PAC: A Novel Self-Adaptive Neuro-Fuzzy Controller for Micro Aerial Vehicles

Md Meftahul Ferdaus\textsuperscript{a}, Mahardhika Pratama\textsuperscript{b}, Sreenatha G. Anavatti\textsuperscript{a}, Matthew A. Garratt\textsuperscript{a}, Edwin Lughofer\textsuperscript{c}

\textsuperscript{a}School of Engineering and Information Technology, UNSW Canberra, ACT 2612, Australia
\textsuperscript{b}School of Computer Science and Engineering, Nanyang Technological University, Singapore, 639798, Singapore
\textsuperscript{c}Department of Knowledge Based Mathematical Systems, Johannes Kepler University, Linz, Austria

Abstract

There exists an increasing demand of a flexible and computationally efficient controller for micro aerial vehicles (MAVs) due to a high degree of environmental perturbations. In this work, an evolving neuro-fuzzy controller namely Parsimonious Controller (PAC) is proposed and features less network parameters than conventional approaches due to the absence of rule premise parameters. PAC is built upon a recently developed evolving neuro-fuzzy system known as parsimonious learning machine (PALM) and adopts new rule growing and pruning modules derived from the approximation of bias and variance. These methods has no reliance on user-defined thresholds, thereby increasing its autonomy for the real-time deployment. PAC adapts the consequent parameters with the sliding mode control (SMC) theory in the single-pass fashion. The stability of our PAC is proven utilizing the Lyapunov stability analysis. Lastly, the controller’s efficacy is evaluated by observing various trajectory tracking performance from a bio-inspired flapping wing micro aerial vehicle (BI-FWMAV) and a rotary wing micro aerial vehicle called hexacopter. Furthermore, it is compared against three distinctive controller. Our PAC outperforms the linear PID controller and generalized regression neural network (GRNN) based nonlinear adaptive controller. Compared to its predecessor, G-controller, the tracking accuracy is comparable but the PAC incurs significantly less parameters to attain similar or better performance than the G-controller.

Keywords: Micro aerial vehicle, Neuro-fuzzy, Parsimonious learning machine, Self-adaptive

Email addresses: m.ferdaus@student.unsw.edu.au (Md Meftahul Ferdaus), mpratama@ntu.edu.sg (Mahardhika Pratama), s.anavatti@adfa.edu.au (Sreenatha G. Anavatti), M.Garratt@adfa.edu.au (Matthew A. Garratt), edwin.lughofer@jku.at (Edwin Lughofer)
1. Introduction

Advancements in portable electronic technology over the past few years encourage researchers to work on Micro Aerial Vehicles (MAVs). Attempts to humanize these MAVs, such as with the capacity of autonomous flying in a confined space is challenging. A positive step to such achievement is low cost of miniature electronic components like sensors, actuators, microprocessors, batteries etc. Another point is their huge applicability in both civilian and military sectors. Usage of MAVs in military sector have intensified in last decades as explored in [1]. Participation of MAVs in civilian applications has also a lot of socioeconomic benefit [2]. To transmute all these potentialities of MAVs into reality, a major concern is to pursue the preferable control autonomy.

First principle techniques (FPTs) are popular in stabilizing and controlling MAVs, but, within a certain range encircling the equilibrium. Among numerous FPTs, linear controllers like Proportional Integral Derivative (PID), Linear Quadratic (LQ) are employed successfully in different MAVs [3] due to simple structure. Nonetheless, linear controllers performance degrade abruptly to deal with environmental disruption due to MAV’s inherent non-linearity and coupled dynamics. In rejecting disturbance of nonlinear MAV dynamics, nonlinear control techniques (namely Backstepping, Sliding Mode techniques, Feedback Linearization (FBL), H∞) perform diligently. A shortfall of all these linear and nonlinear controllers is their foremost dependency on the precise plant dynamics to be controlled. In MAVs, noisy data that may obtain from various sensors and micro-electronic components. Besides, encountering environmental disruptions like wind gust in open space, actuator degradation are usual in MAVs. Integration of such unexpected facts in MAV’s dynamics is laborious or inconceivable. These deficiencies of FPT based controllers tempts research towards mathematical model-free methods.

Among numerous model-free approaches, the Fuzzy Logic systems (FLS), and Neural Networks (NN) are the most commonly used methods in various control application [4], since they can effectively deal with complex nonlinear and chaotic systems. When superiorities of both FLS and NN based controllers are joined into a single structure, they are known as Neuro-Fuzzy (NF) controller, which is also employed in diverse engineering industries [5, 6]. However, development of a FLS, NN, or NF based high performance controller is tough. Among numerous existing methods, a simple way to construct FLS, NN, or NF controller is to train them offline using the known knowledge about the plant dynamics. Furthermore, it can be executed by mimicking a FPT controller like PID through offline training. These trained NF controller possess several deficiencies. Firstly, they consist of a fixed architecture with a definite number of neurons or membership functions, rules or layers. With their static configuration, they can not adapt with time-varying non-stationary plants. Secondly, the controllers have dependency on experts knowledge or training data. Therefore, control performance relies on the accuracy of that knowledge or data. To summarize, these static FLS, NN, or NF controllers do not have any direct association with the plant models, rather there exists a indirect reliance.
To improve the performance of model-free static controllers, researchers have developed adaptive controllers, which can tune their network parameters to alleviate the adverse influence of uncertainties and disturbances. Usually, it is settled by combining conventional nonlinear control techniques such as backstepping [7], sliding mode techniques [8], feedback linearization (FBL), H∞ [9] etc. with FLS, NN, or NF structure. Such mixture provides a mode-free robust and adaptive control scheme. In these control schemes only the network parameters are altered, where they are maintaining a fixed structure. It enforces us to specify the number of nodes, layers, or rules beforehand. It is hard to know the exact number of rules a priori to attain desired control performance. Controllers with only few rules may fail to produce the desired performance, whereas too many of them may originate a over-complex structure of controllers to actualize in real time. To circumvent such shortcomings, NN or NF controller with flexible architecture can be employed. These flexible controllers are not only able to tune parameters, but also to evolve the structure by adding or deleting layers or rules in self-adaptive fashion.

Research on evolving intelligent controllers has begun in the early twenty-first century. In 2003, a self-organizing NF controller was proposed in [10], where they applied system error, $\epsilon$-completeness, and error reduction ratio in their rule evolution scheme. Their controller needed to store all preceding input-output data, which forced to compute a large matrix in each step and yields high computation costs. It makes them impractical to implement in systems like MAVs, where a fast response is expected from controller to emulate the desired commands. Another evolving controller based on evolving Takagi-Sugeno (eTS) model [11] was developed in [12]. Though their controller was evolving in nature, it suffers from several imperfections. First, in their structural evolution mechanism, fuzzy rules were added or replaced only, pruning of rules was missing. Besides, their eTS fuzzy controller needed to memorize the previous data obtained from the plant and controller itself. Such obligation leads to a computationally costly control mechanism and made them unrealistic in swift reaction-based control application. A hybrid evolving controller was developed in [13] by mixing an evolving and a static TS fuzzy system. Despite of their design simplicity, they required to know some parameters of the plant to be controlled. However, there is no guarantee of availability of those parameters during control operation. An evolving controller was also been actualized using model predictive control technique [14]. However, their dependency on plant's dynamic model restricted their application in complex nonlinear systems, where dynamic models may not be known.

Above mentioned evolving FLS, NN, or NF controller are equipped with hyper-sphere-shaped clustering techniques, where mostly univariate Gaussian membership functions are used. As a consequence, those evolving controllers are associated with several rule premise parameters like mean and width. These parameters need to be updated continuously, which rises computational complexities. Recently, hyper-ellipsoid shaped clustering method was exercised to form an evolving NF controller namely G-controller [15]. They had utilized multivariate Gaussian function due to its scale-invariant behavior and ability to
handle both axis parallel and non-axis parallel data distribution. Nevertheless, like the hyper-sphere based counterparts, hyper-ellipsoid based G-controller involved numerous network parameters. To epitomize, a bottleneck of evolving controllers is the engagement of manifold parameters, which strike adversely in furnishing fast response. To lessen parameters of evolving controller, evolving data-cloud based clustering technique was used to develop evolving TS fuzzy controller in [16]. Though they reduced network parameters, still bound to calculate accumulated distance of a particular point to all other points. Obligation of such calculation can not effectively diminish the memory demand of their evolving controller.

To mitigate the computational complexity of the above discussed evolving controllers, hyper-plane-shaped clustering techniques could be a promising avenue, since they are absolutely free from premise parameters. Lately, a hyper-plane-shaped clustering technique based evolving NF architecture namely parsimonious learning machine (PALM) was proposed in [17]. In our proposed controller, though we have implemented the PALM architecture, its rule adaptation module is replaced to originate a simplified evolving controller. The self-constructing clustering and similarity analysis based rule evolution mechanism of PALM has dependency on user-defined parameters. Those are swapped with the so-called network significance method, which is deduced from the idea of bias-variance trade-off [18]. It eliminates the necessity of user-defined parameters. In this rule evolution technique, the statistical contribution of each rule can be estimated to add or prune them on demand, which aids to achieve a desired control performance. Main features of our evolving controller can be uttered as follows:

1. **Flexible structure:** PAC is characterized by fully flexible network architecture, where rules can be augmented, pruned or updated spontaneously on demand. Therefore, it is not degraded by alteration in plant dynamics. Since it does not need any offline training, no previous information about the plant is required.

2. **Curtailed network parameters:** The fuzzy rules of PAC are portrayed by a hyper-planes which corroborates both the rule premise and consequent parts. As a result, it does not have any premise parameter. Such scheme trims the rule-based parameters to the level of $R \times (N + 1)$, where $N$ denotes the number of input dimension and $R$ is the number of fuzzy rules. The imperative network parameters of the proposed controller are considerably lower than the conventional evolving controllers discussed in the literature.

3. **Implementation of new rule growing and rule pruning module:** In PALM [17], self-constructing clustering approach [19] [20] was employed to generate rules, which faces computational complexities to calculate variance and covariance among different variables. Besides, similarity analysis among hyper-planes in terms of distance and orientation between hyper-planes were measured to merge rules. To eliminate such complex calculation of growing and pruning rules, bias-variance concept based simplified
method namely network significance is proposed in PAC. This concept is derived from the idea of network significance [18] estimating the network bias and variance. A new rule is added in the underfitting situation, while the pruning process is triggered by the overfitting case. The key difference of this paper from [18] lies in the estimation of bias and variance for the hyperplane-shaped hidden unit.

4. Elimination of user-defined thresholds: PALM’s [17] rule evolution method have a reliance on four different predefined thresholds. To eliminate such dependency on user-defined thresholds, the bias-variance concept based network significance method is exercised in PAC, where we do not need any user-defined problem-specific thresholds.

