

Data-Driven Type2 Fuzzy Control Using the Distending Function

March 2020

Abstract

The paper presents a novel data-driven type2 fuzzy controller. A new type2 membership function called the type2 Distending function (T2DF) is used. It represents the uncertainties in the measured data using the T2DF parameters. A special algorithm is presented for extracting a few key rules from the training data. These rules cover the whole input space using the T2DFs and Dombi conjunctive operator. The so-called arithmetic-based tpe2 fuzzy controller is designed. The controller design does not include any implication or reduction steps. Furthermore, a rule reduction technique is developed to reduce the number of extracted rules. The computational complexity is low and the controller can be implemented in real time. The efficiency of the algorithms is demonstrated by the altitude control of a quadcopter. The experiment incorporate both the simulation studies and hardware implementations.

Type2 arithmetic-based fuzzy controller, Type2 Distending function, Control and Error surfaces, Parrot mini-drone Mambo.

1 Introduction

Fuzzy theory has found numerous practical applications in the fields of engineering, operational research and statistics [1, 2, 3]. The fuzzy inference engine consists of a fuzzifier, a rule base system, fuzzy operators and defuzzification. These rules describe the dependencies between the input and output variables in the form of IF-THEN statements. In most cases, expert knowledge is not available or it is poorly described. So the exact description of fuzzy rules is not an easy task. However, if the working data of the process is available then a data-driven based design is an attractive option [4, 5, 6]. The data-based identification of a fuzzy model can be divided into two parts, namely qualitative and quantitative identification. Qualitative identification focuses on the number and description of fuzzy rules, while quantitative identification is concerned with the identification of parameter values. These parameters belong to membership functions and fuzzy operators. Soft computing methodologies like evolutionary algorithms, genetic algorithms and swarm optimization [7, 8, 9, 10]

are used for qualitative identification. Neural networks are mostly used in quantitative identification. These leads to the development of Adaptive Neuro-Fuzzy Inference systems (ANFIS) [11]. However, these methods have some limitations related to the identification, rule interpretability and complexity of the controller design. Earlier we presented a data-driven based approach to solve these issues using the Distending function [12].

Type2 fuzzy systems (T2FS) were developed to handle the uncertainty in type-I fuzzy sets [13, 14]. The sources of uncertainties in type-I fuzzy systems are:

1. The words used in the fuzzy rules may convey different meanings to people.
2. The sensing devices may be imprecise and noise appears in the measured signal.
3. The experts do not always agree on the values in the consequents.

Because of these uncertainties, the membership functions are no longer certain i.e. the grade of the membership functions cannot be a crisp value. To overcome this problem, type2 membership functions (T2MF) were introduced. T2MF contains the footprint of uncertainty (FOU) between the upper membership function (UMF) and lower membership functions (LMF). Interval T2FS have been developed to reduce the computational complexity [15]. T2FS has superior properties such as: 1) Better handling of uncertainties [16]; 2) Smooth controller response [17]; 3) Adaptivity [17]; 4) Reduction in the number of fuzzy rules [18]. T2FS have been successfully used in control system design [19], data mining [20] and time series predictions [21]. The design of the interval T2FS consists of:

1. Fuzzification of the inputs using T2MF.
2. Calculation of rule firing strengths.
3. Implication and aggregation to produce rule outputs. These operations produce a type2 fuzzy set.
4. Type reduction to convert type2 fuzzy sets into type1 fuzzy sets.
5. Defuzzification to get a final crisp output value.

The type reduction step is performed using the so-called Karnik Mendel (KM) iterative algorithm [22]. This algorithm defines two switching points for the lower and upper firing strengths. Using these points, the algorithm generates two type-I fuzzy sets. These sets are defuzzified to get a crisp output. This approach has some drawbacks, such as: 1) The choice of T2MFs; 2) Computational complexity of the type reduction step; 3) Difficulty in the optimization process; 4) Controller design complexity. In our previous paper, we offered solutions to these problems by proposing an arithmetic-based type2 fuzzy controller [23].

In this study, we wish to extend our approach to the case where expert knowledge is not available and the training data of the system is uncertain. It leads

to the design of a data-driven T2FS. Quite recently a few techniques have been proposed to tackle this problem [24, 25, 26]. However, these approaches also have following drawbacks:

1. Qualitative identification:
Qualitative identification suffers from the so-called flat structure (curse of dimensionality) problem of the rule-base [20] i.e. if the number of input variables increase, then an exponentially large number of rules are required to accurately model the system. This is due to the fact that the support area of the most frequently used membership function (e.g. triangular, trapezoidal etc) covers a limited area of the input space. To cover the input space completely, a huge number of rules are required. If we have 2 input variables, each with 6 categories, then the number of rules required to cover the whole input space will be 36. Each rule is applicable only within a specific area and its strength is zero outside. If the training data of the system does not fully span the input and output spaces, then this will cause serious problems when modeling the system. If the input falls in these uncovered areas, then the identified rule base does not generate any action. Even if some sort of interpolation technique is applied, the computation complexity will increase [27]. Therefore a global fuzzy model requires a large number of rules and the number of rules depends exponentially on the number of the input variables and this will lead to a huge complexity in data-driven fuzzy models.
2. Quantitative identification:
The computation complexity of the quantitative part of the identified fuzzy model also increases with the number of rules. As the number of rules increases, the number of parameters of the T2MF and operators also grow exponentially. Computing these parameters will then increase the computational cost of the quantitative model.
3. Choice of T2MF:
The choice of T2MF and its systematic connection with the type of uncertainty are not clear. Different type-I membership functions can be combined to generate T2MFs. However, it is not clear which type of membership functions should be used for a particular type of uncertainty.
4. Interpretability:
In most cases, the interpretability of the identified fuzzy rule base is not clear. It is easier to interpret a few rules and get an insight into the working model. However, if the number of rules grows exponentially, then for a given set of input values, it is not possible to predict the response of the model and analyze its performance. The model tends to be more like a black box in these situations.
5. Optimization process:
Although type2 fuzzy logic systems require fewer rules compared to type-

In fuzzy systems, the number of parameters is comparatively large. So optimizing a large number of parameter values is not an easy task.

6. Complexity of the type2 fuzzy controller design:

After the identification of the rule-base, an interval type2 fuzzy controller can be designed using well established techniques [28]. Most of these techniques use the type reduction step. The type reduction step is based on the KM algorithm, which is computationally expensive. Due to its iterative nature, it is ill-suited for on-line applications. There are some alternative solutions which reduce the computation burden, but these are approximations [29]. These techniques also include the implication and aggregation steps. These steps further add to the computation complexity of the type2 fuzzy controller.

Here, we propose solutions to remove some of these drawbacks by extending our previous results [30, 12, 23]. We present a novel approach to model data-driven type2 fuzzy inference systems. The inference scheme is based on a reduced number of identified rules. Using these rules, we present a design of an arithmetic-based interval type2 fuzzy controller. The whole procedure has the following unique features:

1. We use a new type2 membership function called the interval type2 distending function (T2DF) [23]. With a few rules, it can completely cover the whole input space, and this helps overcome the flat structure issue.
2. T2DF has only a few parameters. Most of these parameters are kept fixed and a few are varied during the training process. It reduces the computation burden of quantitative identification.
3. Different types of uncertainties can be modeled using various parameters of T2DF. Therefore, most forms of the uncertainties in fuzzy systems can be represented using T2DF.
4. Our approach identifies a few important fuzzy rules. And we have developed a rule reduction algorithm which can further reduce the number of identified rules and it results in a interpretable model.
5. Because only a few parameters are varied during the design process, the optimization is simple and fast.
6. We use an arithmetic-based interval type2 fuzzy controller [30]. The type reduction, implication and aggregation steps are not involved. Therefore the controller is computationally efficient.

