1. Introduction

The type-1 (T1) fuzzy set (FS) can model a system based on a single expert’s knowledge. With the type-2 (T2) fuzzy set, introduced by Zadeh (1975) and pioneered by Mendel & John (2002), it is possible to accommodate the knowledge of several experts. In type-2 fuzzy systems, the membership functions (MFs) are themselves fuzzy instead of crisp as in type-1 fuzzy sets. Although T2 fuzzy sets possess better handling capabilities for uncertainties and imprecision, their real-world applications require complex practical considerations (Wu & Mendel, 2019). As an intermediate solution, the more practical model for handling uncertainties is the interval type-2 (IT2) fuzzy set. In interval type-2 fuzzy sets, the membership is an interval instead of T1 fuzzy sets as in traditional generalized type-2 fuzzy sets. In the recent past, the modeling of control systems using IT2 FS has attracted increased interest from the research community.

Liang & Mendel (2000) described the theory and the design steps for an interval type-2 fuzzy logic system. Several researches and applications for the IT2 controller were followed thereafter. Wu & Tan (2004) proposed a singleton interval type-2 fuzzy controller to design the liquid-level control. An interval type-2 fuzzy sliding mode controller (IT2FSMC) is proposed in (Hsiao et al., 2008). IT2FSMC is a combination of the interval type-2 fuzzy logic control (IT2FLC) and the sliding-mode control (SMC) which shares the benefits of these two methods. IT2FSMC is used in many control applications, for example, to control a 3-DOF helicopter (Zeghlache et al., 2017), to control a robot manipulator (Nafia et al., 2018) and so on. Using a genetic algorithm, an optimal type-2 fuzzy controller is implemented for the velocity regulation of a DC motor (Maldonado & Castillo, 2012). Behera et al. (2017) have applied an interval type-2 controller in the automatic generation control of a restructured power system. Control of a one-wheel vehicle in the real world has been developed using an adaptive interval type-2 fuzzy controller in (Chiu & Hung, 2020). Zakovorotniy & Kharchenko (2021) designed an optimal controller associated with speed control using the interval type-2 fuzzy set. The interval type-2 controller has also been designed based on the variations of the footprint of uncertainty. Zhou et al. (2021) have analyzed the variation trend of variable gains, in relation to the increase of footprint of uncertainty. Şahin & Ulu (2023) have proposed an IT2 controller with a dynamic footprint of uncertainty. The dynamic footprint of uncertainty has been defined as a function of system error. In (Şahin & Ulu, 2023), the IT2 controller with the dynamic footprint of uncertainty has been applied for controlling the altitude of a quadcopter.

One of the important tasks of an interval type-2 fuzzy set is to generate multiple type-1 fuzzy sets and then generate the upper and
lower membership functions from these type-1 membership functions by using some fuzzy operations. Similarly, an interval type-2 fuzzy logic controller should incorporate multiple type-1 fuzzy control strategies for the construction of the upper and lower membership functions. However, most of the IT2 FLC modeling found in literature either uses standard symmetrical equally spaced membership functions or the parameters of upper and lower membership functions are obtained by applying some optimization techniques. Not much attention is given to generating the primary membership functions to formulate the upper and lower membership functions. This paper addresses the above issue, and proposes a systematic method for generating two sets of membership functions for the interval type-2 controller. First, two primary membership functions are generated by using the clustering techniques and the differential evolution (Storn, 2008; Das & Suganthan, 2011) algorithm. Then, the upper and lower membership functions are generated by taking the fuzzy union and the fuzzy intersection of these two primary membership functions. This work addresses another challenge for interval type-2 controller, i.e., the number of rules. The proposed controller vividly reduces the required number of rules, thereby easing the computation. An armature-controlled DC motor is used to check the performance of the proposed controller. Noise and disturbance are added during simulation to check the robustness of the proposed controller. The remainder of this paper is structured as follows. Section 2 outlines the interval type-2 fuzzy sets and interval type-2 fuzzy logic controllers. Section 3 illustrates the proposed fuzzy controller. Section 4 provides the description of the applied process. The simulation results are presented and discussed in section 5. Section 6 includes the conclusion of this paper and refers to its possible future scope.

