Online identification of a rotary wing Unmanned Aerial Vehicle from data streams

Author:
Ferdaus, MM; Pratama, M; Anavatti, SG; Garratt, MA

Publication details:
Applied Soft Computing Journal
v. 76
pp. 313 - 325
1568-4946 (ISSN); 1872-9681 (ISSN)

Publication Date:
2019-03-01

Publisher DOI:
https://doi.org/10.1016/j.asoc.2018.12.013

Online Identification of a Rotary Wing Unmanned Aerial Vehicle from Data Streams

Md Meftahul Ferdaus[a], Mahardhika Pratama[b], Sreenatha G. Anavatti[a], Matthew A. Garratt[a]

[a] *School of Engineering and Information Technology, UNSW Canberra, ACT 2612, Australia*

[b] *School of Computer Science and Engineering, Nanyang Technological University Singapore, 639798,Singapore*

Abstract

Until now the majority of the neuro and fuzzy modeling and control approaches for rotary wing Unmanned Aerial Vehicles (UAVs), such as the quadrotor, have been based on batch learning techniques, therefore static in structure, and cannot adapt to rapidly changing environments. Implication of Evolving Intelligent System (EIS) based model-free data-driven techniques in fuzzy system are good alternatives, since they are able to evolve both their structure and parameters to cope with sudden changes in behavior, and performs perfectly in a single pass learning mode which is suitable for online real-time deployment. The Metacognitive Scaffolding Learning Machine (McSLM) is seen as a generalized version of EIS since the metacognitive concept enables the what-to-learn, how-to-learn, and when-to-learn scheme, and the scaffolding theory realizes a plug-and-play property which strengthens the online working principle of EISs. This paper proposes a novel online identification scheme, applied to a quadrotor using real-time experimental flight data streams based on McSLM, namely Metacognitive Scaffolding Interval Type 2 Recurrent Fuzzy Neural Network (McSIT2RFNN). Our proposed approach demonstrated significant improvements in both accuracy and complexity against some renowned existing variants of the McSLMs and EISs.

Keywords: Evolving, Fuzzy, Metacognitive, Online Identification,

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)

Quadcopter, Scaffolding

1. Introduction

Unmanned Aerial Vehicles (UAVs) are aircraft with no aviator on-board. UAV autonomy varies from partial to complete, which begins from human operator based partial remote control to fully autonomous control by onboard computers. Autonomy enables UAVs to perform some tasks very well where human involvement would be dangerous, expensive or simply too tedious. Comparatively higher portability, smaller size, simple method of assembly and reconstruction and lower expenditure have caused the rapid growth of UAV applications such as delivery of equipment in hostile environments, infrastructure inspection and environmental monitoring as described in many places in the literature (e.g. [1, 2, 3]).

UAVs are classified into three subdivisions, namely fixed wing, rotary wing and flapping wing, where the Rotary wing UAVs (RUAVs) can be further classified by the number of rotors as a helicopter, quadcopter, hexacopter, octocopter etc. Among various RUAVs, the most commonly used is the quadcopter. The history of the quadcopter is not new; since the first one was built in 1907 [4]. However, from the beginning to the middle of the 20th century all of them were manned vehicles [5]. Advancements in control theory accelerated the research on unmanned quadcopter in the last quarter of the 20th century. Some of the latest research projects on quadcopter are elaborated in [6, 7, 8, 9, 10]. The cross ($\times$) and plus ($+$) are the two main configurations used to construct a quadrotor, and a simple cross-configured quadcopter is shown in Figure 1. Among four rotors, one pair of rotors situated in the two opposite arms rotate clockwise; another pair rotates counter-clockwise for the torque balancing. The four elementary movements of the quadcopter are vertical altitude Z, roll (θ), pitch (ϕ), and yaw (ψ). For the vertical altitude movement, all four rotors need to speed up or down by equal quantity. For rolling to the right on the X-axis, the speed of the left rotors are increased and the right rotor decreased. The opposite differential commands are given to move to the left. Similarly, for the pitching movement (with respect to the Y-axis), the front and rear rotors are utilized in a similar way. In the yaw movement, the quadcopter rotates on the Z-axis. It is accomplished by increasing the speed of one diagonally opposite rotor pair whilst decreasing the speed of the other pair. Due to the

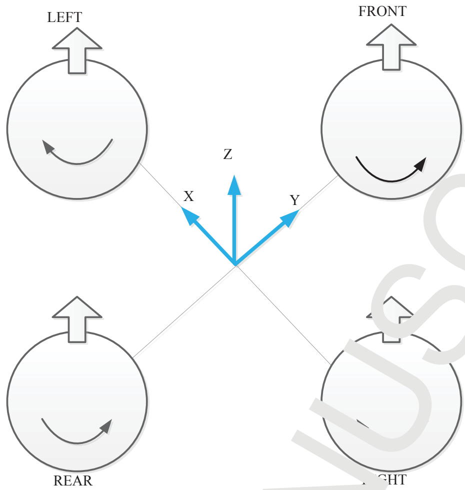


Figure 1: A simple cross-configured quadcopter model.

quadcopters vast applicability in both civilian and military sectors, research interest is increasing to make them more intelligent.

The quadcopter has six degrees of freedoms: they are three translational motions along the X, Y, and Z-axes, and three rotational motions (θ, ϕ, ψ). Besides, the quadcopter system is highly nonlinear and under-actuated. Accurate modeling of quadcopters by considering all the translational and rotational motions and by utilizing the four control input (Z, θ, ϕ, and ψ) is necessary to obtain good control action. Until now, most quadcopter models are based on dynamic equations of the system, where the aggressive trajectories of quadcopters are difficult to integrate. In addition, various non-stationary factors like motor degradation, time varying payload, wind gusts, and rotor damage are extremely difficult to predict and model mathematically and consequently hard to incorporate even for the first principle model based conventional and advanced techniques like PID [11], Linear Quadratic (LQ) techniques [12, 13], Sliding Mode Control (SMC) [14, 15, 16, 17, 18], back-stepping control [19, 20, 21], Feedback Linearization (FBL) [15, 22], H_∞ robust control [23, 24] etc. In more complex systems the physical model may not be possible to derive. These challenges are leading to increasing research interest in data-driven modeling techniques for system identification with real-time sensory data and limited expert knowledge.

In the data-driven techniques, system identification is a vital part. Successful system identification indicates closeness of the input-output behavior of the identified system with the input-output behavior of the actual plant. The data-driven system identification or modeling can play an important role in quadcopter systems, since their counterpart i.e. the model based parameter identification requires several experimental tests to obtain the model parameters. Even some parameters are difficult to obtain from the experiments and problem-dependent. Thus, the model-based system requires a lot of effort for better accuracy. Whereas the data-driven quadcopter model can be used as a generalized model with different motors, propellers or sensor combinations. Some of the commonly used non-linear data-driven system identification techniques are: describing function method, block structured systems, fuzzy logic, neural networks, and Nonlinear Autoregressive Moving Average Model with Exogenous inputs (NARMAX methods). Among these techniques, fuzzy logic [25, 26, 27, 28, 29] and neural network [30] based artificial intelligent systems are promising computational tools since they demonstrate learning capability from a set of data and approximate reasoning trait of human beings which cope with the impression and uncertainty of the decision making process [31]. Furthermore, the fuzzy system offers a highly transparent solution which can be followed easily by the operator [32]. Due to the numerous advantages of fuzzy logic systems [33], they are merged with conventional techniques as fuzzy-PID, fuzzy-PI, fuzzy-PD [34, 35, 36, 37, 38, 39], fuzzy-sliding mode [40, 41], fuzzy back-stepping [42, 43] to model the quadcopter more accurately and consequently to achieve better control action. In these conventional data-driven approaches, the experimental quadcopter flight tests are conducted repetitively on the desired trajectory and from these experimental knowledge, the target trajectory is estimated for the nonlinear time varying quadcopter dynamics. By following this approach, the quadcopter can be identified for a specific trajectory; and further training is required for new trajectory. Actually, this difficulty in identifying quadcopter system is not for the systems stochastic behavior or for the unwanted noise from experimentation. Rather it is for not considering the unobserved data which rely highly on expert domain knowledge because a purely fuzzy system approach without learning capability will have limited generalization power. Besides, a major limitation of these conventional fuzzy logics and neural networks based quadcopter modeling and controlling is the inability to evolve their structure to adapt with sudden changes. They also adopt a batched working principle which has to revisit entire dataset over multiple passes rendering them not scalable for online real-time deployment.

Therefore, to solve the problems that exist with conventional intelligent systems, Evolving Intelligent Systems (EISs) are a good candidate [44, 45], since they learn from scratch with no base knowledge and are embedded with the self-organizing property which adapts to changing system dynamics [46]. EIS fully work in a single-pass learning scenario which is scalable for online real-time requirement under limited computational resources such as UAVs platform [47]. Nevertheless, EISs remain cognitive in nature where they still require scanning all samples regardless of their true contribution and training samples must be consumed immediately with the absence of learning capability to determine ideal periods to learn these samples [48]. The Metacognitive Learning Machine (McLM) technique enhances the adaptability of EIS by interpreting the meta-memory model of [49] where the learning process is developed in three phases, namely what-to-learn, how-to-learn and when-to-learn [50, 51]. The what-to-learn is implemented with a sample selection mechanism which determines whether to accept data samples, the how-to-learn is where the underlying training process takes places, the when-to-learn is built upon a sample reserved mechanism which allows to delay the training process of particular samples when their significance does not suffice to trigger the learning mechanism. Recent advances in the McLM [52, 53, 54] have involved the concept of scaffolding theory as a foundation of the how-to-learn another prominent theory in psychology to help learners to solve complex tasks. The use of scaffolding theory is claimed to generate the plug-and-play property where all learning process are self-contained in the how-to-learn without over-dependence on pre-and/or post-processing steps. It is worth noting that the Scaffolding theory does not hamper the online learning property of EISs since all learning components follow strictly single-pass learning mode which is well-suited for online real-time applications. The scaffolding theory consists of two parts: active supervision and passive supervision. The passive supervision is constructed using parameter learning theories which demand target variables to elicit system errors for correction signals while the active supervision features three components: fading, complexity reduction and problematizing. The complexity reduction alleviates learning complexities by applying feature selection, data normalization, etc. and the problematizing focuses on concept drifts in data distributions while the fading component is meant to reduce the network complexity by discarding inactive components using the pruning and merging scenarios.

A novel online system identification of quadcopter based on a recently developed McSLM [55], namely McSIT2RFNN, is proposed in this paper .

McSIT2RFNN is structured as a six-layered network architecture actualizing interval type-2 Takagi Sugeno Kang (TSK) fuzzy inference scheme. This network architecture features a local recurrent connection which functions as an internal memory component to cope with the temporal system dynamic and to minimize the use of time-delayed input attributes [55]. Note that the local recurrent link does not compromise the local learning property because the spatio-temporal firing strength is generated by feeding previous states of system dynamic back to itself [55]. The rule layer consists of interval type-2 multivariate Gaussian functions with uncertain means which characterizes scale-invariant trait and maintains inter-correlation among input variables.The rule consequent layer is constructed by the nonlinear Chebyshev polynomial up to the second order which expands the degree of freedom of a rule consequent [55]. The polynomial is utilized here to rectify the approximation power of the zero-or first-order TSK rule consequent.

