



A new approach to fuzzy control using the distending function

József Dombi, Abrar Hussain*

Institute of Informatics, University of Szeged, 6720, Szeged, Hungary



ARTICLE INFO

Article history:

Received 13 July 2019

Revised 23 October 2019

Accepted 5 December 2019

Keywords:

Fuzzy arithmetic

Adaptive fuzzy control

Distending function

Membership function

Continuously stirred tank reactor

ABSTRACT

Here, we develop a fuzzy controller using fuzzy arithmetics and a new type of membership function. The proposed new fuzzy control technique is simple, fast and computationally efficient, compared to the classical techniques (Mamdani, Takagi Sugeno) and it can also adapt to the process dynamics. The unique features are: 1) A new class of parametric membership function called the Distending Function (DF) is introduced; 2) A general parametric operator system is used. It utilizes most of the fuzzy operator systems for evaluating the knowledge base; 3) Inference is based on fuzzy arithmetic operations; 4) This leads to a computationally efficient single-step defuzzification. With these concepts, the paradigm of fuzzy control design changes radically. Using this technique with an optimization method, an adaptive fuzzy controller is designed. This adaptive controller adjusts to the changing dynamics of the non-linear processes by tuning our new type of membership function. The effectiveness of the proposed methodology is demonstrated on two industrial processes (a water tank system and continuously stirred tank reactor system).

© 2019 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY license. (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

Fuzzy theory has been an area of extensive research since its inception, nearly half a century ago, by Lotfi A. Zadeh [1,2] and it is providing applications in various areas of daily life [3–6]. To design control systems for complex ill-defined non-linear processes (for which adequate analytical models are not available), is a challenging task. Novel control techniques are proposed to solve such problems [7–10]. However, if a knowledge base is available for these systems, fuzzy theory provides an adequate solution for controller design [11]. For some non-linear processes, the model parameters vary with time or they may have uncertain initial conditions. The control of such non-linear dynamic processes is called adaptive control, where the control law adapts itself to the changing dynamics in order to meet the control objectives [12–14]. An adaptive fuzzy controller organizes the rule base (type and number of rules) and it tunes the parameters of the membership functions if the process dynamics changes over time [15,16]. Now, we will present some basics of conventional fuzzy control techniques and briefly mention our contributions:

The design of fuzzy logic control (FLC) is based on the set of 'If then' rules forming a rule base. The multi-input single output (MISO) fuzzy rule base has the following form:

If x_1 is A_1^i and \dots and x_n is A_n^i then y is B^i , (1)

where x_1, x_2, \dots, x_n, y are the linguistic variables that take the linguistic values from the fuzzy sets A_1, A_2, \dots, A_n, B and $i = 1, \dots, l$ is the number of fuzzy rules. The part of the rule after 'If' is a logical expression, called the antecedent (related to the input) and the part after 'then' is called the consequent (related to the output). The Fuzzy Rule Inference (FRI) applies a fuzzy relation to map the input and output space. Based on the FRI, various types of control techniques have been developed. The two best known are the Mamdani [17] and model-based Takagi Sugeno (TS) [18,19] type fuzzy control systems.

In the Mamdani inference system (also called the Type-I TS system), the output of a rule is a fuzzy set. The operation of the Mamdani inference system consists of five steps. These are: 1) The fuzzification of crisp inputs in order to get fuzzy inputs; 2) The antecedent parts of the rules are based on fuzzy logic operators. These fuzzy logical expressions are evaluated to determine the applicability of the fuzzy rules; 3) Implication is carried out to get fuzzy outputs; 4) Aggregating the fuzzy outputs of all the rules; 5) Defuzzification of the aggregated output is carried out to get the final crisp output. The Mamdani fuzzy controller directly transforms the operator implicit knowledge or expert explicit views into fuzzy rules and generates a control law. The Mamdani approach is intuitive, works well with direct human input and various control tasks can be performed [20,21]. The stability of a closed loop control system is one of the main objectives that should be met. Frequency domain methods are mostly used for the stability analysis of Mamdani fuzzy controllers, such as Popov's method, the circle stability criterion and hyperstability theory [22].

* Corresponding author.

E-mail address: hussain@inf.u-szeged.hu (A. Hussain).

In the case of the TS (Type-II / Type-III) fuzzy controller, the consequent part of the rule (Eq. (1)) is a function (mostly linear) of the inputs or (as a special case) a crisp value i.e. it is not a fuzzy set as in the Mamdani case. The TS fuzzy model consists of a membership function and a set of linear models to form a global nonlinear model. The TS fuzzy controller also has a nonlinear function approximation property [23].

There are some drawbacks with these two control techniques. These are:

1. The input space is not completely covered by the triangular (or trapezoidal) membership functions i.e. they cover only a limited subspace. For example if we have two inputs, each with 7 categories, then 49 rules are required to cover the whole input space. Usually a few rules are applied to decrease the computation load, so a large area of the input space is not covered. If the input value falls in the area which is not covered by any rule, then no action is generated. If the number of input variables increases from two (working in a higher dimensional space), then the problem grows exponentially. Some efforts to overcome this problem have been made by L. Koczy and K. Hirota [24], but all these procedures increase the computational cost.
2. Most of the membership functions are not analytical i.e. the derivative is not defined at every point and higher derivatives do not exist. This is a drawback because the gradient-based optimization techniques cannot be used to tune the parameters of these membership functions as they work only on analytical functions. However in this case, the gradient-free optimization techniques can still be used but these are not very fast, accurate and computationally efficient compared to gradient-based methods.
3. It is not clear how to choose a fuzzy operator system for the antecedent part. Various fuzzy operators can be chosen. Using different fuzzy operators produces different results. Hence the choice of membership function and operator system is completely arbitrary and we cannot get a proper efficient design.
4. Different implication operators are introduced. Surprisingly, the product operator is mainly used. The product operator is a strict t-norm and it is not an implication operator.
5. The result of an evaluation of the consequent part of the rule is not a membership function (it is an α -cut of the membership function). For every input value, each rule is evaluated to get the aggregated output. The aggregation of consequent parts of all the rules is a membership function which does not belong to the same class of the antecedent and consequent membership functions.
6. Centre of gravity (COG) defuzzification usually involves the integral evaluation of the aggregated function and it is computationally expensive. Although there are various defuzzification methods that do not require integral calculations, in general these are less accurate than COG-based methods.
2. The DF is analytical i.e. higher derivatives exist at each point. This property is used in optimization procedures to tune the parameters of the DF.
3. The general parametric fuzzy operator system is used for evaluating the antecedent part of the rule. The operator system and the DF are based on the Dombi operator. Hence, both are consistent with each other.
4. Our approach does not involve the implication step. Instead the activation strength of each rule is multiplied by the consequent DF to get the fuzzy output of each rule.
5. The consequent of each rule is a DF. Aggregation is carried out using the weighted arithmetic mean of these consequent DFs of all the rules. A linear combination is closed for DFs and so the result of aggregation is also a DF.
6. Defuzzification in this case is only a single-step calculation (finding the point that has the highest value of the aggregated DF). This is why it is simple and computationally efficient.
7. Using our proposed approach, we design an adaptive fuzzy controller. It consists of tuning the DF parameters using gradient descent optimization. The adaptive controller can handle the changing process dynamics with increased computational efficiency.

We combine the advantages of Mamdani and TS methods. Here, the antecedent and consequent parts of the rule base are fuzzy sets, so it is very close to direct human linguistic inputs. Related work for designing an adaptive fuzzy controller for a nonlinear system with unknown dynamics was carried out by Ning Wang et al. [13,14]. However, there are two main differences with our approach: 1) Product inference (operator and implication) is used for evaluating the fuzzy rules and designing controller in [14]; 2) Adaptivity is achieved using Retractable Membership Functions (RMFs) in [13], which are symmetric membership functions. In our approach, we have used a more general operator system which can utilize various available fuzzy operators (e.g. min/max, product, Einstein, Hamacher, Dombi, drastic). Fuzzy arithmetic is used instead of implication i.e. we used a regression-like approach. In our study, the symmetric and asymmetric DFs are used. The asymmetric DF provides more flexibility in adaptive controller design. The method described in [13] and our approach overcome the curse of dimensionality issue, but in slightly different ways. The efficiency of this new approach is shown by designing an adaptive control system for a water tank level and vehicle lateral dynamics.

The rest of the paper is structured as follows. In Section II, we briefly introduce the distending function, its properties and an overview of fuzzy arithmetic. In Section III, we explain the proposed design approach using fuzzy arithmetic. In Section IV, we describe the adaptive control design using an optimization method. In Section V, we outline the benchmark systems, simulations and discuss the results. Lastly, in Section VI, we present our main conclusions.

2. The distending function

The Distending Function (DF) is a continuous function which is monotonically increasing in the interval $(-\infty, 0)$ and monotonically decreasing in the interval $(0, +\infty)$ and it takes values in $[0, 1]$. The DF $(\delta_{\epsilon, \nu}^{(\lambda)}(x))$ is a parametric membership function. The parameters are the threshold (ν) , tolerance/error (ϵ) and sharpness (λ) . The DF has a peak value of 1 at $x = 0$. If the input is in the interval $[-\epsilon, \epsilon]$, the value of the DF is greater than ν and also $\delta_{\epsilon, \nu}^{(\lambda)}(\epsilon) = \nu$. When the input is in the interval $[-\epsilon, \epsilon]$, it is treated as a truth value. The threshold (ν) divides the $[0, 1]$ interval into truth and falseness regions. And the parameter λ controls the sharpness of the DF. If $\lambda \rightarrow \infty$, then the DF approaches the characteristic function. With an appropriate values of ν, ϵ and λ ,

Using these techniques, designing an adaptive fuzzy controller is a challenging task. As we mentioned above, both of these techniques have some advantages and disadvantages. There is a need to combine these advantages into a single design approach. We attempted to solve these issues using a new approach, which has the following good features:

1. A new type of parametric membership function called the Distending Function (DF) is introduced. With a few rules, it can cover the whole input space. It has three parameters and each has a semantic meaning. The values of the two parameters are usually fixed and one parameter value is tuned during the design process. It has two types, namely the symmetric and asymmetric DF and both can be utilized for control system design.

all the existing membership functions (Trapezoidal, Gaussian, Sigmoidal, etc) can be approximated using the DF. The membership function can be shifted by a parameter c and we shall use the notation $(\delta_{\varepsilon, \nu}^{(\lambda)}(x - c))$. We can interpret it as x approaches c , i.e. x is equal to c (soft equality). Here c is the center point of the DF $(\delta_{\varepsilon, \nu}^{(\lambda)}(c) = 1)$. Now we will introduce two types of DF namely, symmetric and asymmetric.