2. Preliminaries

2.1 Interval Type-2 Fuzzy Set

Unlike type-1 fuzzy sets, in IT2 FS (Mendel & Wu, 2010), each membership function consists of two MFs, i.e., upper membership function (UMF) and lower membership function (LMF). The area in the interval between the upper MF and the lower MF is called footprint of uncertainty (FOU). The FOU enables IT2 FS to represent the uncertainty in decision making. An example of an IT2 FS is shown in Figure 1 to indicate the different components of an IT2 FS.

![Figure 1. A schematic of an interval type-2 fuzzy set](https://www.sic.ici.ro)

The following definitions related to interval type-2 FS are used throughout the paper.

**Definition 1 [(IT2 FS)]**

A T2 FS $\tilde{A}$ is characterized by $\mu_\tilde{A}(x,m)$ and expressed by the following equation:

$$\tilde{A} = \int_{x \in X} \int_{m \in J_x} \mu_\tilde{A}(x,m) / (x,m)$$

(1)

Here, $X$ is the universe of discourse of the variable $x$, $m \in J_x \subseteq (0,1)$ is the primary membership value of $x$, and $\mu_\tilde{A}(x;m)$ denotes the secondary membership grade that is 1 for IT2 FS.

**Definition 2 [(FOU)]**

The FOU of $\tilde{A}$ is defined as the union of all of its primary MFs, and is expressed by equation (2):

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x = (x,m);$$

where $m \in J_x \subseteq [0,1]$.  

(2)

**Definition 3 [(Upper and lower MF)]**

The upper bound of $FOU(\tilde{A})$ is called the UMF and is denoted by $\mu_{\tilde{A}}(x)$. Similarly, the lower bound of $FOU(\tilde{A})$ is called LMF and is expressed by $\mu_\tilde{A}(x)$. That is,

$$UMF = FOU(\tilde{A}) \quad \text{for} \quad \forall x \in X.$$  

(3)

and

$$LMF = FOU(\tilde{A}) \quad \text{for} \quad \forall x \in X.$$  

(4)

2.2 Interval Type-2 Fuzzy Logic Controller

An interval type-2 fuzzy controller comprises four major building blocks, i.e. a fuzzifier,
Design of an Improved Interval Type-2 Controller Using FCM and Supervised Clustering Algorithms

3. The Proposed Interval Type-2 Controller

The proposed interval type-2 fuzzy controller is developed in the following three stages, namely generation of membership functions, formation of a fuzzy inference engine, and defuzzification.

3.1 Generation of Membership Functions

The common membership functions used in FLCs are the Gaussian, triangular, and trapezoidal one. For the proposed IT2 controller, the Gaussian membership functions are considered for two reasons. First, the Gaussian membership function requires a lower number of parameters in comparison with other membership functions. Secondly, the proposed controller generates membership functions based on traditional and an improved fuzzy c-means clustering, and the shape of the Gaussian function features a higher suitability or proximity to the FCM-like clustering algorithm. Two MFs are generated for each subset of antecedents and the consequence. A detailed step-by-step description of the membership functions’ generation is provided below.

3.1.1 Generation of Control Surface

The control surface consists of error ($e$), change of error ($\Delta e$), and control action ($u$). The MFs of $e$, $\Delta e$ and $u$ are defined in the common interval of $[-1, 1]$ as it is shown in Figure 3. This is one of the unbiased & natural ways of MF distribution. The input space ($e$, $\Delta e$) is uniformly sampled in 1,000 points in the interval of $[-1, 1]$.

Then the respective control actions for the sampled input space are calculated by the modified version of the original MacVicar-Whelan rule base (MacVicar-Whelan, 1976) with the 7 linguistic variables listed in Table 1.