The proposed data-driven type2 fuzzy controller is computationally efficient, more interpretable and it can handle various types of uncertainties.

The rest of the paper is organized as follows. In Section II, we briefly introduce interval T2DF and its properties. In Section III, we explain the proposed controller design approach and rule reduction algorithm. In Section IV, we describe the bench mark system, simulation results, hardware implementation and discuss the results. In Section V, we give a brief summary and conclusion.

2 Interval type2 Distending Function

Zadeh proposed various membership functions and one of them has the following form [31]:

$$\mu(x) = \frac{1}{1 + \left(\frac{x-a}{b}\right)^2} \quad (1)$$

Based on $\mu(x)$, we defined a more general parametric function which models a soft equality and it is called the Distending function (DF). This type of membership function is closely related to the operator systems and in our case (i.e. the Distending function), it is associated with the Dombi operators. DF can be derived from the Kappa function of the Dombi operator [32]. The DF has four parameters, namely ν , ε , λ and c .

It has two forms: 1) Symmetric; 2) Asymmetric. The Symmetric DF (shown in Fig. 1) is symmetric around $x - c$ and it is defined as [30]:

$$\delta_{\varepsilon, \nu}^{(\lambda)}(x - c) = \frac{1}{1 + \frac{1-\nu}{\nu} \left| \frac{x-c}{\varepsilon} \right|^\lambda}, \quad (2)$$

where $\nu \in (0, 1)$, $\varepsilon > 0$, $\lambda \in (1, +\infty)$ and $c \in \mathbb{R}$. $\delta_{\varepsilon, \nu}^{(\lambda)}(x - c)$ is denoted by $\delta_s(x)$.

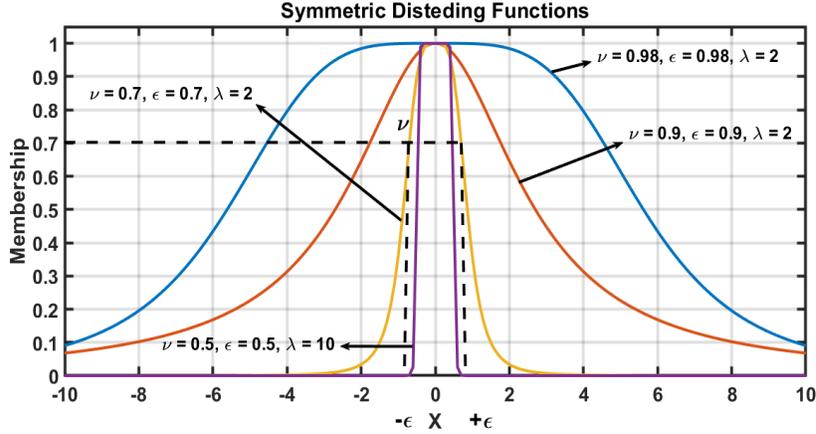


Figure 1: Various shapes of symmetric Distending Functions (here $c = 0$)

The asymmetric DF (shown in Fig. 2) is given by

$$\delta_A(x - c) = \frac{1}{1 + \frac{1-\nu_R}{\nu_R} \left| \frac{x-c}{\varepsilon_R} \right|^{\lambda_R} \frac{1}{1+e^{-\lambda^*(x-c)}} + \frac{1-\nu_L}{\nu_L} \left| \frac{x-c}{\varepsilon_L} \right|^{\lambda_L} \frac{1}{1+e^{\lambda^*(x-c)}}}, \quad (3)$$

where $\nu_R, \nu_L \in (0, 1)$, $\varepsilon_R, \varepsilon_L > 0$, $\lambda_L, \lambda_R \in (1, +\infty)$, $c \in \mathbb{R}$ and $\lambda^* \in (1, +\infty)$. Here, c is the centre point i.e. $\delta_A(c) = 1$.

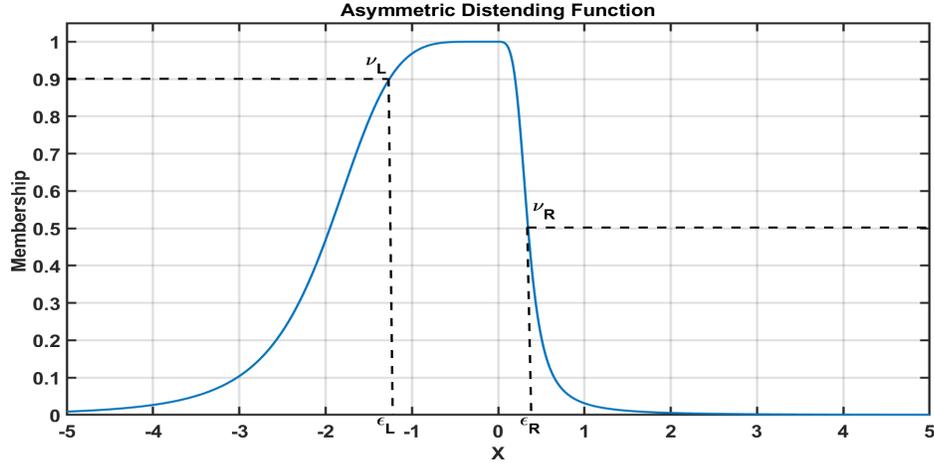


Figure 2: The asymmetric Distending Function ($\nu_L = 0.5$, $\varepsilon_L = 0.5$, $\lambda_L = 5$, $\nu_R = 0.8$, $\varepsilon_R = 0.7$, $\lambda_R = 5$, $\lambda = 5$, $c = 0$)

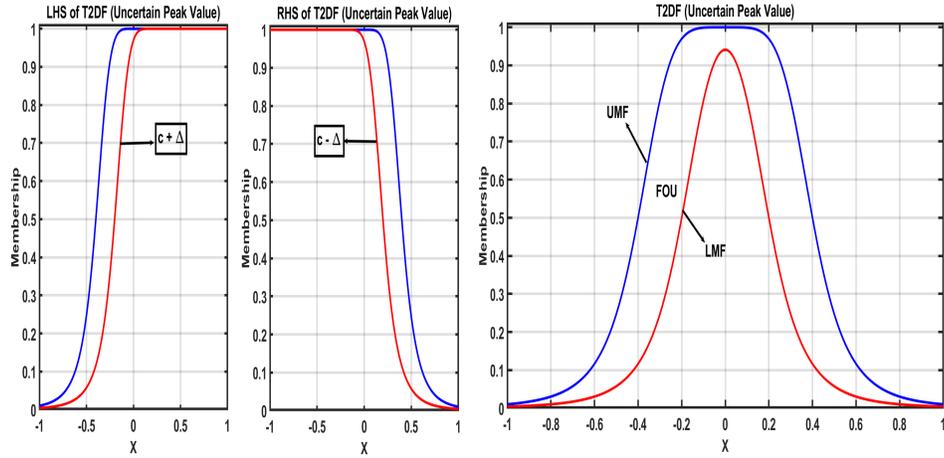


Figure 3: Uncertain peak values T2DF with the footprint of uncertainty (FOU)

2.1 Construction of the interval T2DF

The values of the DF parameters ($\nu, \varepsilon, \lambda, c$) may be uncertain. As a result, these parameters can take various values around their nominal values, within the uncertainty bound (Δ). By varying the parameter values within Δ , various DFs are obtained. The DF with the highest grade values is called the upper membership function (UMF) and that with lowest values is called the lower membership function (LMF). The UMF, LMF and various DFs in between can be combined to form an interval T2DF [23]. If the peak value of the DF becomes

uncertain, then it can be represented using the interval T2DF with an uncertain 'c' value, as shown in Fig. 3.