2.1. The symmetric distending function

The Symmetric DF is symmetric around $x - c$ and is defined as:

Definition 1. The symmetric DF (shown in Fig. 1) is given by

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

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

2.2. The asymmetric distending function

The asymmetric type of the DF can be defined in the following way:

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

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

where $\nu_R, \nu_L \in (0, 1)$, $\varepsilon_R, \varepsilon_L > 0$, $\lambda_R, \lambda_L \in (1, +\infty)$, $c \in \mathbb{R}$ and $\lambda^* \in (1, +\infty)$. λ^* is a technical parameter and its value is very large compared to λ_R and λ_L . ν_L, ε_L and λ_L are the parameters of the left hand side whereas ν_R, ε_R and λ_R are the parameters of the right

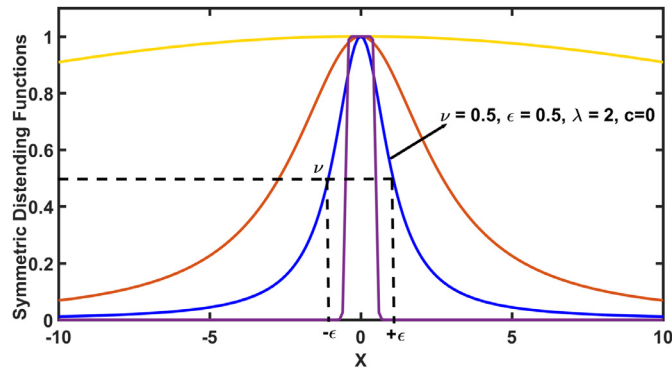


Fig. 1. Various shapes of symmetric Distending Functions depending on parameter values.

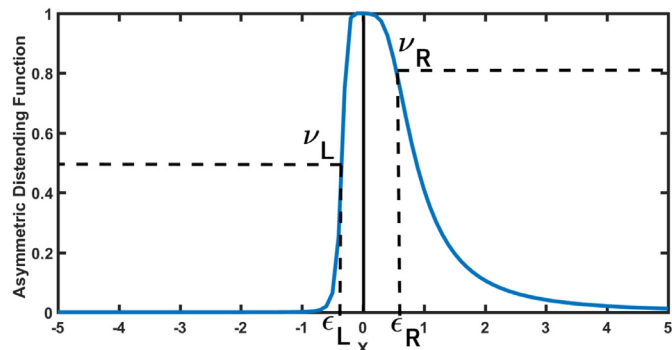


Fig. 2. The asymmetric Distending Function ($\nu_L = 0.5$, $\varepsilon_L = 0.5$, $\lambda_L = 5$, $\nu_R = 0.8$, $\varepsilon_R = 0.7$, $\lambda_R = 5$, $\lambda = 5$, $c = 0$).

hand side of the asymmetric DF. If the input lies between ε_R and 0, then the grade of membership is greater than ν_R and the same is true for ε_L and ν_L . Here, c is the centre point i.e. $\delta_A(c) = 1$.

The asymmetric DF provides more flexibility in control design. Here, the right and left hand sides of the asymmetric DF can be controlled independently. Next, we will give a formula to calculate the coordinate of the Center of Gravity (COG) of the asymmetric DF. The coordinate of the COG is used for defuzzification. First, we calculate the area under the DF. We will consider only one (right hand) side of the DF. Using this area, we derive an expression for the coordinate of the COG for the right hand side. Then we give an expression of the COG for both sides of the asymmetric DF.

2.3. Area under the distending function

The integral of the DF can be written in the form

$$I = \int_0^{+\infty} \frac{1}{1 + \frac{1-\nu}{\nu} \left| \frac{x}{\varepsilon} \right|^\lambda} dx. \quad (4)$$

The result is

$$I = \varepsilon \left(\frac{\nu}{1-\nu} \right)^{\frac{1}{\lambda}} \frac{\pi}{\sin \frac{\pi}{\lambda}}. \quad (5)$$

2.4. Center of gravity (COG) defuzzification

Let the function $\delta_{\varepsilon, \nu, +}^{(\lambda)}(x)$ ("+" means the right hand side) be defined as follows:

$$\delta_{\varepsilon, \nu, +}^{(\lambda)}(x) = \begin{cases} 0, & \text{if } x < 0 \\ \frac{1}{1 + \frac{1-\nu}{\nu} \left| \frac{x}{\varepsilon} \right|^\lambda}, & \text{if } x \geq 0, \end{cases} \quad (6)$$

where $\nu \in (0, 1)$, $\varepsilon > 0$ and $\lambda \in \mathbb{R}$, $\lambda > 1$. Let x^* denote the horizontal coordinate of the COG, as shown in Fig. 3. It is well known that

$$x^* = \frac{\int_{-\infty}^{+\infty} x \delta_{\varepsilon, \nu, +}^{(\lambda)}(x) dx}{\int_{-\infty}^{+\infty} \delta_{\varepsilon, \nu, +}^{(\lambda)}(x) dx}. \quad (7)$$

Then the coordinate x^* of the COG of the area under the curve for $\delta_{\varepsilon, \nu, +}^{(\lambda)}(x)$ is

$$x^* = \frac{1}{2} \varepsilon \left(\frac{\nu}{1-\nu} \right)^{\frac{1}{\lambda}} \frac{1}{\cos \frac{\pi}{\lambda}}. \quad (8)$$

Using this, the coordinate x^* of COG for both sides of asymmetric DF is

$$x^* = \frac{\Delta_L^2 S_1 - \Delta_R^2 S_2}{\Delta_L + \Delta_R}, \quad (9)$$

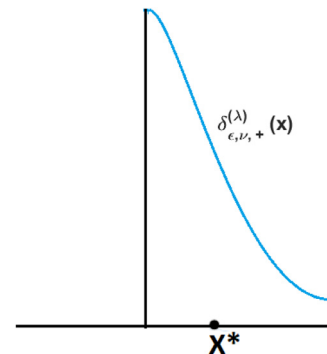


Fig. 3. The COG of the DF (RHS).

where

$$\Delta_L = \varepsilon_L \left(\frac{v_L}{1-v_L} \right)^{\frac{1}{\lambda_L}} \frac{\pi}{\sin \frac{\pi}{\lambda_L}} ; \quad \Delta_R = \varepsilon_R \left(\frac{v_R}{1-v_R} \right)^{\frac{1}{\lambda_R}} \frac{\pi}{\sin \frac{\pi}{\lambda_R}}$$

$$S_1 = \frac{\sin \frac{\pi}{\lambda_L}}{\frac{2\pi}{\lambda_L} \cos \frac{\pi}{\lambda_L}} ; \quad S_2 = \frac{\sin \frac{\pi}{\lambda_R}}{\frac{2\pi}{\lambda_R} \cos \frac{\pi}{\lambda_R}}.$$

Remark 1. Some special cases:

1. If $v_L = v_R = v$ and $\lambda_L = \lambda_R = \lambda$, then

$$x^* = \frac{\varepsilon_L^2 \left(\frac{v}{1-v} \right)^{\frac{1}{\lambda}} - \varepsilon_R^2 \left(\frac{v}{1-v} \right)^{\frac{1}{\lambda}}}{\varepsilon_L + \varepsilon_R} \frac{1}{\cos \frac{\pi}{\lambda}}.$$

2. If $v = 0.5$, then

$$x^* = \frac{1}{2} (\varepsilon_L - \varepsilon_R) \frac{1}{\cos \frac{\pi}{\lambda}}$$

and if $c \neq 0$, then

$$x^* - c = \frac{1}{2} (\varepsilon_L - \varepsilon_R) \frac{1}{\cos \frac{\pi}{\lambda}}. \quad (10)$$

3. For symmetric DFs, $\varepsilon_R = \varepsilon_L$, so

$$x^* = c, \quad (11)$$

which gives the expression for the coordinate of the COG of the symmetric DFs.

Now, we will evaluate the derivatives of DF. These will be used in the optimization process.

2.5. Derivatives of the distending function

Let

$$\delta(x) = \delta_{\varepsilon, v}^{(\lambda)}(x) = \frac{1}{1 + \left| \frac{1-v}{v} \right| \left| \frac{x}{\varepsilon} \right|^{\lambda}}. \quad (12)$$

Then the first derivative of Eq. (12) is:

$$\frac{\partial}{\partial x} \delta(x) = -\frac{\lambda}{x} \delta(x) (1 - \delta(x)).$$

The partial derivatives of the DF with respect to ε is

$$\frac{\partial}{\partial \varepsilon} \delta(x) = -\frac{\lambda}{\varepsilon} \delta(x) (1 - \delta(x)).$$

It is worth mentioning that the derivatives of DF are similar to the derivatives of the sigmoid function and these derivatives can readily be calculated just using the DF.

2.6. Fuzzy arithmetic operations on distending functions

It was suggested by Zadeh [25] that fuzzy quantities can be combined arithmetically using the laws of fuzzy theory. This direction was then explored independently by many researchers [26–28]. Later it was established that fuzzy theory is an extension of the algebra of many-valued logic and interval analysis [29,30]. Thus interest in fuzzy interval domain has increased [31]. Fuzzy arithmetic can be viewed as the arithmetic of α cuts. It handles the fuzzy quantities which are obtained by mapping a real number to real interval $[0, 1]$. We create a α cuts ($\alpha \in [0, 1]$) for each fuzzy quantity and then perform the required operation using the principles of interval arithmetic. Here, instead of intervals, we will use the left and right hand sides of DFs, which are defined on these intervals. This is possible in the case where two sides of the given functions are monotonously increasing or decreasing functions. The details and advantages of using fuzzy arithmetic operations in control design are given in [32].