5. Simple weight adaptation mechanism: Since the proposed controller has no premise parameters, its only consequent parameter i.e. weights are adapted by exerting SMC learning theory to confirm a stable closed-loop system. To secure the uniform asymptotic convergence of tracking error to zero, a robustifying subsidiary control term is affixed in parallel with the PALM network. Conclusively, stability of the proposed PAC is ensured by exploiting the Lyapunov function. To evaluate the controller’s stable and precise tracking performance, it has been geared into a simulated BI-FWMAV and hexacopter plant.

Aforementioned features of the PAC is desiring to achieve desired control autonomy in MAVs. Therefore, the proposed controller is applied to control BI-FWMAV and hexacopter in this work. In addition, the whole code of the PAC is written in C programming language. It is compatible with majority of the MAVs hardware, where its implementation is made accessible publicly in [21].

2. Related work

Recent advent in portable microelectronic technology is encouraging to develop MAVs. Their autonomous flying ability in both natural and human-made environment is aspiring for numerous defense applications. Besides, they can play a vital role in civilian ventures like disaster alleviation, communication, environment conservation etc. Until now, majority of the commercially available MAVs are controlled by FPT based linear controller like PIDs because of their design simplicity and easy employment. In this work, we are dealing with the control performance a hexacopter and BI-FWMAV. In regulating rotary wing [22] and flapping wing MAVs [23, 24], implementations of PID controllers have witnessed profusely in last decades. Though PIDs can control MAVs with a satisfactory accuracy in certain environment, due to their fixed gain, they are not robust against real world uncertainties like wind gust. Another prominent linear control method for MAVs are Linear Quadratic (LQ) technique [25, 26]. These linear controllers suffer severely from attaining desired performance due to the complex nonlinearity in system dynamics. Besides, various un-modeled uncertainties and perturbations have detrimental influence on these precise-model-based linear controllers. Afterwards, to control MAVs researchers had focused
on various nonlinear controllers namely back-stepping [27], sliding mode control [28], feedback linearization (FBL) [29], $\mathcal{H}_\infty$ robust control [30] techniques by virtue of their disturbance rejection. However, like their linear counterparts, nonlinear controllers have direct association with precise dynamic model to be controlled. Such dependency limits their application where the plant dynamics are unknown. In our work, the complete dynamics of the BI-FWMAV is unidentified, where exercising these model-based nonlinear controller is impractical. Besides, BI-FWMAV’s light weight architecture is naive to manifolds un-modeled outdoor uncertainties. Model free control approaches are effective solution to deal with un-modeled uncertainties.

Model-free control approaches for MAVs have commenced with standalone FLS [31, 32], NN [33, 34], or NF based controllers [35]. However, the requirement of offline training, static structure, dependency on experts knowledge confine their applicability in complex systems with uncertain environment, and experienced spillover effect [36]. Having promising quality to handle complex dynamic systems with uncertainties, the evolving intelligent controllers have not yet been employed profusely in regulating MAVs. One of the main reasons is their complex framework and its challenging execution. Apart from that, MAVs require a control method with prompt response, which sometimes difficult to acquire from evolving controllers due to their high memory demand. Lately, an evolving controller using interval type-2 (IT2) NF structure was proposed in [37] to control a quadcopter MAV. Nonetheless, their IT2NF system was just functioning as a uncertainty and perturbation observer, and a PD controller was used to control the attitude and position of the quadcopter. In [38], an evolving NF controller was proposed for a simulated quadcopter MAV plant. Though their controller successfully generated and pruned fuzzy rules on the fly with a satisfied tracking accuracy, they only had considered a simplified dynamic model of the quadcopter to be controlled. One of the latest research on evolving controller in regulating MAVs attitude and altitude was reported in [15]. Unlike the evolving controllers discussed in the literature, they effectually exercised a NF evolving controller in a high dimension nonlinear complex MAV plant. Besides, their controller was evaluated by incorporating uncertain wind gust effect in their MAVs plant.

In evolving controllers, adaptation of consequent parameters play a crucial role to attain desired accuracy. To adapt the rule consequent parameters in evolving controllers, gradient-based methods are typically used [39]. However, the desired performance accuracy of gradient-based controllers were observed only when they were used to control plant with a slow variation in dynamics. Furthermore, gradient-based method like dynamic back-propagation comprises of partial derivatives. Such algorithms can not guarantee fast convergence speed, particularly in complex non-convex search space. In addition, there are chances to be trapped in local minimum [40]. Alternately, evolutionary algorithms were attempted in [41] to tune parameters. Nevertheless, the stability of their proposed controller was not confirmed and fast response was not ensured. Such constraints can be handled simply by imposing SMC theory to adapt the consequent parameters as witnessed in [42]. Inspired by the simplicity of the SMC
theory [15], in our work it has been employed to update the consequent parameters. Like [15], predefined static sliding parameters are replaced with learning rate based self-organizing parameters. It makes the proposed PAC a complete self-adaptive model-free controller.

Arrangement of the remaining sections of this paper is as follows: In Section 3, limitations of existing evolving controllers are analyzed. Section 4 details the network structures of the PALM based evolving controller PAC along with the explanation on rule generation and pruning mechanism. Predicaments in formulating FW AMV or hexacopter model are asserted in Section 5. Experimental results and performance evaluation of the proposed controller are described in Section 6. At last, the paper terminates with concluding reMAVks in Section 7.

3. Problem statement

In a closed-loop control system, data comes in a sequential online manner to the controller, which may contain various uncertainties and disturbances. Such an irregularly distributed sequence of incoming data to the controller can be exposed formally as $e = \{e^1, e^{1+t}, e^{1+2t}, ..., e^f\}$, denoted by "Data stream of $e$" in Fig. 1, where $t$ is the time step (in sec) of the closed-loop control system, $f$ is the final time until which plants need to be controlled. Here, $e$ is indicating the data stream of error, which is the difference between the reference trajectory data stream $Y_r$ and observed corresponding output data stream from the plant $Y$.

The remaining incoming data streams to the closed-loop system as displayed in Fig. 1 are data stream of derivative of error ($\dot{e}$), the integral of error ($\int e$). To tackle irregular sequence of incoming data, the controller should hold some desiring features such as: 1) able to work in single pass mode; 2) deal with various uncertainties of the incoming data; 3) perform with low memory burden and computational complexity to enable real-time deployment under resource
constrained environment. In realm of NF system, such learning proficiency is manifested by evolving NF systems [43]. However, enormous free parameters associated with the evolving NF controller causes complex computation. From the perspective of controlling MAV, a swift response from the controller is much expected, where the tuning of several premise parameters in evolving controller is a hindrance to fulfill such expectation. To decipher the involvement of profuse premise parameters in evolving controller, their premise parameter-dependent hyper-spherical or hyper-ellipsoidal clustering techniques require to substitute with premise parameter-free clustering method. From this research gap, a hyper-plane based clustering method [17] is utilized in our work, which is composed from the consequent parameter rather than any premise parameters as explained in [17]. Another inadequacy of state of the art evolving controllers is their affiliation to user-defined parameters to evolve their structure. Those parameters claims to alter with respect to the corresponding closed-loop system. Our proposed evolving controller is free from such parameter, and they have been superseded with bias-variance concept. To get a clearer view, the flow of data streams in our closed-loop system is displayed in Fig. 1. Unlike the PALM [17], in the fuzzification layer of PAC, we always have incoming target/reference data stream \( Y_r \) as shown in the fuzzification block of the Fig. 1. Therefore, we have not used the recurrent PALM (rPALM) architecture like [17], since it does not have any access to the target value in testing phase. To obtain an explicit impression, the self-evolving formation of our proposed PAC is enumerated in the next section.

4. Structure of PALM based self-adaptive controller

Our proposed PAC is a three-layered NF system and functioning in tandem. It’s evolving architecture is rooted with TS fuzzy model, where classical hyper-spherical [12], hyper-ellipsoidal [44], or data-cloud based [45] clusters are substituted with hyper-plane-based clusters. Utilization of hyper-planes have removed antecedent parameters as explained in [17], which reduces the number of operative parameters in our controller dramatically. Popular hyper-plane-based clustering techniques procedures like fuzzy C-regression model (FCRM) [46], fuzzy C-quadratic shell (FCQS) [47], double FCM [48], inter type-2 fuzzy c-regression model (IT2-FCRM) [49] are non-incremental in nature, thereupon can not entertain evolving hyper-planes. Additionally, they deployed hyper-spherical function for instance Gaussian function to accommodate hyper-planes. To mitigate such inadequacies, a new membership function [49] is used in PAC. To fetch a vivid overview on the mechanism of PALM itself, the architecture of the PALM network is detailed in the following subsection.

4.1. Architecture of PALM

In our developed evolving closed-loop control system, the PALM network is fed by three inputs namely error (\( e \)), derivative of error (\( \dot{e} \)) and actual plants output (\( y \)). Referring to the theory of fuzzy system, these crisp data (\( e, \dot{e}, y \))
needs to be transformed into fuzzy set, which is the initial step in the PALM’s work flow. This fuzzification process is accomplished by adopting hyper-plane-shaped membership function, which is framed through the concept of point-to-plane distance. The employment hyper-plane-shaped membership function can be expressed as follows:

\[ f_{T_1}^j = \mu_B(j) = \exp\left(-\eta \frac{d(j)}{\max\{d(j)\}}\right) \]  

where \( \eta \) is an regulating parameter which adjusts the fuzziness of membership grades. Based on the observation in [17], and empirical analysis with different MAV plant in our work, the range of \( \eta \) is fixed as [1, 100]. This membership function empowers the utilization of hyper-plane-based clusters directly into the PALM network without any rule parameters except the first order linear function or hyperplane. Because a point to plane distance is not unique, the compatibility measure is executed using the minimum point to plane distance. \( d(j) \) in Eq. (1) denotes the distance between the current data point and \( j \)th hyperplane as with Eq. (2). It is determined by following the definition of a point-to-plane distance [50] and is formally expressed as follows [17]:

\[ d(j) = \frac{|y_r - (\sum_{i=1}^{n} a_{ij}x_i + b_{0j})|}{\sqrt{1 + \sum_{i=1}^{n} (a_{ij})^2}} \]  

where \( a_{ij} \) and \( b_{0j} \) are consequent parameters of the \( j \)th rule, \( i = 1, 2, ..., n; n \) is the number of input dimension, and \( y_r \) is the reference trajectory to the plant. In closed-loop control perspective, the target value \( y_r \) for the plant is always known. However, in regression or classification problem, it is not possible to acquire the target value in the testing phase. Therefore, in [17] the target-value dependent PALM was advanced to a recurrent structure namely rPALM. Such recurrent architecture is not required in our PALM based controller since the target trajectory \( y_r \) is continually available up to the end of control performance. It is noteworthy to state that a type-1 fuzzy structure is facilitated in PALM. In light of a MISO system, the IF-THEN rule of PALM can be exposed as follows:

\[ R^j: \text{IF } X_n \text{ is close to } f_{T_1}^j, \text{ THEN } y_j = x_e^j \omega_j \]  

where \( x_e \) is the extended input vector and is expressed by inserting the intercept to the original input vector as \( x_e = [1, e, \dot{e}, y] \); \( e \) is the error i.e. the difference between the reference and actual output of the plant, \( \dot{e} \) is the error derivative i.e. the difference between the present and previous state error value, \( y_r \) is the reference for the plant to be controlled, \( \omega_j \) is the weight vector for the \( j \)th rule, \( y_j \) is the consequent part of the \( j \)th rule. The antecedent part of PALM is simply hyperplane and does not consists of any premise parameters. From Eq. (3), it is observed that the drawback of PALM lies in the high level fuzzy inference scheme which lowers the transparency of fuzzy rule. The intercept of extended input vector dominates the slope of hyperplane which eliminates the untypical gradient dilemma.
In PALM, an analogous consequent part alike the basic T-S fuzzy model’s rule consequent part \( (y_j = b_{0j} + a_{1j}x_1 + ... + a_{nj}x_n) \) is employed. The consequent part for the \( j \)th hyperplane is calculated by weighting the extended input variable \((x_e)\) with its corresponding weight vector as follows:

\[
f^{T}_{j1} = x_e^T \omega_j \tag{4}
\]