Various T2DFs belonging to the same fuzzy variable can be combined together to form a single T2DF. The support of the resultant T2DF will be approximately same as the combined support of the individual T2DFs. The UMF of the T2DF consists of the LHS and RHS (the same is true for the LMF). The LHS and RHS are given as [23]

$$\bar{\delta}_L^2(x-c) = \frac{1}{1 + \frac{1-\nu}{\nu} \left| \frac{x-c}{\varepsilon} \right|^\lambda \frac{1}{1+e^{\lambda^*(x-c)}}}, \quad (4)$$

$$\bar{\delta}_R^2(x-c) = \frac{1}{1 + \frac{1-\nu}{\nu} \left| \frac{x-c}{\varepsilon} \right|^\lambda \frac{1}{1+e^{(-\lambda^*(x-c))}}}. \quad (5)$$

The LHS and RHS of the UMF and LMF can be combined using the Dombi conjunctive operator to get a single T2DF. Consider two T2DFs δ_1^2 and δ_2^2 . The LHS of δ_1^2 and RHS of δ_2^2 can be combined using the Dombi conjunctive operator. This produces a resultant T2DF δ_{result}^2 , as shown in Fig. 4. Combining various T2DF helps to reduce the number of fuzzy rules. This leads to a decrease in the computational complexity of the identified fuzzy model.

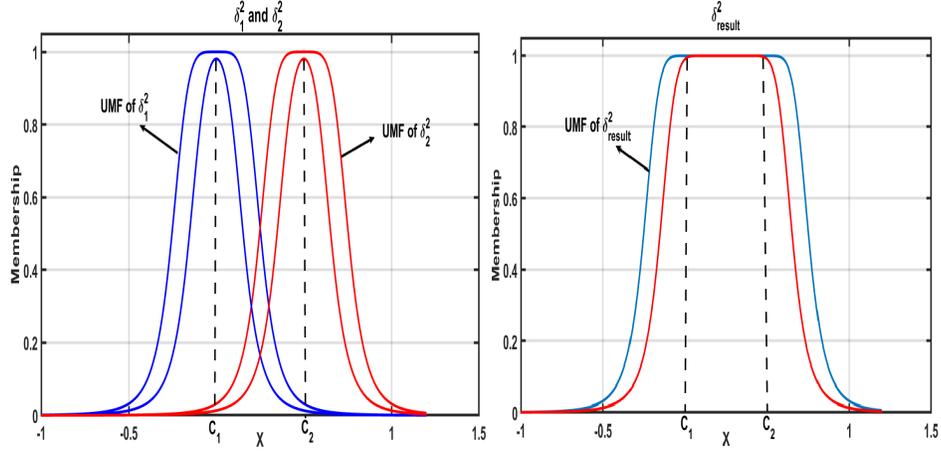


Figure 4: Combining two T2DF (δ_1^2 and δ_2^2) to get a single T2DF (δ_{result}^2)

2.2 Distending Function in a Higher Dimension

Consider n different T2DFs in n different dimensions given by $\delta_{1(\varepsilon_1, \nu_1)}^{2(\lambda_1)}(x_1 - c_1)$, $\delta_{2(\varepsilon_2, \nu_2)}^{2(\lambda_2)}(x_2 - c_2) \dots, \delta_{n(\varepsilon_n, \nu_n)}^{2(\lambda_n)}(x_n - c_n)$. If we apply the Dombi conjunctive operator on these n T2DFs, then the result will also be a T2DF $\delta^2(x_1, x_2, \dots, x_n)$

in n dimensions (shown in Fig. 5). And

$$\bar{\delta}^2(x_1, x_2, \dots, x_n) = \frac{1}{1 + \sum_{i=1}^n \frac{1-\nu_i}{\nu_i} \left| \frac{x_i - a_i}{\epsilon_i} \right|^{\lambda_i}}, \quad (6)$$

$$\underline{\delta}^2(x_1, x_2, \dots, x_n) = \frac{1}{1 + \sum_{i=1}^n \frac{1-\nu_i}{\nu_i} \left(\left| \frac{x_i - (c_i - \Delta)}{\epsilon_i} \right|^{\lambda_i} \sigma_{Li} + \left| \frac{x_i - (c_i + \Delta)}{\epsilon_i} \right|^{\lambda_i} \sigma_{Ri} \right)}, \quad (7)$$

$$\text{where } \sigma_{Li} = \frac{1}{1 + e^{-\lambda_i(x_i - (c_i - \Delta))}} \quad \text{and} \quad \sigma_{Ri} = \frac{1}{1 + e^{\lambda_i(x_i - (c_i + \Delta))}}.$$

Here Δ is the upper bound on the uncertainty in the c value, $\bar{\delta}^2(x_1, x_2, \dots, x_n)$ is the UMF and $\underline{\delta}^2(x_1, x_2, \dots, x_n)$ is the LMF of $\delta^2(x_1, x_2, \dots, x_n)$. (Please see the Appendix for the proof.)

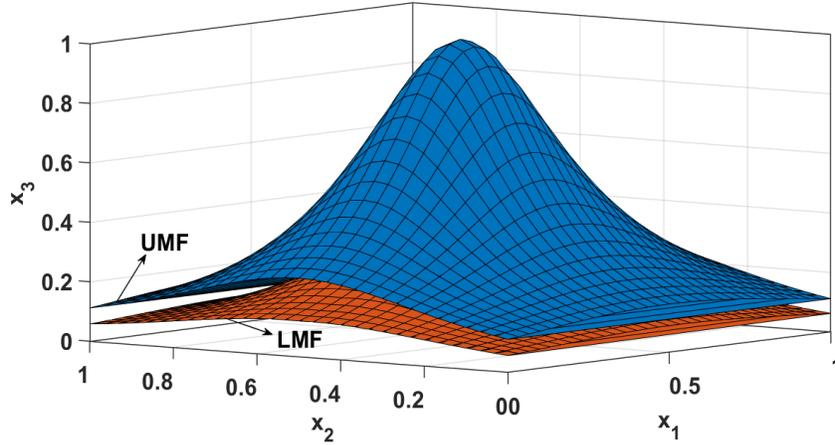


Figure 5: The T2DF in the x3 dimension

3 Data-driven fuzzy controller

The proposed algorithm is motivated by our previous study, where fuzzy arithmetic operations were used to design a fuzzy controller [30]. Here, we suppose that expert knowledge in the form of linguistic rules is not available. However the training (working) data of the process is available. Using the training data, a rule base for a multi-input multi-output (MIMO) system can be written as:

$$\begin{aligned} &\text{if } x_1 \text{ is } U_1^i \text{ and } \dots \text{ and } x_n \text{ is } U_n^i \\ &\text{then } y_1 \text{ is } V_1^i ; \dots ; y_m \text{ is } V_m^i, \end{aligned} \quad (8)$$

where x_1, x_2, \dots, x_n are the input variables and y_1, y_2, \dots, y_m are the m output variables, and the corresponding fuzzy subsets are U_1, U_2, \dots, U_n and V_1, V_2, \dots, V_m , respectively. The index i represents the rule number and there are l fuzzy rules. A MIMO system given by Eq. (8) with m independent outputs can always be replaced by m multi-input single output (MISO) systems of the form

$$\text{if } x_1 \text{ is } U_1^i \text{ and } \dots \text{ and } x_n \text{ is } U_n^i \text{ then } y_s \text{ is } V_s^i, \quad (9)$$

where $s = 1, \dots, m$ are the m outputs. For simplicity, we will consider the case where $s = 1$ and we propose a methodology for generating a crisp control signal using the input and output training data. The methodology can be generalized to m independent outputs.