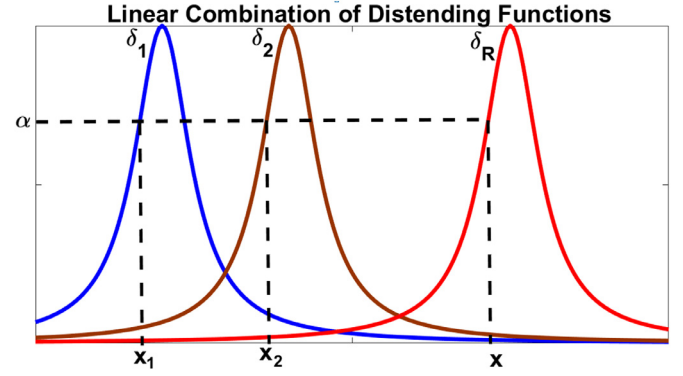


Fig. 4. A linear combination of two DFs δ_1 and δ_2 using fuzzy arithmetic operations on α cuts.

Next, we will show that DFs are closed under a linear combination i.e. a linear combination of DFs is also a DF (see Fig. 4). Let

$$\delta_{i(v, \varepsilon_i, c_i)}^{(\lambda)}(x) = \frac{1}{1 + \left(\frac{1-v}{v} \right) \left| \frac{x-c_i}{\varepsilon_i} \right|^{\lambda}},$$

where $i = 1, \dots, n$ are the DFs that have the same v and λ values. These n DFs can be combined using fuzzy arithmetic operations and the result is

$$\delta_R(x) = \frac{1}{1 + \frac{1-v}{v} \left| \frac{x-c_R}{\varepsilon_R} \right|^{\lambda}}. \quad (13)$$

δ_R is the resultant DF obtained from the linear combination of n DFs with c_R and ε_R given by

$$\varepsilon_R = \sum_{i=1}^n w_i \varepsilon_i, \quad c_R = \sum_{i=1}^n w_i c_i. \quad (14)$$

3. Control design approach

Our design methodology is motivated by our previous study where a fuzzy control was designed using fuzzy arithmetic operations [32]. It has all the desired properties, such as:

- Range independence,
- The effect of fuzziness is fully incorporated,
- The computation speed is very high compared to conventional COG defuzzification-based control.

Using expert knowledge or process data, the multi-input multi-output system (MIMO) is described by:

$$\text{If } x_1 \text{ is } A_1^i \text{ and } \dots \text{ and } x_n \text{ is } A_n^i \\ \text{then } y_1 \text{ is } B_1^i; \dots; y_m \text{ is } B_m^i, \quad (15)$$

where x_1, x_2, \dots, x_n are the input linguistic variables which take the values from the input fuzzy subsets A_1, A_2, \dots, A_n . The variables y_1, y_2, \dots, y_m are the output linguistic variables which take the values from the output fuzzy subsets B_1, B_2, \dots, B_m . $i = 1, \dots, l$ is the number of fuzzy rules. If the output variables are independent of each other, then each rule of the rule base given by Eq. (15) can be written as m multi input single output (MISO) rules

$$\text{If } x_1 \text{ is } A_1^i \text{ and } \dots \text{ and } x_n \text{ is } A_n^i \text{ then } y \text{ is } B^i. \quad (16)$$

We will try to find the fuzzy inference mechanism to map the input-output space and generate a crisp output for the control signal. In our approach, we handle the antecedent and consequent parts of the rule separately.

3.0.1. The antecedent part

The antecedent part of the i th fuzzy rule is

$$\mathcal{L}(\delta_1(x_1)^i, \delta_2(x_2)^i, \dots, \delta_n(x_n)^i) = \hat{w}_i(x), \quad (17)$$

where $\hat{w}_i(x)$ is the rule applicability function. \mathcal{L} is the fuzzy logical expression and it may include *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 use a very general parametric operator [33]

$$D_\gamma(x) = \frac{1}{1 + \left(\frac{1}{\gamma} \left(\prod_{i=1}^n \left(1 + \gamma \left(\frac{1 - \delta(x_i)}{\delta(x_i)} \right)^\alpha \right) - 1 \right) \right)^{\frac{1}{\alpha}}}. \quad (18)$$

Most of the conjunctive or disjunctive operators used (e.g. min/max, product, Einstein, Hamacher, Dombi, drastic) are covered by Eq. (18).

For a specific input values \underline{x}^* , Eq. (17) can be evaluated and this results in a single numeric value $\hat{w}_i(\underline{x}^*)$

$$\mathcal{L}(\delta_1(x_1^*)^i, \delta_2(x_2^*)^i, \dots, \delta_n(x_n^*)^i) = \hat{w}_i(\underline{x}^*), \quad (19)$$

where $\hat{w}_i(\underline{x}^*)$ is called the strength of the i th rule. We normalize these strengths (to compare the rules) to get the firing strengths $w_i(\underline{x}^*)$. The firing strength gives the probability of the rule. The firing strength of the i th rule is

$$w_i(\underline{x}^*) = \frac{\hat{w}_i(\underline{x}^*)}{\sum_{i=1}^l \hat{w}_i(\underline{x}^*)}, \quad (20)$$

where

$$\sum_{i=1}^l w_i(\underline{x}^*) = 1. \quad (21)$$

3.0.2. The consequent part

This part of the rule is a fuzzy set represented by a single DF. The firing strength of each rule (calculated from the antecedent part) is multiplied by the consequent part. Also the fuzzy output of each rule is a DF. By combining all the rules, we can generate an aggregated output. If $w_1(\underline{x}^*), w_2(\underline{x}^*), \dots, w_l(\underline{x}^*)$ are the firing strengths and $\delta_{1o}(x), \delta_{2o}(x), \dots, \delta_{lo}(x)$ are the l consequents, then the aggregated output of the l fuzzy rule is given by:

$$\delta_a(x) = \sum_{i=1}^l w_i(\underline{x}^*) \delta_{io}(x). \quad (22)$$

Of course, $\delta_a(x)$ is also a DF. We calculate the parameters of the DF using Eq. (14). The aggregated output DF $\delta_a(x)$ has the following form

$$\delta_a(x) = \frac{1}{1 + \frac{1-\nu}{\nu} \left| \frac{x-c_a}{\varepsilon_a} \right|^\lambda}, \quad (23)$$

where

$$c_a = \sum_{i=1}^l w_i(\underline{x}^*) c_i, \quad \varepsilon_a = \sum_{i=1}^l w_i(\underline{x}^*) \varepsilon_i, \quad (24)$$

where c_i and ε_i are the parameters of the i th consequent. The crisp control output will be the COG of the aggregated DF $\delta_a(x)$.

The whole procedure is summarized in Algorithm 1.

Our control design approach has the following unique features:

- A general parametric operator is used for calculating the firing strength of rules. This single operator can be employed to calculate the Zadeh, product, Einstein, drastic and Dombi t-norm and t-conorms with appropriate values of the parameters.
- The inputs are fuzzified using DFs. These DFs can approximate most types of bell-shaped membership functions. Also, the input member functions might have different shapes (Gaussian, Trapezoidal, Sigmoidal) at the same time for different categories.

Algorithm 1: Algorithm for synthesis of fuzzy controller using DF.

- Step 1:** Define the DFs for the input and output linguistic variables.
- Step 2:** Fuzzify the crisp inputs using Eq. (2) or Eq. (3).
- Step 3:** Construct the rule base from the expert knowledge using Eq. (16).
- Step 4:** Calculate the strength of each rule using Eq. (18) by choosing the appropriate fuzzy conjunctive/disjunctive operators.
- Step 5:** Calculate the l firing strengths using Eq. (20).
- Step 6:** Calculate the parameters (c_a, ε_a) of the aggregated output DF by using Eq. (24).
- Step 7:** Generate the aggregated output DF using Eq. (23).
- Step 8:** Get the crisp output control signal u by calculating the COG of the aggregated output DF using Eq. (9) or Eq. (11).

- The aggregate functions is also a DF. This is due to the fact that a linear combination of DFs is a also a DF.
- The design procedure is simplified because it does not include implication. The aggregation is carried out using fuzzy arithmetic operations.
- Defuzzification is a single-step calculation.

Hence, the proposed approach is computationally efficient.

4. Adaptive fuzzy control design

If the process parameters vary with time, then the adaptive controller changes the control signal in accordance with the change in process dynamics. Therefore, the adaptive controller works even when the values of the parameters are outside the desired range. Here, we shall present a hybrid scheme for adaptive control. We call it a hybrid because it utilizes the knowledge base and an optimization method. The hybrid scheme selects one fuzzy rule from the knowledge base and it tunes the parameters of the antecedent part. This rule will be selected based on the firing strength. The optimization technique is used to tune the parameters of the antecedent part.