Table 1. Rule base for generating the control action

<table>
<thead>
<tr>
<th>$\Delta e/e$</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>NM</td>
<td>NB</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>ZE</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>ZE</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>PS</td>
<td>NM</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>PM</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

By combining, ($e$, $\Delta e$) with the control action $u$ obtained in Table 1, a control surface is generated as it is illustrated in Figure 4.
3.1.2 Clustering of the Control Surface

Next, the control surface is clustered into $c$ number of clusters. As each process input $e$ and $\Delta e$ can be divided into 3 major groups, i.e. negative, zero, & positive, $c$ is selected as $3 \times 3 = 9$ to cover the control surface significantly. The control surface is first clustered into 9 groups by traditional fuzzy $c$-means clustering. It provides one set of means for the Gaussian membership functions. A second set of cluster centers is obtained by using a newly developed supervised clustering algorithm. The details on supervised clustering can be found in (Bandyopadhyay et al., 2007; Mukhopadhyay & Maulik, 2009). The two major steps of supervised clustering are to first segregate the core and boundary data points, and then to apply supervised learning in clustering. From the membership matrix of FCM, the membership values of the data points for all the clusters are found. Among the membership values of a data point for all the clusters, let us assume that the highest membership of a data point ($x_i$) is represented by $\mu_{ik}$ and the second highest membership is denoted by $\mu_{ik}^\Delta$ and the difference between these two membership values is represented by $\Delta \mu_i$. Data points with a higher $\Delta \mu$ are assumed to be near a cluster center and are designated as core data points. On the other hand, the data points with a lower $\Delta \mu_i$ are considered to be at the overlap of at least two clusters and are defined as boundary data points. The membership-related threshold for deciding if a data point belongs either to the core data set or to the boundary data set plays a significant role in the final cluster solution. Mallick & Mukhopadhyay (2018) proposed a scheme, namely Clusterwise $mean$ to separate the core data points and boundary data points. In Clusterwise $mean$, the core data set is formed by taking core data points from each cluster. Hence, the membership-related threshold is selected cluster-wise. In this method, the threshold of a cluster is the mean value of membership differences $\Delta \mu_i$ for all the data points belonging to that cluster only, and is determined by equation (5).

$$
\Delta \mu_{\text{threshold}} (j) = \frac{1}{n_j \times \mu_{\text{core}}^j} \sum \Delta \mu_p
$$

(5)

Here, $C_i$ denotes the $i^{th}$ cluster, $n_i$ is the number of data points in $i^{th}$ cluster, $p$ is the index of data points belonging to $j^{th}$ cluster, $i$ is a set given by $\{1,2,3,...,n\}$, and $n$ is the number of all data points. The data points with $\Delta \mu_i \geq \Delta \mu_{\text{threshold}}$ of that cluster are regarded as core data. The overall core data set is formed by amalgamating the core data points coming from each cluster. After the segregation of core and boundary data points, the class labels of the boundary data points are predicted by a $k$-$nn$ classifier based on the class labels of the core data points. In this paper, the value of $k$ in the $k$-$nn$ classifier is set to 5. The final cluster centers are found by averaging the data points belonging to that cluster. The cluster centers obtained by traditional FCM and by Clusterwise $mean$ provide two sets of means for the primary Gaussian membership functions.

3.1.3 Selection of the Standard Deviations of the Primary MFs

The standard deviations of the Gaussian membership functions are optimally selected by differential evolution (DE). Here, each chromosome consists of 27 genes (9 genes for the standard deviation of the $e$ component, the next 9 genes for the $\Delta e$ component, and the last 9 genes for the $u$ component). In each run, DE optimizes the value of standard deviations by minimizing the sum of squared errors (SSE) of the data points of the control surface.

$$
SSE = \sum_{i=1}^{k} (\text{act}(u_i) - \text{est}(u_i))^2
$$

(6)

Here, $k$ is the number of data points, $\text{est}(u_i)$ is the estimated control action for a population with a certain set of standard deviations, and $\text{act}(u_i)$ is the output found from the control surface. The best solution found at the last iteration of DE is translated as the standard deviations of the respective Gaussian MFs. The pseudo-code of the differential evolution algorithm used here is given in Algorithm 1.