McSIT2RFNN features unique online learning techniques where a synergy between the metacognitive learning scenario and the Scaffolding theory comes into picture while retaining computationally light working principle through a fully one-pass learning scenario for online real-time applications. The learning process starts from the what-to-learn process using an online active learning mechanism, which actively extracts relevant training samples for training process while ruling out inconsequential samples for the training process. Selected training samples are then processed further in the how-to-learn designed under the scaffolding concept. The problematizing facet of the scaffolding theory is depicted by the rule growing mechanism which assesses statistical contribution of data points to be a candidate of a new rule. This scenario controls stability-and plasticity dilemma in learning from data streams since it guides to proper network complexity for a given problem and addresses changing data distributions by introducing a new rule when a change is detected. A rule recall scenario is put forward to represent the problematizing aspect which tackles the temporal or recurring drift. This learning mechanism plays a vital role during real-flight missions of the UAV because previously seen flight conditions often re-appear again in the future. The complexity reduction component is portrayed by an online feature selection scenario which puts into perspective relevance and redundancy of input features. This learning component lowers the input dimension which contributes positively to models generalization and computational complexity. The fading process relies on the rule merging scenario and the rule pruning scenario. The rule pruning scenario removes obsolete rules which are no longer relevant to current training concept by studying mutual information between fuzzy rule and the target variable. Significantly overlapping rules are coalesced into a single rule by the rule merging scenario and this mechanism is capable of cutting down network complexity and improving interpretability of rule semantics. The efficacy of our proposed methodology was carefully investigated through simulations using real-world flight data as well as real-time flight tests. Our algorithm was benchmarked with several prominent algorithms, and it was shown that our algorithm produced the most encouraging performance in attaining a trade-off between accuracy and complexity.

Our proposed methodology carry the following advantages: 1) it is compatible for online real-time deployment in the real flight tests of a quadcopter since it works fully in the single-pass learning mode. Furthermore, McSIT2RFNN does not necessarily see all sensory data streams due to its what-to-learn component further substantiating scalability of McSIT2RFNN in handling online data streams; 2) it features a highly flexible foundation which self-evolves its network structure and parameters in accordance with variations of data streams no matter how slow, rapid, gradual, and temporal a change in data streams is; 3) McSIT2RFNN is created from a combination between the interval type-2 fuzzy system which is more robust to face uncertainties than its type-1 counterparts and the recurrent network architecture which is capable of coping with temporal system dynamics and lagged input variables; 4) it actualizes a plug-and-play working principle where all learning modules are embedded in only a single training scenario without the requirement of pre-and/or post-training steps. The major contributions of this paper are summarized as follows: 1) a novel online system identification of quadcopter based on a psychologically inspired learning machine, namely McSIT2RFNN, is proposed; 2) real-time flight tests were done where real-world flight data were obtained and preprocessed. We also made these flight data publicly available for the convenience of readers; 3) Experimental validations of the proposed approach were carried out to inspect the efficacy of the proposed approach. This includes simulations using real-world flight data and real-time flight tests [56].

The remaining part of the paper is organized as follows: Section 2 describes the learning policy of the McSIT2RFNN technique by describing both the cognitive and meta-cognitive components. In section 3, the details of the quadcopter flight experiment and system identification is explained. Finally, the paper ends with concluding remarks in section 4.

2. Online learning policy of McSIT2RFNN

This section describes the learning policy of Meta-cognitive Scaffolding Based Interval Type 2 Recurrent Fuzzy Neural Network (McSIT2RFNN) [55]. The McSIT2RFNN has two components namely cognitive and meta-cognitive. The cognitive component corresponds to the network structure of McSIT2RFNN while the metacognitive component consists of learning scenarios to fine-tune the cognitive component.

2.1. Cognitive mechanism of McSIT2RFNN

In McSIT2RFNN, a six-layered recurrent network structure with a local recurrent connection is utilized for the hidden layer. The first layer is known as the input layer, which passes the fed input to the second layer as follows:

$$\big(\eta_{out}\big)_k^1 = \big(\eta_{atv}\big)^1(x_k) = x_k \tag{1}$$

where η_{out} represents output a layer, and η_{atv} denotes the forward activation function of a layer.

Unlike the conventional neuro-fuzzy system, the univariate Gaussian function is replaced by an interval-valued multivariate Gaussian function with uncertain mean and then it is utilized in the second layer of the McSIT2RFNN, which is also known as the rule layer. This Gaussian function consequently generates an interval-valued firing strength as follows:

$$\widetilde{\eta}_{out}{}^2 = \big(\eta_{atv}{}^2\big)\big(\eta_{out}{}^1\big) = \exp(-(\Gamma_n^2 - \widetilde{\zeta}_i)\Sigma_i^{-1}(\Gamma_n^2 - \widetilde{\zeta}_i)) \tag{2}$$

where, $\widetilde{\eta}_{out}{}^2 = [\underline{\eta_{out}}^2, \overline{\eta_{out}}^2]$, $\widetilde{\zeta}_i = [\underline{\zeta}_i, \overline{\zeta}_i]$, and $\widetilde{\zeta}_i$ is the uncertain centroid of the ith rule abiding by the condition $\underline{\zeta}_i < \overline{\zeta}_i$. If we consider to model or identify a Multi-Input-Single-Output system, the If-Then rule of the McSIT2RFNN can be expressed as follows:

$$\Re_j : \text{If} \quad X_n \text{ is } \widetilde{\eta}_{out_j}{}^2 \text{ Then } y_j = x_e^j \Omega_j \tag{3}$$

where x_e^j, Ω_j are respectively an extended input variable resulted from a nonlinear mapping of the wavelet coefficient ($x_e \in \Re^{1\times(2\mu+1)}$) and a weight vector ($\Omega \in \Re^{1\times(2\mu+1)}$). The consequent part of the rule is explained in the fifth layer. However, the rule presented in Eq. (3) is not transparent enough to expose atomic clause of the human-like linguistic rule [57]. It operates in a totally high dimensional space, therefore cannot be represented in fuzzy set. Since the non-axis-parallel ellipsoidal rule cannot be expressed directly in interval type-2 fuzzy environment, a transformation strategy is required [58, 59]. Such transformation technique should have the capability of formulating the fuzzy set for the non-axis parallel ellipsoidal cluster [60, 61]. The transformation strategy developed in [62] is extended in this work in terms of interval type-2 system, which can be expressed as follows:

$$\sigma_i = \frac{(\underline{r}_i + \overline{r}_i)}{2\sqrt{\Sigma_{ii}}} \tag{4}$$

where Σ_{ii} represents the diagonal element of the covariance matrix and $\tilde{r}_i$ denotes the Mahalanobis distance, which is $\widetilde{r}_i = (X_n - \widetilde{\zeta}_i)\Sigma_i^{-1}(X_n - \widetilde{\zeta}_i) = [\overline{r}_i \;\; \underline{r}_i]$, where $\overline{r}_i$ is the upper and $\underline{r}_i$ is the lower Mahalanobis distance and $\widetilde{\zeta}_i = [\underline{\zeta}_i, \overline{\zeta}_i]$. No transformation is required for the mean or centroid $(\widetilde{\zeta}_i)$ of the multivariate Gaussian function, since it can be directly applied to the fuzzy set level. After successfully presenting the interval-valued multi-variable Gaussian function into fuzzy set, the fuzzification process of the the upper and lower Gaussian membership functions with uncertain means $\widetilde{\zeta}_{j,i} = [\underline{\zeta}_{j,i}, \overline{\zeta}_{j,i}]$ is exhibited as follows:

$$\widetilde{\eta}^2_{out_{j,i}} = \exp\left(-\left(\frac{\eta^2_{atv_i} - \widetilde{\zeta}_{j,i}}{\sigma_{j,i}}\right)^2\right) \quad N(\widetilde{\zeta}_i^j, \sigma_i^j, \eta^2_{atv_i}) \;\; \widetilde{\zeta}_j = [\overline{\zeta}_i^j, \; \underline{\zeta}_i^j] \tag{5}$$

$$\overline{\eta}^2_{out_{j,i}} = \begin{cases} N(\overline{\zeta}_i^j, \sigma_{j,i}; \eta) & \eta^2_{atv_i} < \overline{\zeta}_i^j \\ 1 & , \quad \overline{\zeta}_i^j < x_i < \underline{\zeta}_i^j \\ N(\widetilde{\zeta}_i^j, \sigma_{j,i}; \eta^2_{atv_i}) & \eta^2_{atv_i} > \underline{\zeta}_i^j \end{cases} \tag{6}$$

$$\underline{\eta}^2_{out_{j,i}} = \begin{cases} N(\underline{\zeta}_i^j, \sigma_i^j; \eta) & x_i \leq \frac{(\overline{\zeta}_i^j + \underline{\zeta}_i^j)}{2} \\ N(\overline{\zeta}_i^j, \sigma_i^j; \eta^2_{atv_i}) & x_i > \frac{(\overline{\zeta}_i^j + \underline{\zeta}_i^j)}{2} \end{cases} \tag{7}$$

After getting the above expression of the fuzzy set, the fuzzy rule exposed in Eq. (3) can be transformed (2) into a more interpretable form as follows:

$$R_j : \text{If } \; X_1 \text{ is } \widetilde{\eta}_{out_1}{}^2 \text{ and } X_2 \text{ is } \widetilde{\eta}_{out_2}{}^2 \text{ and ... and } X_{nu} \text{ is } \widetilde{\eta}_{out_{nu}}{}^2 \text{ Then } y_j = x_e^i \Omega_i \tag{8}$$

where j is the number of rules, nu is the number of inputs. This transformation technique has overcame the issue of transparency of the multi-variable Gaussian function. The validity of $\widetilde{\eta}^2_{out_{j,i}} = [\overline{\eta}^2_{out_{j,i}}, \underline{\eta}^2_{out_{j,i}}]$ is proven in [55].

In the third layer, the upper and lower bound of membership degrees are connected using the product $t-norm$ operator in each fuzzy set and generates an interval-valued spatial rule firing strength as follows.

$$\left(\underline{\eta_{out}}\right)_i^3=\prod_{k=1}^{nu}\left(\underline{\eta_{atv}}\right)_{i,k}^3=\prod_{k=1}^{nu}\left(\underline{\eta_{out}}\right)_{i,k}^{2.1}\ ,\ \left(\overline{\eta_{out}}\right)_i^3=\prod_{k=1}^{nu}\left(\overline{\eta_{atv}}\right)_{i,k}^3=\prod_{k=1}^{nu}\left(\overline{\eta_{out}}\right)_{i,k}^{2.1} \tag{9}$$

The forth layer is known as temporal firing layer. In this layer of McSIT2RFNN a local recurrent connection is observed, where the spatial firing strength of previous observation is fed back to itself and generates a temporal firing strength as follows:

$$\left(\overline{\eta_{out}}\right)_{i,o}^4=\Lambda_i^o\left(\overline{\eta_{atv}}\right)_i^4+(1-\Lambda_i^o)\left(\bar{\eta}_{out}\right)_i^4(n-1) \tag{10}$$

$$\left(\underline{\eta_{out}}\right)_{i,o}^4=\Lambda_i^o\left(\underline{\eta_{atv}}\right)_i^4+(1-\Lambda_i^o)\left(\underline{\eta_{out}}\right)_i^4(n-1) \tag{11}$$

where $\Lambda_i^o\in[0,1]$ denotes a recurrent weight for the ith rule of the oth class. The fifth layer of McSIT2RFNN is the consequent layer, where the Chebyshev polynomial up to the second order is utilized to construct the extended input feature x_e [63]. This Chebyshev polynomial is expressed in Eq. (12).

$$\tau_{n+1}(x)=2x_k\tau_n(x_k)-\tau_{n-1}(x_k) \tag{12}$$

If X is considered as a 2-D input composition like $[x_1,x_2]$, then the extended input vector can be presented as $x_e=[1,x_1,\tau_2(x_1),x_2,\tau_2(x_2)]$, where $x_e\in\Re^{1\times(2\mu+1)}$, and μ represents the input dimension. This layer functions as an enhancement layer that maps to the original input vector to high dimensional space to rectify the mapping capability of the rule consequent. The extended input variable x_e is weighted and generates an output of the consequent layer as follows:

$$\left(\eta_{out}\right)_i^5=x_e^i\Theta_i \tag{13}$$

where Θ_i is a connection weight between the temporal firing layer and the output layer. In the output layer, type reduction mechanism is observed, where q design coefficient method is used instead of commonly used Karnik-Mendel (KM) technique. The final crisp output of the McSIT2RFNN can be expressed as follows:

$$y_{out}=\left(\eta_{out}\right)^6=\frac{(1-q_{out})\left(\overline{\eta_{out}}\right)_{i,o}^4\left(\eta_{out}\right)_i^5}{\sum_{i=1}^R\left(\underline{\eta_{out}}\right)_{i,o}^4}+\frac{q_{out}\left(\underline{\eta_{out}}\right)_{i,o}^4\left(\eta_{out}\right)_i^5}{\sum_{i=1}^R\left(\overline{\eta_{out}}\right)_{i,o}^4} \tag{14}$$

where R represents the number of fuzzy rules and q is the design factor $q \in \Re^{1\times no}$. The q design factor based type reduction mechanism performs by altering the proportion of the upper and lower rules to the final crisp output of McSIT2RFNN, where the normalization term of the original q design factor [64] is modified to overcome the invalid interval as shown in [55].