Here we have three tunable parameters, namely ν , λ and ε . We shall fix the values of ν and λ and we will tune the ε parameter. The knowledge base of a fuzzy controller consists of "If then" rules that have the following form:

$$\left[\begin{array}{l} \text{If } x_1 \text{ is } A_{11} \text{ and } \dots, x_n \text{ is } A_{1n} \text{ then } y \text{ is } B_1 \\ \vdots \\ \text{If } x_1 \text{ is } A_{i1} \text{ and } \dots, x_n \text{ is } A_{in} \text{ then } y \text{ is } B_i \\ \vdots \\ \text{If } x_1 \text{ is } A_{l1} \text{ and } \dots, x_n \text{ is } A_{ln} \text{ then } y \text{ is } B_l \end{array} \right], \quad (25)$$

where $i = 1, 2, \dots, l$ is the number of rules. $A_{i1}, A_{i2}, \dots, A_{in}$ are the input linguistic terms and B_i are the output linguistic terms. All the input and outputs are associated with corresponding DFs. Here, y is the fuzzy output. Now let $\hat{w}_1, \hat{w}_2, \dots, \hat{w}_l$ be the strengths of the l fuzzy rules and c_1, c_2, \dots, c_l be the COG values of l corresponding DFs. The crisp value Y (25) is

$$Y = \frac{c_1 \hat{w}_1 + c_2 \hat{w}_2 + \dots + c_l \hat{w}_l}{\hat{w}_1 + \hat{w}_2 + \dots + \hat{w}_l}. \quad (26)$$

Let the squared error function has the form

$$E = \frac{1}{2} (Y_{ref} - Y)^2, \quad (27)$$

where Y_{ref} is the known reference control signal and Y is the control signal (crisp output) generated by the fuzzy controller. Now

the adaptive control problem reduces to the following optimization task

$$\text{Minimise}_{\varepsilon > 0}(E). \quad (28)$$

Using the gradient descent method,

$$\varepsilon_{t+1} = \varepsilon_t - \eta_s \frac{\partial}{\partial \varepsilon}(E). \quad (29)$$

From Eq. (27), we have

$$\varepsilon_{t+1} = \varepsilon_t + 2\eta_s E \frac{\partial}{\partial \varepsilon} \left(\frac{c_1 \hat{w}_1 + c_2 \hat{w}_2 + \dots + c_l \hat{w}_l}{\hat{w}_1 + \hat{w}_2 + \dots + \hat{w}_l} \right), \quad (30)$$

where η_s is the step size of the optimization. To reduce the complexity and computational cost, we will select one rule and tune the antecedent part at the same time. The rule will be selected on the basis of rule strength ($\hat{w}_1, \hat{w}_2, \dots, \hat{w}_l$). Now we will explain this selection and the tuning procedure.

Let the i th rule has the highest firing strength. The remaining $(l-1)$ strengths will be constant. Eq. (30) can be written as

$$\varepsilon_{t+1} = \varepsilon_t + 2\eta_s E \frac{\partial}{\partial \varepsilon_i} \left(\frac{c_i \hat{w}_i + k_1}{\hat{w}_i + k_2} \right), \quad (31)$$

where

$$k_1 = c_1 \hat{w}_1 + \dots + c_{i-1} \hat{w}_{i-1} + c_{i+1} \hat{w}_{i+1} + \dots + c_l \hat{w}_l$$

$$k_2 = \hat{w}_1 + \dots + \hat{w}_{i-1} + \hat{w}_{i+1} + \dots + \hat{w}_l.$$

From Eq. (31), we get

$$\frac{\partial}{\partial \varepsilon_i} \left(\frac{c_i \hat{w}_i + k_1}{\hat{w}_i + k_2} \right) = \frac{(\hat{w}_i + k_2)(c_i \hat{w}_i') + (c_i \hat{w}_i + k_1) \hat{w}_i'}{(\hat{w}_i + k_2)^2}, \quad (32)$$

where

$$\hat{w}_i' = \frac{\partial}{\partial \varepsilon_i}(\hat{w}_i).$$

By using Eq. (18),

$$\hat{w}_i' = \frac{\partial}{\partial \varepsilon_i} \left(\frac{1}{1 + \left(\sum_{i=1}^n \left(\frac{1 - \delta_i(x_i)}{\delta_i(x_i)} \right)^\alpha \right)^{\frac{1}{\alpha}}} \right). \quad (33)$$

Here $\delta_1(x_1), \dots, \delta_n(x_n)$ are the n antecedent DFs in the i th rule. To reduce the computation cost, the ε parameter of only one antecedent DF will be tuned at a time. The DF to be tuned is selected by comparing the grade of membership of these n DFs. The DF with the highest grade is selected for tuning. The remaining $(n-1)$ antecedent DFs in the i th rule will be treated as constants. Let ε_i be the parameter of the i th antecedent DF having the highest grade value. Then Eq. (33) can be written as

$$\hat{w}_i' = \frac{\partial}{\partial \varepsilon_i} \left(\frac{1}{1 + \left(\left(\frac{1 - \delta_i(x_i)}{\delta_i(x_i)} \right)^\alpha + k_3 \right)^{\frac{1}{\alpha}}} \right), \quad (34)$$

where

$$k_3 = \sum_{k=1}^{i-1} \left(\frac{1 - \delta_k(x_k)}{\delta_k(x_k)} \right)^\alpha + \sum_{k=i+1}^n \left(\frac{1 - \delta_k(x_k)}{\delta_k(x_k)} \right)^\alpha.$$

For the Dombi conjunctive operator, $\alpha = 1$. Since k_3 is independent of ε_i ,

$$\begin{aligned} \hat{w}_i' &= \frac{\partial}{\partial \varepsilon_i} \delta_i(x_i) \\ &= -\frac{\lambda}{\varepsilon_i} \delta_i(x_i)(1 - \delta_i(x_i)). \end{aligned} \quad (35)$$

Eq. (31) can be written as

$$\varepsilon_{t+1} = \varepsilon_t - \frac{2\lambda \eta_s E \delta_i(x_i)(1 - \delta_i(x_i))(2c_i \hat{w}_i + c_i k_2 + k_1)}{\varepsilon_i (\hat{w}_i + k_2)^2}, \quad (36)$$

which is the update for ε of the i th antecedent DF of the i th rule. The procedure for designing the adaptive fuzzy controller is summarized in Algorithm 2.

Algorithm 2: Algorithm for the synthesis of the adaptive fuzzy controller using DF.

- Step 1:** Define a tolerance τ as an acceptable upper bound on error E .
- Step 2:** Calculate the error E between the fuzzy control Y and reference control Y_{Ref} .
- Step 3:** If $E \geq \tau$, then perform the following steps (steps 4 to 9) else exit.
- Step 4:** Calculate the rule strengths $\hat{w}_1, \dots, \hat{w}_l$ of l fuzzy rules.
- Step 5:** Select the rule with the highest rule strength.
- Step 6:** Calculate the grade of membership of n antecedent DF within in the selected rule using Eq. (2) or Eq. (3).
- Step 7:** Select the antecedent DF with the highest grade of membership.
- Step 8:** Update the parameter ε of the selected antecedent DF using Eq. 36.
- Step 9:** Go to Step 2.

5. Simulation, results and discussion

The effectiveness of the proposed technique is demonstrated using the simulation case studies of two practical systems.

5.1. Water tank level control

5.1.1. Water tank model

Consider a water tank system, as shown in Fig. 5. Water continuously flows in and out of the tank. There is also a valve at the inlet pipe to control the inflow to the tank. The rate of change of the water volume V inside the tank is

$$\frac{dV}{dt} = q_{in} - q_o, \quad (37)$$

where q_{in} and q_o are the inflow and outflow rates. If A_t is the area of the tank base and l is the height of the liquid in the tank, then

$$A_t \frac{dl}{dt} = q_{in} - A_o \sqrt{2gl}. \quad (38)$$

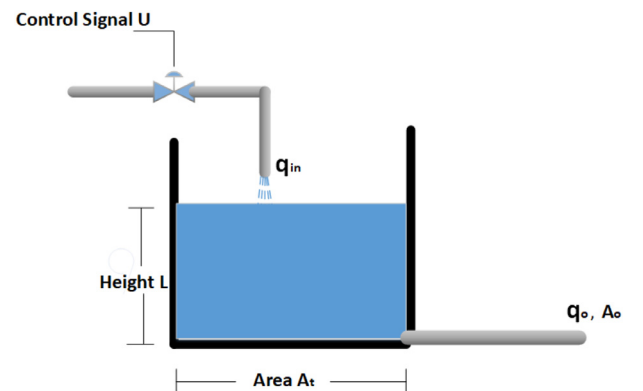


Fig. 5. Water Tank level system.

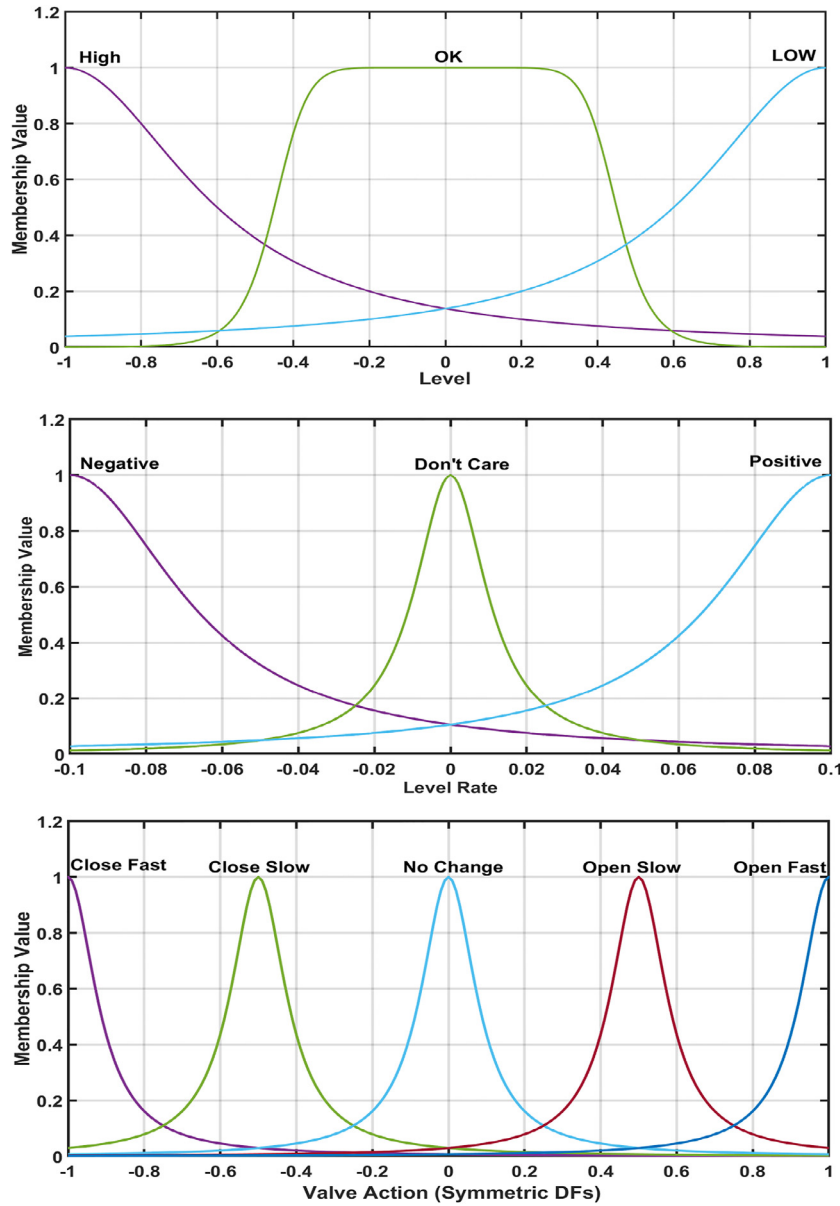


Fig. 6. Antecedent and Consequent DFs for the Water Tank level controller (Symmetric).

where A_o is the cross sectional area of the outlet and g is the acceleration due to gravity. A control signal u is sent to the valve located on the inlet pipe and the height of water column inside the tank is controlled by the change in the ratio of inlet and outlet flow rates. The level l of the water in the tank is measured using a level sensor.