The weight vector in Eq. (4) is updated recursively by the SMC theory based adaptation laws, which ensures a smooth alteration in the weight value. In the next step, the rule firing strength is normalized and added with the rule consequent to produce the end-output of PALM. The final defuzzified crisp output of the PALM can be expressed as follows:

\[
u_{PALM} = \frac{\sum_{j=1}^{R} f^1_{j1} f^2_{j1}}{\sum_{i=1}^{R} f^1_{i1}} \tag{5}
\]

The normalization term in Eq. (5) assures the partition of unity where the sum of normalized membership degree is one. The above described PALM network of PAC has similar structure like [17]. Nonetheless, a simplified learning mechanism is used in PAC than the original PALM as explained in the next subsection.

4.2. **Automatic constructive mechanism of PAC**

In PALM [17], the self-constructive clustering technique was adopted to grow the rule. The rule significance was determined by measuring input and output coherence, where the coherence was calculated by investigating the correlation between the existing data samples and the target concept. Again, the computation of correlation has dependency on finding variance and covariance among different variables. In addition, the PALM’s rule growing method was regulated by two predefined threshold. On the other hand, PALM’s rules were merged by measuring the similarity between the hyperplane-shaped fuzzy rules. The similarity among rules were measured by observing the angle and minimum distance among them. The merging strategy was also controlled by predefined thresholds. This clearly shows high computational cost of PALMs rule evolving mechanism and made them incompatible in fast response based control applications. To subjugate such complexity, a simplified rule evolution technique is implemented in PAC bottomed by the network significance method, which is formulated from the concept of bias-variance [18]. Here, no predefined threshold are required to control the generation or pruning of its fuzzy rules. This network significance method based rule growing and deletion modules of PAC are clarified in subsequent paragraphs of this section.

4.2.1. **New Mechanism of rule generation in PALM**

The strength of PAC can be analyzed by witnessing the tracking error which can be written in terms of mean square error (MSE) as follows:

\[
e_{MSE} = \frac{1}{T} \sum_{t=1}^{T} \left(y_r(t) - y(t)\right)^2 \tag{6}
\]
where \( e_{MSE} \) is denoting the mean square error, \( y_r(t) \) is expressing the desired trajectory and \( y(t) \) is attained output from the plant to be controlled. Such formulation experiences two obstructions in learning mechanism of an evolving controller, such as: 1) it needs to memorize all data points to get a clearer view about the controller’s constructive mechanism; 2) though recursive calculation of \( e_{MSE} \) excluding preceding data is possible, it does not investigate the strength of reconstruction for uncertain upcoming data. In simple words, it does not consider the generalization capacity of evolving controllers. To mitigate such hindrance, let us consider that \( Y_0 \) is the observed plant’s output for the reference input \( Y_r \), and \( E[Y] \) is the plant’s expected output. According to the definition of expectation \([21]\), if \( x \) is continuous, then the expectation of \( f(x) \) can be formulated as: \( E[f(x)] = \int_{-\infty}^{\infty} f(x) p(x) dx \), where \( p(x) \) is the probability density function of \( x \). Then the network significance (NS) method can be defined as follows:

\[
NS = \int_{-\infty}^{\infty} (Y_r - Y)^2 p(t) dt
\]  

In effect, the NS method is the expectation of squared tracking error, and verbalized as follows:

\[
NS = E[(Y_r - Y)^2] = E[(Y - E[Y] + E[Y] - Y_r)^2]
\]  

After several mathematical operation, NS can be formulated with the bias and variance concept as follows:

\[
NS = E[(Y - E[Y])^2] + (E[Y] - Y_r)^2 = \text{Var}(Y) + \text{Bias}(Y)^2
\]  

In Eq. 9, the variance of \( Y \) i.e. \( \text{Var}(Y) \) can be presented as \( \text{Var}(Y) = E[(Y - E[Y])^2] = \int_{-\infty}^{\infty} (Y - E[Y])^2 p(t) dt = E[Y^2] - E[Y]^2 \). In establishing the NS in PAC, a rule firing strength of unity is assumed. It simplifies the network outcome as \( Y = \sum_{i=1}^{n} x_e \omega_i \) because the expectation of the normalized firing strength of PAC does not have unique solution. Here \( x_e \) is the extended input and can be exposed as \( x_e = [1, x_1, x_2, ..., x_n] \), where \( n \) is the number of inputs to the network; \( \omega_i \) is the weight vector of the \( i \)th rule. Now the expectation of \( Y \) can be formulated as \( E[Y] = \int_{-\infty}^{\infty} x_e \omega p(x) dx \). Suppose that the normal distribution applies, it leads the expectation to the following expression.

\[
E[Y] = \omega_i \mu_e
\]  

where \( \mu_e \) is mean of the extended input \( x_e \). Factually, the integration of \( x_e \) over \(-\infty \) to \( \infty \) originates the mean \( \mu_e \).

Let us call back the formula of variance \( \text{Var}(Y) = E[Y^2] - E[Y]^2 \), where the second term of the right hand side is simply \( E[Y]^2 = E[y] \times E[y] \), and the first term can be formulated as \( E[Y^2] = \omega_i \mu_e^2 \). Incorporation of all these results in NS formula of Eq. 9 settle the final manifestation of NS. Since NS consists of both bias and variance, a high value of NS may indicate a high variance (over-complex network with profuse fuzzy rules) or a high bias (over-simplified network) problem. Such phenomenon can not be elucidated simply
by system error index. Augmentation of a new rule is inferred to subjugate the high bias dilemma. Nonetheless, such phenomenon is not convenient for high variance context since the addition of rules magnify the network complexity. To retain a compact network structure with a satisfactory tracking performance, the concept of bias-variance trade-off is inserted in our work to calculate the NS. Such regulation of the rules of our controller has no reliance to user-defined parameters \[52, 53\]. By confirming the fundamental objective of rule growing procedure to ease the high bias dilemma, the condition of growing rules in our work is expressed as follows:

\[\mu_k^{ba} + \sigma_k^{ba} \geq \mu_{ba}^{min} + \Gamma \sigma_{ba}^{min}\]  (11)

where \(\mu_{ba}^{k}\) is denoting mean of bias and \(\sigma_{ba}^{k}\) is standard deviation of bias at the \(k\)th observation whilst \(\mu_{ba}^{min}\) and \(\sigma_{ba}^{min}\) are pointing the minimum of mean and standard deviation up to \(k\)th time instant. In computing these variables, no preceding data are required. Their values are being updated directly based on the availability of upcoming signals to the PALM. When Eq. (11) is satisfied, the values of \(\mu_{ba}^{min}\) and \(\sigma_{ba}^{min}\) are to be reset. To perceive an improved tracking performance from the commencing of PAC’s control operation, a rapid decay in the bias value is expected. It is retained in formulating the settings of bias in Eq. (11) as long as the plant does not encounter any uncertainties or disturbances. The presence of any disruptions in control system will elevate value of bias, which cannot be addressed directly by adapting the consequent parameter of the PAC. To elucidate such hindrance, Eq. (11) is originated from the adaptive sigma rule, where \(\Gamma\) controls the degree of confidence of the sigma rule. In our work, the \(\Gamma\) is expressed as \(\Gamma = 1.3 \exp(-\text{bias}^2) + 0.7\), which revolves \(\Gamma\) between 1 and 2. Consequently it obtains the level of confidence from around 68% to 96%. Such scheme enhances the flexibility in rule-growing module to adapt with the environmental perturbations. It also eliminates the dependency of evolving controller on user-defined problem dependent parameters. To sum up, a high bias usually signifies an oversimplified network, which is solved by adding rules. However, it is avoided in case of low bias since it may magnify the variance.

4.2.2. New rule pruning module of PALM

The structure of PALM may become highly complex as a result of high variance. On that ground, control of variance is essential to reduce the network complexity by pruning the fuzzy rules. Since a high variance indicates the overfitting condition, the rule pruning scheme initiates from the evaluation of variance. Like the rule growing mechanism of PAC, a statistical process control technique is embraced in rule pruning module to trace the high variance dilemma as follows:

\[\mu_k^{var} + \sigma_k^{var} \geq \mu_{var}^{min} + 2\pi \sigma_{var}^{min}\]  (12)

where \(\mu_{var}^{k}\) is denoting mean and \(\sigma_{var}^{k}\) is standard deviation of variance at the \(k\)th observation whilst \(\mu_{var}^{min}\) and \(\sigma_{var}^{min}\) are pointing the minimum of mean and standard deviation up to \(k\)th time instant. Here, the term \(\pi\) is adopted as \(\pi = 1.3 \exp(-\text{var}) + 0.7\), \(\pi\) is a dynamic constant and regulating the degree of
confidence in the sigma rule. The term 2 is Eq. (12) halts the direct pruning after growing. Furthermore, \( \mu_{\text{var}}^{\text{min}} \) and \( \sigma_{\text{var}}^{\text{min}} \) are reset the condition in Eq. (12) is fulfilled.

After the execution of Eq. (12), the significance of each rule is examined via the idea of network significance, and inconsequential rules are pruned to reduce the overfitting condition. The significance of rules are tested via the concept of network significance, adapted to evaluate each rules statistical contribution. The significance of \( i \)th rule is determined as its average activation degree for all possible incoming data samples or its expected values as expressed in Eq. (9).

Considering normal distribution assumption, the importance of \( i \)th rule can be expressed as \( HS_i = \omega_i \mu_r \). A small value of \( HS_i \) indicates that the \( i \)th rule plays an small role to recover the clean input attributes. Therefore, it can be pruned with a very insignificant loss of tracking accuracy. Since the contribution of the \( i \)th rule is calculated in terms of the expectation \( E(Y) \), the least contributing rule having the lowest \( HS \) is regarded inactive. When the overfitting condition occurs or Eq. (12) is satisfied, the rule with lowest \( HS \) is pruned and can be expressed as follows:

\[
\text{Pruning} \rightarrow \min_{i=1,...,R} HS_i
\]

The condition in Eq. (13) targets to mitigate the overfitting situation by deleting the least significant rule. It also indicates that the desired trajectory tracking performance can still be achieved with the rest \( R-1 \) rules. Furthermore, this rule pruning strategy enhance the generalization power of PAC by reducing its variance, which helps to deal with variety of disturbances.

