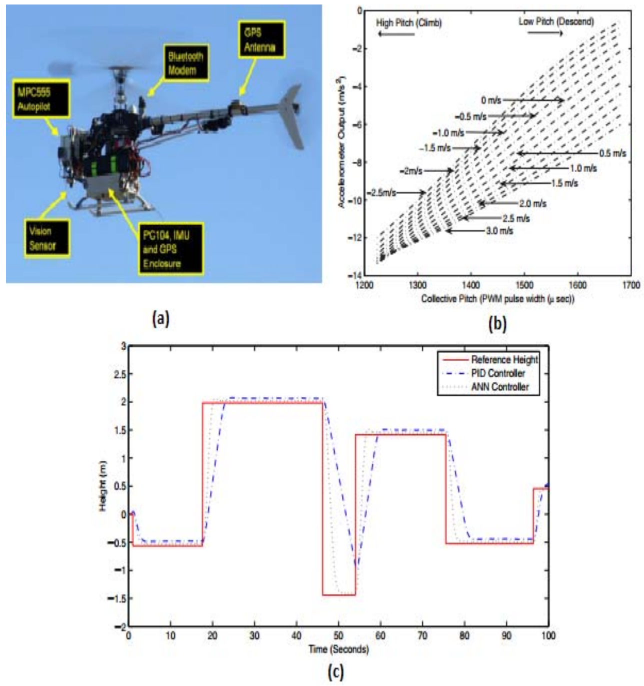

Fig. 3. NN control of the heave dynamic of an unmanned helicopter according to [89]. (a) Eagle helicopter platform. (b) Nonlinear characteristics of the main rotor blade pitch command signal with respect to vertical velocity and acceleration. (c) Performance of the NN control system versus conventional PD control.

in desired altitude based on a certain optimality index and by considering the limitation of the plant. The MATLAB optimization toolbox is employed to generate the desired trajectories and the NN is trained to mimic the optimized control input. The training data are affected by the nonlinear relation of the collective with respect to vertical velocity and acceleration, as given in Fig. 3(b). The research by Garratt and Anavatti [90] indicates that the NN autopilot achieves similar trajectories to the one generated by the optimization process, but with significantly shorter computational time. Their research indicates the capability of the NN autopilot to overcome severe disturbances and to outperform traditional PD controller [see Fig. 3(c)]. This also eliminates the need for modeling and system identification technique.

Tijani *et al.* [90] proposed a hybrid of conventional backpropagation training for nonlinear autoregressive with exogenous inputs networks (NARX) and multiobjective differential evolution algorithms for identification of the nonlinear model of a small-scale helicopter UAV from real-time flight data. Their research was motivated by the challenging demand to achieve a high-fidelity model of an unmanned helicopter system due to complex nonlinear characteristics of the systems while addressing the problem of determining optimal/suboptimal network architecture performance. Unlike other EA, DE solves the optimization problem using the fundamental concepts of population, initialization, mutation, crossover, and selection. The model developed was able to satisfy constraints between the design objectives. Compared with a similar approach using NARX networks, the proposed MODE-NARX model can enhance accuracy by 55% compared with the previous model architecture. The system was able to achieve a set of Pareto solutions with delicate balance (optimum solution) between accuracy and complexity. Apart from it, the proposed design algorithm can support timely development of the nonparametric model of the helicopter system. The proposed algorithm can serve as an automatic network selection architecture to overcome the trial-and-error selection of architecture selection of the parameters, which may require several days to achieve similar results depending on the experience of the designer.

Zhang *et al.* [94] proposed a memory based learning policy that can access salient information from previous observations. The system works by augmenting the state of the system with continuous memory states that has the policy to read and write information, suitable for high-dimensional continuous systems. The main research challenge in creating such a policy is related to the need to determine the most important information and to memorize it each step. Zhang *et al.* [95] demonstrate that through the guided policy search, achieved by decomposing the search policy into a trajectory optimization phase, effective memorization and recall techniques can be achieved. While the supervised learning employs memorization actions to develop memory states, trajectory optimization employs its optimization phase to choose an appropriate value of the states in order to determine the policy, leading to the right actions in future states. Based on continuous control task in manipulating and navigation settings, they demonstrate the effectiveness of their proposed method in dealing with complex policies, e.g., to remember target position by remembering past sensory inputs. The algorithm can also outperform other algorithms such as the long short-term memory neural networks as they are more difficult to train.

Zhang *et al.* [95] developed a hybrid control technique comprising MPC assisted with reinforcement learning in the framework of guided policy search. Their research aims to improve the practicality of MPC technique, which is effective for controlling aerial robots, yet being computationally intensive, and often require the state estimation of the system. While MPC is employed to generate training data, the data can be subsequently used to train a deep policy of neural networks to access observation from the on-board sensors. The MPC algorithm minimizes the cost function against matching up the policy of the neural network. Accordingly, the neural network can be employed to control the vehicle in the absence of the full state knowledge, resulting in substantially lower computational cost of MPC. By employing MPC for offline trajectory optimization, the system can be made robust to error and hence the catastrophic failures on the training phase can be avoided. By employing supervised learning for the final policy, one can train complex and high dimensional policies, representing the behavior of the complex dynamic systems. However, the system shares limitations of the prior guided policy method, e.g., the requirement to have a full state observation at training phase to perform MPC, although the resulting neural network policy can employ the on-board sensors to perform control tasks in the real world. Thus, although this technique is practical for many robotic tasks, some UAV maneuvers cannot be learned within an instrumented training