5.1.2. Control scenario (changing the water level)

A fuzzy controller is designed using our approach to control the water level of the tank at the specified height (reference) by opening/closing the inlet valve. The difference in the measured level l and the reference height (called the Level Error) is fed to the fuzzy controller as an input. To control the level efficiently, the rate of change of the level in the tank is also fed to the controller as a second input. The controller then generates the control signal for opening and closing the valve. The controller rule base is shown in the Table 1. The multiple inputs are fuzzified using the DF (Eq. (2) or Eq. (3)). DFs are also defined for the output control i.e. the valve opening/closing signal. The DFs for the antecedent

Table 1

The rule base for the Water Tank level fuzzy controller.

Rate of level			
Level error	Positive	Negative	Don't care
High	-	-	Close fast
Low	-	-	Open fast
OK	Close slow	Open slow	No change

and consequent parts are shown in Fig. 6. The control action is generated using Algorithm 1. Here, Fig. 8 shows the control surface of the fuzzy controller. A reference signal for changing the level of water in the tank between 5m and 15m is fed to the system. Figs. 9 and 10 show the response comparison of the proposed and the conventional Mamdani controllers. The conventional controller is based on Gaussian membership functions, product conjunction and implication operators, max aggregation and COG defuzzification. Fig. 11 shows the performance of the proposed fuzzy controllers using symmetric and asymmetric DFs. The consequent DFs for the asymmetric DF-based controller are shown in Fig. 7.

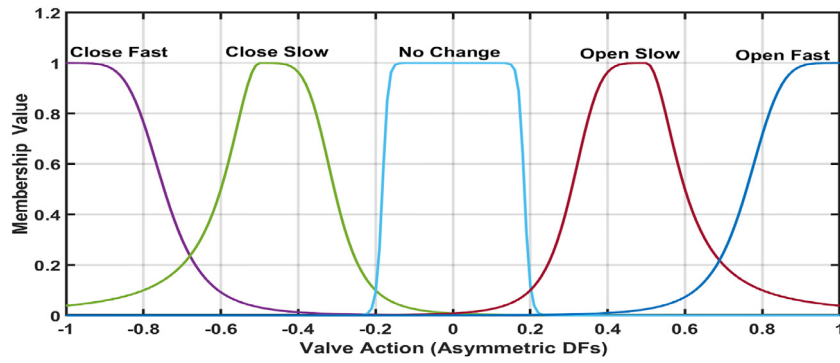


Fig. 7. Consequent DFs for the Water Tank level controller (Asymmetric).

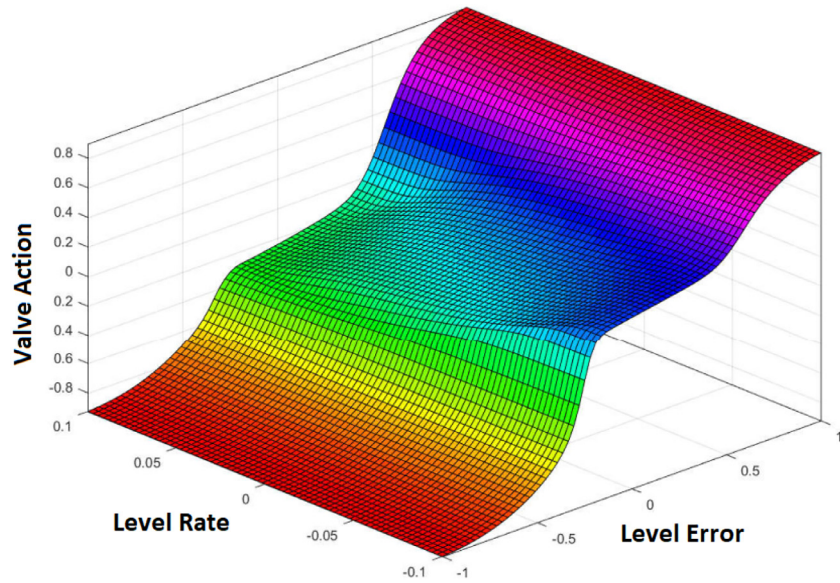


Fig. 8. The control surface for the Water Tank level controllers.

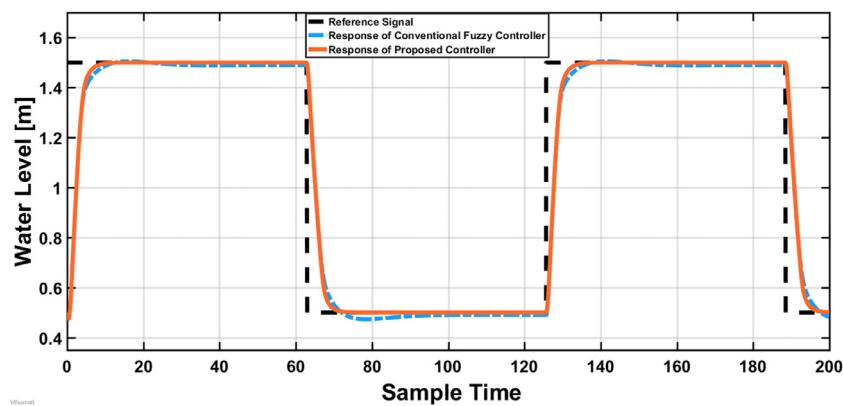


Fig. 9. The response comparison of our proposed controller with a conventional fuzzy controller.

5.2. Temperature control of the continuously stirred tank reactor (CSTR)

5.2.1. CSTR dynamical model

CSTR is a chemical reactor (shown in Fig. 12) which converts a hazardous chemical A into an acceptable product B, which is then

disposed of in the natural environment. The reactor consists of a tank, a cooling jacket and a continuous stirring mechanism. The volume of the chemical inside the tank is usually kept constant. The tank temperature (T_R) and concentration C_A of the chemical A in the outlet stream are the important variables. The reaction is exothermic and irreversible. The tank is continuously stirred for

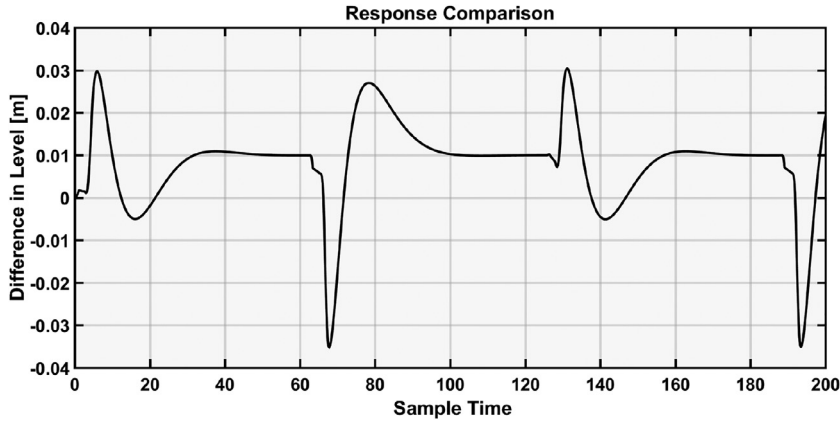


Fig. 10. The difference in response between the proposed controller and the conventional controller (proposed - conventional).

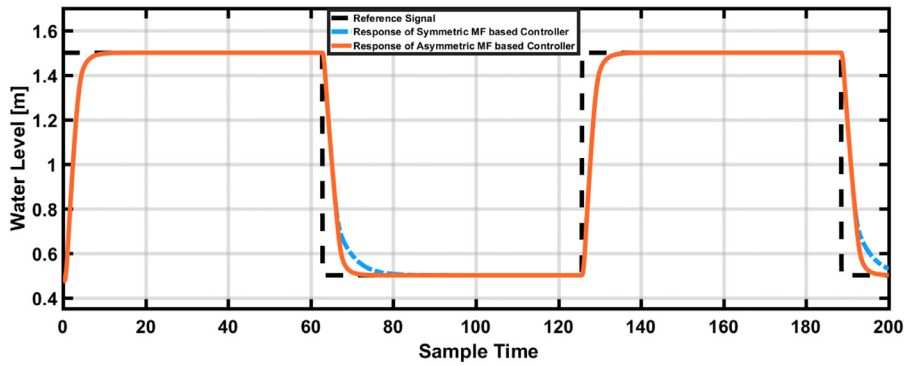


Fig. 11. The response comparison of the symmetric and asymmetric DF-based fuzzy controllers.