3.1.4 Formation of $UMF$ and $LMF$

As the control surface is clustered into 9 groups, each antecedent ($e$, $\Delta e$), and consequence ($u$) consist of 9 membership functions. Two sets of primary Gaussian MFs for the antecedents and the consequence are formed by considering cluster centers as the means, and the solution of differential evolution as the standard deviations. These two Gaussian membership functions shall be defined as $GMF_1$ and $GMF_2$. Then the upper membership functions ($UMFs$) are generated by using the fuzzy MAX operation over $GMF_1$ and $GMF_2$ as it is given in equation (7).

$$
UMF = \max_{\forall i} \mu_{GMF_1}(x(i)), \mu_{GMF_2}(x(i))
$$

(7)

for $\forall x \in X$
Algorithm 1: Pseudocode to generate standard deviations of the Gaussian MFs

Require: Control surface data, \textit{popsize}, \textit{Maxiter}, means of the Gaussian MFs

Ensure: Standard deviations of the Gaussian MFs

Randomly generate initial population \( x_{i,G} \), \( i=1, 2, \ldots, \text{popsize} \)

Evaluate cost \( f(x_{i,G}) \) using procedure \( \text{SSE} \)

Find the best population \( \text{pbest} \) of the population \( x_{i,G} \)

\( G \leftarrow 0 \)

while \( G < \text{Maxiter} \) do

\( G \leftarrow G + 1 \)

for \( i = 1 \) to \( \text{popsize} \) do

Select randomly \( r1 \) and \( r2 \) such that \( i \neq r1 \neq r2 \)

Generate \( F \) from normal distribution

\( v_{i,G+1} = x_{i,G} + F \left( \text{pbest} - x_{i,G} + x_{r1,G} - x_{r2,G} \right) \)

Calculate cost \( f(v_{i,G+1}) \)

if \( f(v_{i,G+1}) < f(x_{i,G}) \) then

\( x_{i,G+1} = v_{i,G+1} \)

else \( x_{i,G+1} = x_{i,G} \)

end

end

Find the \( \text{pbest} \) of the population \( x_{i,G+1} \)

end

Procedure \( \text{SSE}(f) \)

Formulate UMFs and LMFs using Means and translating genes of the chromosome as the standard deviations

Evaluate the estimated outputs \( (u) \) for all the input data points \( (e, \Delta e) \) of the control surface

Calculate \( f = \sum_{i=1}^{k1} \left( \text{act}(u_i) - \text{est}(u_i) \right)^2 \);

\( k1: \) no. of data points on the control surface

Return \( f \)

End

Figure 5. Membership functions (MFs) of the proposed IT2 FLC: UMFs of (a) error \( (e) \); (b) change of error \( (\Delta e) \) and (c) control action \( (u) \); LMFs of (d) error \( (e) \); (e) change of error \( (\Delta e) \) and (f) control action \( (u) \)
Similarly, lower membership functions (LMFs) are obtained by taking the minimum of \(GMF_1\) and \(GMF_2\) as in equation (8).

\[
LMF = \min_{\forall i} \mu_{GMF_1}(x(i)), \mu_{GMF_2}(x(i))
\]

\[
\text{for } \quad \forall x \in X
\]

The upper and lower MFs obtained for the antecedents and the consequence of the proposed IT2 FLC are shown in Figure 5.

### 3.2. Fuzzy Inference System

Let us assume that the process input at an instant be denoted by \(e_{in}\) and \(\Delta e_{in}\). The membership intervals of \(e_{in}\) are computed by means of projection of \(e_{in}\) to the \(e\) component of all the 9 rules. Similarly, the membership intervals of \(\Delta e_{in}\) are also calculated. Then the firing interval \([f^{kL}, f^{kU}]\) at \(k\)th rule is estimated by minimum \(t\)-norm. In Example 1, the procedure of the fuzzy inference system is illustrated.