TABLE I

STATE-OF-THE-ART BIOINSPIRED FLIGHT CONTROL SYSTEMS

No	UAV Platfoms	Purpose	Algorithms	Pros	Cons
1	Yamaha R-Max helicopter [74]	To improve the position tracking performance	Pseudo Control Hedging (PCH), neural network	The outer loop can continue the adaptation process irrespective of the dynamics of the inner loop, robustness in the face of modelling error, accurate tracking performance, Ability to adapt rapidly changing environment	The use of neural networks as a curve fitting technique may lead to computational complexity due to the tendency of over-fitting as a result of its black box nature.
2	A network of multiple UAVs [77]	To maximise the controllability of complex UAV coordination	Particle Swarm Optimisation (PSO) combined with a global search scheme and a modified simulated binary crossover (MSBX)	Reliability and better performance compared to other EAs	Rather slow convergence rate, only focuses on local controllability, neglected the effects of time delay.
3	Aerosonde UAV [92]	Autonomous navigation	To develop an adaptive neuro-fuzzy inference system (ANFIS) flight control system	Fast and accurate control system. To facilitate automated learning process to the fuzzy system	For some flight conditions, the ANFIS controller has resulted in unstable performance. It lacks practical consideration.
4	AR.Drone Quadcopter [78]	To propose an obstacle avoidance system in order to address see-and-avoid scenario	Cross-Entropy optimisation-based fuzzy logic controller	Improved performance, reduced rules by 64 %	Current system has no omnidirectional sensing capability.
5	Hypothetical non-minimum phase fixed-wing UAV [79]	To achieve robust altitude holding	Fuzzy-GA	Better robustness and more accurate time domain performance	There is no practical implementation of the proposed autopilot.
6	Quadcopter drone [25]	To perform automatic online tuning for PID autopilots	Novel evolutionary swarm algorithm, namely, Artificial Bee Colony (ABC)	Simple, efficient, suitable for online tuning scenario, can outperform conventional GA in some cases	For complex systems there is a small chance to get trapped in a local minima.
7	Birotor helicopter [86]	To introduce an intelligent autopilot system	Combined Type-2 fuzzy logic system and the Ant Colony Optimisation	Better capability of handling uncertainties, superior performance compared to the conventional fuzzy autopilot and the PSO.Conceptual simplicity and high efficiency.	May not be suitable for system with limited payload.
8	Individual and group of UAVs (hypothetical) [93]	To propose a collision avoidance algorithm in individual and group of UAVs	Swarm intelligence-based collision avoidance algorithm	Stability and robustness, built-in redundancy compared to the leader-follower method	Parameters can have dramatic effect on the collective behavior of the system.
9	Hypothetical UAV [94]	To develop an adaptive route planning for a single UAV	Modified Firefly Algorithm (MFA)	Simple, faster convergence rate, flexible, better robustness. The proposed algorithm is not sensitive to the parameter light absorption coefficient and discovery rate, eliminating the need for fine tuning those parameters	It lacks practical implementation.
10	Sikorsky S-61 helicopter model [88]	To propose high-performance neuro-fuzzy controller in the hover and forward flight modes	Hybrid fuzzy logic controller (FLC) and general regression neural network (GRNN)	Computationally efficient and rapid response	It requires large training samples. Analysis was based on simulation only.
11	Hypothetical aircraft [89]	To develop a high performance autopilot	Hybrid evolutionary-H_∞ and gain scheduling techniques	The system is able to address the time domain constraint. The system also perform reasonably well outside its nominal condition based on local information only	Relatively limited portion of the flight enveloped was analysed.

setup. One way to overcome this issue is by combining state estimation techniques with guided policy search.

Li *et al.* [96] developed a hybrid hierarchical control strategy based on adaptive RBFNNs and the integral sliding mode technique for attitude and position control of a quadcopter UAV in the face of disturbance and uncertainties. The proposed control system uses the integral double loop sliding mode technique to track desired attitude with zero steady-state error while using an RBFNN to approximate arbitrary functions to generate an aggressive control law so that the state variables can converge faster and to quickly reject disturbances. Furthermore, Li *et al.* [97] employ Lyapunov's stability technique to prove the null tracking error performance and adaptive estimation capability of the system. The effectiveness of the proposed control system is compared with respect to the performance of the traditional PD/integral sliding model and backstepping/nonlinear H_∞ controllers.

Howard and Merz [97] proposed an evolutionary autopilot for a quadcopter UAV to tune the performance of the cascaded on-board PID controller. The system works by means of the differential evolution technique. There are multiple benefits of the proposed platform. First, the system is reliable since it can continuously operate for over 24 h. The controller can also be implemented straight away without the need for modeling. Furthermore, the system also offers flexibility since the optimized parameters can stay beyond its estimated bounds. Unlike the traditional tuning method, there is no need for separation between positions from the altitude loop. This will pave the way to a new research direction, e.g., investigating the effects of different payload configurations on the optimality of gain and comparison with different EAs in hardware.

Howard and Elfes [98] investigated the development of the spiking neural networks autopilot, comprising a self-adaptive evolutionary algorithm to generate weight combination in the networks, in addition to a synaptic plasticity mechanism. They studied the effects of pairing a dynamic controller with a problem of the same properties. This work is the first in the literature that employs evolutionary computing to develop spiking neurocontrollers for a quadcopter. It further shows that even simple plasticity schemes can be employed for adaptation in dynamic scenario, allowing the spiking networks to operate at a higher frequency compared to one with traditional encoding techniques. Their research also indicates that the spiking controllers can achieve reasonably less tracking error than conventional PID controllers and the feedforward neural networks.