Let us now assume that the input and output databases have the following form:

$$U = \begin{bmatrix} a_1^1 & a_2^1 & \dots & a_n^1 \\ a_1^2 & a_2^2 & \dots & a_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ a_1^l & a_2^l & \dots & a_n^l \end{bmatrix}, \quad V = \begin{bmatrix} b^1 \\ b^2 \\ \vdots \\ b^l \end{bmatrix}, \quad (10)$$

where U and V contains the l data points of each input and output variable. a_1, a_2, \dots, a_n are the data points belonging to the input fuzzy subsets U_1, U_2, \dots, U_n respectively and b_1 is included in the output fuzzy subset V . Each column of the U matrix corresponds to a unique feature (input variables) of the process. Therefore the U matrix forms an n dimensional input feature space. Each column of the training matrix U is normalized by transforming it into the $[0, 1]$ interval. As a result, the features values are comparable on the same scale.

Fuzzy rule consists of an antecedent and a consequent part. Here, the antecedent part contains a row of U and the consequent part is an element of V . The antecedent part of the i th fuzzy rule is given by the following relation:

$$\mathcal{L}(\delta_1^2(x_1)^i, \delta_2^2(x_2)^i, \dots, \delta_n^2(x_n)^i), \quad (11)$$

where \mathcal{L} is the fuzzy logical expression and it may contain AND ($x_1 \in A_1$ and $x_2 \in A_2$), OR ($x_1 \in A_1$ or $x_2 \in A_2$) and NOT ($x_1 \notin A_1$) operators. Here, we can use a very general class of the fuzzy operators [10]:

$$D_\gamma(x) = \frac{1}{1 + \left(\frac{1}{\gamma} \left(\prod_{i=1}^n \left(1 + \gamma \left(\frac{1 - \delta^2(x_i)}{\delta^2(x_i)} \right)^\alpha \right) - 1 \right) \right)^{\frac{1}{\alpha}}} \quad (12)$$

This operator describes a wide range of fuzzy operators e.g. min/max, Einstein, Hamacher, product and drastic. But here we will use the Dombi conjunctive/disjunctive operator.

In our approach the fuzzy rules will be based on sample values in the U and V

matrices. Therefore, a few rows from the data base matrix U are selected. These can be selected randomly, but from a practical point of view it is beneficial to choose those rows which contain the extremum (around 0 and 1) and average (around 0.5) values of the input variables. These rows and the corresponding elements in the V matrix are used to construct the rule base. It is called the boundary-value rule base (R_b) because it mostly contains those values of the inputs that lie on the boundary of the control domain.

In our procedure, two different control surfaces are constructed. These are called the estimated and the fuzzy control surfaces. The estimated control surface is constructed directly from the database (Eq. (10)). For missing data values, linear interpolation is used. The estimated control surface is denoted by G . The fuzzy control surface is generated from R_b and it is denoted by G^* . These two surface are then used to create a third surface called the error surface E . Next, we will describe the procedure used to construct the surfaces G^* and E using R_b .

3.1 Construction of Fuzzy Control Surface G^* and Error Surface E

Each selected row from the database matrix U corresponds to a single rule. It is a row vector and it consists of unique values of all the input variables (features). We will construct T2DFs for all input variables. The parameter c (peak value coordinate) of T2DF is given and it is equal to the value of the corresponding input variable. The value of λ can be chosen between 1 to ∞ , but for practical applications $\lambda = 3$ serves as a good initial value. The value of ε depends upon the number of rules in R_b ($\varepsilon = \frac{1}{\text{no. of rules in } R_b}$) and it ensures that the whole input space is covered. The input variables are usually measured using the feedback sensors. The Δ value of each sensor depends on the tolerance intervals of the corresponding sensor. All the Δ values are transformed into the $[0, 1]$ interval to make these compatible with the values of the input variables. T2DFs have a long tail. Consequently each T2DF influences the other existing T2DFs. The ν value of each T2DFs will be calculated based on the principle of minimum influence on all the other T2DFs. This influence can never be zero, but it can be decreased by a factor k . For less influence, a large value of k should be chosen. However from practical point of view, a value of 10 is sufficient. It means that the influence will decrease 10 folds. The influence of the i th T2DF at the peak value of the j th T2DF is given by

$$\frac{1}{1 + \frac{1-\nu}{\nu} \left(\left| \frac{x_{i1}-x_{j1}}{\varepsilon} \right|^\lambda + \dots + \left| \frac{x_{in}-x_{jn}}{\varepsilon} \right|^\lambda \right)} = \frac{1}{k}, \quad (13)$$

where x_{i1}, \dots, x_{in} are the n coordinates of the peak value of the i th T2DF and x_{j1}, \dots, x_{jn} are the coordinates of the peak value of the j th T2DF. This equation can be used for single dimensional T2DFs as well as n dimensional

T2DFs. Then the required value of ν can be calculated using

$$\nu = \frac{1}{1 + \frac{k-1}{d}}, \quad (14)$$

where $d = \left(\left| \frac{x_{i1} - x_{j1}}{\epsilon} \right|^\lambda + \dots + \left| \frac{x_{in} - x_{jn}}{\epsilon} \right|^\lambda \right)$. Obviously it is a distance measure if $\epsilon = 1$ and $\lambda = 2$.

In the antecedent part of *ith* rule, there are n T2DFs corresponding to n input variables. Each rule is evaluated using the Dombi conjunctive/disjunctive operator. By applying the Dombi conjunctive/disjunctive operator over the n input T2DFs, we get a single T2DF. This is called the output T2DF. The UMF and LMF of this output T2DF are given by Eq. (6) and Eq. (7), respectively. Here, l output T2DFs will be generated from the l rules. All these output T2DFs are superimposed in the input space to generate a fuzzy control surface G^* .

An error surface E is defined as the difference between the estimated control surface G and the fuzzy control surface G^* . That is,

$$E(x_1, \dots, x_n) = G(x_1, \dots, x_n) - G^*(x_1, \dots, x_n). \quad (15)$$

3.2 Extending the rule base

We shall decrease the magnitude of E below a chosen threshold τ_E ($|E| < \tau_E$). This is achieved by an iterative procedure of adding new rules to R_b . To add a new fuzzy rule, the coordinates of the maximum value on E are located. The corresponding row in the database containing these coordinates is selected. This row is then added to R_b as a new rule. This rule is evaluated to generate an output T2DF. The ν value of this output T2DF is calculated using Eq. (14). This T2DF is superimposed in G^* . This will modify the surface G^* in such a way that the magnitude of the maximum error on the surface E at these coordinates will decrease. This process is repeated in an iterative manner until the error surface E is within the tolerance limit τ_E . For a very small value of τ_E , a large number of rules have to be extracted from the training data and vice versa. Therefore there is a compromise between the value of the threshold τ_E and the number of rules in R_b .

If the number of rules in R_b is large, then some of the rules can be merged to reduce the computational complexity. This is achieved using a reduction procedure.