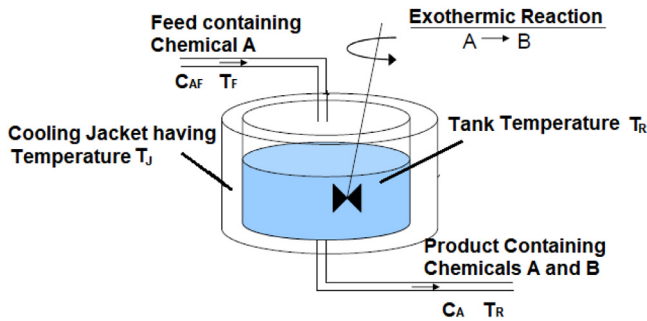


Fig. 12. Continuously Stirred Tank Reactor.

Table 2
CSTR model parameters.

S.No	Paramter	Description
1	T_J	Temperature of cooling jacket (K)
2	q	Volumetric Flowrate (m^3/sec)
3	V	Volume of liquid in CSTR (m^3)
4	ρ	Density of A \rightarrow B Mixture (kg/m^3)
6	C_p	Heat capacity of A-B Mixture ($J/kg - K$)
7	ΔH_r	Heat of reaction for A \rightarrow B (J/mol)
8	k_0	Pre-exponential factor (1/sec)
9	U_A	Overall heat transfer coefficient ($U = W/m^2 - K$)
10	C_{AF}	Feed Concentration (mol/m^3)
11	T_F	Feed Temperature (K)
12	C_A	Concentration of A in CSTR (mol/m^3)
13	T_R	Temperature in CSTR (K)
14	R_T	Residence Time (Sec)

proper mixing to get uniform temperature and concentration profiles. By changing the jacket temperature (T_J), the tank temperature T_R and concentration C_A can be controlled. If T_R reaches the high threshold limit then temperature runaway can occur in the reactor and result in an unsafe operation. The dynamical model of CSTR is derived using mole balance and energy balance equations. It leads to the following state equations:

$$\begin{aligned} \dot{x}_1 &= T_F - x_1 + x_2 \frac{\Delta H_r k_0 e^{\frac{-E}{RT}}}{\rho C_p} + U_A (T_J - x_1); \\ \dot{x}_2 &= \frac{q}{V} (C_{AF} - x_2) - x_2 k_0 e^{\frac{-E}{RT}}, \end{aligned} \quad (39)$$

Here, x_1 and x_2 are the states of the process and represents T_R and C_A respectively. T_J is the control variable. The relevant parameters of this model are given in Table 2. An open source Matlab package was used for the simulations of CSTR [34].

5.2.2. Control scenario (tracking the reactor temperature T_R)

A fuzzy controller based on DFs is designed to track T_R at a set point of 365K and C_A ratio in the outlet stream below 0.3. The threshold for the reactor temperature is 400K. T_R must remain below this threshold to avoid temperature runaway. The controller generates the change in T_J to achieve the desired T_R . The rule base consists of seven rules, as shown in Table 3. Inputs are fuzzified using the DF (Eq. (3)) and rules are evaluated using Dombi operators (see Eq. (18)). The reference signal governing the reactor temperature is subtracted from the actual reactor temperature T_R of the CSTR to generate an error signal E . The error in the reactor temperature E , rate of change of the error signal dE and feed temperature T_F form the input to the fuzzy controller. The antecedents and consequent DFs are shown in Fig. 16. The control

Table 3

Rule base of CSTR fuzzy controller. PL- Positive Large, NL- Negative Large, PM- Positive Medium, NM- Negative Medium, PS- Positive Small, NS- Negative Small, M- Minor, NC- No Change.

Rule no.	Temperature error	Negative rate of temperature error	Feed temperature	Change in jacket temperature
1	PL	-	-	PL
2	NL	-	-	NL
3	M	-	-	NC
4	M	PL	-	NM
5	M	NL	-	PM
6	PL	-	PL	NS
7	PL	-	NL	PS

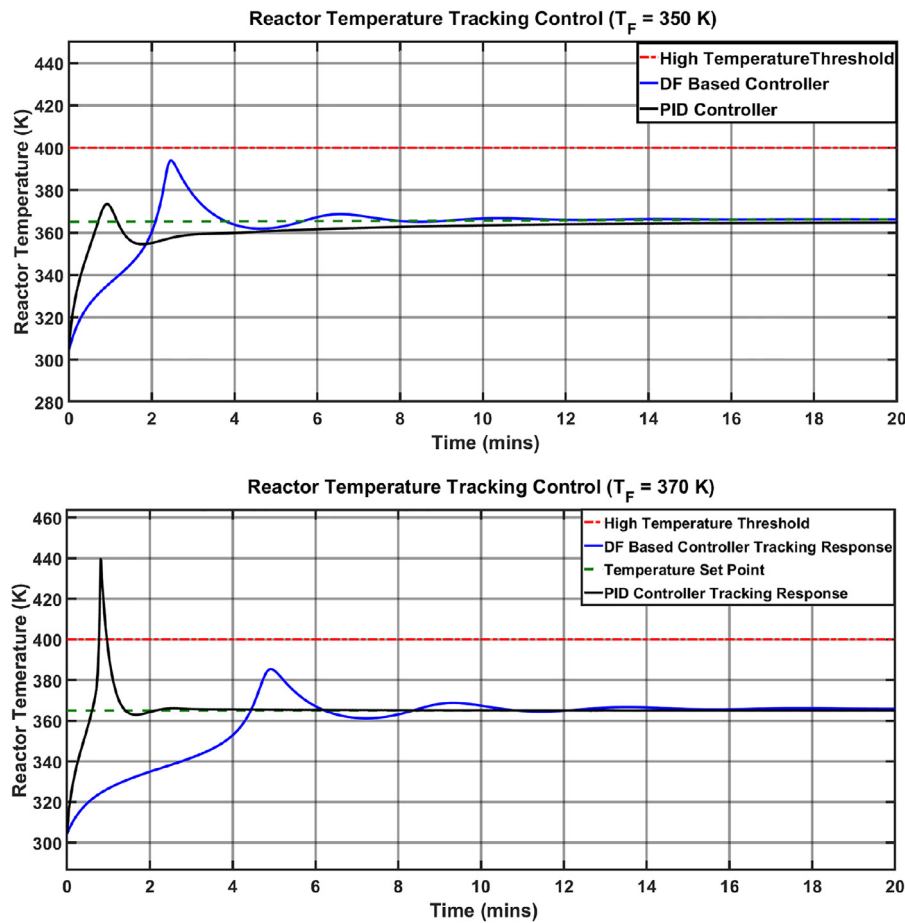


Fig. 13. Comparison of tracking responses generated using the PID ($P = 4$, $I = .8$, $D = 0.5$, $N = 100$) and DF-based controller for different feed temperatures. Both controllers track the reactor temperature when $T_F = 350\text{K}$ and they are within the safe operating temperature limits (Top). Only the DF-based controller tracks and keeps the reactor temperature below the high threshold when $T_F = 370\text{ K}$ (Bottom).

surface (for two inputs; E and T_F) of the fuzzy controller is shown in Fig. 15. A reference signal tells the controller to achieve the desired T_R by changing T_J . The response of the tuned PID controller ($P = 4$, $I = .8$, $D = 0.5$, $N = 100$) has also been plotted for comparison purposes. Fig. 13 shows the reactor temperature during the simulation scenario. The top figure shows the response of the PID and DF-based controllers when the feed temperature is 350K. The bottom figure shows the responses when the feed temperature is increased to 370K. The parameters of the PID and DF-based controller were kept the same. It is evident from the responses that the PID response exceeds the high temperature threshold (400K) when the feed temperature is increased. However the DF-based controller keeps the reactor temperature within the threshold limits, but with a slightly slow response. The concentration C_A has been plotted in Fig. 14 (the $T_F = 350\text{K}$ case).

5.3. Adaptive control

The effectiveness of the proposed adaptive fuzzy controller is examined in a reactor temperature tracking situation where the process dynamics change. The volume V of liquid in the reactor is an important parameter and it must remain constant. This is usually ensured by a separate level control system which keeps the liquid in the reactor at a constant level and hence the volume remains unchanged. However it may happen that the reactor liquid volume increases or decreases due to a fault in the liquid level controller or sudden increase/decrease in the chemical A inventory. In such cases, the reactor temperature will not follow the desired set point and it may lead to runaway situation. Now, consider the case in which the reactor liquid volume increase by 10 percent. Without adaptive control, the reactor temperature is shown in Fig. 17. It is

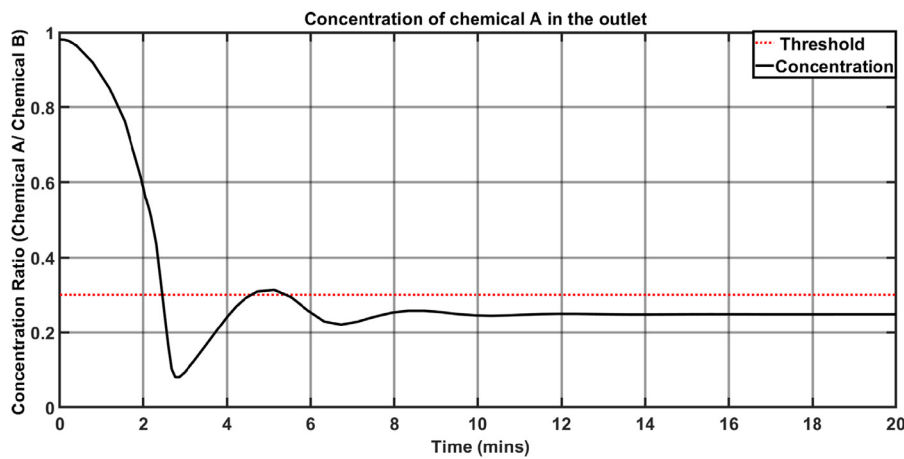


Fig. 14. The concentration of chemical A in the outlet ($T_f = 350K$).