2.2. Meta-cognitive learning mechanism of McSIT2RFNN

In meta-cognitive learning policy, incoming training data streams $\big((X_n,),$ where X_n is an input variable vector $\big)$ are fed into the what-to-learn section. In this section, the probability of a sample to stay in the existing cluster is calculated as:

$$P_r(X_n \in N_i) = \frac{\frac{1}{N_i}\sum_{n=1}^{N_i} S_M(X_N, X_n)}{\sum_{i=1}^{R}\sum_{n=1}^{N_i}\frac{S_M(X_N,X_n)}{N_i}} \tag{15}$$

where, X_N is representing the current incoming data stream and X_n is indicating the nth support of the ith cluster, meanwhile $S_M(X_N, X_n)$ is defining the similarity measure. Since Eq. (15) requires to revisit previously seen samples, its recursive form is formulated as follows:

$$\begin{aligned}\frac{\sum_{n=1}^{N_i} S_M(X_N, X_n)}{N_i} &= \frac{\sum_{n=1}^{N_i-1}\sum_{j=1}^{u}(X_{n,j}-X_{N,j})^2}{(N_i-1)u} \\ &= \frac{(\sum_{j=1}^{u}(N_i-1)x_{N,j}{}^2 - 2\sum_{j=1}^{u}x_{N,j}K_{i,j} + v_{N_i})}{(N_i-1)u}\end{aligned} \tag{16}$$

where, $K_{N_i,j} = K_{N_i-1,j} + x_{N_i-1,j}$, and $v_{N_i} = v_{N_i-1} + \sum_{j=1}^{u} x_{N_i-1,j}{}^2$.

The necessity of a data sample to be trained by the how-to-learn section is monitored by the what-to-learn section through computation of the sample's entropy which portrays the level of uncertainty caused by the samples as follows:

$$H_{tr}(N|X_n) = -\sum_{i=1}^{R} P_r(X_n \in N_i)\log P_r(X_n \in N_i) \tag{17}$$

In McSIT2RFNN structure, a highly uncertain data stream is accepted as a training sample since it helps to mitigate the uncertainties in learning the target function. However, it opens the door for outliers to be fed to the how-to-learn section. To overcome this shortcoming, the entropy or uncertainty measured by Eq. (17) can be weighted by its average distance to the R, which are the densest local regions of the cognitive part [65]. Thus, computation of an average distance between the enquired sample and focal point can be expressed as:

$$A_d(X) = \frac{\sum_{i=1}^{R} similarity(X, C_i)}{R} \tag{18}$$

where *similarity* (X, C_i) is a distance function that computes the pair-wise similarity value between two examples like Cosine, Euclidean, etc. Finally, combining the concept of Eq. (18) in Eq. (17) the H_{tr} can be modified as:

$$H_{tr} = H_{tr}(N|X_n) \times A_d(X) \tag{19}$$

Acceptance of a data stream depends on the magnitude of H_{tr} in Eq. (19), where H_{tr} should be greater than or equal to a threshold as follows:

$$H_{tr} \geq \delta \tag{20}$$

where δ denotes an uncertainty threshold which is not constant rather it is adjusted dynamically. In this method, δ is set as $\delta_{N+1} = \delta_N(1 \pm s_s)$, where $\delta_{N+1} = \delta_N(1 + s_s)$ creates augmentation by admitting training data from the training process for minimizing the computational load and vice versa. The value of the step size s_s is set as 0.01, which refers to the thumb rule in [66]. This tuning scenario is necessary notably in non-stationary environments since a concept change directly hits the sample consumption.

After satisfying the condition of Eq. (20), a data stream is fed to the how-to-learn phase. The how-to-learn phase of McSIT2RFNN is derived from the Scaffolding theory. It encompasses both parameter and structural learning scenarios which are done in the strictly single-pass manner.

2.2.1. Mechanism of growing rule

The feature of growing rules in the how-to-learn section is governed by the Generalized Type-2 Datum Significance (GT2DQ) method forming a modification of the neuron significance of [67, 68] to the context of interval-valued multi-variable fuzzy rule. Gaussian Mixture Model (GMM) is used in this method as the input density function to cope with complex and even irregular data clouds. The extended formula of the neuron significance for the multi-variable Gaussian neuron [68] is further extended to the generalized interval-valued neuron in [55], which is utilized in the rule growing mechanism of this work. To express the significance of ith multi-variable interval-valued rule, the $L_u - norm$ of the error function is weighted by the input density function which can be presented as follows:

$$\mathcal{E}_i = \|\Omega_i\|_u(1-q)\left(\int_{\Re^{nu}} \exp(-u||x-\overline{\zeta}_i||^2_{\Sigma_i} p(x)dx)\right)^{1/u} + \|\Omega_i\|_u\, q\left(\int_{\Re^{nu}} \exp(-u||x-\underline{\zeta}_i||^2_{\Sigma_i} p(x)dx)\right)^{1/u} \tag{21}$$

where the Gaussian term under the integral can be written as follows:

$$(2\pi/u)^{nu/2}\det(\Sigma_i)^{-1/2} \times N(x;\, \widetilde{\zeta}_i\Sigma_i^{-1}/u),\; \widetilde{\zeta}_i = [\underline{\zeta}_i, \overline{\zeta}_i]$$

Therefore, it can be realized that the neuron significance depends on the input density $p(x)$. Usually, the input density $p(x)$ is considered to follow simple data distributions as explained in [69] or uniform data distribution as described in [70]. Utilizing the concept of GMM, $p(x)$ is able to cope with complex data distributions and can be expressed as follows:

$$p(x) = \sum_{m=1}^{M} \alpha_m N(x;\, v_m,\, \Sigma_m) \tag{22}$$

where $N(x;\, v_m,\, \Sigma_m)$ denotes multi-variable Gaussian probability density function with mean vector $v_m \in \Re^{1\times nu}$ and covariance matrix $\Sigma_m \in \Re^{nu\times nu}$, α_m denotes the mixing coefficients which satisfies the condition $\sum_{m=1}^{M}\alpha_m = 1,\ \alpha_m > 0$. Now using the GMM in the input density $p(x)$, the further derivation can be expressed as follows:

$$\begin{aligned}\mathcal{E}_i =& \|\Omega_i\|_u(1-q)((2\pi/u)\det(\Sigma_i)^{-1/2} \\ &\times \sum_{m=1}^{M} \alpha_m \int_{\Re^{nu}} N(x;\overline{\zeta}_i\Sigma_i^{-1}/u)N(x;\, v_m,\, \Sigma_m)dx)^{1/u} \\ &+ \|\Omega_i\|_u q((2\pi/u)\det(\Sigma_i)^{-1/2} \\ &\times \sum_{m=1}^{M} \alpha_m \int_{\Re^{nu}} N(x;\underline{\zeta}_i\Sigma_i^{-1}/u)N(x;\, v_m,\, \Sigma_m)dx)^{1/u}\end{aligned} \tag{23}$$

The integral term of Eq. (23) is a product of two Gaussian distributions and can be solved as $\int_{\Re^{nu}} N(x;\widetilde{\zeta}_i\Sigma_i^{-1}/u)N(x; v_m, \Sigma_m)dx = N(\widetilde{\zeta}_i - v_m;\ 0, \Sigma_i^{-1}/u + \Sigma_m)$. Accordingly, the final formula of the GT2DQ method to express the significance of the ith interval-valued multivariable rule [55] is expressed as:

$$\begin{aligned}\mathcal{E}_i = \|\Omega_i\|_u (1-q)\big\{(2\pi/u)^{n/2}\det(\Sigma_i)^{-1/2}\overline{N}_i\gamma^T\big\}^{1/u} \\ + \|\Omega_i\|_u\ q\big\{(2\pi/u)^{n/2}\det(\Sigma_i)^{-1/2}\underline{N}_i\gamma^T\big\}^{1/u}\end{aligned} \tag{24}$$

In (24) the mixing coefficient is denoted by γ and can be expressed as:

$$\gamma = [\alpha_1, ..., \alpha_m, ..., \alpha_M] \in \Re^{1\times m} \tag{25}$$

In Eq. (24), $\overline{N}_i$ and $\underline{N}_i$ are defined as $\overline{N}_i = \lfloor N(\overline{\zeta}_i - v_1; 0, \Sigma_i^{-1}/u + \Sigma_1), (\overline{\zeta}_i - v_2; 0, \Sigma_i^{-1}/u + \Sigma_2), ..., (\overline{\zeta}_i - v_m; 0, \Sigma_i^{-1}/u + \Sigma_m), ..., (\overline{\zeta}_i - v_M; 0, \Sigma_i^{-1}/u + \Sigma_M)\rfloor$, $\underline{N}_i = \lfloor N(\underline{\zeta}_i - v_1; 0, \Sigma_i^{-1}/u + \Sigma_1), (\underline{\zeta}_i - v_2; 0, \Sigma_i^{-1}/u + \Sigma_2), ..., (\underline{\zeta}_i - v_m; 0, \Sigma_i^{-1}/u + \Sigma_m), ..., (\underline{\zeta}_i - v_M; 0, \Sigma_i^{-1}/u + \Sigma_M)\rfloor$.

L_2-norm is utilized in McSIT2RFNN where $u = 2$ since the majority of the researchers are using the same technique. Besides, some parameters of the Gaussian Mixture Model (GMM), namely the mean v_m, the covariance matrix Σ_m, the mixing coefficients α_m, and the number of mixing models M, are acquired using previously recorded data points $N_{prerecord}$ like [69, 67, 68]. In today's world of big data, having access to the $N_{prerecord}$ is easy. Furthermore, the total number of training data samples is noticeably larger than that of the pre-recorded data samples. The proposed method's sensitivity with regards to an altered number of prehistory samples is analyzed in [55], which proves that the $N_{prerecord}$ is not case sensitive.

In McSIT2RFNN, the generation of hypothetical rule depends upon an incoming data stream and therefore, $\overline{c}_i$, $\underline{c}_i$, Σ_i^{-1} are substituted with $\overline{c}_{R+1}$, $\underline{c}_{R+1}$, Σ_{R+1}^{-1}. The formula for crafting a hypothetical rule can be expressed as follow:

$$\widetilde{C}_{R+1} = X_N \pm \Delta X,\ diag(\Sigma_{R+1}) = \frac{\max((C_i - C_{i-1}), (C_i - C_{i+1}))}{\sqrt{\frac{1}{\ln(\epsilon=0.5)}}} \tag{26}$$

where ϵ is a predefined constant with a set value of 0.5. The ϵ regulates the proportion of rule base plenitude. ΔX is the uncertainty factor which initializes the footprint of uncertainty. In McSIT2RFNN, the value of ΔX is fixed at 0.1 for simplicity, although one can also use an optimization technique to adjust the uncertainty factor. A hypothetical rule can be added as a new rule by utilizing Eq. (26), only if the condition of Eq. (27) is satisfied as follows:

$$\max_{i=1,\dots,R}(\mathcal{E}_i) \leq (\mathcal{E}_{P+1}) \tag{27}$$

However, this condition itself does not suffice to be the only criteria to judge the contribution of a hypothetical rule because of the fact where limited information in respect to the spatial proximity of a data sample to existing rules is included. The distance information is required to delineate its relevance to current training concept. To overcome the limitation, another rule growing condition need to be satisfied as follows:

$$Fz \leq \rho, \text{ where } Fz = \max_{i=1,\dots,R}\left(q\left(\underline{\eta_{out}}\right)_i^3 + (1-q)\left(\overline{\eta_{out}}\right)_i^3\right) \tag{28}$$

where ρ denotes a critical value of the chi-square distribution χ^2 with nu degrees of freedom and a significant α level. In [55], the ρ is expressed as $\rho = \exp(-\chi^2(\alpha))$, which is similar to the expression of [71]. In McSIT2RFNN, the value of α is set as 5%. To compute the Fz of Eq. (28), the q design factors are applied for considering the effect of lower and upper rules. When a newly added rule satisfies the condition of Eq. (28), the new rule is sufficiently away from the existing rules, and consequently, has a low risk of overlapping. A similar approach is observed in [69, 67, 68]. However, McSIT2RFNN utilizes the spatial firing strength instead of measuring point to point distance [69, 67, 68]. The second section of Eq. 28 indicates the maximum spatial firing strength, which is also known as the winning rule. Finally, a hypothetical rule is added as a new rule by complying Eq. (26), Eq. (27) and Eq. (28), where the consequent part of the new rule is expressed as follows:

$$\Omega_{R+1} = \Omega_{win}, \quad \Psi = \overline{\omega} \tag{29}$$

where $\overline{\omega}$ is a large positive constant of magnitude of 10^5.