### 4.3. Adaptation of weights in PAC

Unlike the conventional evolving controller, PAC does not possess any premise parameter, consequently free from the computation of tuning those parameters. Weight is its particular consequent parameter need to be adjusted to realize desired control efficacy. Inspired by the smooth employment and regulations of SMC theory in various neuro-fuzzy systems [54, 55, 56], in our work SMC learning theory is functioned to adapt the PAC’s weight, which ascertains stability in the closed-loop control system. Besides, it confirms adequate robustness in a system against exterior obstructions, parameter variations and uncertainties [15]. In designing SMC, a time-varying sliding surface that restricts motion of a system to a plane can be exposed as follows:

\[
S_{ss}(u_{PALM}, u) = u_{src}(t) = u_{PALM}(t) + u(t)
\]

In this work, the sliding surface for BI-FWMAV and hexacopter plant to be controlled can be expressed as:

\[
s_l = e + \gamma_1 \dot{e} + \gamma_2 \int_0^t e(t) dt
\]
where, $\gamma_1 = \frac{\alpha_2}{\alpha_1}$, $\gamma_2 = \frac{\alpha_3}{\alpha_1}$, $\epsilon$ is the error i.e. the divergence from the trajectory obtained from the plant to the reference one. Here, the sliding parameters are initiated with tiny values such as $\alpha_1 = 1 \times 10^{-2}$, $\alpha_2 = 1 \times 10^{-3}$, $\alpha_3 \approx 0$. These parameters are further evolved by using dissimilar learning rates. Proper assignment of these rates support to secure the desired parameters with minimal time, which affirms to gain stability in the closed-loop system swiftly. Engagement of these self-organizing sliding parameters shapes a fully model free controller. Similar definition as of G-controller [15] is maintained in this work as follows:

Definition: After a specific time $t_k$, a sliding motion will be formed on the sliding manifold $S_{ss}(u_{PALM}, u) = u_{src}(t) = 0$, where the state $S_{ss}(t)\dot{S}_{ss}(t) = u_{src}(t)\dot{u}_{src}(t) < 0$ to be convinced for the entire time period with some non-trivial semi-open sub-interval of time expressed as $[t, t_k) \subset (0, t_k)$.

To enforce the above-mentioned definition of sliding mode condition, weights of the proposed PAC are adapted accordingly. This adaptation process of the proposed controller is summarized below.

Theorem 1. The adaptation laws for the consequent parameter i.e. weights of the PAC are elected as:

$$\dot{\omega}(t) = -\alpha_1 F(t)\psi(t)s(t), \text{ where } \omega(0) = \omega_0 \in \mathbb{R}^{nR \times 1}$$ (16)

where the term $F(t)$ is updating recursively as follows:

$$\dot{F}(t) = -F(t)\psi(t)\psi^T(t)F(t), \text{ where } F(0) = F_0 \in \mathbb{R}^{nR \times nR}$$ (17)
where \( n \) is the number of inputs to the PALM, and \( R \) is the number of induced rules. These weight adaptation laws assure a stable closed-loop control system, where the plants to be controlled are of different order.

**Proof:** In our proposed controller, the reliance of the subsidiary robustifying control term on the sliding surface can be formulated as follows:

\[
u_{src}(t) = \alpha_1 s_l(18)\]

This subsidiary robustifying control term \( u_{src} \) may endure high-frequency oscillations in contributing to the control input. Such repulsive occurrence in sliding mode control theory is termed as chattering effect. To suppress this chattering effect, control systems are primarily facilitated with saturation or sigmoid functions. In this work, due to simplicity a saturation function is used to alleviate those detrimental consequence.

The final outcome from PALM \( (u_{PALM}) \) in PAC can be expressed as follows:

\[
u_{PALM}(t) = \psi^T(t)\omega(t) (19)\]

The overall control signal as observed in Fig. 2 can be declared as follows:

\[
u(t) = u_{src}(t) - u_{PALM}(t) (20)\]

The cost function can be determined as follows:

\[
J(t) = \int_0^t s_l^2(\tau)d\tau \\
= \frac{1}{\alpha_1^2} \int_0^t (u(\tau) + u_{PALM}(\tau))^2d\tau \\
= \frac{1}{\alpha_1^2} \int_0^t (u(\tau) + \psi^T(t)\omega(\tau))^2d\tau (21)
\]

The gradient of \( J \) with respect to \( \omega \) is settled as follows:

\[
\nabla_\omega J(t) = 0 \\
\Rightarrow \int \psi(\tau)u(\tau)d\tau + \omega(t) \int_0^t \psi(\tau)\psi^T(\tau)d\tau = 0 \\
\Rightarrow \omega(t) = \left[ \int_0^t \psi(\tau)\psi^T(\tau)d\tau \right]^{-1} \int_0^t \psi(\tau)u(\tau)d\tau \\
\Rightarrow \omega(t) = -F(t) \int_0^t \psi(\tau)u(\tau)d\tau \quad (22) \\
\Rightarrow F^{-1}(t) \omega(t) = -\int_0^t \psi(\tau)u(\tau)d\tau \quad (23)
\]

where,

\[
F(t) = \left[ \int_0^t \psi(\tau)\psi^T(\tau)d\tau \right]^{-1}
\]
\[ F^{-1}(t) = \int_{0}^{t} \psi(\tau)\psi^{T}(\tau)d\tau \] (24)

The derivative of Eq. (24) is as follows:

\[ F^{-1}(t)\dot{F}(t)F^{-1}(t) = -\psi(t)\psi^{T}(t) \]

\[ \dot{F}(t) = -F(t)\psi(t)\psi^{T}(t)F(t) \] (25)

From Eq. (25), it is noticed that \( \dot{F}(t) \) is a negative definite and \( F(t) \) is decreasing over time, therefore \( F(t) \in l_{\infty} \). Now executing the time derivative of Eq. (22) and utilizing Eq. (18), (19), (20), and (23) the following is retrieved:

\[ \dot{\omega}(t) = \dot{F}(t)F^{-1}(t)\omega(t) - F(t)\psi(t)u(t) \]

\[ = -F(t)\psi(t)\psi^{T}(t)\omega(t) - F(t)\psi(t)u(t) \]

\[ = -F(t)\psi(t)\left(\psi^{T}(t)\omega(t) + u(t)\right) \]

\[ = -\alpha_{1}F(t)\psi(t)s_{l}(t) \] (26)

4.3.1. Stability Analysis of the PAC

Definition: PALM performs as an universal function approximator. Therefore, an existence of \( \omega^{*} \) without any loss of generality is assumed such that:

\[ u(t) = \psi^{T}\omega^{*}(t) + \varepsilon_{f}^{*}(z) \] (27)

where \( \varepsilon_{f}^{*}(z) = [\varepsilon_{f1}^{*}, \varepsilon_{f1}^{*}, ... , \varepsilon_{f1}^{*}]^{T} \in \mathbb{R}^{k} \) is the minimal functional approximator error. In this work, \( \dot{\omega}(t) = \omega(t) - \omega^{*} \) is defined. In addition, \( s_{l}(t) = \psi^{T}\tilde{\omega}(t) \) is assumed too.

Lemma 2.

\[ \frac{d(F^{-1}(t)\tilde{\omega}(t))}{dt} = -F^{-1}(t)\dot{F}(t)F^{-1}(t)\tilde{\omega}(t) + F^{-1}(t)\dot{\tilde{\omega}}(t) \]

\[ = \psi(t)\psi^{T}(t)\tilde{\omega}(t) - \psi(t)s_{l}(t) \]

\[ = \psi(t)s_{l}(t) - \psi(t)s_{l}(t) \]

\[ = 0 \] (28)

This is indicating that \( F^{-1}(t)\tilde{\omega}(t) \) is not altering with respect to time, and therefore \( F^{-1}(t)\tilde{\omega}(t) = F^{-1}(0)\omega(0) \), \( \forall t > 0 \).

\[ \lim_{t \to \infty} \tilde{\omega}(t) = \lim_{t \to \infty} F(t)F^{-1}(0)\omega(0) \] (29)

Since \( F(t) \) is decreasing and \( \tilde{\omega}(t) \in l_{\infty}, \omega(t) \in l_{\infty} \). In this work the following Lyapunov function is considered:

\[ V(t) = \frac{1}{2}\tilde{\omega}(t)F^{-1}(t)\tilde{\omega}(t) \] (30)
The time derivative of this Lyapunov function is expressed as follows:

\[
\dot{V}(t) = \frac{1}{2} \tilde{\omega}^T(t) F^{-1} \dot{\tilde{\omega}}(t) + \frac{1}{2} \tilde{\omega}^T(t) \dot{F}^{-1} \tilde{\omega}(t)
\]

\[
= -\tilde{\omega}^T(t) \psi(t) s_I(t) - \frac{1}{2} \tilde{\omega}^T(t) \psi(t) \psi^T(t) \dot{\tilde{\omega}}(t)
\]

\[
= -s_I^2(t) - \frac{1}{2} s_I^2(t)
\]

\[
= -\frac{3}{2} s_I^2(t) \leq 0
\]  

From Eq. (30) and Eq. (31), it is witnessed that \( V(t) > 0 \), and \( \dot{V}(t) \leq 0 \). Furthermore, Eq. (31) depicts that \( \dot{V}(t) = 0 \), if and only if \( e(t) = 0 \). It is signifying the assurance of the system’s global stability by the Lyapunov theorem. By exploiting Barbalat’s lemma \[57\], \( e(t) \to 0 \) as \( t \to \infty \) is also observed here. It is guaranteeing the asymptotic stability of the system. Thus, a convergence of the plant’s tracking error to zero is confirmed.

5. Plant dynamics of MAVs

The hexacopter plant, developed at UAV laboratory of UNSW Canberra and a BI-FWMAV plant developed by Defence Science and Technology Group (DSTG) are engaged in our work to evaluate the proposed PAC’s performance. In this section, we are initiating with hexacopter’s nonlinear complex plant dynamics.