3.3 Reducing the rule base

Here, we describe a heuristic approach used to decrease the number of rules in R_b . Rules reduction will lead to a lower computational cost and higher interpretability. Various output T2DFs which are close to each other in the input space can be combined to get a single T2DF (as shown in Fig. 4). The procedure is explained as follows:

One of the input variable is selected and we call it a principle feature. In a

control system, the error measure of the control variable is usually chosen as the principle feature. To ensure that the rule reduction procedure does not change the fuzzy control surface significantly, the input space is divided into two half spaces. The space where the principle feature value is less than 0.5 lies in the first half and the rest of the space lies in the second second half. Within each half, the output T2DFs are segregated into different groups. If the Euclidean distance between the peak value coordinates of various output T2DFs is less than a predefined distance D then these T2DFs are placed in the same group, where for each half:

$$D = \frac{\text{Sum of euclidean distances b/w peak value coordinates of T2DFs}}{\text{Total no. of T2DFs in the same half}}. \quad (16)$$

Each output T2DF is obtained by applying a unique rule in R_b . The output T2DFs in the same group are combined together to produce a single T2DF. Consequently the rules associated with all these output T2DFs are eliminated and replaced by a single new rule. Therefore the number of rules in R_b decreases. Now it is called a reduced rule base R_r . Using R_r , a new fuzzy control surface is constructed and it is denoted by G_r^* . Then a reduced error surface (E_r) is obtained using

$$E_r(x_1, \dots, x_n) = G(x_1, \dots, x_n) - G_r^*(x_1, \dots, x_n). \quad (17)$$

This procedure is performed in an iterative way as long as $E_r(x_1, \dots, x_n)$ is within a chosen threshold τ_R . If the threshold value τ_R is high then a large number of rules can be eliminated to get a much simpler and interpretable model. However, the accuracy of such an identified fuzzy model decreases. So the τ_R value should be chosen based on a compromise between model accuracy and interpretability. A high τ_R value leads to more interpretability but less accuracy and vice versa.

3.4 Designing the Interval type2 fuzzy controller

The reduced rule base R_r is used to design an arithmetic-based interval type2 fuzzy controller [23]. Each rule in R_r has two parts (antecedent and consequent). We evaluate these two parts separately to generate a crisp control signal. For a specific values of input variables, the antecedent part is evaluated (Eq. 11) and it results in an interval $[\hat{v}_i(\underline{x}^*) \ \bar{v}_i(\underline{x}^*)]$

$$\begin{aligned} \mathcal{L}(\bar{\delta}_1^2(x_1^*)^i, \bar{\delta}_2^2(x_2^*)^i, \dots, \bar{\delta}_n^2(x_n^*)^i) &= \bar{v}_i(\underline{x}^*), \\ \mathcal{L}(\underline{\delta}_1^2(x_1^*)^i, \underline{\delta}_2^2(x_2^*)^i, \dots, \underline{\delta}_n^2(x_n^*)^i) &= \hat{v}_i(\underline{x}^*). \end{aligned}$$

Here, $\hat{v}_i(\underline{x}^*)$ is the lower strength and \bar{v}_i is the upper strength of the i th rule. $\underline{\delta}_n^2(x_n^*)^i$ is the LMF and $\bar{\delta}_n^2(x_n^*)^i$ is the UMF of the n th T2DF. The rule strengths

are normalized to get the lower and upper firing strengths ($v_i(\underline{x}^*)$, $\bar{v}_i(\underline{x}^*)$):

$$v_i(\underline{x}^*) = \frac{\hat{v}_i(\underline{x}^*)}{\sum_{i=1}^k \hat{v}_i(\underline{x}^*)}, \quad \bar{v}_i(\underline{x}^*) = \frac{\hat{\bar{v}}_i(\underline{x}^*)}{\sum_{i=1}^k \hat{\bar{v}}_i(\underline{x}^*)}, \quad (18)$$

$$\text{where} \quad \sum_{i=1}^k v_i(\underline{x}^*) = 1, \quad \sum_{i=1}^k \bar{v}_i(\underline{x}^*) = 1,$$

and k is the total number of rules in R_r .

The consequent part of each rule in R_r is a single numeric value in the data matrix V . Let b^1, \dots, b^k be the consequent values, $\bar{v}_1^*, \dots, \bar{v}_k^*$ be the upper firing strengths and v_1^*, \dots, v_k^* be the lower firing strengths of the k fuzzy rules in R_r . Then the crisp output control U_{crisp} is generated by

$$U_{crisp} = \frac{\underline{c}_a + \bar{c}_a}{2}, \quad (19)$$

$$\text{where } \bar{c}_a = \sum_{i=1}^k \bar{v}_i(\bar{x}^*) b^i, \text{ and } \underline{c}_a = \sum_{i=1}^k v_i(\underline{x}^*) b^i.$$

The whole procedure is summarized in Algorithm 1.

4 Benchmark System, Simulations Results, Hardware implementation and discussion

The effectiveness of the proposed technique is demonstrated by designing an altitude control system for a quadcopter (Parrot mini-drone). Using the training data of the quadcopter, a data-driven type2 controller is designed based on the proposed approach. Simulation studies are carried in Matlab Simulink to control the altitude of the quadcopter in the presence of noisy sensor measurements. Later the designed controller is deployed on the mini-drone hardware to check the real-time performance performance.

4.1 Quadcopter Model

Parrot mini-drone Mambo is used in this study. The Matlab Simulink Aerospace block set provides the simulation model of this quadcopter [33]. The simulation consists of the airframe model, sensors model, environment model and flight controller. The airframe model is schematically shown in Fig. 6. It consists of axis parameters (rotational (ϕ, θ, ψ) and translational (x, y, z)), mass, torques, and rotors. The environment model describes the effects of external factors on the quadcopter. It consists of atmosphere and gravity models. The sensor model includes three sensors, namely 1) Sonar for altitude measurement; 2) A camera for optical flow estimation; 3) IMUs to measure the linear and rotational motions. The flight control system contains the roll ϕ , pitch θ , yaw ψ and altitude z controllers. The mathematical model of the system is given as

$$\dot{x} = F(x, u) + N, \quad (20)$$

Algorithm 1: Algorithm for Data Riven Based Type2 Fuzzy Controller

- Step 1:** Obtain the input and output databases of the process (Eq. (10)).
- Step 2:** Transform each column of U into the $[0, 1]$ interval.
- Step 3:** Construct the estimated control surface G .
- Step 4:** Generate a boundary value rule base R_b from the input and output databases.
- Step 5:** Assign a T2DF to each input point in the antecedent part of the rule base R_b . Choose $k = 10$ and calculate ν using Eq. (13) and Eq. (14).
- Step 6:** Calculate the output T2DF for each rule using Eq. (6) and Eq. (7).
- Step 7:** Generate fuzzy control surfaces G^* from all the output T2DFs.
- Step 8:** Construct the error surface E using Eq. (15). If E is within the threshold τ_E , then go to step 10.
- Step 9:** Find the maximum value coordinates on E . Add a new rule in R_b corresponding to these coordinates in the databases. Go to Step 6.
- Step 10:** Calculate the Euclidean distance D and create groups of output T2DFs.
- Step 11:** Combine the T2DFs in each group using the Dombi conjunctive operator.
- Step 12:** Construct the reduced error surface E_r using Eq. (17). If E_r is within the threshold τ_R , then go to step 10.
- Step 13:** Generate the crisp control signal using Eq. (19).

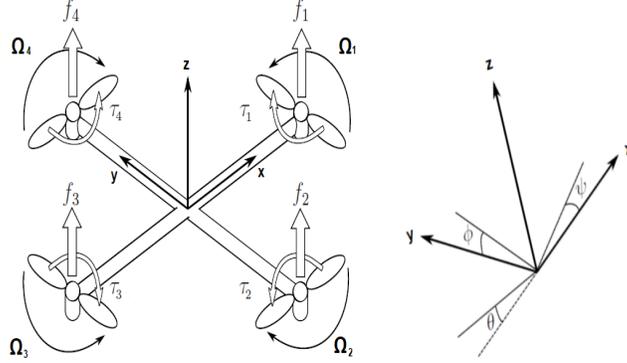


Figure 6: Airframe model of the quadcopter structure [34]

where

$$\begin{aligned}
 x &= [x \ y \ z \ \phi \ \theta \ \psi]^T, \\
 u &= [u_1 \ u_2 \ u_3 \ u_4]^T, \\
 N &= [n_1 \ n_2 \ n_3 \ n_4 \ n_5 \ n_6]^T.
 \end{aligned}$$

Here x is the state vector consisting of translational and rotational components, N contains the external disturbances affecting the system states and u represents the model inputs. Let $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ are the angular speeds of the four rotors of the quadcopter. Then

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_r \end{bmatrix} = \begin{bmatrix} b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ b(-\Omega_2^2 + \Omega_4^2) \\ b(\Omega_1^2 - \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{bmatrix}.$$

Here, u_2, u_3, u_4 controls the roll, pitch and yaw angles. u_1 is the total thrust input and controls the altitude z of the quadcopter. b is the thrust coefficient, d is the drag coefficient and u_r is the residual angular speed.