When a hypothetical rule does not satisfy the condition of neither Eq. 27 nor Eq. 28, then it is not added as a new rule. Nonetheless, the rule is then utilized by fine tuning its antecedent part. This tuning helps to absorb information carried by the latest data stream, while it maintains the existing network architecture as follows:

$$\widetilde{C}_{win}{}^{N} = \frac{N_{win}{}^{N-1}}{N_{win}{}^{N-1}+1}\widetilde{C}_{win}{}^{N-1} + \frac{(X_N - \widetilde{C}_{win}{}^{N-1})}{N_{win}{}^{N-1}+1} \tag{30}$$

$$\Sigma_{win}(N)^{-1}7 = \frac{\Sigma_{win}(N-1)^{-1}}{1-\alpha} + \frac{\alpha}{1-\alpha}$$
$$\frac{(\Sigma_{win}(N-1)^{-1}(X_N - \widehat{C}_{win}{}^{N-1}))(\Sigma_{win}(N-1)^{-1}(X_N - \widehat{C}_{win}{}^{N-1}))^T}{1+\alpha(X_N - \hat{C}_{win}{}^{N-1})\Sigma_{win}(old)^{-1}(X_N - \hat{C}_{win}{}^{N-1})^T} \tag{31}$$

$$N_{win}{}^{N} = N_{win}{}^{N-1} + 1 \tag{32}$$

where $\alpha = 1/(N_{win}{}^{N-1}+1)$, $\widetilde{C}_{win} = [\underline{C}_{win}, \overline{C}_{win}]$, and $\widehat{C}_{win} = (\underline{C}_{win} + \overline{C}_{win})/2$. This adaptation technique is extracted from the idea of the sequential maximum likelihood principle with an extension for incorporating the interval valued multivariate Gaussian function. Here the mid-point of uncertain centroids are utilized to adapt the certain input covariance matrix. The inverse covariance matrix is adjusted directly with no re-inversion process. This re-inversion process slows down the model update. Moreover, it may cause unstable computation in the presence of an ill-defined covariance matrix. A constant, k_{fs}, is applied in our practical implementation where it aims to replace big values of inverse covariance matrix causing numerical instability. Note that big values of covariance matrix means very small values of covariance matrix implying the Gaussian function with a very small width. In relationship to Scaffolding theory, the rule growing and adaptation technique described in this sub-section can be categorized as the problematizing component of active supervision due to its relationship with the drift handling approach due to the capability of updating the model with respect to the learning context. To overcome the drift, McSIT2RFNN embraces a passive approach by upgrading its structure continuously in accordance with the new incoming samples, and does not depend upon a dedicated drift detection approach like [72].

2.2.2. Mechanism of pruning rule

The idea of the neuron significance is also used in the rule pruning scheme due to its capability of detecting a superfluous fuzzy rule which does not have a significant role during its lifespan. Generalized Type-2 Rule Significance (GT2RS) method is utilized in McSIT2RFNN, which is an enhanced version of the T2ERS method through the utilization of the interval-valued multivariate Gaussian function [73]. The GT2RS technique follows the same principle like its rule growing counterpart, where a fuzzy rule's contribution is evaluated based on its statistical significance presented in Eq. (27). To sum up, a rule is pruned from the training process after satisfying a condition as follows:

$$\mathcal{E}_i < mean(\mathcal{E}_i) - 2std(\mathcal{E}_i),\ mean(\mathcal{E}_i) = \frac{\sum_{n=1}^{N}\mathcal{E}_{i,n}}{N},$$
$$std(\mathcal{E}_{i,n}) = \sqrt{\frac{\sum_{n=1}^{N}(\mathcal{E}_{i,n} - mean(\mathcal{E}_i))^2}{N-1}} \tag{33}$$

The calculation of mean and standard deviation of Eq. (33) can be done easily in a recursive way. The condition of Eq. (33) analyzes not only the statistical contribution of ith rule during its lifetime, but also the down-trend of the statistical contribution of that rule. The GT2RS method can approximate the rule significance rigorously by considering the overall training region, which verifies the methods effectiveness. In addition, the capability of handling complex and irregular data distributions of the GMM based input density function $p(x)$ is indicating that the future contribution of the ith fuzzy rule is also taken into account during the estimation of the rule significance. Furthermore, by utilizing Eq. (27) in GT2RS method, the influence of the local sub-model Ω_i is considered, which is usually ignored by most of the rule pruning techniques. It is worth-noting that the contribution of a fuzzy rule to the overall system output is highly affected by the output weight. Low output weight forces the output of a fuzzy rule to be negligible. The GT2RS method is representing the fading component of active supervision in Scaffolding theory.

In this work, using the default threshold values of growing and pruning module, only two rules are generated and no rules are pruned to identify the quadcopter from data streams with a very insignificant RMSE. Therefore, to observe the rule pruning mechanism clearly, the rule pruning threshold has been reduced from 0.9 to 0.4 and rule growing threshold from 0.45 to 0.25 in case of modeling quadcopter with 27000 samples. After that, the number of generated and pruned rules have been witnessed graphically as follows:

2.2.3 Mechanism of forgetting and recalling rule

Type-2 Relative Mutual Information (T2RMI) method is utilized in McSIT2RFNN for detecting the obsolete rules, where the main idea is to examine

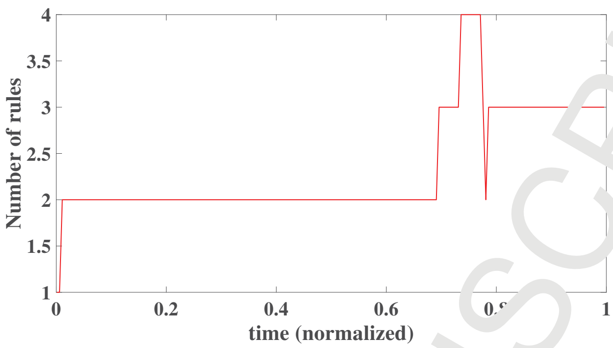


Figure 2: Number of added and pruned rules with a reduced threshold (in case of quad-copter model with 27000 samples)

the correlation between the fuzzy rules and the target concept. This T2RMI method is an improved version of RMI method in [74] with respect to the sequential working framework of the T2RMI. Moreover, the T2RMI method is tailored to cope up with the methodology of interval type-2 fuzzy system. Unlike the RMI, in T2RMI the maximum compression index (MCI) [75] is utilized, which ameliorates the robustness of the linear correlation measure. In comparison with other linear correlation measures like Pearson coefficient, the MCI is not affected by rotation. The MCI is another improved characteristic of the T2RMI method with respect to the RMI method since the RMI method is still supported by the classic symmetrical uncertainty approach. The T2RMI also has the ability to detect the outdated fuzzy rules by analyzing their relevance to the current data progression. In McSIT2RFNN, the T2RMI method is expressed as follows:

$$\xi\big((\widetilde{\eta}_{out})_i^3, y_{out}\big) = q_0\xi\big((\underline{\eta}_{out})_i^3, y_{out}\big) + (1-q_0)\xi\big((\overline{\eta}_{out})_i^3, y_{out}\big) \tag{34}$$

$$\xi\big((\underline{\eta}_{out})_i^3, y_{out}\big) = \frac{1}{2}\big(\mathrm{var}(\underline{\eta}_{out})_i^3\big) + \mathrm{var}(y_{out}) - \sqrt{(\mathrm{var}(\underline{\eta}_{out})_i^3 + \mathrm{var}(y_{out}))^2 - 4\mathrm{var}(\underline{\eta}_{out})_i^3\,\mathrm{var}(y_{out})\big(1-\rho\big((\underline{\eta}_{out})_i^3, y_{out}\big)^2\big)} \tag{35}$$

$$\rho\big((\underline{\eta}_{out})_i^3, y_{out}\big) = \frac{\mathrm{cov}\big(\underline{\eta}_{out}\big)_i^3, y_{out}\big)}{\sqrt{\mathrm{var}\big(\underline{\eta}_{out}\big)_i^3\big)\mathrm{var}(y_{out})}} \tag{36}$$

where $\mathrm{var}\big(\underline{\eta}_{out}\big)_i^3$, $\mathrm{cov}\big(\underline{\eta}_{out}\big)_i^3$, $\rho\big(\underline{\eta}_{out}\big)_i^3$ respectively represent the variance, covariance, Pearson index and output variable of the ith fuzzy rule with lower bound. Similar technique is also applied to the upper bound of the fuzzy rule $\xi\big(\big(\overline{\eta}_{out}\big)_i^3, y_{out}\big)$. Since the spatial firing strength extracts the relevance of the fuzzy rule in the input space, the fuzzy rule is represented by the spatial firing strength. In principle, $\xi\big(\big(\widetilde{\eta}_{out}\big)_i^3, y_{out}\big)$ implies the eigenvalue for the normal direction to the principal component of two variables $\big(\big(\widetilde{\eta}_{out}\big)_i^3, y_{out}\big)$, where maximum data compression is attained in time of projection of information along its principal component direction. Therefore, the MCI has the ability to categorize the cost of discarding the ith rule from the training process, aiming to achieve the maximum amount of information compression. Some interesting features of the MCI is exposed in [55]. A fuzzy rule is regarded as obsolete, or the rule is forgotten after satisfying the condition as follows:

$$\xi_{i,o} < mean(\xi_{i,o}) - 2std(\xi_{i,o}), \; mean(\xi_{i,o}) = \frac{\sum_{n=1}^{N} \xi_{i,o}^{n}}{N},$$
$$std(\xi_{i,o}) = \sqrt{\frac{\sum_{n=1}^{N}(\xi_{i,o}^{n} - mean(\xi_{i,o}))^2}{N-1}} \tag{37}$$

The T2RMI method is also utilized to recall the discarded rules when they become relevant again to the output of the system. This function is supported by the fact that correlation of a rule to target concept is influenced by the environments. In other words, in McSIT2RFNN a fuzzy rule is not permanently deleted and is added to a list of rules pruned by the T2RMI method $R^* = R^* + 1$, where R^* is the number of deactivated rules by the T2RMI method. Such a rule may recall in the future when it becomes relevant again to the systems output. This rule recall scenario makes the T2RMI method effective to deal with the cyclic drift by remembering old data distribution, which increases the relevance of the obsolete rules. The rule recall technique is activated after satisfying the condition as follows:

$$\max(\xi_{i^*}) > \max(\xi_i), \text{ where } i^* = 1, ..., R^* \text{ and } i = 1, ..., R \tag{38}$$

From Eq. (38) it is obvious that a rule is recalled when the validity of the obsolete rule is higher than any of the existing rules. Therefore, an obsolete rule brings the most compatible concept to describe the current data trend and should be reactivated as follows:

$$\widetilde{C}_{R+1} = \widetilde{C}_{i^*}, \quad {\Sigma_{R+1}}^{-1} = {\Sigma_{i^*}}^{-1}, \quad \Psi_{R+1} = \Psi_{i^*}, \Omega_{R+1} = \Omega_{i^*} \tag{39}$$

Unlike the rule recall concept in [76], the rule recall scenario of McSIT2RFNN works as another rule growing mechanism as clarified in [55]. The rule recall scenario can be categorized as the problematizing part of the Scaffolding theory.