IV. Discussion: Opportunities and Challenges

In this section, we will highlight several research opportunities and potential challenges in intelligent aerial robotics.

A. Computational Demand and System Capabilities

Considering the practical aspects of the intelligent control algorithms, one needs to carefully consider the trade-off between the required computational demand (upper and lower bounds) and the processing capabilities of the systems, including the data acquisition rate and the capabilities of the sensors. For instance, a 6-DOF aerial robot that has m state variables represented with n linguistic variables will require approximately n^m fuzzy rules, compared with several GA techniques (e.g., *shortest path* algorithm that has $\mathcal{O}(n^3)$ complexity [99] and one maximum algorithm that has $\mathcal{O}(n \log n)$ complexity [100], where n indicates the input of the systems). Although intelligent systems may be demanding in terms of the computational demand and the processing time, we are confident that this issue has become progressively less relevant due to substantial increase in computational power, speed, processors, and the size of memory as clearly stated by Moore's Law. The use of parallel computation as in [44] is also a possible solution to speed up the computational time.

B. Online Learning: Structural and Parametric

As a universal approximator, fuzzy inference systems provide an alternative solution to overcome the drawbacks of model-based control systems in the absence of complex mathematical models. Its reasoning capability can be performed by imitating human expertise as represented by its rule base and database. However, since the reasoning process heavily relies on the *a priori* knowledge about the system, it is desirable to automate the acquisition process of such knowledge, particularly for complex and large-scale systems, where manual development is impractical. This learning task can be performed by integrating neurofuzzy systems, that is, to train the fuzzy system by a learning algorithm derived from the neural network theory. While classical neurofuzzy systems employ a static model derived based on its offline data, current research directions have been shifted toward the use of dynamic model, which provides a more realistic solution to deal with nonstationary environments. For instance, dynamic environments (e.g., atmospheric turbulence) can force the aircraft to move to a new operating regime. This circumstance can lead to a demand to synthesize new knowledge since the previously learned knowledge is no longer appropriate—offline training data only provide limited coverage of a certain operating condition.

Addressing the limitation of the static models, dynamic models have been implemented by means of a two-stage training process, namely, structure and parameter learning. While structure learning can be initially performed using offline data (before its parameters can be fine-tuned using a certain error reduction scheme, e.g., genetic-based structure learning or backpropagation method), this process turns out to be insufficient to meet the dynamics of the incoming data streams. The remedy of this issue is given in the form of online structure learning [101]. Such processes require the development of more advanced learning algorithms such as the integration of the fuzzy role induction algorithm with a rough set-assisted feature reduction method. Some examples of other learning algorithms can be found in the works of Salehfar *et al.* [103] and Soukkou *et al.* [104], where they propose the use of learning vectors and reduced Takagi–Sugeno rule base, respectively.

Although online training can be used to improve the accuracy of the offline model, its accuracy may not be sufficient due to uncertainties and error in measurements. Thus, to get the best of both worlds, one needs to develop robust and accurate offline model first. Furthermore, the accuracy of the offline model can be improved by performing online training.

C. Expedited Learning Time

Another issue to look at is related to the *efficiency* of the learning algorithms in terms of computational bits and learning time [104]. Computationally complex algorithms can lead to intractable processing, especially for large data stream. This will be of little value especially for fast and challenging systems, such as in UAV systems, where high control bandwidth is often required to cope with fast dynamics and real-time performance is essential. One way to achieve an efficient learning mechanism is to overcome the issue of *overfitting*, a common problem with the increasing of the rule base to encompass all variations in incoming data.

Another issue worth considering is related to the questions of how to filter out the extraneous superfluous incoming data that may not lead to new knowledge of the existing model, in addition to having the *ability to simplify the rule base*. Thus, the development of the *rule–recall* mechanism is of critical importance to alleviate the necessity for new learning mechanism and thus increasing computational efficiency. This step will enhance the suitability of the algorithms to cope with (near) real-time and online demands. Without loss of generality, this procedure may restrict the algorithm to update models and statistical information in a samplewise manner using as little prior data as possible.

D. Uncertainties in Data Distribution and Representation

Real-world data streams, which are characterized by their inexact, inaccurate, and uncertain nature [105] (e.g., due to disagreements in expert knowledge, measurement, and process noise), can lead to the concept of drifting, where the input and/or output concepts do not follow a predictable data distribution. Meanwhile, disagreement in expert knowledge and the measurement noise can cause uncertainty in its representation. Because data may not necessarily represent the system dynamics, the identification of proper structures (e.g., type-2 fuzzy rules) representing such uncertainties becomes an important task. Considering the issue of classification, uncertainty can lead to confusion or overlapping class in the feature space. Although some mature evolutionary algorithms have the ability to capture rapidly changing environments to cope with abrupt drifts, the systems may still have poor robustness in the face of gradual, incremental, or cyclic drifts.

These types of drifts are more difficult to overcome since they cannot be detected by global drift detection approaches in both structural and parametric learning scenarios. This circumstance can interfere with current data distribution and may severely suppress the accuracy of the system. Moreover, cyclic drift can change the functional characteristics of the system over time, e.g., forgetting previously learned knowledge. A missed component (i.e., neuron or branch) in a certain training episode cannot be recovered in the future. Many evolutionary systems do not have good robustness because of this issue.