4.2 Designing the data driven altitude controller

The objective is to control the altitude z by generating an appropriate total thrust u_1 . The thrust u_1 depends on the height (sonar measurement) and rate of change of the height of the quadcopter. The input and output databases are generated for an altitude control scenario using a PD control. The rule base R_b is constructed from the normalized data using Algorithm 1. Here, 26 rules are identified in R_b . The T2DFs corresponding to R_b are shown in Fig. 7. The

upper and lower control surfaces constructed using R_b are shown in Fig. 8. The threshold τ_E is set at .15. The final error surface E for R_b is constructed using Eq. (15) (Fig. 9). Then rule reduction is applied to get a reduced rule base R_r . The number of rules is reduced to 17. The upper and lower control surfaces constructed using R_r are shown in Fig. 11. Fig. 10 shows the T2DFs contained in R_r , and the error surface E_r is shown in Fig. 12. An arithmetic based controller is designed using R_r (Eq. 18 and Eq. 19). The controller is then used to regulate the altitude of the quadcopter. Fig. 13 shows the surface plot of the arithmetic based controller. The quadcopter is programmed to takeoff and reach to an altitude of 1m, then rise to altitude of 1.5m and finally descend to 1m. Fig. 14 shows the altitude response of the controller quadcopter during this simulation study.

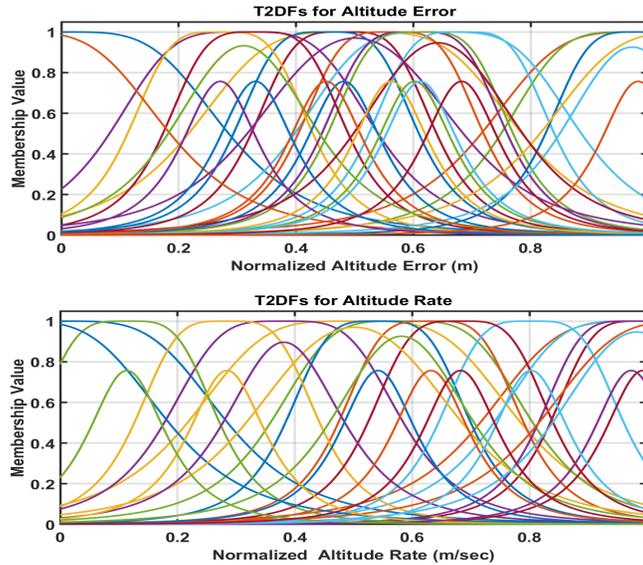


Figure 7: T2DFs for the Normalized Inputs using rule base R_b (Extended rule base)

4.3 Hardware Implementation

Matlab provides a support package for parrot quadcopter drones (Mambo FPV and Bebop2). It connects with the quadcopter over bluetooth/Wifi and can send control commands. Matlab Simulink includes the simulation model of the quadcopter. The model contains the algorithm for the flight control system (FCS). FCS algorithm implements roll, pitch, yaw and altitude controllers. The support package generates the C code of FCS and deploys it in the quadcopter. The algorithm can access the on-board sensors such as the accelerometer, gyroscopes, camera and sonar. The flight data (altitude, images etc) is

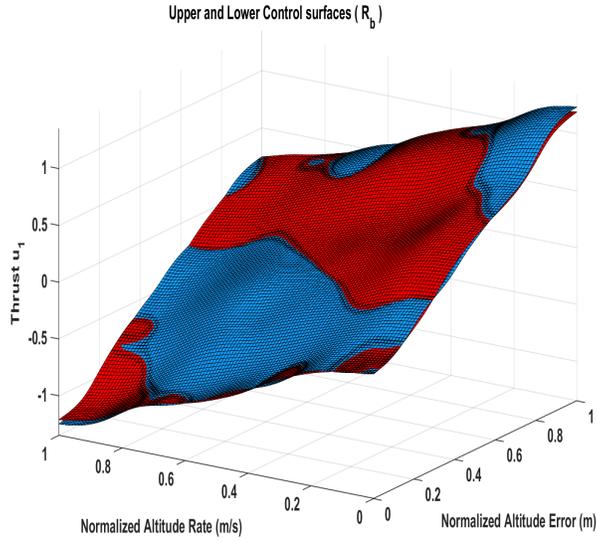


Figure 8: Upper (blue) and Lower (Red) control surfaces using rule base R_b

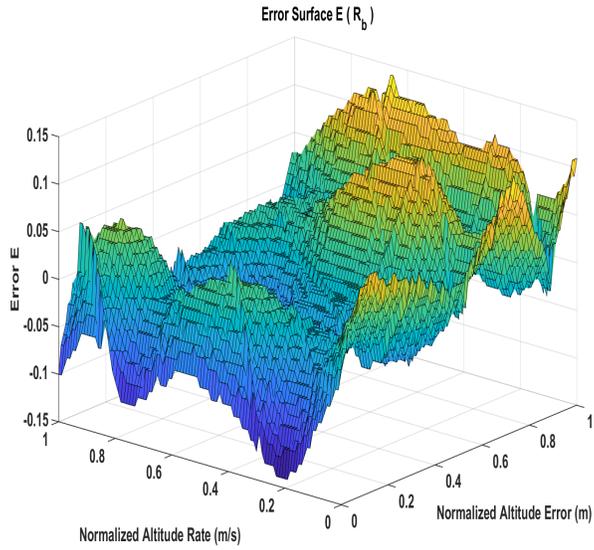


Figure 9: Error surfaces E using rule base R_b

saved in on-board storage and can be retrieved from the quadcopter at the end of the flight. The altitude controller in the FCS was replaced with the proposed controller. The C code of the FCS was generated and deployed in the quad-

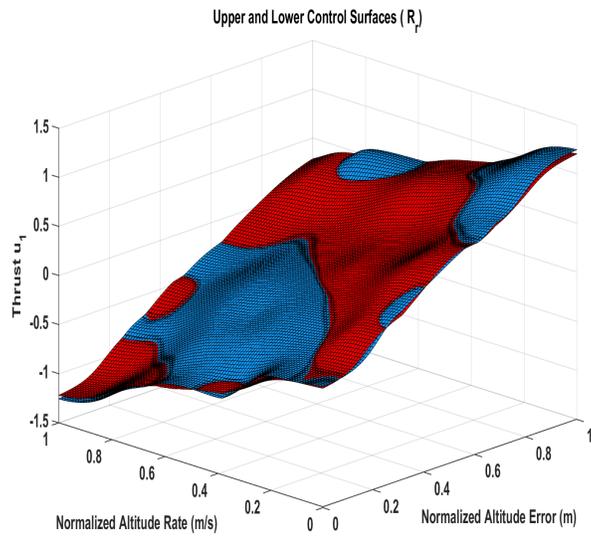


Figure 10: Upper (blue) and Lower (Red) control surfaces using rule base R_r

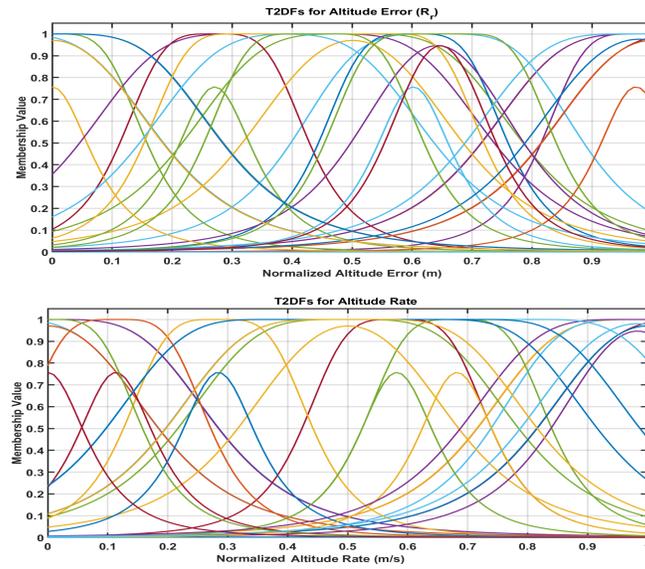


Figure 11: T2DFs for the Normalized Inputs using rule base R_r (Reduced rule base)

copter via bluetooth. A fixed flight trajectory corresponding to the simulation scenario was also coded. The uncertainty in the controller input (altitude) data