2.2.4. Mechanism of Merging rule

In the online identification of a quadcopter, a complete dataset may not be available. This phenomenon creates an opportunity for two rules to move together which may cause a significant overlapping as a result of the continuous adaptation of fuzzy rules [77]. Therefore, an online rule merging mechanism is required to reduce the system's complexity and to improve rule interpretability. Recently, the idea of online rule merging has been introduced in EIS by [63, 78]. However, in these approaches, an over-dependence on a problem-specific predefined threshold to determine an acceptable level of overlapping is observed, which limits the flexibility of EIS.

A novel online rule merging technique called Type-2 Geometric Criteria (T2GC) is utilized in McSIT2RFNN. T2GC is an extended version of geometric criteria of [79], which was developed for the type-1 fuzzy system. This T2GC not only observes at the overlapping degree between rules but also looks at their geometric interpretation in the product space thoroughly. Two important properties of this T2GC are the overlapping degree and homogeneity. These two criteria are applied mainly to examine the similarity of the winning rule in light of the fact that the winning rule is the only one to receive the rule premise adaptation expressed in Eq. (34)-(36) and a major underlying reason of overlapping. Hence, this procedure targets to relieve the computational burden.

Overlapping Degree. The overlapping degree examines the similarity level of two rules to analyze their possibility of being redundant. Because of the necessity of developing a threshold-free rule merging process and the construction of McSIT2RFNN is with multi-variable Gaussian function, the Bhattacharyya distance is utilized[80]. The benefits of using the Bhattacharyya distance is that it can analyze whether two clusters are exactly disjoint, touching, or overlapping without any trouble in selecting predefined threshold. The overlapping degree between the winning rule and other rules $i = \{1, ..., R\} \setminus \{win\}$ can be expressed as:

$$s_1(win, i) = (1 - q)\overline{s}_1(win, i) + q_o\underline{s}_1(win, i) \tag{40}$$

$$\overline{s}_1(win,i) = \frac{1}{8}(\overline{c}_{win} - \overline{c}_i)^T \Sigma^{-1} (\overline{c}_{win} - \overline{c}_i) + \frac{1}{2}\ln \frac{\det(\Sigma^{-1})}{\sqrt{\det(\overline{\Sigma_{win}^{-1}})(\overline{\Sigma_i^{-1}})}} \tag{41}$$

$$\underline{s}_1(win,i) = \frac{1}{8}(\underline{c}_{win} - \underline{c}_i)^T \Sigma^{-1} (\underline{c}_{win} - \underline{c}_i) + \frac{1}{2}\ln \frac{\det(\Sigma^{-1})}{\sqrt{\det(\overline{\Sigma_{win}^{-1}})(\overline{\Sigma_i^{-1}})}} \tag{42}$$

where $\Sigma^{-1} = (\Sigma_{win}^{-1} + \Sigma_i^{-1})/2$. The conditions such as $s_1(win,i) > 0, s_1(win,i) < 0$, and $s_1(win,i) = 0$ exhibit respectively the overlapping, disjointing, and touching phenomenon of two clusters. In the McSIT2RFNN, the rule merging process is considered mandatory when two rules are overlapping and/or touching as follows:

$$s_1(win,i) \geq 0 \tag{43}$$

It is important to mention that the utilization of Bhattacharyya distance is suitable for the McSIT2RFNNs rule since the multivariate Gaussian function in the Bhattacharyya distance has a one-to-one relationship with that of the McSIT2RFNN.

Homogeneity Criterion. Homogeneity of clusters has an important role in merging two clusters, since the merging of non-homogeneous clusters may cause cluster delamination, undermining generalization and representation of local data clouds [81, 79]. The Cluster delamination is indicating an oversized cluster that covers two or more distinguishable data clouds. The measure of homogeneity of clusters in McSIT2RFNN is formulated by examining the volume of the merged clusters in contrast with their individual volume as follows:

$$\underline{\nu}_{merged} + \overline{\nu}_{merged} < u(\overline{\nu}_i + \underline{\nu}_i + \overline{\nu}_{win} + \underline{\nu}_{win}) \tag{44}$$

Finally after satisfying the condition of Eq. (43), and Eq. (44), the rules are merged. Eq. (44) also presents a minor chance of cluster delamination since the volume of the merged cluster is less than the volume of two independent clusters, and therefore the two clusters form a joint homogeneous region. The term u is involved in Eq. (44) to combat obstruct the curse of dimensionality.

After satisfying all rule merging conditions, two merging candidates are combined. Since a rule containing more supports should have higher influence to ultimate shape and orientation of the merged cluster, the rule merging procedure is directed by the weighted average strategy [77] as follows:

$$\widetilde{C}_{merged}{}^{new} = \frac{\widetilde{C}_{win}{}^{old} N_{win}{}^{old} + \widetilde{C}_i{}^{old} N_i{}^{old}}{N_{win}{}^{old} + N_i{}^{old}},$$
$$\widetilde{C}_i = [\underline{C}_i + \overline{C}_i], \quad N_{merged}{}^{new} = N_{win}{}^{old} + N_i{}^{old} \tag{45}$$

$$\Sigma_{merged}^{-1}{}^{new} = \frac{\Sigma_{win}^{-1}{}^{old} N_{win}{}^{old} + \Sigma_i^{-1}{}^{old} N_i{}^{old}}{N_{win}{}^{old} + N_i{}^{old}},$$
$$\Omega_{merged}{}^{new} = \frac{\Omega_{win}{}^{old} N_{win}{}^{old} + \Omega_i{}^{old} N_i{}^{old}}{N_{win}{}^{old} + N_i{}^{old}} \tag{46}$$

2.2.5. Mechanism of Online Feature Selection

The feature selection mechanism plays an important role to improve the performance of EIS by reducing computational complexity and makes modeling problems easier to solve. Therefore, feature selection characterizes the complexity reduction part of Scaffolding theory. Majority of these feature selection mechanisms are a part of a pre-processing step. However, very recently research in online feature selection mechanism of EIS is being conducted [63, 77, 81]. These techniques can minimize the significance of inconsequential features by assigning a low weight. But they still keep superfluous input attributes in the memory. Therefore the complexity issue remains unsolved. Besides, in recent EISs the online feature selection mechanism only measures the relevance between input attributes and target variables. They do not consider the redundancy among input attributes. A novel online feature selection mechanism, called Sequential Markov Blanket Criterion (SMBC) is utilized in McSIT2RFNN, which is able to mitigate all the above mentioned limitations of the existing EIS and works completely with the single-pass learning environment. It is an improved version of MBC [82].

By analyzing the Markov blanket theory, four different types of input feature are obtained in respect to their contribution; they are namely: irrelevant, weakly relevant, weakly relevant but non-redundant, and strongly relevant. The SMBC targets to eliminate irrelevant, weakly relevant input features from the training process, while keeps weakly relevant but non-redundant, and strongly relevant features in the training process. In SMBC, C-Correlation and F-Correlation tests are developed and then utilized to deal with the issue of irrelevance and redundancy respectively. These two correlations are defined as follows:

Definition 1 (C-Correlation) [82]: *The relevance of the input feature is indicated by the correlation of input feature x_k and target variable t_{out}, which are measured by the C-correlation $C(x_k, t_{out})$.*

Definition 2 (F-Correlation) [82]: *The issue of redundancy is signified by the similarity degree of two different input variables $x_k, x_{k1}, k \neq k1$. The measure of similarity between two input attributes is called the F-correlation $F(x_k, x_{k1})$.*

The MCI method exposed in Eq. (34)-(36) is adopted to analyze the C and F-correlation. It is accomplished by just replacing $\left((\widetilde{\eta}_{out})_i^3, y_{out}\right)$ in Eq. (34)-(36) with (x_k, t_{out}), and (x_k, x_{k1}). The SMBC is implemented in two stages, where in the first stage the F-correlation eliminates the inconsequential features, and consequently reduce complexity. It helps the next step, the C-correlation, to run with a smaller number of input variables. The working procedure in the F-correlation and C-correlation in McSIT2RFNN is described elaborately in [55].

2.2.6. Mechanism of Adapting q design factor and recurrent weight

Adaptation or fine tuning of the free network parameters of the McSIT2RFNN, namely the design coefficients and the recurrent weights are accomplished by utilizing the zero-order density maximization (ZEDM) method. This ZEDM method is an improved version gradient descent technique since ZEDM utilizes error entropy as cost function unlike the mean square error (MSE) in gradient descent technique, therefore leads to a more accurate prediction. Since the accurate model of the error entropy is too complex to be derived with the first-principle technique, the cost function is formulated by utilizing the Parzen Window density estimation method and can be expressed as follows:

$$\hat{f}(0) = \frac{1}{N\phi\sqrt{2\pi}} \sum_{n=1}^{N} \exp\left(-\frac{{e_{n,0}}^2}{2\phi^2}\right) = \frac{1}{N\phi\sqrt{2\pi}} \sum_{n=1}^{N} K\left(\frac{-{e_{n,0}}^2}{2\phi^2}\right) \qquad (47)$$

where $e_{n,0}$ represents the system error of the oth output variable, T denotes a smoothing parameter, fixed as 1 for simplicity and N is the total number of samples seen so far. It is worth noting that a recursive expression can be derived to satisfy the one-pass learning requirement. The detailed adaptation process is explained in [55].

2.2.7. Mechanism of Adapting Rule Consequent

The adaptation of the rule consequent represents the passive supervision of Scaffolding theory because it relies on the system error, actualizing the action-consequent mechanism. For adapting the rule consequent the Fuzzily Weighted Generalized Recursive Least Square (FWGRLS) method [62] is used in McSIT2RFNN. FWGRLS is an improved version of the Generalized Recursive Least Square (GRLS) method [83] and performs locally. This local learning scenario provides a flexible mechanism and greater robustness, because each rule is fine-tuned separately. Thereby, entire learning procedures of a particular rule do not affect the stability and convergence of remaining rules. The local learning scenario also raises the interpretability of the TSK fuzzy rule as explained in [84]. The details of the FWGRLS method is elaborated in [62, 63, 70].

3. Quadcopter flight experiment and online system identification results

3.1. Experimental set up of quadcopter flight

Our quadcopter experiments were accomplished in the indoor UAV laboratory at the University of New South Wales, Canberra campus. We use a Pixhawk autopilot framework developed by an open and independent hardware project called PX4. The Pixhawk flight control unit (FCU) is manufactured and sold by 3D robotics and has three onboard sensors; namely gyroscope, accelerometer, and magnetometer. The experimental quadcopter model is displayed in Figure 3. To record quadcopter flight data the Robot Operating System (ROS), running under the Ubuntu 16.04 version of Linux was used. A ROS package called MAVROS was utilized in this work, where the MAVROS had enabled communication between the PX4 and a ROS enabled computer. By utilizing the ROS, a well-structured communication layer was introduced into the quadcopter that reduced the burden of having to reinvent necessary software.

During the real time flight testing accurate vehicle position, velocity, and orientation were the required information for verification of the proposed McSIT2RFNN based on-line system identification of quadcopter. In order to track the quadcopter in three dimensional space, a VICON optical motion capture system was employed to track the UAV motion with sub-millimetre accuracy. The indoor VICON motion capture system consisted of a volume that was $10 \times 10 \times 4.3$ m^3 and formed by a netted truss framework.