**Example 1.** Let the projections of \(e_{in}\) on the \(k\)th LMF and \(k\)th UMF be expressed by \(fL1\) and \(fU1\), respectively, and the positions of \(\Delta e_{in}\) on the \(k\)th LMF and \(k\)th UMF be expressed by \(fL2\) and \(fU2\), respectively. Then the firing strength of \(LMF\) of \(k\)th rule denoted by \(f^k\) is calculated as it is given in equation (9).

\[
f^k = \min(fL1, fL2)
\]

Similarly, the firing strength of \(UMF\) of \(k\)th rule expressed by \(f^k\) is estimated by equation (10).

\[
f^k = \min(fU1, fU2)
\]

Next, the inferences due to firing of \(k\)th rule are obtained from the intersection of \(f^k\) with \(LMF_k(\tilde{u})\), and from the intersection of \(f^k\) with \(UMF_k(\tilde{u})\). Let the consequence fuzzy output of \(LMF\) and \(UMF\) of \(k\)th rule be denoted by \(LMF^k\) and \(UMF^k\), respectively. Thus,

\[
LMF^k = \min(f^k, LMF_k(\tilde{u}))
\]

and

\[
UMF^k = \min(f^k, UMF_k(\tilde{u})).
\]

### 3.3. Defuzzification

The total output of \(LMF\) (and \(UMF\)) for a discrete output point is obtained by \(MAX\) aggression of all the rules. Let the output \(LMF\) and \(UMF\) at \(n\)th data point be denoted by \(\bar{r}^n\) and \(\bar{r}^n\), respectively.

Then,

\[
\bar{r}^n = \max(LMF^1, ..., LMF^k, ..., LMF^N),
\]

and

\[
\bar{r}^n = \max(UMF^1, ..., UMF^k, ..., UMF^N),
\]

where \(N\) is the total number of rules. Next, the fuzzy output is defuzzified. In this paper, the Nie-Tan method of defuzzification (Nie & Tan, 2008; Wu, 2012) was used. In this method, the crisp output \(y\) of an IT2 FLC is estimated by the following in equation (15).

\[
y = \frac{\sum_{n=1}^{P} y_n (\bar{r}^n + \bar{r}^n)}{\sum_{n=1}^{P} (\bar{r}^n + \bar{r}^n)}
\]

where \(P\) is the number of output discrete points.

### 4. The Applied Process

The armature-controlled DC motor is widely used in many industrial applications. Hence, researchers still use DC motors as a process for validating their proposed controller (Sankeshwari & Chillé, 2019; Li et al., 2021). The block diagram of the armature-controlled DC motor is shown in Figure 6 and the transfer function of the armature-controlled DC motor is given by equation (16).

\[
\frac{\theta_m(s)}{E_a(s)} = \frac{K_f}{R_a L_a s^3 + (R_a J + L_a B) s^2 + (R_a B + K_a K_f) s}
\]

![Figure 6](https://www.sic.ici.ro)

Here, \(\theta_m\): angular position of the motor shaft, \(E_a\): applied voltage, \(K_f\): torque constant of motor, \(J\): the moment of inertia, \(K_a\): voltage constant of the motor, \(B\): viscous friction coefficient, \(L_a\): armature inductance, and \(R_a\): armature resistance. In an armature-controlled DC motor, the armature inductance is negligibly small compared to armature resistance (Mandal, 2014). Thus,
the transfer function in equation (16) can be approximated as equation (17).

\[
\frac{\theta_a(s)}{E_a(s)} = \frac{K_T / R_a J}{s^2 + (B / J + K_a K_T / R_a J)s}
\]  

(17)

## 5. Results and Discussion

For uniformity purposes, all the simulations were carried out by a fourth-order Runge-Kutta solver with a fixed interval of 0.01 seconds. In order to illustrate the performance of the proposed model, a conventional IT2 FLC is considered for comparison purposes. The membership functions of the conventional IT2 FLC are considered to be symmetrical triangles (except for the two MFs at the extreme ends) as it is shown in Figure 7.