E. Stability–Plasticity Dilemma

Stability–plasticity dilemma is a well-known constraint in neural networks in both biological and artificial systems [106]. Given the nature of large and unbounded data streams, uncertainties could occur when the system is trying to achieve a delicate balance between integrating new knowledge and keeping the existing one. While learning in parallel and distributed systems requires plasticity for the integration of new knowledge, stability is also needed to prevent forgetting previous knowledge. Catastrophic forgetting is an emerging issue in artificial neural networks. This takes place when the systems completely forget previously learned information as soon as it is exposed to new information. Too much plasticity can lead to constant forgetfulness of the previously encoded data, while too much stability can suppress the efficient coding of the data at synapses level.

One way to overcome this issue is by implementing the concept of double-loop dynamic systems as in the work of Lecerf [107], where he introduced a double-loop concept as a core of a structural and dynamic model in large neural networks so that the system has a built-in capability to address this dilemma. Meanwhile, Tung and Quek [108] introduced a generic self-organizing fuzzy NN that has not only strong noise rejection capability due to the proposed discrete incremental clustering technique,but it has also the ability to overcome the stability-plasticity dilemma as a result of the consistency and compactness of its fuzzy rule-base, as it has a built-in mechanism to identify and remove redundant and out-of-date rules.

F. Curse of Dimensionality

The curse of dimensionality can be regarded as a joint problem between the data and the algorithm being applied. It happens when the algorithm does not scale to high-dimensional data due to high processing time or memory requirements that grow exponentially with respect to the data dimension. This issue makes it difficult if not impossible to setup a rule-base with more than three inputs while maintaining the interoperability of the end user [30].

Although this issue can be addressed by means of the decentralized systems, it is apparent that reducing the curse of dimensionality and avoiding overfitting can improve the accuracy of the model. The feature selection process can be carried out offline using prerecorded samples. An input feature that cannot be recalled in the future once discarded can cause the discontinuity of the training process that justifies the need of retraining phase from scratch.

Considering an n-dimensional fuzzy controller, which performs n-to-one mapping, the number of input variables can exponentially increase the number of fuzzy rules or parameters. Therefore, designing a high-dimensional fuzzy controller is not a trivial task, particularly for systems with a large number of inputs [109]. Addressing this issue remains a major research challenge in evolutionary algorithms. One potential solution of this problem is given in Du *et al.* [109], where they investigated a class of high-dimensional zero-order Takagi–Sugeno fuzzy controllers, originally proposed by Lewis and Liu [110], which are able to overcome the curse of dimensionality. They also explored the local stability and bounded-input bounded-output stability of the fuzzy systems.

G. Limited Payloads and Challenging Flight Environments

To be able to successfully navigate in challenging environments, such as in cluttered environments (e.g., indoor building or urban area), UAVs are often required to possess high control bandwidth. However, small UAVs are always subject to limited computational and mechanical payloads. This will signify the importance of having a control algorithm that is not only optimal and intelligent but also capable of performing efficient self-learning in the face of uncertainties, actuator constraints, and variations in data trends.

V. Conclusion

We have explored several important aspects of intelligent aerial robotics both from methodological and technological point of view, such as processing speed, reliability, flexibility, and learning capability. We believe that bioinspired computation still has much room to grow since current progress is still in its infancy. It is considered among the most powerful computation techniques and is believed to be the next generation of computing. While there are great opportunities in exploring new concepts, the main challenge associated with this topic is related to its interdisciplinary nature. Thus, collaborations of research studies from different communities (e.g., computer science, life science, and artificial intelligence) may be required.

While we believe the future trend of small UAV will be moving toward simpler, faster, more robust, and lower cost designs, we also envisage that evolutionary algorithms will have played more vital roles in addressing the shortcomings of model-based control, e.g., due to better robustness, practicality, and flexibility in the absence of complex mathematical models.

For instance, in complex dynamic systems, it is very difficult, if not impractical, to derive complete mathematical models from first principles. Although it is possible to perform experiment-based modeling by means of the system identification technique, the resulting mathematical models will contain some uncertainties. The use of linear models will only be able to capture the dynamics of the system for a limited range around its linearization point. This problem can be easily tackled using the concept of *evolutionary* systems, offering more flexibility and robustness to overcome the uncertain, imprecise, and inaccurate nature of real-world problems.

Considering multiple tradeoffs, one must carefully address the specific demand of a certain mission. There are various demands in the UAV market place, e.g., cost, speed, reliability and accuracy. The *no-free-lunch* theorem states that all optimization methods, on average, demonstrate a similar performance over all possible instances. Hence, we believe that there is no one-size-fits-all system, since each algorithm possesses its merits and demerits. Leveraging on the advantages of multiple evolutionary strategies, we envisage that hybrid evolutionary systems such as fuzzy/GA, fuzzy/NN, and PSO/GA will predominate future markets, especially for UAVs, where designers are often required to meet multiple optimization objectives. This way, one can easily offset the limitations of a certain technique with the advantages of other methods.

REFERENCES

[1] X. C. Ding, A. R. Rahmani, and M. Egerstedt, "Multi-UAV convoy protection: An optimal approach to path planning and coordination," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 256–268, Apr. 2010.

[2] A. Pahsa, P. Kaya, G. Alat, and B. Baykal, "Integrating navigation & surveillance of unmanned air vehicles into the civilian national airspaces by using ADS-B applications," in *Proc. Integr. Commun., Navigat. Surveilance Conf. (ICNS)*, May 2011, pp. J7-1–J7-7.

[3] B. S. Morse, C. H. Engh, and M. A. Goodrich, "UAV video coverage quality maps and prioritized indexing for wilderness search and rescue," in *Proc. 5th ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, Mar. 2010, pp. 227–234.