5.1. Dynamics of hexacopter plant

Rigid body dynamics of our hexacopter plant is formulated using Newton’s second law of motion, where we have calculated correlations between the forces and moments acting on the hexacopter and translational and rotational accelerations. We have assumed our hexacopter plant as a traditional mass distribution, where the plane of symmetry was xz plane. Such consideration makes the cross product of moments of inertia in yz and xy plane zeros. After this simplified implementation, the equations of forces \( (F_x, F_y, F_z) \) and moments \( (L, M, N) \) in X, Y and Z axes are exposed in Eq. (32). For further clarifications, readers can go through \[58\], where equations in (32) are derived elaborately.

\[
\begin{align*}
F_x & = m(\dot{u} + qw - rv) \\
F_y & = m(\dot{v} + ru - pw) \\
F_z & = m(\dot{w} + pv - qu) \\
L & = I_x \ddot{\phi} - I_x \dot{z}^2 + qr(I_y - I_y) - I_{xx} pq \\
M & = I_y \ddot{\theta} + rp(I_x - I_z) + I_{xz}(p^2 - r^2) \\
N & = -I_{xz} \ddot{\phi} + I_x \dot{z} \dot{r} + pq(I_y - I_x) + I_{xz} qr
\end{align*}
\]  

(32)
\begin{equation}
\begin{align*}
I_x &= \int \int \int (y^2 + z^2) \, dm \\
I_y &= \int \int \int (x^2 + z^2) \, dm \\
I_z &= \int \int \int (x^2 + y^2) \, dm \\
I_{xy} &= \int \int xy \, dm \\
I_{xz} &= \int \int xz \, dm \\
I_{yz} &= \int \int yz \, dm
\end{align*}
\end{equation}

where \( m \) is the body mass in kg; \( I_x, I_y, I_z \) are hexacopter’s mass moments of inertia with regard to x, y, and z-axis respectively in kgm\(^{-2}\); \( I_{xz} \) is the product of inertia. In our simulated plant, the practiced values of the above parameters are as: \( m = 3 \, kg \), \( I_x = 0.04 \, kgm^{-2} \), \( I_y = 0.04 \, kgm^{-2} \), \( I_z = 0.06 \, kgm^{-2} \), \( I_{xz} = 0 \, kgm^{-2} \).

5.2. Complexities in hexacopter’s over-actuated plant dynamics

Unlike the conventional hexacopter plant with 6 degrees of freedom (DOF) rigid body dynamics, 8 DOF over-actuated hexacopter plant with medium fidelity is considered in this work. Two surplus DOF are induced from two moving masses. The masses can slide along their own rail aligned in lengthwise and sideways consecutively. To get a synopsis about the over-actuated simulated Hexacopter plant, its high-level diagram is shown in Fig. 3. The roll and pitch command to the "control mixing" block of Fig. 3 is driven by "attitude controller". In attitude control mechanism, the inner loop is controlled by a linear PID controller and outer loop is regulated by PAC. The thrust command of "control mixing" is provided by PAC. Moving of mass to the longitudinal direction shifts the center of gravity (CoG) to X-axis, which is denoted by \( CG_X \) in Fig. 3 and \( CG_Y \) is expressing the shift of CoG to Y-axis due to the movement of mass to the lateral direction. Both the movements \( CG_X \) and \( CG_Y \) are controlled by PAC. In control mixing block, a simple linear mixing composition is utilized to convert the roll, pitch, yaw, and thrust commands to the required speed of motors. These signals are used to calculate the required thrust and torque of individual rotor based on the relative airflow faced by each of them and the commanded motor speed. Afterwards, the total vertical force and yawing torque is of the plant are calculated by summing up the thrust and torque of individual rotor. The product of thrust to a single rotor and moment arm yields the rolling and pitching moments acting on the hexacopter. Finally, the controlled thrust along with the yawing torque, and rolling and pitching moments are fed to the rigid body dynamics to update the body state accordingly. Hexacopter’s detailed rigid body dynamics, nonlinear aerodynamics along with associated complexities are discoursed in the supplementary document.

A user friendly graphical mask for the rigid body dynamics block allows users to alter the mass, moments of inertia and initial states on demand. During experimentation, errors are obtained by measuring the difference from actual to reference altitude and attitude. These errors, their derivatives, and reference
Dynamics of BI-FWMAV plant

Dynamics of the BI-FWMAV plant is highly nonlinear and expresses higher complexity than the hexacopter. It is mainly due to its light weight and smaller size. Our four-winged BI-FWMAV plant was developed by the DSTG \[59\]. The top-level diagram of the simulated BI-FWMAV plant is shown in Fig. 4.
Four wings of the BI-FWMAV is operated by four actuators as exhibited in Fig. 4. From the analysis on dragonfly flight in [60, 61], the influence of seven different flapping parameters on the wings and actuators as well, are considered in developing the BI-FWMAV plant. These parameters are namely stroke plane angle \((F_{p_{spa}})\) (in rad), flapping frequency \((F_{p_{ff}})\) (in Hz), flapping amplitude \((F_{p_{fa}})\) (in rad), mean angle of attack \((F_{p_{aoa}})\) (in rad), amplitude of pitching oscillation \((F_{p_{po}})\) (in rad), phase difference between the pitching and plunging motion \((F_{p_{pd}})\), time step \((F_{p_{ts}})\) (in sec). Exploring a variety of combinations of these parameters, the BI-FWMAV can carry out take-off, rolling, pitching, and yawing as explained in [15]. Since our proposed controller is used to regulate the altitude of the BI-FWMAV, it is essential to know the dominant parameter in determining the altitude. After a successful parametric analysis, the flapping amplitude has turned out to be the dominant one in altitude tracking. Individual forces and moments of actuators are combined to provide the demanded force and moment to the rigid body dynamics based on the relative airflow acting on each wing and the commanded actuator speed. The combined force utilized in our work can be formulated as follows:

\[
F_T = F_{a_1} + F_{a_2} + F_{a_3} + F_{a_4} + (mg \times DCM) \tag{34}
\]

where, \(m\) is the mass, \(g\) is the acceleration due to gravity, \(DCM\) is the direction cosine matrix, \(F_{a_i}\) (where \(i = 1, 2, 3, 4\)) is the force provided by individual actuator. Similarly, the total moment necessary for the rigid body can be demonstrated as follows:

\[
M_T = M_{a_1} + M_{a_2} + M_{a_3} + M_{a_4} \tag{35}
\]

where, \(M_{a_i}\) (where \(i = 1, 2, 3, 4\)) is presenting the individual momentum of each wing and can be articulated as

\[
M_{a_i} = F_{a_i} \times (CG - CP_i) \tag{36}
\]

where \(i = 1, 2, 3, 4\); \(CG = [0 \ 0 \ 0]\); and \(CP_1 = [0.08 \ 0.05 \ 0]\); \(CP_2 = [0.08 \ 0.05 \ 0]\); \(CP_3 = [0.08 - 0.05 \ 0]\); \(CP_4 = [-0.08 - 0.05 \ 0]\); and \(\times\) is presenting \((3 \times 3)\) cross product. Finally, the accumulated force and moment are transformed into the body coordinate system, and all the required body states like three dimensional angular displacements \((\phi, \theta, \psi)\), angular velocities \((\omega_{bx}, \omega_{by}, \omega_{bz})\) and accelerations \((\alpha_{bx} = \frac{d\omega_{bx}}{dt}, \alpha_{by} = \frac{d\omega_{by}}{dt}, \alpha_{bz} = \frac{d\omega_{bz}}{dt})\) and linear displacements \((X_b, Y_b, Z_b)\), linear velocities \((v_{bx}, v_{by}, v_{bz})\) and accelerations \((a_{bx} = \frac{dv_{bx}}{dt}, a_{by} = \frac{dv_{by}}{dt}, a_{bz} = \frac{dv_{bz}}{dt})\) are acquired, and the BI-FWMAV states are updated. Complex nonlinear wing dynamics and aerodynamics module of the BI-FWMAV has been clarified in Section 2 of the supplementary document.

From the above discussion, and description of the supplementary document, it is obvious that the BI-FWMAV plant associates profuse parameters with high nonlinearity, though we have omitted some complexity in revealing precise wing kinematics. Deriving a precise mathematical model of such highly nonlinear, complex, and over-actuated plant is exceptionally laborious, where inclusion of
uncertainties and uncharted disruption is more difficult or unfeasible in some cases. These perspectives necessitate a controller that performs precisely with a minimum or no knowledge about the system. Being model-free and self-evolving, our developed PAC is a suitable candidate. More importantly, the impediment of conventional evolving controllers i.e. involvement of numerous free parameters is resolved here since our controller do not have any premise parameters and only depends on one consequent parameter namely weights of the network. With such simplistic evolving structure, our prosed controller provides comparable and satisfactory tracking performance.

6. Numerical experiments

In our work, the proposed evolving PAC is used to regulate an over-actuated hexacopter and BI-FWMAV plant, where numerous altitude and attitude trajectories are tracked for both the MAVs. To be specific, PAC is appraised in tracking altitude of six different trajectories for BI-FWMAV. On the other hand, in hexacopter the performance of PAC is witnessed both for tracking variant altitude and attitude, which are detailed in upcoming subsections.

6.1. Experimental results from BI-FWMAV

Our proposed evolving PAC was inspected for numerous tracking signals and their consequent outcomes were contrasted with a recently developed evolving controller called G-controller [19], a General Regression Neural Network (GRNN) based nonlinear adaptive controller, and a linear controller PID. Our developed PAC’s code was written in C programming language and made openly accessible in [21]. The performance of all these controllers were observed in BI-FWMAV plant for a duration of 100 seconds, where the characteristics of six separate altitude trajectories were as: 1) an unaltered height of 10 meters, can be exposed as \( Y_r(t) = 10 \text{ m} \); 2) variable heights with sharp edges, where the heights were altering from 3 m to 6 m, and to 9 m from there, and vice versa afterwards. The duration of hovering at a particular height was 20 seconds; 3) variable heights with smooth edges; 4) sum of sines function, which was an amalgamation of a sine wave with a frequency of 0.3 rad sec\(^{-1}\), amplitude of 4 m, bias of 6 m, and a cosine wave having a frequency of 0.5 rad sec\(^{-1}\), amplitude of 3 m and bias of 3 m; 5) a periodic square wave pulse, where the amplitude was varying between 1 m to 11 m, and its frequency is 0.2 rad sec\(^{-1}\); 6) a stair-case function, where each steps had a duration of 20 seconds. The individual heights of first three steps are same i.e. 3 m, which is 2 m in the last step. In these numerical experiments, GRNN based nonlinear adaptive controller operates better than its linear counterpart PID. Again, both the evolving controllers manifested better tracking performance than the GRNN controller. To acquire a deeper understanding of these manifestations, some of their desired features
Table 1: Measured features of various controllers in operating the DI flying robot (RT: rise time, ST: settling time, CH: constant height, VH: variable height, SS: sum of sine, RMSE: root mean square error, ms: millisecond, m: meter, MA: maximum amplitude, PSW: periodic square wave)