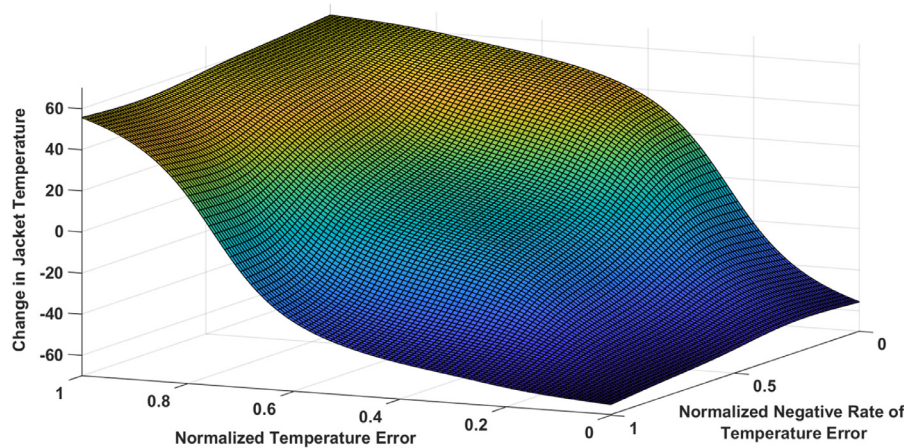


Fig. 15. The control surface of the CSTR fuzzy controller. Two inputs (temperature error and negative rate of temperature error) are used here.

clear that reactor temperature exceeds the higher threshold limit (resulting in unsafe operating conditions) when there is a slight change in the liquid volume. We apply Algorithm 2 to tune the parameters of DFs. The response of the adaptive controller is shown in Fig. 18. The adaptive algorithm is able to tune the parameters of DFs to such a degree that the reactor temperature does not reach the higher threshold and the reactor temperature follows the desire set point even when there is a change in the reactor liquid volume.

5.4. Discussion

From the simulation results, it follows that the proposed fuzzy controller follows the reference command signal very efficiently. The control surfaces indicates that the fuzzy controller has very smooth transitions for these nonlinear processes (Figs. 8 and 15). The performance is much better than the conventional fuzzy controller and PID. Compared to the conventional fuzzy controller, the proposed controller has a short rise time, peak time and percentage overshoot (Figs. 9 and 10). The symmetric and asymmetric DF-based controllers converge to a zero steady state error, but the response of the asymmetric DF-based controller is better due to its short rise and peak times (Fig. 11). The left and right hand sides of the asymmetric DF can be shaped independently, resulting in a more flexible control. Computation efficiency is the major advantage

of the proposed method and the reasons are: 1) The grade of membership can be calculated very quickly the DF. 2) There is no implication, aggregation is based on fuzzy arithmetic operations and defuzzification is a single step calculation. Table 4 shows the computation cost of the classical fuzzy controller based on Mamdani inference model and the proposed arithmetic based controller. The conventional controller used Gaussian membership functions, product conjunction and implication operators, max aggregation and COG defuzzification. Different cases based on the number of fuzzy rules and the numerical resolution (n) of the output control surface have been considered. Each entry is the average result of 100 simulations. It is shown that when we have 8 fuzzy rules and a very high numerical resolution (i.e. $n = 1001 \times 1001$), the DF-based fuzzy arithmetic control technique is at least 21 times faster than the conventional Mamdani controller. As shown in Fig. 18, the adaptive algorithm converges in only 8 iterations. The proposed adaptive controller needs less computation time and it has a rapid convergence. For adaptation convergence, an upper bound on ϵ can be defined using the range of the input signal (i.e. $\frac{\text{Input range}}{4}$). In our simulations, the ϵ parameter always converged within a few iterations. New rules can be added in the knowledge base if the adaptation procedure fails to find the value of ϵ between the upper and lower bounds. This indicates that the process dynamics has changed significantly and existing rules cannot control the system even after adaptation.

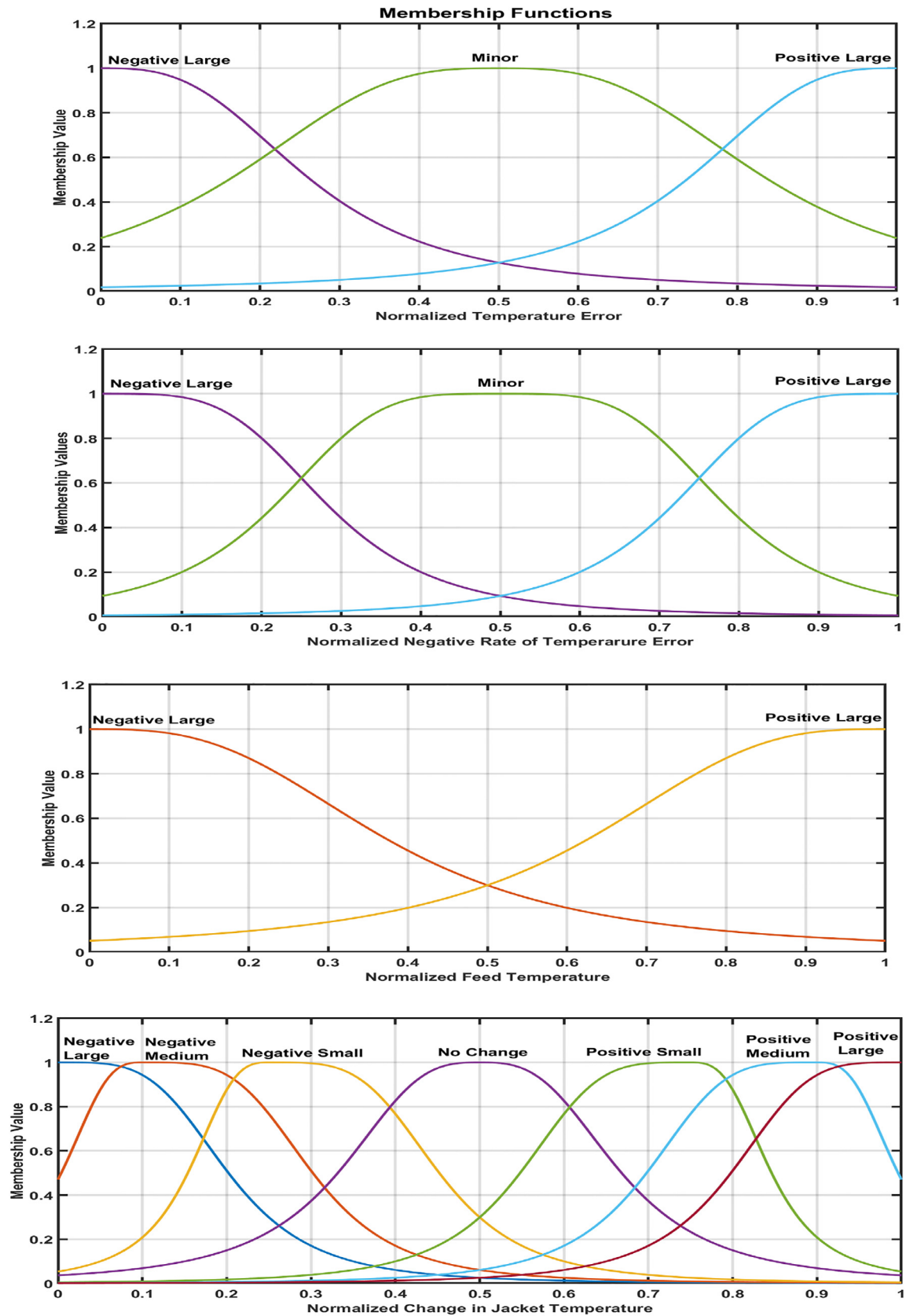


Fig. 16. The antecedent and consequent DFs of the CSTR controller.

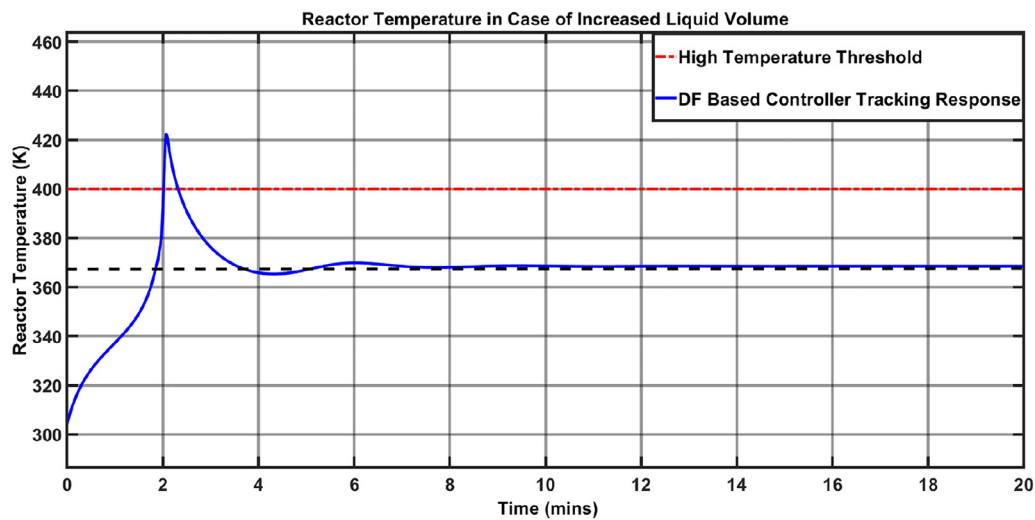


Fig. 17. The reactor temperature exceeded the high temperature threshold when the liquid volume increased.