[4] B. S. Faiçal, G. Pessin, G. P. R. Filho, A. C. P. L. F. Carvalho, G. Furquim, and J. Ueyama, "Fine-tuning of UAV control rules for spraying pesticides on crop fields," in *Proc. 26th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2014, pp. 527–533.

[5] R. A. Zahawi, J. P. Dandois, K. D. Holl, D. Nadwodny, J. L. Reid, and E. C. Ellis, "Using lightweight unmanned aerial vehicles to monitor tropical forest recovery," *Biol. Conservation*, vol. 186, pp. 287–295, Jun. 2015.

[6] J. Nikolic, M. Burri, J. Rehde, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A UAV system for inspection of industrial facilities," in *Proc. IEEE Aerosp. Conf.*, Mar. 2013, pp. 1–8.

[7] M. Liu, G. K. Egan, and F. Santoso, "Modeling, autopilot design, and field tuning of a UAV with minimum control surfaces," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2353–2360, Nov. 2015.

[8] F. Santoso, M. A. Garratt, and S. G. Anavatti, "VisualâŁ"inertial navigation systems for aerial robotics: Sensor fusion and technology," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 260–275, Jan. 2017.

[9] O. Spinka, S. Kroupa, and Z. Hanzalek, "Control system for unmanned aerial vehicles," in *Proc. 5th IEEE Int. Conf. Ind. Inform.*, vol. 1, Jun. 2007, pp. 455–460.

[10] F. Santoso, M. Liu, and G. Egan, "H_2 and H_∞ robust autopilot synthesis for longitudinal flight of a special unmanned aerial vehicle: A comparative study," *IET Control Theory Appl.*, vol. 2, no. 7, pp. 583–594, 2008.

[11] F. Santoso, M. Liu, and G. Egan, "Optimal control linear quadratic synthesis for an unconventional aircraft," in *Proc. 12th Austral. Int. Aerosp. Congr. (AIAC)*, Mar. 2006.

[12] L. Singh and J. Fuller, "Trajectory generation for a UAV in urban terrain, using nonlinear MPC," in *Proc. Amer. Control Conf.*, vol. 3, Jun. 2001, pp. 2301–2308.

[13] J. R. Azinheira and A. Moutinho, "Hover control of an UAV with backstepping design including input saturations," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 3, pp. 517–526, May 2008.

[14] R. G. Hernández-García and H. Rodríguez-Cortés, "Transition flight control of a cyclic tiltrotor UAV based on the gain-scheduling strategy," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Jun. 2015, pp. 951–956.

[15] K. J. Åström and B. Wittenmark, *Adaptive Control*. Reading, MA, USA: Addison-Wesley, 1995.

[16] C. Tan, G. Tao, and R. Qi, "Multiple-model based adaptive control design for parametric and matching uncertainties," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2014, pp. 2353–2358.

[17] I. Petersen, V. A. Ugrinovskii, and A. V. Savkin, *Robust Control Design Using H_∞ Methods*. Berlin, Germany: Springer-Verlag, 2000.

[18] A. T. Gaitan and Y. Bolea, "Modeling and robust attitude control of a quadrotor system," in *Proc. 10th Int. Conf. Comput. Sci. Autom. Control (CCE)*, Sep./Oct. 2013, pp. 7–12.

[19] L. Qiu, G. Fan, J. Yi, and W. Yu, "Robust hybrid controller design based on feedback linearization and μ synthesis for UAV," in *Proc. 2nd Int. Conf. Intell. Comput. Technol. Autom.*, Oct. 2009, pp. 858–861.

[20] F. Santoso, M. Liu, and G. K. Egan, "Robust μ-synthesis loop shaping for altitude flight dynamics of a flying-wing airframe," *J. Intell. Robot. Syst.*, vol. 79, no. 2, pp. 259–273, Aug. 2015.

[21] S. Kanchanavally, R. Ordónez, and C. J. Schumacher, "Path planning in three dimensional environment using feedback linearization," in *Proc. Amer. Control Conf.*, Jun. 2006, pp. 1–6.

[22] A. J. Calise and R. T. Rysdyk, "Nonlinear adaptive flight control using neural networks," *IEEE Control Syst.*, vol. 18, no. 6, pp. 14–25, Dec. 1998.

[23] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of evolutionary fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 109–119, Apr. 1999.

[24] A. Abraham, "Neuro fuzzy systems: State-of-the-art modeling techniques," *Connectionist Models Neurons, Learn. Process., Artif. Intell., Lecture Notes Comput. Sci.*, vol. 2084, J. Mira, A. Prieto, Eds. Berlin, Germany: Springer-Verlag, 2001.

[25] P. Ghiglino, J. L. Forshaw, and V. J. Lappas, "Online evolutionary swarm algorithm for self-tuning unmanned flight control laws," *J. Guid., Control Dyn.*, vol. 38, no. 4, pp. 772–782, Apr. 2015.

[26] Y.-J. Gong *et al.*, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, Oct. 2016.

[27] E. Bonabeau, G. Theraulaz, and M. Dorigo, *Swarm Intelligence: From Natural to Artificial Systems* (Santa Fe Institute Studies on the Sciences of Complexity). New York, NY, USA: Oxford Univ. Press, 1999.

[28] J. Vieira, F. M. Dias, and A. Mota, "Neuro-fuzzy systems: A survey," in *Proc. 4th WSEAS Int. Conf. Neural Netw.*, 2004, pp. 1–6.

[29] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

[30] P. Albertos and A. Sala, "Fuzzy logic controllers. Advantages and drawbacks," in *Proc. 8th Congr. Latinoamericano Control Autom.*, 1998, pp. 833–844.