| Desired trajectory | Measured features | Control method | PID | GRNN | G-control | PAC |
|--------------------|-------------------|----------------|-----|------|-----------|-----|
| CH (MA 10 m)       | RMSE              | 0.6460         | 0.5880 | 0.6631 | 0.6656    |
|                    | RT (ms)           | 50.772         | 39.2681 | 41.208 | 44.792     |
|                    | ST (ms)           | 560.98         | 188.55  | 127.15 | 84.701     |
|                    | Peak (m)          | 12.246         | 12.785  | 10.813 | 10.163     |
| VH with sharp change (MA 9 m) | RMSE | 0.3303         | 0.3443  | 0.3324 | 0.3503     |
|                    | RT (ms)           | 23.931         | 50.811  | 50.892 | 38.586     |
|                    | ST (ms)           | 8176.4         | 8167.3  | 8133.2 | 8166.1     |
|                    | Peak (m)          | 9.3732         | 9.0083  | 9.0061 | 9.2265     |
| VH with smooth change (MA 13 m) | RMSE | 0.0895         | 0.0366  | 0.0228 | 0.0373     |
|                    | RT (ms)           | 8.8573         | 5.1052  | 4.187  | **0.7204** |
|                    | ST (ms)           | 9984.3         | 9872.2  | 9871.9 | **9870.5** |
|                    | Peak (m)          | 13.007         | 13.019  | 13.019 | **13.005** |
| SS function (MA 11 m) | RMSE | 0.4730         | 0.4595  | 0.4963 | 0.5008     |
|                    | RT (ms)           | 21.109         | 23.847  | 18.674 | 18.848     |
|                    | ST (ms)           | 9960.5         | 9960.4  | 9960.1 | 9960.2     |
|                    | Peak (m)          | 11.468         | 11.457  | 11.431 | **11.430** |
| PSW function (MA 11 m) | RMSE | 1.6701         | 1.6035  | 1.5042 | 1.5083     |
|                    | RT (ms)           | **16.905**     | 49.052  | 60.416 | 55.291     |
|                    | ST (ms)           | 9813.3         | 9637.8  | 9522.6 | 9588.7     |
|                    | Peak (m)          | 13.584         | 12.548  | 11.244 | 11.332     |
| Staircase function (MA 12 m) | RMSE | 0.3073         | 0.3594  | 0.2885 | 0.3031     |
|                    | RT (ms)           | 5996.0         | 5994.8  | 5998.4 | 6000.3     |
|                    | ST (ms)           | 8384.2         | 8066.8  | 8056.6 | **8056.1** |
|                    | Peak (m)          | 12.453         | 12.228  | 12.007 | 12.189     |
Figure 5: Performance observation of different controllers in tracking altitude of BI-FWMAV, when the trajectories are (a) constant hovering, (b) variable heights with sharp edges, (e) periodic square wave function, (f) staircase function, rule evolution corresponding to (c) constant hovering, (d) variable heights with sharp edges.
like root mean squared error (RMSE), rising time (RT) in milliseconds, settling
time (ST) in milliseconds, and peak value of the overshoot are captured and
tabulated in Table 1. All these anticipations are pictured in Fig. 5 and detailed
in the next paragraph.

In Fig. 5 (a), controllers were facilitated to track a 10 m height trajectory,
where both from PID and GRNN controller higher overshoot with peak val-
ues more than 12 m were attested. Better performance with a peak overshoot
around 11 m was noticed from the self-adaptive controllers. With regards to
peak overshoot, the lowest values were exhibited by evolving controllers in all
six different scenario of Fig. 5, which is evidently signifying the superiority
of their evolving structure. Unlike the benchmarked evolving G-controller, our
proposed one does not consist of any premise parameters. A network with fewer
parameters supports PAC to procure prompter settlement, which was witnessed
from the lowest settling time of 84.7 seconds in Fig. 5 (a). It was considerably
faster than the benchmarked evolving G-controller since it demands 127.15
seconds to settle. In most cases of Fig. 5, lowest or very comparative set-
tling time was observed from our proposed controller. The tracking accuracy
of the PAC in terms of RMSE and rising time was not always the least one.
Nonetheless, their achievements were still comparable and sometimes surpassed
benchmarked controllers.

6.2. Experimental results from hexacopter plant

All the controllers were assessed to track both the altitude and attitude
(in terms of rolling and pitching) of the over-actuated hexacopter plant. Four
separate trajectories of altitude were as follows: 1) a constant peak with a value
of 4 m; 2) altering heights with sharp edges, where the peak was of 9 m; 3)
a periodic square wave function; and 4) a staircase function with a peak of 12
m. In all conditions, a higher overshoot was perceived from the linear PID
controller at each sharp changes as depicted in Fig. 6. Peak of this overshoot
was lessen while the linear controller was replaced with the nonlinear adaptive
GRNN controller. Yet GRNN’s performance was not consistent in all cases.
Especially, in dealing with the square wave trajectory, performance deteriorates
significantly as portrayed in Fig 6 (c). This issue was managed by the evolving
controllers effectively owing to self-adaptive architecture. Quick settlements
were also testified from evolving controllers as recorded in Table 2.

Furthermore, the rolling and pitching position (in rad) was observed with a
sum of sine trajectory, which was a fusion of a sine wave with a frequency of 0.3
radsec$^{-1}$, amplitude of 0.3 m, and a cosine wave possessing a frequency of 0.5
radsec$^{-1}$, amplitude of 0.5 m. The amplitude of the cosine wave was substituted
with 0.4 m in rolling mode, where a far precise tracking was witnessed from the
PAC than PID. Interestingly, the adaptive GRNN controller failed to track the
Table 2: Measured features of various controllers in regulating the hexacopter (RT: rise time, ST: settling time, CH: constant height, VH: variable height, ms: millisecond, m: meter, MA: maximum amplitude, PSW: periodic square wave, rad: radian)

| Desired trajectory         | Measured features | Control method | PID  | GRNN | G-control | PAC  |
|----------------------------|-------------------|----------------|------|------|-----------|------|
| CH (MA 4 m)                | RMSE              | 0.3513         | 0.3348 | 0.4201 | 0.4198    |      |
|                            | RT (ms)           | 91.605         | 240.14 | 146.63 | 143.04    |      |
|                            | ST (ms)           | 553.28         | 352.09 | 240.14 | 286.69    |      |
|                            | Peak (m)          | 5.1417         | 4.1473 | 4.0027 | 4.1111    |      |
| VH with sharp change (MA 9 m) | RMSE              | 0.6556         | 0.7568 | 0.6513 | 0.6487    |      |
|                            | RT (ms)           | 91.979         | 143.89 | 122.60 | 118.93    |      |
|                            | ST (ms)           | 8612.9         | 8464.1 | 8256.8 | 8245.7    |      |
|                            | Peak (m)          | 9.9313         | 9.0533 | 9.0007 | 9.0113    |      |
| PSW function (MA 11 m)     | RMSE              | 2.4173         | 3.8872 | 3.3908 | 3.4174    |      |
|                            | RT (ms)           | 101.33         | 1603.1 | 96.061 | 80.451    |      |
|                            | ST (ms)           | 9887.5         | 10050  | 9947.2 | 9951.1    |      |
|                            | Peak (m)          | 12.987         | 12.593 | 11.295 | 11.453    |      |
| Staircase function (MA 12 m) | RMSE              | 0.5490         | 0.4793 | 0.6074 | 0.6005    |      |
|                            | RT (ms)           | 4117.8         | 6015.8 | 6005.6 | 6005.6    |      |
|                            | ST (ms)           | 8378.9         | 8221.1 | 8181.4 | 8175.4    |      |
|                            | Peak (m)          | 12.737         | 12.034 | 12.0007| 12.0006   |      |
| Pitching                   | RMSE              | 0.0451         | 0.0486 | 0.0466 | 0.0109    |      |
|                            | RT (ms)           | 14.907         | 14.686 | 65.469 | 10.135    |      |
|                            | ST (ms)           | 10057          | 10057  | 9982.9 | 10053     |      |
|                            | Peak (rad)        | 0.5615         | 0.5758 | 0.5398 | 0.5469    |      |
| Rolling                    | RMSE              | 0.1673         | N/A   | 0.0290 | 0.0259    |      |
|                            | RT (ms)           | 166.116        | N/A   | 118.75 | 91.596    |      |
|                            | ST (ms)           | 10037          | N/A   | 9978.9 | 9979.9    |      |
|                            | Peak (rad)        | 0.3907         | N/A   | 0.4822 | 0.4852    |      |
Figure 6: Performance observation of different controllers in tracking altitude of hexacopter, when the trajectories are (a) constant hovering, (b) variable heights with sharp edges, (d) staircase function, and (c) evolution of rules corresponding to constant hovering rolling trajectory, which is the reason of their absence in Fig. 7 (a). Insertion of PAC yielded better tracking of pitching position, which is obvious from the lowest RMSE of 0.01. The captured RMSE for G-controller was much higher, nearly 0.04 as recorded in Table 2.

To sum up, superior or comparative tracking of trajectories were witnessed in our proposed evolving controller. Additionally, faster responses were obtained than the benchmarked evolving G-controller, testifying the benefits of having evolving structure with minimal network parameters.

6.3. Robustness against uncertainties and noise
In this work, variety of disturbances were inserted in both BI-FWMAV and hexacopter’s plant to verify controllers robustness against those disturbances. For instance, in the plant dynamics of BI-FWMAV, a sudden noise with a peak of 7 m and duration of 0.1 second, and an artificial wind gust with a maximum velocity of 4 $m/s^{-1}$ immediately after 2 seconds was embedded in the plant. Effects of both wind gust and sudden peak noise was observed for all six different altitude trajectories of BI-FWMAV, which are depicted in Fig. 8. From a closer view, an obvious performance degradation in dealing with disturbances was witnessed from the non-adaptive PID controller in all cases. In
Figure 7: Performance observation of different controllers in tracking desired (a) rolling, (b) pitching of the hexacopter MAV; (c) evolution of rules in tracking rolling, and (d) pitching in hexacopter.
GRNN controller, due to the adaptation of the network parameters, it performed better than the PID. However, it suffered severely in tracking trajectories with sharp changes because of the absence of structure adaptation mechanism. On the other hand, sharper settlement and recovery from the adverse effect of gust were sighted from evolving controllers in our experiments. At the same time, in rejecting sudden peak noise, proposed PAC dominated the G-controller and other benchmarked controllers as well since the lowest peak was viewed from the PAC. Such accomplishments were possible due to their evolving structure with adaptation of fewer parameters than the benchmarked evolving controller.

In the hexacopter dynamics, a sharp peak noise with an amplitude of 7 m and period of 0.1 second was implanted to observer robustness of the controllers. Effects of disturbance was witnessed for four different altitude trajectories of hexacopter. RMSE, settling time, rise time and peak overshoot values for all those trajectories were tabulated for all benchmarked and proposed controller in Table 4. Such perturbation was handled effectively by evolving controllers than their static counterparts as attested in Fig. 9. A high peak and slow settlement was detected in PID and GRNN controller. On the contrary, a negligible overshoot with a rapid settlements were inspected from the evolving controllers. For example, after closely observing the constant altitude trajectory in Fig. 9 (a), recorded values of settling time were less than 300 ms from the evolving controllers, which was more than 340 ms in GRNN and nearly 1000 ms in PID controller. Similar phenomenon was observed in remaining trajectories, which is evidently declaring the improved robustness against uncertainties of the PAC in contrast with the benchmarked static controller.