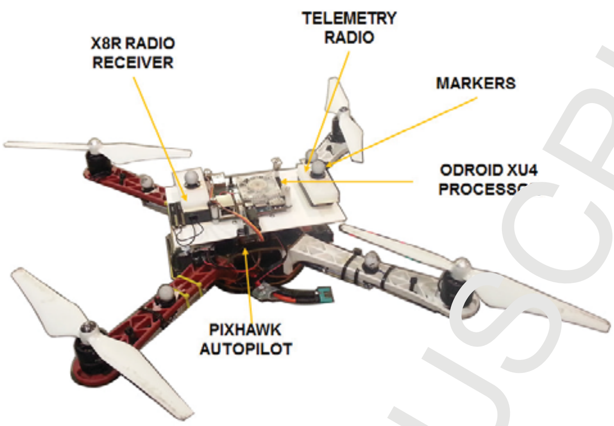


Figure 3: Pixhawk autopilot based experimental quadcopter model

The object tracking information was routed to the quadcopter via a custom SDK UDP package to the desired IP address which in our platform was an Odroid single board computer. At each time step, position, velocity and orientation information was recorded. During testing the pilot controlled the quadcopter RUAV manually from an RC transmitter using pitch, roll and yaw and thrust commands. To record key published topics, the *rosbag* recording tool was used. The *rosbag* enables us to record and synchronize all critical experimental data via published topics that are required for online system identification. Figure 4 represents the way of communicating of the experimental quadcopter UAV system during all the flight tests, where the dotted lines represents wireless communication and solid lines represents wired connection.

3.2. Online system identification results

For system identification of the quadcopter, a variety of quadcopter flight data have been utilized. Among them, there are three different datasets of quadcopter's altitude, consisting of approximately 9,000, 27,000 and 66,000 samples. Using these datasets, the quadcopter's altitude based multi-input-single-output (MISO) online system identification model of quadcopter has been constructed by utilizing the proposed McSIT2RFNN technique [55]. The proposed technique [55] based online data-driven quadcopter model has been also structured from four inputs and output datasets (vertical altitude and the three rotational movements (θ, ϕ, ψ)). The time step of all the

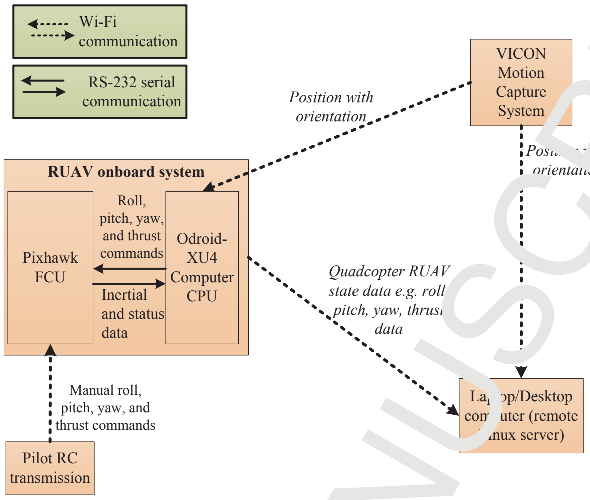


Figure 4: Pixhawk quadcopter RUAV's communication flow

dataset was 0.0198 sec. For comparing and validating the accuracy of the proposed technique, the quadcoper's data driven model has also constructed with eight different renowned EIS based neuro-fuzzy system, namely: eTS [44], simp_eTS [85], DFNN [86], GDFNN [87], FAOSPFNN [88], GENEFIS [62], Adaptive Neuro Fuzzy Inference System (ANFIS) [89], and PANFIS [70]. For the performance analysis the RMSE, number of network parameters, number of training samples, fuzzy rules, and execution time have been considered for each algorithm. All the results are summarized in Tables from 1 to 4. Table 1 to Table 3 express the results for a MISO quadcopter model with approximately 27,000, 66,000, and 9,000 samples of quadcopter's altitude for three different flight respectively. Table 4 summarizes the results of the MIMO quadcopter model. It is clearly observed from the results that, among these eight different algorithms the proposed McSIT2RFNN algorithm performs the best, since the lowest RMSE, fastest execution is observed. Besides, by utilizing the what-to-learn mechanism the McSIT2RFNN has reduced the number of samples required to train, which helps to reduce the execution time as observed in the Tables from 1 to 5. This sample deletion mechanism is not utilized in any other renowned variants of EIS discussed in this article. The altitude tracking performance of the proposed McSIT2RFNN algorithm based MISO quadcopter model with nearly 27,000 and 66,000 samples are displayed in Figure 6a and 6a respectively. A MIMO quadcopter model with nearly 9,000 samples for identifying thrust, roll, pitch, and yaw are displayed from Figure 7a to 7d correspondingly. From those Figures it is clearly observed that the proposed online models output are following the desired dataset collected experimentally very closely in all the cases. The evolving and online nature of the proposed McSIT2RFNN technique helps to track the quick changes in the desired trajectory.

In this work, to model the quadcopter with approximately 27,000, 66,000, and 9,000 samples two inputs ($X(t)$ and $Y(t-1)$) are utilized and in all cases two rules are generated. The If-Then expression of the rule in this work is exposed in Eq. (3). However, the rule presented in Eq. (3) is not transparent enough to expose atomic clause of the human-like linguistic rule. It operates in a totally high dimensional space, therefore cannot be represented in fuzzy set directly. To express such fuzzy rules with multidimensional kernel, the phrase "close to" is conventionally used [90, 91]. As a solution, a transformation strategy is employed in this work as expressed in Eq. (4) to convert the high dimensional space based rules to a lower dimensional human-like linguistic rules. After transformation, rules can be expressed in a conventional interval type-2 fuzzy set environment as exposed in Eq. (5), (6), and (7). Utilizing those fuzzy set expression of Eq. (5), (6), and (7), a more interpretable fuzzy rule is exhibited in Eq. (8).

Now the non-axis-parallel ellipsoidal rule generated in the high dimensional space by the McSIT2RFNN in case of quadcopter model with 27000 samples can be expressed as follows:

$$\begin{aligned}
&R_1 : \text{If} \quad X \text{ is close to} \\
&\widetilde{\eta}_{out}\left(\underline{\eta}_{out} = \left[\begin{bmatrix} -0.5934 \\ -0.2458 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right], \overline{\eta}_{out} = \left[\begin{bmatrix} 0.0066 \\ 0.3542 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right]\right) \\
&\text{Then } y_1 = -0.0043 - 0.0039X_1 - 0.0046\tau(X_1) + 0.9999X_2 - 0.0002\tau(X_2)
\end{aligned} \tag{48}$$

where $\widetilde{\eta}_{out} = \left[\underline{\eta}_{out}, \overline{\eta}_{out}\right]$ is the rule antecedent of the multi-variable Gaussian function, which consists of uncertain centroids $\widetilde{\zeta} = [\underline{\zeta}, \overline{\zeta}]$, where $\underline{\zeta} = \begin{bmatrix} -0.5934 \\ -0.2458 \end{bmatrix}$, and $\overline{\zeta} = \begin{bmatrix} 0.0066 \\ 0.3542 \end{bmatrix}$, and the inverse co-variance matrix $\Sigma^{-1} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$; y_i is denoting the rule consequent of the ith rule obtained from $y_i = x_e^i \Theta_i$, where Θ_i is a connection weight between the temporal firing layer and the output layer; In this work, X is a 2-D input

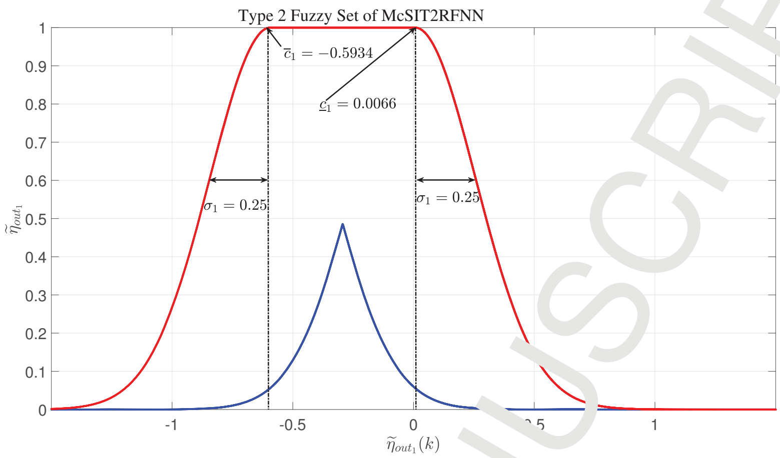


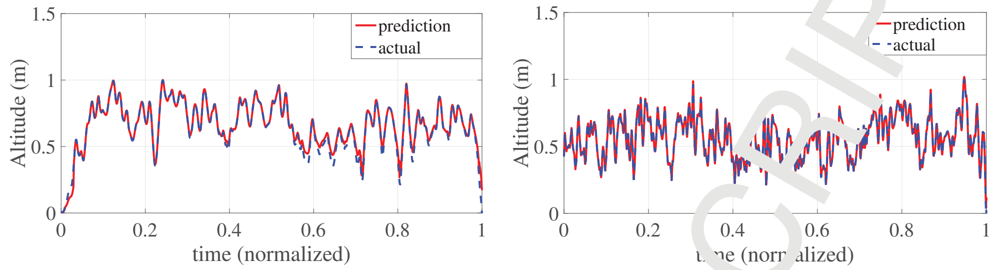
Figure 5: 1st Membership function (with uncertain centroid and certain width) of rule 1 of McSIT2RFNN

composition like $[x_1,\ x_2]$, then x_e is the extended input vector obtained using Chebyshev polynomial and can be expressed as $x_e = [1,\ x_1,\ \tau(x_1),\ x_2,\ \tau(x_2)]$, where, $x_e \in \Re^{1\times(2\mu+1)}$, μ is expressing the input dimension.

After transformation of the rule presented in Eq. (48) to a lower dimensional space, it can be expressed as follows:

$$
\begin{aligned}
&R_1 : \text{If} \quad X_1 \text{ is close to } \widetilde{\eta}_{out_1}\Big(\underline{\eta}_{out_1}(-0.5934,\ 0.25), \overline{\eta}_{out_1}(0.0066,\ 0.25)\Big)\\
&\text{and } X_2 \text{ is close to } \widetilde{\eta}_{out_2}\Big(\underline{\eta}_{out_2}(-0.2458,\ 0.20), \overline{\eta}_{out_2}(0.3542,\ 0.20)\Big),\\
&\text{Then } y_1 = -0.0043[\text{illegible}] - 0.00587X_1 - 0.00457\tau(X_1) + 0.99995X_2 - 0.00022\tau(X_2)
\end{aligned} \tag{49}
$$

where $\widetilde{\eta}_{out_1}$ and $\widetilde{\eta}_{out_2}$ stand for the interval valued Gaussian membership function corresponding to X_1, X_2. The uncertain centroid of $\widetilde{\eta}_{out_1}$ is $\widetilde{c}_1 = [\underline{c}_1, \overline{c}_1] = [-0.5934,\ 0.0066]$, and width is $\sigma_1 = 0.25$, and for $\widetilde{\eta}_{out_2}$ those parameters are $\widetilde{c}_2 = [\underline{c}_2, \overline{c}_2] = [-0.2458,\ 0.3542]$, $\sigma_2 = 0.20$. The 1st membership function of our rule 1 is shown in Figure (5). The plotted membership function has uncertain centroid of $\widetilde{c}_1 = [\underline{c}_1, \overline{c}_1] = [-0.5934,\ 0.0066]$, and certain width of $\sigma_1 = 0.25$.