[31] S. Razavi and B. A. Tolson, "A new formulation for feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1588–1598, Oct. 2011.

[32] H. Zhang, Z. Wang, and D. Liu, "A comprehensive review of stability analysis of continuous-time recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1229–1262, Jul. 2014.

[33] P. A. Estevez, R. Hernandez, C. A. Perez, and C. M. Held, "Gamma-filter self-organising neural networks for unsupervised sequence processing," *Electron. Lett.*, vol. 47, no. 8, pp. 494–496, Apr. 2011.

[34] Y. Li, Z. Tang, G. Xia, and R. Wang, "A positively self-feedbacked Hopfield neural network architecture for crossbar switching," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 1, pp. 200–206, Jan. 2005.

[35] Y. Wang, Y. Chenxie, J. Tan, C. Wang, Y. Wang, and Y. Zhang, "Fuzzy radial basis function neural network PID control system for a quadrotor UAV based on particle swarm optimization," in *Proc. IEEE Int. Conf. Inf. Autom.*, Aug. 2015, pp. 2580–2585.

[36] S. Basterrech and G. Rubino, "A more powerful random neural network model in supervised learning applications," in *Proc. Soft Comput. Pattern Recognit. (SoCPaR)*, Hanoi, Vietnam, Dec. 2013, pp. 201–206.

[37] Q. Yang and S. Jagannathan, "Online reinforcement learning-based neural network controller design for affine nonlinear discrete-time systems," in *Proc. Amer. Control Conf.*, New York, NY, USA, Jul. 2007, pp. 4774–4779.

[38] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1975.

[39] X. Meng and B. Song, "Fast genetic algorithms used for PID parameter optimization," in *Proc. Int. Conf. Autom. Logistics*, Jinan, China, Aug. 2007, pp. 2144–2148.

[40] T. Shima, S. J. Rasmussen, and A. G. Sparks, "UAV cooperative multiple task assignments using genetic algorithms," in *Proc. Amer. Control Conf.*, Jun. 2005, pp. 2989–2994.

[41] C.-W. Chou, C.-F. Chien, and M. Gen, "A multiobjective hybrid genetic algorithm for TFT-LCD module assembly scheduling," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 692–705, Jul. 2014.

[42] D. M. Mukhopadhyay, M. O. Balitanas, A. Farkhod, S.-H. Jeon, and D. Bhattacharyya, "Genetic algorithm: A tutorial review," *Int. J. Grid Distrib. Comput.*, vol. 2, no. 3, pp. 25–32, Sep. 2009.

[43] A. Chipperfield and P. Fleming, "An overview of evolutionary algorithms for control systems engineering," Dept. Automat. Control Syst. Eng., Univ. Sheffield, Sheffield, U.K., Tech. Rep., 1995.

[44] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[45] G. J. Barlow, C. K. Oh, and E. Grant, "Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Dec. 2004, pp. 689–694.

[46] J. C. Kahn and J. L. Tangorra, "Application of a micro-genetic algorithm for gait development on a bio-inspired robotic pectoral fin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 3784–3789.

[47] N. S. Pal and S. Sharma, "Robot path planning using swarm intelligence: A survey," *Int. J. Comput. Appl.*, vol. 83, no. 12, pp. 5–12, Dec. 2013.

[48] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[49] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, Nagoya, Japan, Oct. 1995, pp. 39–43.

[50] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[51] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, Aug. 2009.

[52] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792. Berlin, Germany: Springer-Verlag, 2009, pp. 169–178.

[53] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," *Int. J. Soft Comput. Eng.*, vol. 2, no. 2, pp. 137–151, May 2012.

[54] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Nov. 2007.

[55] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd ed. Bristol, U.K.: Luniver Press, 2010.

[56] J.-C. Zufferey, S. Hauert, T. Stirling, S. Leven, J. Roberts, and D. Floreano, "Aerial collective systems," in *Handbook of Collective Robots*, S. Kernbach, Ed. Singapore: Pan Stanford, 2013, pp. 609–660.

[57] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Jul. 1987.

[58] S. Hauert *et al.*, "Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 5015–5020.

[59] G. Castellano, C. Castiello, A. M. Fanelli, and L. Jain, "Evolutionary neuro-fuzzy systems and applications," in *Advances in Evolutionary Computing for System Design* (Studies in Computational Intelligence), vol. 66. Berlin, Germany: Springer-Verlag, 2007, pp. 11–45.

[60] C.-T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1320–1336, Dec. 1991.

[61] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

[62] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 724–740, Sep. 1992.

[63] D. Nauck and R. Kruse, "Neuro-fuzzy systems for function approximation," *Fuzzy Sets Syst.*, vol. 101, no. 2, pp. 261–271, Jan. 1999.

[64] S. Tano, T. Oyama, and T. Arnould, "Deep combination of fuzzy inference and neural network in fuzzy inference software—FINEST," *Fuzzy Sets Syst.*, vol. 82, no. 2, pp. 151–160, Sep. 1996.

[65] S. M. Sulzberger, N. Tschichold-Gurman, and S. J. Vestli, "FUN: Optimization of fuzzy rule based systems using neural networks," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1, Mar./Apr. 1993, pp. 312–316.

[66] X.-Z. Wang and C.-R. Dong, "Improving generalization of fuzzy IF–THEN rules by maximizing fuzzy entropy," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 3, pp. 556–567, Jun. 2009.

[67] J. Lampinen and I. Zelinka, "On stagnation of the differential evolution algorithm," in *Proc. 6th Int. Conf. Soft Comput. (MENDEL)*, Brno, Czech Republic, 2000, pp. 76–83.