6.4. Self-adaptive mechanism of PAC

Based on the bias variance concept explained in section 4, rules of the PAC have been evolved dynamically in different experiments. Before analyzing the evolution of structure of our proposed controller, we have tried to summarize shortfalls of the benchmarked controllers used in this work. The linear PID controller’s realization is based upon three gain parameters namely proportional, integral and differential gain. They are typically denoted as $K_p$, $K_i$, and $K_D$. It requires to set values for those parameters in offline before utilizing in control operation, which may oblige repetitious efforts. Besides, those parameters can not be tuned online. Before performing the control operation, the adaptive GRNN controller requires offline training encouraged by the PID controller’s input-output datasets. Though, it can adapt its network parameter during operation, it has a fixed structure with a hidden layer consists of ten fixed
Figure 8: Performance observation of different controllers in tracking altitude of BI-FWMAV by considering sudden noise and wind gust uncertainty, when the trajectories are (a) constant hovering, (b) variable heights with sharp edges, (c) variables height with smooth edges, (d) sum of sine function, (e) periodic square wave function, and (f) staircase function.
Table 3: Measured features of various controllers in operating the DI flying robot by considering sudden peak noise and wind gust perturbation (RT: rise time, ST: settling time, CH: constant height, VH: variable height, SS: sum of sine, ms: millisecond, m: meter, MA: maximum amplitude, PSW: periodic square wave)

| Desired trajectory | Measured features | Control method | PID | GRNN | G-control | PAC |
|--------------------|------------------|----------------|-----|------|------------|-----|
| CH (MA 10 m)       | RMSE             | 0.6539         | **0.5292** | 0.6656 | 0.6671     |
|                    | RT (ms)          | 50.772         | 48.592 | **41.208** | 44.792     |
|                    | ST (ms)          | 1034.2         | 699.36 | 634.72 | **633.06** |
|                    | Peak (m)         | 12.247         | **10.879** | 11.016 | 11.004     |
| VH with sharp change (MA 9 m) | RMSE             | 0.3438         | 0.4892 | **0.3350** | 0.3607     |
|                    | RT (ms)          | **23.931**     | 47.670 | 50.892 | 38.586     |
|                    | ST (ms)          | 8176.2         | 8167.4 | **8133.2** | 8166.1     |
|                    | Peak (m)         | 9.3111         | 12.203 | **9.0122** | 9.2265     |
| VH with smooth change (MA 13 m) | RMSE             | 0.1265         | 0.1005 | **0.0595** | 0.1026     |
|                    | RT (ms)          | 8.8573         | 3.5530 | 4.188 | **0.7204** |
|                    | ST (ms)          | 9884.3         | 9872.2 | **9870.5** | 9871.9     |
|                    | Peak (m)         | 13.007         | 13.009 | 13.019 | **13.006** |
| SS function (MA 11 m) | RMSE             | 0.4730         | **0.4690** | 0.4995 | 0.5041     |
|                    | RT (ms)          | 21.109         | 23.847 | **18.675** | 18.848     |
|                    | ST (ms)          | 9960.1         | 9960.4 | **9959.8** | **9959.8** |
|                    | Peak (m)         | **11.468**     | 11.586 | 11.495 | 11.501     |
| PSW function (MA 11 m) | RMSE             | 1.6714         | 1.5965 | **1.5052** | 1.7941     |
|                    | RT (ms)          | **16.905**     | 49.739 | 60.416 | 55.265     |
|                    | ST (ms)          | 9813.3         | 9637.0 | **9522.8** | 9613.7     |
|                    | Peak (m)         | 13.584         | 13.512 | **11.080** | 11.332     |
| Staircase function (MA 12 m) | RMSE             | 0.3213         | 0.3600 | 0.2914 | 0.3074     |
|                    | RT (ms)          | 5996.3         | 6000.1 | **5999.2** | 6000.3     |
|                    | ST (ms)          | 8384.3         | 8076.4 | 8056.5 | **8056.0** |
|                    | Peak (m)         | 12.453         | 12.275 | **12.007** | 12.189     |
Figure 9: Performance observation of different controllers in tracking altitude of hexacopter considering sudden noise, when the trajectories are (a) constant hovering, (b) variable heights with sharp edges, (c) periodic square wave function, and (d) staircase function.

Table 4: Measured features of various controllers in regulating the hexacopter by considering sudden peak noise (RT: rise time, ST: settling time, CH: constant height, VH: variable height, ms: millisecond, m: meter, MA: maximum amplitude, PSW: periodic square wave)

| Desired trajectory                      | Measured features | Control method |
|-----------------------------------------|-------------------|----------------|
| CH (MA 4 m)                             | RMSE 0.3532       | PID 0.3329     | GRNN 0.4185 | G-Controller 0.4187 |
|                                         | RT (ms) 87.605    | 113.28         | 140.66     | 138.04 |
|                                         | ST (ms) 976.45    | 341.09         | 236.27     | 280.69 |
|                                         | Peak (m) 5.141    | 4.1473         | 4.0020     | 4.1111 |
| VH with sharp change (MA 9 m)           | RMSE 0.5831       | PID 0.6957     | GRNN 0.6557 | G-Controller 0.6506 |
|                                         | RT (ms) 87.979    | 135.89         | 122.60     | 118.93 |
|                                         | ST (ms) 8588.9    | 8471.1         | 8299.8     | 8288.7 |
|                                         | Peak (m) 9.9313   | 9.0533         | 9.0007     | 9.0113 |
| PSW function (MA 11 m)                  | RMSE 2.4301       | PID 3.8653     | GRNN 3.3907 | G-Controller 3.4174 |
|                                         | RT (ms) 101.33    | 1610.7         | 1571.0     | 80.451 |
|                                         | ST (ms) 9981.0    | 10093          | 9932.5     | 9965.1 |
|                                         | Peak (m) 12.988   | 12.5934        | 11.295     | 11.453 |
| Staircase function (MA 12 m)            | RMSE 0.6306       | PID 0.106      | GRNN 0.5775 | G-Controller 0.6074 |
|                                         | RT (ms) 4156.3    | 6052.7         | 6005.6     | 6043.6 |
|                                         | ST (ms) 8411.0    | 8264.7         | 8181.4     | 8219.4 |
|                                         | Peak (m) 12.714   | 12.0343        | 12.0007    | 12.0006 |
To attain more robustness against disturbances, the benchmarked G-controller can adapt both its network and parameters. Nonetheless, obviously it needs to deal with lots of free parameters in both antecedent and consequent parts. Besides, the evolution of G-controller is regulated by some user-defined thresholds. Sometimes, it is causing a large settling time. On the contrary, our proposed evolving controller has no premise parameters. The only parameter that needs to be adapted is weight, which is adapted here using SMC theory.

The structure evolution in terms of added or pruned rules for some trajectories of BI-FWMAV and hexacopter were disclosed graphically in Fig. 5 (c) and (d), Fig. 6 (c), and in Fig. 7 to get a vivid insight about the evolution of rules in PAC. For further clarification, the fuzzy rule extracted by PAC in controlling the BI-FWMAV can be uttered as follows:

$$R^1: \text{IF } X_n \text{ is close to } [1, e, \dot{e}, y_r] \times [0.0121, 0.0909, 0.4291, 0.6632]^T, \quad (37)$$

with

\[ e, \dot{e}, y_r \]

where \( e \) is the error i.e. the difference between the reference and actual output of the plant, \( \dot{e} \) is the error derivative i.e. the difference between the present and previous state error value, \( y_r \) is the reference for the plant to be controlled. Since PAC is targeted to minimize the tracking error to zero or very close to zero, it needs information about the error as an input to the closed-loop system. It is also witnessed in PAC’s rule as exposed in (37). When PAC was controlling the BI-FWMAV in tracking a constant altitude of 10 m, it generated 3 rules within 1 second at the beginning of control operation. Since the reference is unaltered and stability of the plant is achieved, PAC does not add or prune any extra rule later on as witnessed from the Fig. 5 (c). While BI-FWMAV was following a variable height trajectory with sharp changes at edges, the PAC starts operating by producing only one rule. After 8 seconds it adds two more rules and achieved system stability. After that the changes in trajectories are handled by PAC only through tuning of weights only. It does not need any further structure evolution as observed in Fig. 5 (d). Successful evolution of rules by confirming system stability was also achieved by PAC in controlling the hexacopter plant. For instance, PAC supports hexacopter to track a constant altitude of 4 m with two rules as displayed in Fig. 6 (c). While PAC was regulating the rolling of hexacopter for a sum of sine trajectory, it started operating with only one rule. Immediately after 17 seconds, it added another two rules, however one of them was pruned at 19 seconds to minimize the overfitting phenomenon, while maintained system stability with two rules only. With these two rules, it tracked the trajectory efficiently through the SMC based weight adaptation. Similar scenario was witnessed in controlling pitching position of the hexacopter since the same trajectory was employed. In sum up, by adapting both the structure ans weights, the PAC was controlling the MAVs effectively to follow the desired trajectories very closely and by preserving the stability of the closed-loop system.
7. Conclusion

Based on the research gap in controlling MAVs in cluttered environments, a completely model-free evolving controller namely PAC is proposed in this work. A bottleneck of the existing evolving controllers is the utilization of numerous free parameters and their tuning. Such inadequacy has been mitigated in our PAC since it has no premise parameters. The only parameter used in our evolving controller to acquire desired tracking is the weight. For instance, in our experiment the benchmarked evolving G-controller with three rules requires 48 network parameters, whereas PAC needs only 12 parameters with three rules. Apart from that, conventional evolving controllers adhere to user-defined problem-based thresholds to shape their structure. In PAC, rather than predefined parameters, the bias-variance concept based network significance method is utilized to determine it’s structure. The PAC has been verified by implementing them in various MAV plants namely BI-FWMAV and hexacopter to track diverse trajectories. Achievements are contrasted with a commonly utilized linear controller PID, an adaptive nonlinear GRNN controller, en evolving controller namely G-controller. Furthermore, controller’s robustness against uncertainties and disruptions is ascertained by injecting a wind gust and sudden peak to the MAVs dynamics. In controlling both plants with uncertainties, lower or comparable overshoot and settling time were observed from PAC with a simplified evolving structure, which is testifying its robustness against uncertainties and compatibility in regulating MAVs. Inspired by PAC’s efficient performance in controlling the simulated BI-FWMAV and hexacopter plant, it will be employed in MAVs hardware in future.

Acknowledgment

The authors would like to thank the Australian Defense Science and Technology Group for providing the simulated BI-FWMAV plant, unmanned aerial vehicle laboratory of the University of New South Wales Canberra for supporting with the hexacopter plants, and the computational support from the Computational Intelligence Laboratory of Nanyang Technological University (NTU) Singapore. This work was financially supported by NTU start-up grant and MOE tier-1 grant.

References

References

[1] S. Azeem, Autonomous unmanned aerial vehicles, The George Washington University, Tech. Rep.

[2] D. Floreano, R. J. Wood, Science, technology and the future of small autonomous drones, Nature 521 (7553) (2015) 460.
[3] S. Bouabdallah, Design and control of quadrotors with application to autonomous flying, Tech. rep., Epfl (2007).

[4] M. M. Ferdaus, S. G. Anavatti, M. Pratama, M. A. Garratt, Towards the use of fuzzy logic systems in rotary wing unmanned aerial vehicle: a review, Artificial Intelligence Review (2018) 1–34.