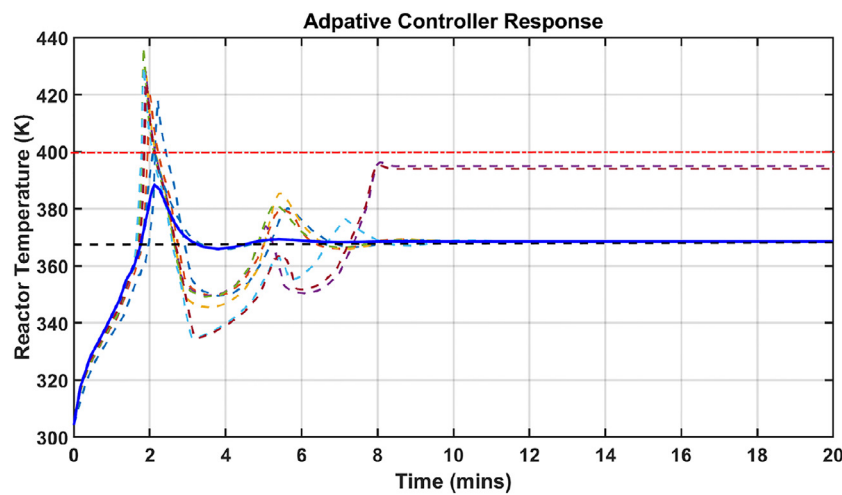


Fig. 18. Adaptive controller performance. The dashed lines shows the results of various iterations. The solid blue line shows the final response of the adaptive controller. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4

A speed comparison of a conventional Mamdani controller and Arithmetic-based controllers (for both symmetric and asymmetric types).

	$n = 11 \times 11$			$n = 101 \times 101$			$n = 1001 \times 1001$		
	3 Rules	5 Rules	8 Rules	3 Rules	5 Rules	8 Rules	3 Rules	5 Rules	8 Rules
Conventional Mamdani Inference	0.0712	0.0748	0.0932	5.2351	5.735	6.8722	20.68	22.34	27.30
Arithmetic Symmetric DF	0.00078	0.0013	0.0014	0.0950	0.1116	0.1150	1.184	1.2563	1.2787
Arithmetic Asymmetric DF	0.00079	0.0013	0.0015	0.0954	0.1131	0.1161	1.1895	1.2637	1.314
Speed ratio (Mamdani / Symmetric DF)	90	55	65	55	50	60	17	17	21

6. Conclusions

A novel technique for the design of a fuzzy controller has been proposed. It is based on a new type of parametric membership function called the DF. The DF is a continuous and differentiable function (it is analytical). It has a few parameters and it covers the input space with a few rules. A general parametric fuzzy operator is used to calculate the firing strengths. Also, the operator system and the DF are consistent with each other. The design process is simplified by handling the antecedent and consequent parts separately. The proposed approach does not include any type of implication. Aggregation is performed using fuzzy arithmetic operations,

more precisely using a linear combination of the DFs. The result of aggregation is also a DF. Also, defuzzification is just a single step calculation. The technique is simple, computationally efficient and overcomes some of the drawbacks with the existing established techniques.

Based on this new fuzzy control design technique and the gradient-based optimization method, a hybrid adaptive fuzzy controller is presented. It tunes the parameters of the DF using the gradient descent technique. The calculation is fast and the adaptation process converges within a few iterations. The adaptive fuzzy controller can satisfactorily achieve the objectives even when the process dynamics change. Computation efficiency is one of the

main advantages of the DF-based controller. It is 20 to 50 times faster than the conventional fuzzy controllers. Lastly, the effectiveness of the proposed approach has been demonstrated using a water tank system and a continuously stirred reactor system.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgments

This study was supported by the project “Integrated program for training new a generation of scientists in the fields of computer science”, no. EFOP-3.6.3-VEKOP-16-2017-0002. The project was supported by the European Union and co-funded by the European Social Fund.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.jprocont.2019.12.005](https://doi.org/10.1016/j.jprocont.2019.12.005).

References

- [1] L.A. Zadeh, Fuzzy sets, *Inf. Control* 8 (3) (1965) 338–353.
- [2] L.A. Zadeh, *Toward a theory of fuzzy systems*, 1432, National Aeronautics and Space Administration, 1969.
- [3] P. Melin, O. Castillo, J. Kacprzyk, M. Reformat, W. Melek, *Fuzzy logic in intelligent system design: theory and applications*, 648, Springer, 2017.
- [4] G.S. Atsalakis, I.G. Atsalaki, F. Pasiouras, C. Zopounidis, Bitcoin price forecasting with neuro-fuzzy techniques, *Eur. J. Oper. Res.* 276 (2) (2019) 770–780.
- [5] X. Xiang, C. Yu, L. Lapiere, J. Zhang, Q. Zhang, Survey on fuzzy-logic-based guidance and control of marine surface vehicles and underwater vehicles, *Int. J. Fuzzy Syst.* 20 (2) (2018) 572–586.
- [6] H. Zhou, J. Wang, H. Zhang, Multi-criteria decision-making approaches based on distance measures for linguistic hesitant fuzzy sets, *J. Oper. Res. Soc.* 69 (5) (2018) 661–675.
- [7] N. Wang, H.R. Karimi, Successive waypoints tracking of an underactuated surface vehicle, *IEEE Trans. Ind. Inf.* (2019) (2019), doi:[10.1109/TII.2019.2922823](https://doi.org/10.1109/TII.2019.2922823). Early Access.
- [8] N. Wang, H.R. Karimi, H. Li, S. Su, Accurate trajectory tracking of disturbed surface vehicles: a finite-time control approach, *IEEE/ASME Trans. Mechatron.* 24 (3) (2019) 1064–1074.
- [9] N. Wang, S. Su, X. Pan, X. Yu, G. Xie, Yaw-guided trajectory tracking control of an asymmetric underactuated surface vehicle, *IEEE Trans. Ind. Inform.* 15 (6) (2019) 3502–3513.
- [10] N. Wang, G. Xie, X. Pan, S. Su, Full-state regulation control of asymmetric underactuated surface vehicles, *IEEE Trans. Ind. Electron.* 66 (11) (2019) 8741–8750, doi:[10.1109/TIE.2018.2890500](https://doi.org/10.1109/TIE.2018.2890500).
- [11] D. Driankov, R. Palm, *Advances in Fuzzy Control*, 16, Physica, 2013.
- [12] Y. Li, S. Sui, S. Tong, Adaptive fuzzy control design for stochastic nonlinear switched systems with arbitrary switchings and unmodeled dynamics, *IEEE Trans. Cybernet.* 47 (2) (2016) 403–414.
- [13] N. Wang, S. Su, J. Yin, Z. Zheng, M.J. Er, Global asymptotic model-free trajectory-independent tracking control of an uncertain marine vehicle: an adaptive universe-based fuzzy control approach, *IEEE Trans. Fuzzy Syst.* 26 (3) (2017) 1613–1625.
- [14] N. Wang, J. Sun, M.J. Er, Tracking-error-based universal adaptive fuzzy control for output tracking of nonlinear systems with completely unknown dynamics, *IEEE Trans. Fuzzy Syst.* 26 (2) (2017) 869–883.
- [15] C. Caraveo, F. Valdez, O. Castillo, Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation, *Appl. Soft Comput.* 43 (2016) 131–142.
- [16] L. Amador-Angulo, O. Castillo, A new fuzzy bee colony optimization with dynamic adaptation of parameters using interval type-2 fuzzy logic for tuning fuzzy controllers, *Soft Comput.* 22 (2) (2018) 571–594.
- [17] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *Int. J. Man-Mach. Stud.* 7 (1) (1975) 1–13.
- [18] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst. Man Cybernet. SMC-15* (1) (1985) 116–132, doi:[10.1109/TSMC.1985.6313399](https://doi.org/10.1109/TSMC.1985.6313399).
- [19] S. Cao, N.W. Rees, G. Feng, Analysis and design of fuzzy control systems using dynamic fuzzy-state space models, *IEEE Trans. Fuzzy Syst.* 7 (2) (1999) 192–200.
- [20] D.N.M. Abadi, M.H. Khooban, Design of optimal mamdani-type fuzzy controller for nonholonomic wheeled mobile robots, *J. King Saud Univ.-Eng. Sci.* 27 (1) (2015) 92–100.
- [21] N.M. Raharja, Iswanto, O. Wahyunggoro, A.I. Cahyadi, Altitude control for quadrotor with mamdani fuzzy model, in: 2015 International Conference on Science in Information Technology (ICSITech), 2015, pp. 309–314, doi:[10.1109/ICSITech.2015.7407823](https://doi.org/10.1109/ICSITech.2015.7407823).
- [22] A. Piegat, *Fuzzy Modeling and Control*, 69, Physica, 2013.
- [23] Q. Gao, X. Zeng, G. Feng, Y. Wang, J. Qiu, Ts fuzzy model-based approximation and controller design for general nonlinear systems, *IEEE Trans. Syst. Man Cybernet. Part B (Cybernet.)* 42 (4) (2012) 1143–1154.
- [24] L. Kóczy, K. Hirota, Ordering, distance and closeness of fuzzy sets, *Fuzzy Set. Syst.* 59 (3) (1993) 281–293.
- [25] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning–i, *Inf. Sci.* 8 (3) (1975) 199–249.
- [26] D. Dubois, H. Prade, Operations on fuzzy numbers, *Int. J. Syst. Sci.* 9 (6) (1978) 613–626.
- [27] R. Jain, Tolerance analysis using fuzzy sets, *Int. J. Syst. Sci.* 7 (12) (1976) 1393–1401.
- [28] M. Mizumoto, K. Tanaka, The four operations of arithmetic on fuzzy numbers, *Syst. Comput. Control.* 7 (5) (1976) 73–81.
- [29] D. Dubois, H. Prade, The mean value of a fuzzy number, *Fuzzy Set. Syst.* 24 (3) (1987) 279–300.
- [30] R.C. Young, The algebra of many-valued quantities, *Mathematische Annalen* 104 (1) (1931) 260–290.
- [31] R. Mesiar, Triangular-norm-based addition of fuzzy intervals, *Fuzzy Set. Syst.* 91 (2) (1997) 231–237.
- [32] J. Dombi, T. Szepe, Arithmetic-based fuzzy control, *Iranian J. Fuzzy Syst.* 14 (4) (2017) 51–66.
- [33] J. Dombi, Towards a general class of operators for fuzzy systems, *IEEE Trans. Fuzzy Syst.* 16 (2) (2008) 477–484.
- [34] J. Hedengren, Simulink CSTR simulation and control, <https://www.mathworks.com/matlabcentral/fileexchange/48018-simulink-cstr-simulation-and-control>, Accessed: 2019-09-15.