[68] O. Hassanein, S. G. Anavatti, and T. Ray, "Black-box tool for nonlinear system identification based upon fuzzy system," *Int. J. Comput. Intell. Appl.*, vol. 12, no. 2, p. 1350009, Jun. 2013.

[69] O. Hassanein, S. G. Anavatti, H. Shim, and S. A. Salman, "Auto-generating fuzzy system modelling of physical systems," in *Proc. IEEE Multi-Conf. Syst. Control*, Sep. 2015, pp. 1142–1147.

[70] M. Russo, "FuGeNeSys—A fuzzy genetic neural system for fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 3, pp. 373–388, Aug. 1998.

[71] A. Abraham, "EvoNF: A framework for optimization of fuzzy inference systems using neural network learning and evolutionary computation," in *Proc. IEEE Int. Symp. Intell. Control (ISIC)*, Oct. 2002, pp. 327–332.

[72] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, Mar. 2008.

[73] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Optimization in continuous domains by learning and simulation of Gaussian networks," in *Proc. Genet. Evol. Comput. Conf.*, 2000, pp. 201–204.

[74] E. N. Johnson and S. K. Kannan, "Adaptive trajectory control for autonomous helicopters," *J. Guid., Control, Dyn.*, vol. 28, no. 3, pp. 524–538, May/Jun. 2005.

[75] E. N. Johnson and A. J. Calise, "Pseudo-control hedging: A new method for adaptive control," in *Proc. Workshop Adv. Guid. Control Technol.*, Redstone Arsenal, AL, USA, Nov. 2000, pp. 1–23.

[76] Y. Tang, H. Gao, J. Kurths, and J.-A. Fang, "Evolutionary pinning control and its application in UAV coordination," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 828–838, Nov. 2012.

[77] C. Fu, M. A. Olivares-Mendez, P. Campoy, and R. Suarez-Fernandez, "UAS see-and-avoid strategy using a fuzzy logic controller optimized by cross-entropy in scaling factors and membership functions," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, May 2013, pp. 532–541.

[78] A. R. Babaei, M. Mortazavi, and M. H. Moradi, "Classical and fuzzy-genetic autopilot design for unmanned aerial vehicles," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 365–372, Jan. 2009.

[79] B. Whithehead and S. Bieniawski, "Model reference adaptive control of a quadcopter UAV," in *Proc. AIAA Guid., Navigat., Control Conf.*, Aug. 2010, pp. 1–13.

[80] M. Achtelik, T. Bierling, J. Wang, L. Höcht, and F. Holzapfel, "Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties," in *Proc. Infotech@Aerospace*, Mar. 2011, pp. 1–15.

[81] I. Sadeghzadeh, A. Mehta, and Y. Zhang, "Fault/damage tolerant control of a quadcopter helicopter UAV using model reference adaptive control and gain-scheduled PID," in *Proc. AIAA Guid., Navigat., Control Conf.*, Aug. 2011, pp. 1–13.

[82] G. Armano and M. R. Farmani, "Clustering analysis with combination of artificial bee colony algorithm and k-means technique," *Int. J. Comput. Theory Eng.*, vol. 6, no. 2, pp. 141–145, 2014.

[83] S. Lee *et al.*, "A hybrid of artificial bee colony and flux balance analysis for identifying optimum knockout strategies for producing high yields of lactate in *Echerichia coli*," in *Advances in Biomedical Infrastructure* (Studies in Computational Intelligence), vol. 477, A. S. Sidhu and S. K. Dhillon, Eds. Berlin, Germany: Springer-Verlag, 2013.

[84] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 50–66, Jan. 2010.

[85] A. Rezoug, Z. Achour, and M. Hamerlain, "Ant colony optimization of type-2 fuzzy helicopter controller," in *Proc. Int. Conf. Robot. Biomimetics*, Bali, Indonesia, Dec. 2014, pp. 1548–1553.

[86] O. Castillo and P. Melin, *Type-2 Fuzzy Logic: Theory and Applications*. Berlin, Germany: Springer-Verlag, 2008.

[87] T. G. B. Amaral and M. M. Crisotomo, "Neuro-fuzzy controller for helicopter motion control," in *Proc. 10th IEEE Int. Conf. Fuzzy Syst.*, vol. 3, Dec. 2001, pp. 594–597.

[88] G. Avanzini and E. A. Minisci, "Evolutionary design of a full-envelope flight control system for an unstable fighter aircraft," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[89] M. Garratt and S. Anavatti, "Non-linear control of heave for an unmanned helicopter using a neural network," *J. Intell. Robot. Syst.*, vol. 66, no. 4, pp. 495–504, Sep. 2012.

[90] I. B. Tijani, R. Akmeliawati, A. Legowo, and A. Budiyono, "Nonlinear identification of a small scale unmanned helicopter using optimized NARX network with multiobjective differential evolution," *Eng. Appl. Artif. Intell.*, vol. 33, pp. 99–115, Aug. 2014.

[91] S. Kurnaz, O. Cetin, and O. Kaynak, "Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1229–1234, Mar. 2010.

[92] R. Sharma and D. Ghose, "Swarm intelligence based collision avoidance between realistically modelled UAV clusters," in *Proc. Amer. Control Conf.*, Jul. 2007, pp. 3892–3897.

[93] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "A modified firefly algorithm for UCAV path planning," *Int. J. Hybrid Inf. Technol.*, vol. 5, no. 3, pp. 123–143, Jul. 2012.

[94] M. Zhang, Z. McCarthy, C. Finn, S. Levine, and P. Abbeel, "Learning deep neural network policies with continuous memory states," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 520–527.