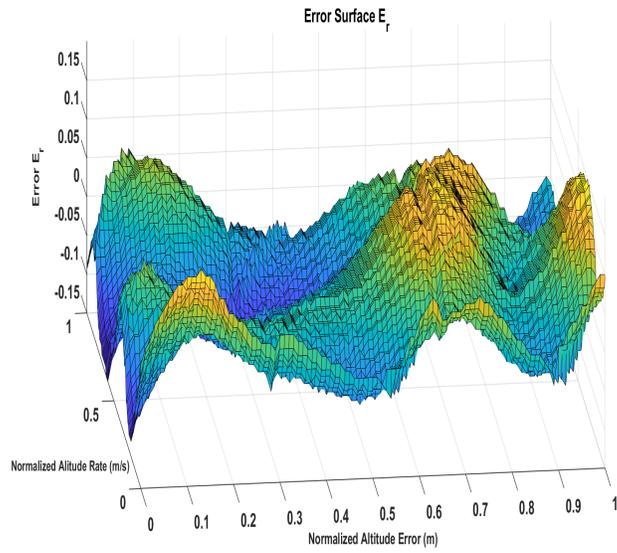


Figure 12: Error surfaces E_r using rule base R_r .

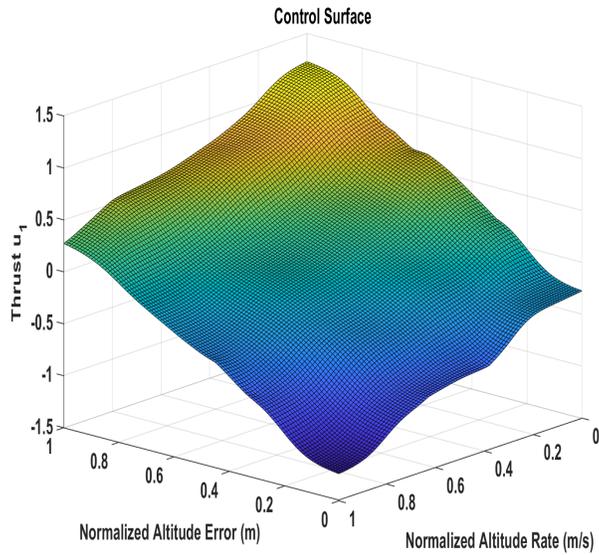


Figure 13: Control surface of the Arithmetic based controller

was created by adding white noise to the sonar measurements. The altitude measurements together with the added noise signal are shown in Fig. 15. The noisy data was the input to the proposed type2 controller in the FCS. The drone flight

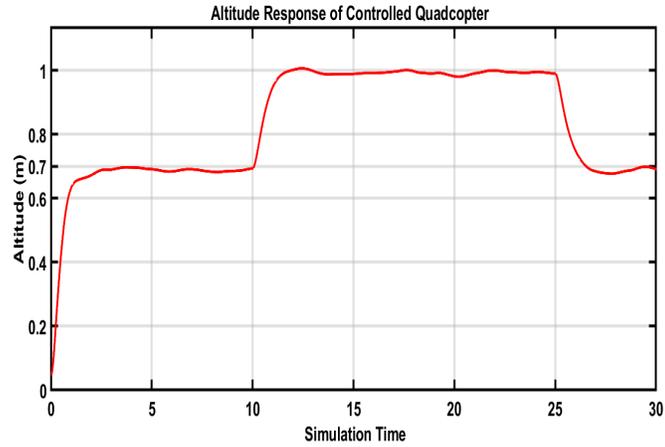


Figure 14: Simulated altitude response of Quadcopter (Matlab Simulink)

was tested in a protected environment. The data recorded during the flight was retrieved at the end (Flight data: <https://github.com/Abrarlaghari/Mambo-Quadcopter-Simulink.git>). Fig. 16 shows the altitude measurements during the test flight. This shows that the proposed controller successfully followed the desired trajectory, even in the presence of the noisy sensor data.

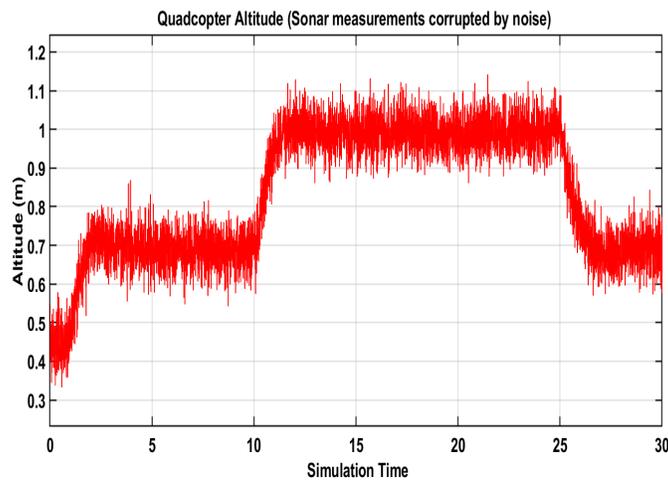


Figure 15: Noisy altitude measurements by on-board sonar sensor

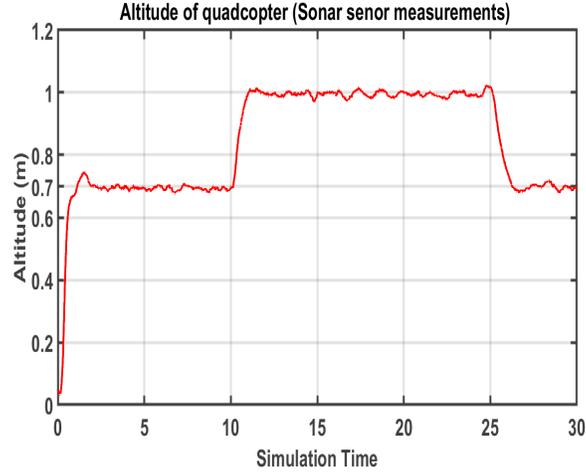


Figure 16: Altitude trajectory followed by Mambo quadcopter

4.4 Result discussions

Control surface (Fig. 8 and Fig. 10) with smooth transitions can be obtained using T2DF and the Dombi conjunctive operator. The whole input space is covered using the rule base R_b/R_r . The λ and ϵ parameters are kept fixed, the c parameter is determined directly from the data and ν is calculated using Eq. (14). This reduced the number of parameter calculations per T2DF. There is a slight difference between the surfaces obtained using R_b and R_r . However, the number of rules in R_r are two-thirds compared to R_b . This demonstrates the efficiency of the rules reduction procedure. The simulation results in Fig. 14 tell us that the proposed controller is able to regulate the altitude of the quadcopter very precisely. Hardware implementations showed very promising results for the real-time control applications. The controller performed the altitude control function in the presence of the uncertain (noisy) measurement data. This demonstrates the effectiveness of the proposed data-driven design based on T2DF.