(a) with approx. 27,000 samples (b) with approx. 66,000 samples

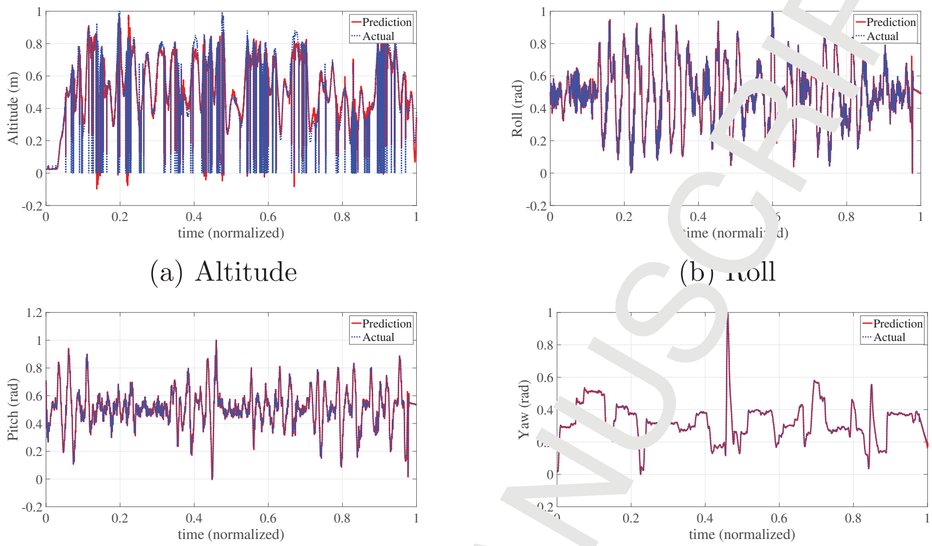
Figure 6: System identification of Quadcopter MISO model

3.3. Online system identification with noisy samples

To prove the robustness of the McSIT2RFNN against uncertainties, another quadcopter flight experiment has been accomplished considering some noise from VICON optical motion capture system. The quadcopter flight dataset consists of nearly 27000 samples with a noisy 1000 samples, which is utilized to model the quadcopter. The adaptation power of the proposed algorithm against noise is clear from the lowest obtained RMSE compared to its type-1 counterparts. Furthermore, with the noisy data still it can model the quadcopter with only 546 data samples, where its type-1 variants need all the training samples i.e. 16433 (60% of the total samples). Thereby, lowest execution time in modeling the quadcopter is also observed from the type-2 fuzzy based proposed McSIT2RFNN algorithm, which is only 2.13 seconds. Therefore, the results are clearly indicating its improved performance and uncertainty handling capacity than the type-1 counterpart. The results are summarized in Table 5.

4. Conclusion

EIS is an appropriate candidate for modeling a complex and highly nonlinear system like quadcopter RUAV. The incorporation of McSLM with EIS make it more appropriate. Such an advanced EIS called McSIT2RFNN is utilized to model the quadcopter with uncertainties from experimental quadcopter flight data. In McSIT2RFNN, a new local recurrent network architecture is driven by the interval-valued multivariate Gaussian function in the hidden node and the nonlinear Chebushev function in the consequent node. As with its predecessors, the McSIT2RFNN characterizes an open structure, which has the ability to grow, prune, adjust, merge, recall its hidden node



(a) Altitude (b) Roll

(c) Pitch (d) Yaw

Figure 7: System identification of Quadcopter MIMO model with approx. 9,000 samples

Table 1: Online system identification result comparison of MISO quadcopter model (approx. 27,000 samples)

Algorithm	Reference	RMSE	Network Parameters	Training Samples	Fuzzy Rule	Execution Time (sec)
DFNN	[86]	0.0780	10	16483	1	194.86
GDFNN	[87]	0.0067	10	16483	1	329.93
FAOSPFNN	[88]	0.0280	12	16483	2	38.10
eTS	[84]	0.0021	40	16483	4	9.39
simp_eTS	[85]	0.0020	13	16483	1	3.50
GENEFIS	[62]	0.0020	63	16483	1	3.29
PANFIS	[70]	0.0020	5	16483	1	2.92
ANFIS	[89]	0.0061	36	16483	6	33.01
McSIT2RFNN	[55]	**0.0013**	32	**1279**	2	**2.27**

automatically and to select relevant data samples of quadcopter flight on the fly using an online active learning methodology. The McSIT2RFNN is also

Table 2: Online system identification result comparison of MISO quadcopter model (approx. 66,000 samples)

Algorithm	Reference	RMSE	Network Parameters	Training Samples	Fuzzy Rule	Execution Time (sec)
DFNN	[86]	0.0810	10	39640	1	1113.69
GDFNN	[87]	0.0080	10	39640	1	1666.33
FAOSPFNN	[88]	0.0150	12	39640	2	135.59
eTS	[44]	0.0016	40	39640	2	21.03
simp_eTS	[85]	0.0015	13	39640	1	11.20
GENEFIS	[62]	0.0015	4	39640	1	7.63
PANFIS	[70]	0.0015	5	39640	1	6.93
ANFIS	[89]	0.0050	30	39640	5	34.93
McSIT2RFNN	[55]	**0.0008**	32	**2329**	2	**5.5**

Table 3: Online system identification result comparison of MISO quadcopter model (approx. 9,000 samples)

Algorithm	Reference	RMSE	Network Parameters	Training Samples	Fuzzy Rule	Execution Time (sec)
DFNN	[86]	0.15	10	5467	1	19.77
GDFNN	[87]	0.14	10	5467	1	23.64
FAOSPFNN	[88]	0.21	12	5467	2	28.58
eTS	[44]	0.14	40	5467	4	2.10
simp_eTS	[85]	0.13	13	5467	4	1.77
GENEFIS	[62]	0.13	26	5467	1	1.10
PANFIS	[70]	0.135	5	5467	1	1.15
ANFIS	[89]	0.46	48	5467	8	36.94
McSIT2RFNN	[55]	0.13	32	769	2	**0.91**

equipped with the online dimensionality reduction technique to cope with the curse of dimensionality. All learning mechanisms are carried out in the single-pass and local learning mode and actualize the plug-and-play learning

Table 4: Online system identification result comparison of MIMO quadcopter model (approx. 9000 samples)

Algorithm	Reference	RMSE1	RMSE2	RMSE3	RMSE4	Network Parameters	Training Samples	Fuzzy Rule	Execution Time (sec)	Input attribute
DFNN	[86]	0.26	0.18	0.14	0.15	10	5753	1	49.98	4
GDFNN	[87]	0.23	0.18	0.14	0.13	10	5753	1	85.75	4
FAOSPFNN	[88]	0.55	0.28	0.14	0.13	12	5753	1	12.11	4
eTS	[44]	0.22	0.20	0.15	0.10	292	5753	14	20.06	4
simp_eTS	[85]	0.29	0.31	0.29	0.18	104	5753	5	8.11	4
GENEFIS	[62]	0.24	0.19	0.19	0.11	4	5753	1	6.1	1
PANFIS	[70]	0.24	0.17	0.14	0.13	3	5753	1	5.4	4
McSIT2RFNN	[55]	**0.23**	**0.17**	**0.14**	**0.10**	66	**461**	2	**4.5**	4

Table 5: Online system identification result comparison of MISO quadcopter model (approx. 27,000 samples with 1000 noisy samples)

Algorithm	Reference	RMSE	Network Parameters	Training Samples	Fuzzy Rule	Execution Time (sec)
DFNN	[86]	0.0583	10	16483	1	211.72
GDFNN	[87]	0.0141	10	16483	1	328.56
FAOSPFNN	[88]	0.0242	12	16483	2	216.47
eTS	[44]	0.0072	40	16483	4	9.39
simp_eTS	[85]	0.0212	13	16483	6	14.36
GENEFIS	[62]	0.0037	63	16483	1	3.20
PANFIS	[70]	0.0097	5	16483	1	2.79
ANFIS	[89]	0.0061	36	16483	6	32.29
McSIT2RFNN	[55]	**0.0011**	32	**546**	2	**2.13**

principle, which aims to minimize the use of pre-and/or post-training steps. These features help the McSIT2RFNN method to identify the quadcopter RUAV more accurately than other variants of EIS. Thus, the accurate MISO and MIMO quadcopter modeling or better online identification results from McSIT2RFNN verifies their feasibility in modeling UAVs. In future research, a flexible controller for UAVs based on McSIT2RFNN will be developed and validate experimentally.

References

[1] Z. Sarris, S. Atlas, Survey of UAV applications in civil markets, in: Proceedings of the 9th Mediterranean Conference on Control and Automation, pp. 1–11.

[2] E. Hernandez-Martinez, G. Fernandez-Anaya, E. Ferreira, J. Flores-Godoy, A. Lopez-Gonzalez, Trajectory tracking of a quadcopter UAV with optimal translational control, in: 11th IFAC Symposium on Robot Control, pp. 228–233.

[3] C. E. Rash, P. A. LeDuc, S. D. Manning, Human factors in US military unmanned aerial vehicle accidents, in: Human Factors of Remotely Operated Vehicles, Emerald Group Publishing Limited, 2006, pp. 117–131.

[4] W. R. Young, et al., The Epic of Flight: The Helicopters, Time-Life, Chicago (1982).

[5] G. J. Leishman, Principles of helicopter aerodynamics, Cambridge university press, 2006.

[6] I. Kroo, F. Prinz, M. Shantz, P. Kunz, G. Fay, S. Cheng, T. Fabian, C. Partridge, The Mesicopter: A miniature rotorcraft concept–phase ii interim report, Stanford university (2000).

[7] R. Naldi, P. Pounds, S. De Marco, L. Marconi, Output tracking for quadrotor-based aerial manipulators, in: American Control Conference (ACC), 2015, IEEE, pp. 1855–1860.

[8] J. du Plessis, P. E. Pounds, Rotor Flapping for a Triangular Quadrotor, in: Australasian Conference on Robotics and Automation 2014, The University of Melbourne.

[9] P. Pounds, R. Mahony, P. Hynes, J. M. Roberts, Design of a four-rotor aerial robot, in: Proceedings of the 2002 Australasian Conference on Robotics and Automation (ACRA 2002), Australian Robotics & Automation Association, pp. 145–150.

[10] K. Rudin, M.-D. Hua, G. Ducard, S. Bouabdallah, A robust attitude controller and its application to quadrotor helicopters, IFAC Proceedings Volumes 44 (2011) 10379–10384.

[11] M. Sahul, V. N. Chander, T. Kurian, A novel method on disturbance rejection PID controller for quadcopter based on optimization algorithm, IFAC Proceedings Volumes 47 (2014) 192–199.

[12] F. Rinaldi, S. Chiesa, F. Quagliotti, Linear quadratic control for quadrotors UAVs dynamics and formation flight, Journal of Intelligent & Robotic Systems (2013) 1–18.

[13] P. Bader, S. Blanes, E. Ponsoda, Structure preserving integrators for solving (non-) linear quadratic optimal control problems with applications to describe the flight of a quadrotor, Journal of Computational and Applied Mathematics 262 (2014) 223–233.

[14] R. Xu, U. Ozguner, Sliding mode control of a quadrotor helicopter, in: 2006 45th IEEE Conference on Decision and Control, IEEE, pp. 4957–4962.

[15] D. Lee, H. J. Kim, S. Sastry, Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter, International Journal of control, Automation and systems 7 (2009) 419–428.

[16] S. Bouabdallah, R. Siegwart, Backstepping and sliding-mode techniques applied to an indoor micro quadrotor, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005, IEEE, pp. 2247–2252.

[17] S. L. Waslander, G. M. Hoffmann, J. S. Jang, C. J. Tomlin, Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005), IEEE, pp. 3712–3717.

[18] J.-J. Xiong, G. Zhang, Discrete-time sliding mode control for a quadrotor UAV, Optik-International Journal for Light and Electron Optics 127 (2016) 3718–3722.

[19] B. Ahmed, H. R. Pota, M. Garratt, Flight control of a rotary wing UAV using backstepping, International Journal of Robust and Nonlinear Control 20 (2010) 639–658.

[20] C. K. Tan, J. Wang, Y. C. Paw, T. Y. Ng, Tracking of a moving ground target by a quadrotor using a backstepping approach based on a full state cascaded dynamics, Applied Soft Computing 47 (2016) 47–62.

[21] T. Madani, A. Benallegue, Backstepping control for a quadrotor helicopter, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 3255–3260.

[22] T. Taniguchi, L. Eciolaza, M. Sugeno, Quadrotor control using dynamic feedback linearization based on piecewise bilinear models, in: 2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA), IEEE, pp. 1–7.

[23] M. Chen, M. Huzmezan, A combined mbpc/2 DOF H infinity controller for a quad rotor UAV, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 5520.

[24] D. Walker, I. Postlethwaite, Advanced helicopter flight control using two-degree-of-freedom H∞ optimization, Journal of Guidance, Control, and Dynamics 19 (1996) 461–468.

[25] A. Babaei, M. Mortazavi, M. Moradi, Classical and fuzzy-genetic autopilot design for unmanned aerial vehicles, Applied Soft Computing 11 (2011) 365–372.