[5] D. Zhang, Z. Zhou, X. Jia, Networked fuzzy output feedback control for discrete-time takagi–sugeno fuzzy systems with sensor saturation and measurement noise, Information Sciences.

[6] C.-J. Lin, A ga-based neural fuzzy system for temperature control, Fuzzy Sets and Systems 143 (2) (2004) 311–333.

[7] S. Tong, Y. Li, P. Shi, Observer-based adaptive fuzzy backstepping output feedback control of uncertain mimo pure-feedback nonlinear systems, IEEE Transactions on Fuzzy Systems 20 (4) (2012) 771–785.

[8] W.-J. Chang, H.-Y. Qiao, C.-C. Ku, Sliding mode fuzzy control for nonlinear stochastic systems subject to pole assignment and variance constraint, Information Sciences 432 (2018) 133–145.

[9] J. Cheng, X.-H. Chang, J. H. Park, H. Li, H. Wang, Fuzzy-model-based hâŁđ control for discrete-time switched systems with quantized feedback and unreliable links, Information Sciences 436 (2018) 181–196.

[10] Y. Gao, M. J. Er, Online adaptive fuzzy neural identification and control of a class of mimo nonlinear systems, IEEE Transactions on Fuzzy Systems 11 (4) (2003) 462–477.

[11] P. P. Angelov, D. P. Filev, An approach to online identification of takagi-sugeno fuzzy models, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 (1) (2004) 484–498.

[12] P. Angelov, A fuzzy controller with evolving structure, Information Sciences 161 (1-2) (2004) 21–35.

[13] J.-C. de Barros, A. L. Dexter, Evolving fuzzy model-based adaptive control, in: Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International, IEEE, 2007, pp. 1–5.

[14] H.-G. Han, X.-L. Wu, J.-F. Qiao, Real-time model predictive control using a self-organizing neural network, IEEE transactions on neural networks and learning systems 24 (9) (2013) 1425–1436.

[15] M. Ferdaus, M. Pratama, S. G. Anavatti, M. A. Garratt, Y. Pan, Generic evolving self-organizing neuro-fuzzy control of bio-inspired unmanned aerial vehicles, arXiv preprint arXiv:1802.00635.
[16] I. Škrjanc, S. Blazic, P. Angelov, Robust evolving cloud-based pid control adjusted by gradient learning method, in: 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 2014, pp. 1–8.

[17] M. Pratama, M. Ferdaus, S. G. Anavatti, M. A. Garratt, Palm: An incremental construction of hyperplanes for data stream regression, arXiv preprint arXiv:1805.04258.

[18] M. Pratama, Incremental learning of denoising autoencoder for evolving data streams, submitted.

[19] R.-F. Xu, S.-J. Lee, Dimensionality reduction by feature clustering for regression problems, Information Sciences 299 (2015) 42–57.

[20] J.-Y. Jiang, R.-J. Liou, S.-J. Lee, A fuzzy self-constructing feature clustering algorithm for text classification, IEEE transactions on knowledge and data engineering 23 (3) (2011) 335–349.

[21] M. M. Ferdaus, M. Pratama, Evolving controller pac (2018). URL https://www.researchgate.net/publication/326416125-PAC

[22] M. M. Ferdaus, S. G. Anavatti, M. A. Garratt, M. Pratama, Fuzzy clustering based nonlinear system identification and controller development of pixhawk based quadcopter, in: Advanced Computational Intelligence (ICACI), 2017 Ninth International Conference on, IEEE, 2017, pp. 223–230.

[23] M. S. Couceiro, N. Ferreira, J. Machado, Modeling and control of a dragonfly-like robot, Journal of Control Science and Engineering 2010 (2010) 5.

[24] A. Alaimo, V. Artale, C. L. R. Milazzo, A. Ricciardello, Pid controller applied to hexacopter flight, Journal of Intelligent & Robotic Systems 73 (1-4) (2014) 261–270.

[25] L. D. Minh, C. Ha, Modeling and control of quadrotor mav using vision-based measurement, in: Strategic Technology (IFOST), 2010 International Forum on, IEEE, 2010, pp. 70–75.

[26] S. Bouabdallah, A. Noth, R. Siegwart, Pid vs lq control techniques applied to an indoor micro quadrotor, in: Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS), IEEE, 2004, pp. 2451–2456.

[27] C. A. Arellano-Muro, L. F. Luque-Vega, B. Castillo-Toledo, A. G. Loukianov, Backstepping control with sliding mode estimation for a hexacopter, in: Electrical Engineering, Computing Science and Automatic Control (CCE), 2013 10th International Conference on, IEEE, 2013, pp. 31–36.
[28] S. Busarakum, V. Srichatrapimuk, The design of sliding mode control of a hexarotor, in: Systems, Process and Control (ICSPC), 2014 IEEE Conference on, IEEE, 2014, pp. 47–52.

[29] S. Sieberling, Q. Chu, J. Mulder, Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction, Journal of guidance, control, and dynamics 33 (6) (2010) 1732–1742.

[30] G. V. Raffo, M. G. Ortega, F. R. Rubio, Backstepping/nonlinear hâĹđ control for path tracking of a quadrotor unmanned aerial vehicle, in: American Control Conference, 2008, IEEE, 2008, pp. 3356–3361.

[31] L. Doitsidis, K. P. Valavanis, N. C. T sourveloudis, M. Kontitsis, A framework for fuzzy logic based uav navigation and control, in: Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on, Vol. 4, IEEE, 2004, pp. 4041–4046.

[32] J. Bacik, D. Perdukova, P. Fedor, Design of fuzzy controller for hexacopter position control, in: Artificial Intelligence Perspectives and Applications, Springer, 2015, pp. 193–202.

[33] B. S. Kim, A. J. Calise, Nonlinear flight control using neural networks, Journal of Guidance, Control, and Dynamics 20 (1) (1997) 26–33.

[34] V. Artale, M. Collotta, G. Pau, A. Ricciardello, Hexacopter trajectory control using a neural network, in: AIP Conference Proceedings, Vol. 1558, AIP, 2013, pp. 1216–1219.

[35] S. Kurnaz, O. Cetin, O. Kaynak, Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles, Expert Systems with Applications 37 (2) (2010) 1229–1234.

[36] H. Leipholz, On the spillover effect in the control of continuous elastic systems, Mechanics research communications 11 (3) (1984) 217–226.

[37] X. Chen, D. Li, Z. Xu, Y. Bai, Robust control of quadrotor mav using self-organizing interval type-ii fuzzy neural networks (soit-iifns) controller, International Journal of Intelligent Computing and Cybernetics 4 (3) (2011) 397–412.

[38] C. Dong, N. Wang, M. J. Er, Self-organizing adaptive robust fuzzy neural attitude tracking control of a quadrotor, in: Control Conference (CCC), 2016 35th Chinese, IEEE, 2016, pp. 10724–10729.

[39] C.-M. Lin, T.-Y. Chen, Self-organizing cmae control for a class of mimo uncertain nonlinear systems, IEEE Transactions on Neural Networks 20 (9) (2009) 1377–1384.

[40] T. Poston, C.-N. Lee, Y. Choie, Y. Kwon, Local minima and back propagation, in: Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on, Vol. 2, IEEE, 1991, pp. 173–176.
[41] A. V. Topalov, K.-C. Kim, J.-H. Kim, B.-K. Lee, Fast genetic on-line learning algorithm for neural network and its application to temperature control, in: Evolutionary Computation, 1996., Proceedings of IEEE International Conference on, IEEE, 1996, pp. 649–654.

[42] E. Kayacan, E. Kayacan, H. RAMON, W. Saeys, Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm, IEEE transactions on cybernetics 43 (1) (2013) 170–179.

[43] C.-F. Juang, C.-T. Lin, An online self-constructing neural fuzzy inference network and its applications, IEEE Transactions on Fuzzy Systems 6 (1) (1998) 12–32.

[44] M. Pratama, S. G. Anavatti, P. P. Angelov, E. Lughhofer, Panfis: A novel incremental learning machine, IEEE Transactions on Neural Networks and Learning Systems 25 (1) (2014) 55–68.

[45] M. Pratama, P. P. Angelov, E. Lughhofer, M. J. Er, Parsimonious random vector functional link network for data streams, Information Sciences 430 (2018) 519–537.

[46] R. J. Hathaway, J. C. Bezdek, Switching regression models and fuzzy clustering, IEEE Transactions on Fuzzy Systems 1 (3) (1993) 195–204.

[47] R. Krishnapuram, H. Frigui, O. Nasraoui, Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. i, IEEE Transactions on Fuzzy Systems 3 (1) (1995) 29–43.

[48] E. Kim, M. Park, S. Ji, M. Park, A new approach to fuzzy modeling, IEEE Transactions on Fuzzy Systems 5 (3) (1997) 328–337.

[49] W. Zou, C. Li, N. Zhang, A T-S Fuzzy Model Identification Approach based on a Modified Inter Type-2 FRM Algorithm, IEEE Transactions on Fuzzy Systems.

[50] wolfram, Point-plane distance (2018). URL https://mathworld.wolfram.com/Point-PlaneDistance.html

[51] K. Imai, Expectation and functions of random variables (2006). URL https://imai.fas.harvard.edu/teaching/files/Expectation.pdf

[52] J. Gama, R. Fernandes, R. Rocha, Decision trees for mining data streams, Intelligent Data Analysis 10 (1) (2006) 23–45.

[53] J. Gama, I. Žliobaité, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM computing surveys (CSUR) 46 (4) (2014) 44.
[54] E. Kayacan, O. Cigdem, O. Kaynak, Sliding mode control approach for online learning as applied to type-2 fuzzy neural networks and its experimental evaluation, IEEE transactions on industrial electronics 59 (9) (2012) 3510–3520.

[55] V. I. Utkin, Sliding modes in control and optimization, Springer Science & Business Media, 2013.

[56] E. Kayacan, R. Maslim, Type-2 fuzzy logic trajectory tracking control of quadrotor vtol aircraft with elliptic membership functions, IEEE/ASME Transactions on Mechatronics 22 (1) (2017) 339–348.

[57] J.-J. E. Slotine, W. Li, et al., Applied nonlinear control, Vol. 199, Prentice hall Englewood Cliffs, NJ, 1991.

[58] R. C. Nelson, Flight stability and automatic control, Vol. 2, WCB/McGraw Hill New York, 1998.

[59] J. Kok, J. Chahl, A low-cost simulation platform for flapping wing mavs, in: SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring, International Society for Optics and Photonics, 2015, pp. 94290L–94290L.

[60] A. L. Thomas, G. K. Taylor, R. B. Srygley, R. L. Nudds, R. J. Bomphrey, Dragonfly flight: free-flight and tethered flow visualizations reveal a diverse array of unsteady lift-generating mechanisms, controlled primarily via angle of attack, Journal of Experimental Biology 207 (24) (2004) 4299–4323.

[61] J. Kok, J. Chahl, Systems-level analysis of resonant mechanisms for flapping-wing flyers, Journal of Aircraft 51 (6) (2014) 1833–1841.