5 Conclusion

Here, we presented a data-driven interval type2 fuzzy controller for real-time control applications. A new type2 membership function called the type2 Distending function (T2DF) is used in conjunction with Dombi operator. It can model various type of uncertainties using its parameters. The whole input space can be covered just using a few rules. The rules are identified from the input and output databases directly. Then we presented a rule reduction procedure. It is based on combining the T2DFs in the close vicinity and it reduces the number of rules to two-thirds. A procedure is proposed to design arithmetic based

interval type2 fuzzy controller using the reduced rule base. The controller design does not include the implication and type reduction steps. This greatly reduces the computational complexity and paves the way for the real-time implementation of the proposed design. The effectiveness of the whole procedure was demonstrated by designing an altitude controller for Parrot Mambo quadcopter. Simulations carried out in Matlab Simulink showed promising results. Later the designed controller was deployed and tested in the flight control system on the quadcopter hardware. Real-time hardware implementations gave the same results as were obtained in the simulations. The controller regulated the altitude of the quadcopter even in the presence of noisy (uncertain) sensor measurements. This robustness to the noisy data is due the use of T2DFs. The main achievement of the study was a novel robust type2 fuzzy controller that can be derived directly from the data of the system. The controller is suitable for real-time control applications due to its low computation complexity and design simplicity.

References

- [1] R. R. Yager, L. A. Zadeh, An introduction to fuzzy logic applications in intelligent systems, Vol. 165, Springer Science & Business Media, 2012.
- [2] L. Yu, Y.-Q. Zhang, Evolutionary fuzzy neural networks for hybrid financial prediction, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 35 (2) (2005) 244–249.
- [3] M. Mahfouf, M. F. Abbod, D. A. Linkens, A survey of fuzzy logic monitoring and control utilisation in medicine, *Artificial intelligence in medicine* 21 (1-3) (2001) 27–42.
- [4] C. Li, J. Zhou, L. Chang, Z. Huang, Y. Zhang, T-s fuzzy model identification based on a novel hyperplane-shaped membership function, *IEEE Transactions on Fuzzy Systems* 25 (5) (2017) 1364–1370.
- [5] 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.
- [6] S.-H. Tsai, Y.-W. Chen, A novel identification method for takagi-sugeno fuzzy model, *Fuzzy Sets and Systems* 338 (2018) 117–135.
- [7] L. dos Santos Coelho, B. M. Herrera, Fuzzy identification based on a chaotic particle swarm optimization approach applied to a nonlinear yo-yo motion system, *IEEE Transactions on Industrial Electronics* 54 (6) (2007) 3234–3245.
- [8] R. Zhang, J. Tao, A nonlinear fuzzy neural network modeling approach using an improved genetic algorithm, *IEEE Transactions on Industrial Electronics* 65 (7) (2018) 5882–5892.

- [9] H. Hellendoorn, D. Driankov, *Fuzzy model identification: selected approaches*, Springer Science & Business Media, 2012.
- [10] R.-E. Precup, H. Hellendoorn, A survey on industrial applications of fuzzy control, *Computers in Industry* 62 (3) (2011) 213–226.
- [11] A. Piegat, *Fuzzy modeling and control*, Vol. 69, Physica, 2013.
- [12] J. Dombi, A. Hussain, Data-driven arithmetic fuzzy control using the distending function, in: *International Conference on Human Interaction and Emerging Technologies*, Springer, 2019, pp. 215–221.
- [13] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning—i, *Information sciences* 8 (3) (1975) 199–249.
- [14] N. N. Karnik, J. M. Mendel, Q. Liang, Type-2 fuzzy logic systems, *IEEE transactions on Fuzzy Systems* 7 (6) (1999) 643–658.
- [15] Q. Liang, J. M. Mendel, Interval type-2 fuzzy logic systems: theory and design, *IEEE Transactions on Fuzzy systems* 8 (5) (2000) 535–550.
- [16] J. M. Mendel, Computing with words: Zadeh, turing, popper and occam, *IEEE computational intelligence magazine* 2 (4) (2007) 10–17.
- [17] D. Wu, W. W. Tan, Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers, *Engineering Applications of Artificial Intelligence* 19 (8) (2006) 829–841.
- [18] H. Hagrass, Type-2 flcs: A new generation of fuzzy controllers, *IEEE Computational Intelligence Magazine* 2 (1) (2007) 30–43.
- [19] O. Castillo, P. Melin, A review on interval type-2 fuzzy logic applications in intelligent control, *Information Sciences* 279 (2014) 615–631.
- [20] A. Niewiadomski, A type-2 fuzzy approach to linguistic summarization of data, *IEEE Transactions on Fuzzy Systems* 16 (1) (2008) 198–212.
- [21] F. Gaxiola, P. Melin, F. Valdez, O. Castillo, Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction, *Information Sciences* 260 (2014) 1–14.
- [22] J. Mendel, *Uncertain rule-based fuzzy logic systems: Introduction and new directions*. Ed. USA: Prentice Hall (2000) 25–200.
- [23] J. Dombi, A. Hussain, Interval type-2 fuzzy control using distending function, in: *Fuzzy Systems and Data Mining V: Proceedings of FSDM 2019*, IOS Press, 2019, pp. 705–714. doi:10.3233/FAIA190240.
- [24] H. Hassani, J. Zarei, Interval type-2 fuzzy logic controller design for the speed control of dc motors, *Systems Science & Control Engineering* 3 (1) (2015) 266–273. doi:10.1080/21642583.2015.1013644.

- [25] J. Zheng, W. Du, I. Nascu, Y. Zhu, W. Zhong, An interval type-2 fuzzy controller based on data-driven parameters extraction for cement calciner process, *IEEE Access* 8 (2020) 61775–61789.
- [26] S. Bhattacharyya, D. Basu, A. Konar, D. Tibarewala, Interval type-2 fuzzy logic based multiclass anfis algorithm for real-time eeg based movement control of a robot arm, *Robotics and Autonomous Systems* 68 (2015) 104 – 115. doi:<https://doi.org/10.1016/j.robot.2015.01.007>.
- [27] K. W. Wong, D. Tikk, T. D. Gedeon, L. T. Kóczy, Fuzzy rule interpolation for multidimensional input spaces with applications: A case study, *IEEE transactions on fuzzy systems* 13 (6) (2005) 809–819.
- [28] K. Tai, A.-R. El-Sayed, M. Biglarbegian, C. I. Gonzalez, O. Castillo, S. Mahmud, Review of recent type-2 fuzzy controller applications, *Algorithms* 9 (2) (2016) 39.
- [29] E. Ontiveros-Robles, P. Melin, O. Castillo, New methodology to approximate type-reduction based on a continuous root-finding karnik mendel algorithm, *Algorithms* 10 (3) (2017) 77.
- [30] J. Dombi, A. Hussain, A new approach to fuzzy control using the distending function, *Journal of Process Control* 86 (2020) 16–29.
- [31] L. A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on systems, Man, and Cybernetics* (1) (1973) 28–44.
- [32] J. Dombi, T. Jónás, Kappa regression: an alternative to logistic regression, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 28 (02) (2020) 237–267.
- [33] M. M. hardware team, Parrot Drone Support from MATLAB, <https://www.mathworks.com/hardware-support/parrot-drone-matlab.html>, Accessed: 2020-03-11.
- [34] T. Luukkonen, Modelling and control of quadcopter, Independent research project in applied mathematics, Espoo 22 (2011).