[26] D. Domingos, G. Camargo, F. Gomide, Autonomous fuzzy control and navigation of quadcopters, IFAC-PapersOnLine 49 (2016) 73–78.

[27] F. Santoso, M. A. Garratt, S. G. Anavatti, Adaptive Neuro-Fuzzy Inference System identification for the dynamics of the AR. Drone Quadcopter, in: 2016 International Conference on Sustainable Energy Engineering and Application (ICSEEA), IEEE, pp. 55–60.

[28] F. Santoso, M. A. Garratt, S. G. Anavatti, Fuzzy logic-based self-tuning autopilots for trajectory tracking of a low-cost quadcopter: A comparative study, in: 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), IEEE, pp. 64–69.

[29] 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.

[30] K. S. Hatamleh, M. Al-Shabi, A. Al-Ghasem, A. A. Asad, Unmanned aerial vehicles parameter estimation using artificial neural networks and iterative bi-section shooting method, Applied Soft Computing 36 (2015) 457–467.

[31] M. Sundhararajan, X.-Z. Gao, H. Vahdat Nejad, Artificial intelligent techniques and its applications, Journal of Intelligent & Fuzzy Systems 34 (2018) 755–760.

[32] X. Ban, X. Gao, X. Huang, H. Yin, Stability analysis of the simplest Takagi-Sugeno fuzzy control system using Popov criterion, in: Soft Computing in Industrial Applications, Springer, 2007, pp. 63–71.

[33] L. A. Zadeh, Fuzzy logic, neural networks, and soft computing, Communications of the ACM 37 (1994) 77–85.

[34] M. Zareb, R. Ayad, W. Nouibat, Fuzzy-PID hybrid control system to navigate an autonomous mini-quadrotor, in: 2013 3rd International Conference on Systems and Control (ICSC), IEEE, pp. 906–913.

[35] C. Siyuan, Z. Ting, C. Yanchun, C. Jianliang, S. Junyi, L. Jin, A research on quadrotor attitude control based on fuzzy-PI method, in: 2012 31st Chinese Control Conference (CCC), IEEE, pp. 3548–3551.

[36] D. Gautam, C. Ha, Control of a quadrotor using a smart self-tuning fuzzy PID controller, International Journal of Advanced Robotic Systems 10 (2013) 380.

[37] T. Sangyam, P. Laohapiengsak, W. Chongcharoen, I. Nilkhamhang, Path tracking of UAV using self-tuning PID controller based on fuzzy logic, in: Proceedings of SICE Annual Conference 2010, IEEE, pp. 1265–1269.

[38] M. R. Mehranpour, O. Emamgholi, A. M. Shahri, M. Farrokhi, A new fuzzy adaptive control for a quadrotor flying robot, in: 2013 13th Iranian Conference on Fuzzy Systems (IFSC),, IEEE, pp. 1–5.

[39] L. Yanjun, X. Tianqi, Z. Xiaodong, A fuzzy self-tuning PD controller for a quadrotor: Design and implementation, in: 2016 Chinese Control and Decision Conference (CCDC), IEEE, pp. 2448–2453.

[40] Y. Yang, Y. Yan, Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control, Aerospace Science and Technology 54 (2016) 208–217.

[41] C.-L. Hwang, C. Jan, Fuzzy decentralized sliding mode under-actuated trajectory-tracking control for quadrotor unmanned aerial vehicle, in: 2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, pp. 1–10.

[42] H. Khebbache, M. Tadjine, Robust fuzzy backstepping sliding mode controller for a quadrotor unmanned aerial vehicle, Journal of Control Engineering and Applied Informatics 15 (2013) 3–11.

[43] F. J. Niroumand, A. Fakharian, M. S. Seyedsajadi, Fuzzy integral backstepping control approach in attitude stabilization of a quadrotor UAV, in: 2013 13th Iranian Conference on Fuzzy Systems (IFSC), IEEE, pp. 1–6.

[44] 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 (2004) 484–498.

[45] 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 (2018).

[46] E. D. Lughofer, FLEXFIS: A robust incremental learning approach for evolving Takagi-Sugeno fuzzy models, IEEE Transactions on Fuzzy Systems 16 (2008) 1393–1410.

[47] P. Angelov, Fuzzily connected multimodel systems evolving autonomously from data streams, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 41 (2011) 898–910.

[48] R. Elwell, R. Polikar, Incremental learning of concept drift in nonstationary environments, IEEE Transactions on Neural Networks 22 (2011) 1517–1531.

[49] T. O. Nelson, Metamemory: A theoretical framework and new findings, Psychology of Learning and Motivation 26 (1990) 125–173.

[50] D. P. Josyula, H. Vadali, B. J. Donahue, F. C. Hughes, Modeling metacognition for learning in artificial systems, in: World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009, IEEE, pp. 1419–1424.

[51] J. H. Flavell, M. H. S. Patto, J. Piaget, A psicologia do desenvolvimento de Jean Piaget, 1996.

[52] R. M. Isaacson, F. Fujita, Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflections on learning, Journal of Scholarship of Teaching and Learning 6 (2006) 39–55.

[53] S. Suresh, K. Dong, H. Kim, A sequential learning algorithm for self-adaptive resource allocation network classifier, Neurocomputing 73 (2010) 3012–3019.

[54] C. Zain, M. Pratama, E. Lughofer, S. G. Anavatti, Evolving Type-2 Web News Mining, Applied Soft Computing (2017).

[55] M. Pratama, E. Lughofer, M. J. Er, S. Anavatti, C.-P. Lim, Data driven modelling based on recurrent interval-valued metacognitive scaffolding fuzzy neural network, Neurocomputing 262 (2017) 4–27.

[56] 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: 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), IEEE, pp. 223–230.

[57] K. Cpałka, Design of Interpretable Fuzzy Systems, volume 684, Springer, 2017.

[58] R. P. Paiva, A. Dourado, Interpretability and learning in neuro-fuzzy systems, Fuzzy Sets and Systems 147 (2004) 17–38.

[59] M. Eftekhari, S. Katebi, M. Karimi, A. Jahanmiri, Eliciting transparent fuzzy model using differential evolution, Applied Soft Computing 8 (2008) 466–476.

[60] M. J. Gacto, R. Alcalá, F. Herrera, Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures, Information Sciences 181 (2011) 4340–4360.

[61] F. Rudziński, A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers, Applied Soft Computing 38 (2016) 118–133.

[62] M. Pratama, S. G. Anavatti, E. Lughofer, GENEFIS: toward an effective localist network, IEEE Transactions on Fuzzy Systems 22 (2014) 547–562.

[63] M. Pratama, M. J. Er, S. G. Anavatti, E. Lughofer, N. Wang, I. Arifin, A novel meta-cognitive-based scaffolding classifier to sequential non-stationary classification problems, in: 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, pp. 369–376.

[64] R. H. Abiyev, O. Kaynak, Type 2 fuzzy neural structure for identification and control of time-varying plants, IEEE Transactions on Industrial Electronics 57 (2010) 4147–4159.

[65] M. S. Hajmohammadi, R. Ibrahim, A. Selamat, H. Fujita, Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples, Information Sciences 317 (2015) 67–77.

[66] I. Zliobaite, A. Bifet, B. Pfahringer, G. Holmes, Active learning with drifting streaming data, IEEE Transactions on Neural Networks and Learning Systems 25 (2014) 27–39.

[67] M. Bortman, M. Aladjem, A growing and pruning method for radial basis function networks, IEEE Transactions on Neural Networks 20 (2009) 1039–1045.

[68] N. Vuković, Z. Miljković, A growing and pruning sequential learning algorithm of hyper basis function neural network for function approximation, Neural Networks 46 (2013) 210–226.

[69] G.-B. Huang, P. Saratchandran, N. Sundararajan, A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, IEEE Transactions on Neural Networks 16 (2005) 57–67.

[70] M. Pratama, S. G. Anavatti, P. P. Angelov, E. Lughofer, PANFIS: a novel incremental learning machine, IEEE Transactions on Neural Networks and Learning Systems 25 (2014) 55–68.

[71] K. Tabata, M. Sato, M. Kudo, Data compression by volume prototypes for streaming data, Pattern Recognition 43 (2010) 3162–3176.

[72] G. Ditzler, R. Polikar, Incremental learning of concept drift from streaming imbalanced data, IEEE Transactions on Knowledge and Data Engineering 25 (2013) 2283–2301.

[73] M. Pratama, J. Lu, S. Anavatti, E. Lughofer, C. P. Lim, An incremental meta-cognitive-based scaffolding fuzzy neural network, Neurocomputing 171 (2016) 89–105.

[74] H. Han, X.-L. Wu, J.-F. Qiao, Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm, IEEE Transactions on Cybernetics 44 (2014) 554–564.

[75] P. Mitra, C. Murthy, S. K. Pal, Unsupervised feature selection using feature similarity, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 301–312.

[76] M. Pratama, S. G. Anavatti, J. Lu, Recurrent classifier based on an incremental metacognitive-based scaffolding algorithm, IEEE Transactions on Fuzzy Systems 23 (2015) 2048–2066.

[77] E. Lughofer, Evolving fuzzy systems-methodologies, advanced concepts and applications, volume 53, Springer, 2011.

[78] E. Lughofer, On-line incremental feature weighting in evolving fuzzy classifiers, Fuzzy Sets and Systems 163 (2011) 1–23.

[79] E. Lughofer, C. Cernuda, S. Kindermann, M. Pratama, Generalized smart evolving fuzzy systems, Evolving Systems 6 (2015) 269–292.

[80] A. Bhattacharyya, On a measure of divergence between two statistical population defined by their population distributions, Bulletin Calcutta Mathematical Society 35 (1943) 28.

[81] E. Lughofer, M. Sayed-Mouchaweh, Autonomous data stream clustering implementing split-and-merge concepts-Towards a plug-and play approach, Information Sciences 304 (2015) 54–79.

[82] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5 (2004) 1205–1224.

[83] Y. Xu, K.-W. Wong, C.-S. Leung, Generalized RLS approach to the training of neural networks, IEEE Transactions on Neural Networks 17 (2006) 19–34.

[84] E. Lughofer, On-line assurance of interpretability criteria in evolving fuzzy systems-Achievements, new concepts and open issues, Information Sciences 251 (2013) 22–46.

[85] P. Angelov, D. Filev, Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models, in: Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on, IEEE, pp. 1068–1073.

[86] S. Wu, M. J. Er, Dynamic fuzzy neural networks-a novel approach to function approximation, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 30 (2000) 358–364.

[87] S. Wu, M. J. Er, Y. Gao, A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks, IEEE Transactions on Fuzzy Systems 9 (2001) 578–594.

[88] N. Wang, M. J. Er, X. Meng, A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks, Neurocomputing 72 (2009) 3818–3829.

[89] J.-S. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man, and Cybernetics 23 (1993) 665–685.

[90] M. Pratama, J. Lu, G. Zhang, et al., Evolving type-2 fuzzy classifier., IEEE Transactions on Fuzzy Systems 24 (2016) 574–589.

[91] M. Pratama, G. Zhang, M. J. Er, S. Anavatti, An incremental type-2 meta-cognitive extreme learning machine, IEEE Transactions on Cybernetics 47 (2017) 339–353.

- This paper presents at the first time the real-world application of a newly developed algorithm, namely McSIT2RFNN for online identification of rotary wing UAV.
- Real-world experiments were carried out under real flight tests with a real-world quadcopter and different flight conditions. Our algorithm was deployed to perform online identification of UAV dynamic, namely position, velocity and orientation. Furthermore, another numerical validation using artificially injected noise was performed.
- McSIT2RFNN characterises the plug-and-play characteristic where all learning components are integrated in a single dedicated learning modules without the need of pre-and/or post-training steps. It also features the what-to-learn and when-to-learn scenario which makes possible to reduce the number of training samples leading to faster training speed while producing encouraging accuracies.
- The advantage of McSIT2RFNN was experimentally validated using real-world flight data and comparisons with state-of-the art algorithms. It is shown that our algorithm delivered the highest accuracy while imposing the lowest space and memory complexities.