[95] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 528–535.

[96] S. Li, Y. Wang, J. Tan, and Y. Zheng, "Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft," *Neurocomputing*, vol. 216, pp. 126–134, Dec. 2016.

[97] D. Howard and T. Merz, "A platform for the direct hardware evolution of quadcopter controllers," in *Proc. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep./Oct. 2015, pp. 4614–4619.

[98] D. Howard and A. Elfes, "Evolving spiking networks for turbulence-tolerant quadrotor control," in *Proc. Int. Conf. Synth. Simulation Living Syst.*, 2016, pp. 1–8.

[99] J. Scharnow, K. Tinnefeld, and I. Wegener, "Fitness landscapes based on sorting and shortest paths problems," *Parallel Problem Solving Nature, Lecture Notes Comput. Sci.*, vol. 2439, J. J. M. Guervós, P. Adamidis, H. G. Beyer, H. P. Schwefel, J. L. Fernández-Villacañas, Eds. Berlin, Germany, Springer-Verlag, 2002.

[100] H. Muhlenbein, "How genetic algorithms really work: I. Mutation and hillclimbing," in *Proc. Parallel Problem Solving Nature*, 1992, pp. 15–26.

[101] Q. Shen and A. Chouchoulas, "A rough-fuzzy approach for generating classification rules," *Pattern Recognit.*, vol. 35, no. 11, pp. 2425–2438, Nov. 2002.

[102] H. Salehfar, N. Bengiamin, and J. Huang, "A systematic approach to linguistic fuzzy modeling based on input-output data," in *Proc. Winter Simul. Conf.*, vol. 1. Orlando, FL, USA, Dec. 2000, pp. 480–486.

[103] A. Soukkou, A. Khellaf, and S. Leulmi, "Systematic design procedure of TS-type fuzzy controllers," *Int. J. Comput. Intell. Appl.*, vol. 6, no. 4, pp. 531–549, Dec. 2006.

[104] Y. Takahashi, M. Asada, and K. Hosoda, "Reasonable performance in less learning time by real robot based on incremental state space segmentation," in *Proc. Intell. Robots Syst.*, vol. 3. Osaka, Japan, Nov. 1996, pp. 1518–1524.

[105] D. T.-C. Wang, L. F. Ochoa, and G. P. Harrison, "Modified GA and data envelopment analysis for multistage distribution network expansion planning under uncertainty," *IEEE Trans. Power Syst.*, vol. 26, no. 2, pp. 897–904, May 2011.

[106] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers Psychol.*, vol. 4, p. 504, Aug. 2013.

[107] C. Lecerf, "Tackling the stability/plasticity dilemma with double loop dynamic systems," in *Proc. Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, Apr. 1999, pp. 153–158.

[108] W. L. Tung and C. Quek, "GenSoFNN: A generic self-organizing fuzzy neural network," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1075–1086, Sep. 2002.

[109] X. Du, N. Zhang, and H. Ying, "Analysis and design for a class of TS fuzzy controllers that circumvent curse of dimensionality," in *Proc. Annu. Meeting North Amer. Fuzzy Inf. Process. Soc. (NAFIPS)*, Jun. 2005, pp. 489–493.

[110] F. L. Lewis and K. Liu, "Towards a paradigm for fuzzy logic control," in *Proc. Fuzzy Inf. Process. Soc. Biannu. Conf., Ind. Fuzzy Control Intell. Syst. Conf., NASA Joint Technol. Workshop Neural Netw. Fuzzy Logic*, San Antonio, TX, USA, Dec. 1994, pp. 94–100.

Fendy Santoso received the master's degree in electrical and computer systems engineering from Monash University, Melbourne, VIC, Australia, in 2007, supported by a highly competitive Australian Development Scholarship from the Australian Government, and the Ph.D. degree in electrical engineering from the University of New South Wales (UNSW), Sydney, NSW, Australia, in 2012, under a highly competitive University International Postgraduate Awards Scholarship.

He held an appointment as a Visiting Researcher at the Graduate School of Biomedical Engineering, UNSW, in 2014. Since 2015, he has been a Post-Doctoral Researcher with the School of Engineering and Information Technology, Australian Defense Force Academy, UNSW, Canberra, ACT, Australia. His current research interests include control systems and robotics.

Dr. Santoso has been a Reviewer for several control and robotics journals such as the IEEE ROBOTICS AND AUTOMATION LETTERS and the *International Journal of System, Control and Communications* as well as numerous conferences.

Matthew A. Garratt received the Ph.D. degree in robotics from Australian National University (ANU), Canberra, ACT, Australia, in 2007.

He was an Aeronautical Engineer with the Royal Australian Navy, Canberra, before working for two years in the field of computer-aided engineering tools. In 1999, he joined ANU, where he helped build and test an autonomous helicopter using biologically inspired vision for a Defense Advanced Research Projects Agency (DARPA) funded project. Since 2001, he has been with the University of New South Wales, Canberra, where he is currently an Associate Professor. His main research areas are sensing and control for autonomous systems.

Sreenatha G. Anavatti was an Associate Professor with the Indian Institute of Technology, Mumbai, India. He moved to Australia in 1998. He is a currently a Senior Lecturer with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT, Australia. He has supervised a number of Ph.D. scholars over these years. His research interests include the application of artificial intelligence techniques such as fuzzy and neural systems for UAVs, underwater vehicles, and ground vehicles, identification and control of dynamic systems, navigation and path planning for autonomous vehicles, and evolutionary fuzzy systems and their applications to dynamic systems.