The simulation is done in Matlab R2016a for the block diagram shown in Figure 8. For an unbiased comparison, the values of all the gain parameters, namely \(G_e\), \(G_{\Delta e}\) and \(G_{PD}\) in Figure 8, were optimized using differential evolution (DE) for both the proposed model and for the conventional fuzzy logic controller. In DE, Integral square error (ISE) is considered as the objective function given in equation (18).

\[
ISE = \sum_{i=1}^{nL} e^2 (i),
\]  

(18)

where \(e(i)\) is the error, i.e. the difference between the desired step response and the actual step response at the \(i\)th discrete point for a set of \([G_e, G_{\Delta e}, G_{PD}]\); \(n\) is the number of discrete data points, and \(L\) is the sampling time, i.e. 0.01 seconds in this paper. Here, \(n\) is considered to be 3000, as after 3 seconds the fluctuation in the response reduces sufficiently. The algorithm of differential evolution used here for the optimized selections of the gain parameters is similar to Algorithm 1, except that the cost function is calculated using equation (18).

Figure 7. MFs of conventional IT2 controller; solid lines: UMFs, dashed lines: LMF of \(e/\Delta e/ u\)

The value of \(R_a\) is set at 0.4 ohm, that of \(J\) at 0.5 kg-m2, that of \(K_b\) at 1.25 V-s/rad, the value of \(K_T\) at 1 N-m/A, and that of \(B\) at 0.01N-m/rad (Rahman, 2017). The simulation results for the step response with the conventional IT2 controller and with the proposed controller are shown in Figure 9(a) and in Figure 9(b), respectively. From Figure 9, it can be seen that the proposed IT2 controller has a faster response with a lower overshoot, and also settles earlier in comparison with the conventional IT2 controller. The robustness of the proposed controller is checked in two ways; by applying a random noise, and by adding a sudden impulse disturbance. The mean and the variance of the random noise are 0 and 0.1, respectively, and the applied disturbance is of 30-unit impulse.

Figure 8. Matlab simulation block diagram

Figure 9. Step responses of the DC motor in the no noise and no disturbance conditions for (a) the conventional IT2 controller; (b) the proposed IT2 controller
The simulation results for the step response in the presence of the noise and disturbance are shown in Figure 10 and in Figure 11, respectively.

![Figure 10](image1)

**Figure 10.** Step responses of the DC motor in the presence of random noise with a variance of 0.1 for (a) the conventional IT2 controller; (b) the proposed IT2 controller

From Figure 10, it appears that the fluctuation in response due to the random noise for the proposed controller is much lower than for the conventional controller. Figure 11 indicates that the proposed controller is more aggressive than the conventional controller at the event of disturbance. During the simulation, it was found that the conventional controller required 0.54 seconds, whereas the proposed controller takes only 0.06 seconds to settle after applying the disturbance. It was also seen that the proposed controller features a lower overshoot (2.81%) in comparison with an overshoot of 9.89% for the conventional controller due to the disturbance.

The values of different time domain performance indices, namely rise time ($t_r$), peak overshoot ($M_p$), settling time ($t_s$), integral absolute Error ($IAE$), integral time absolute error ($ITAE$), integral square error ($ISE$), and integral time squared error ($ITSE$) are included in Table 2.

<table>
<thead>
<tr>
<th>Noise/Disturbance</th>
<th>Time domain performance index</th>
<th>Conventional</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>No noise and no disturbance applied</td>
<td>$t_r$ (sec.)</td>
<td>1.22</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>$M_p$ %</td>
<td>4.59624</td>
<td>4.29711</td>
</tr>
<tr>
<td></td>
<td>$t_s$ (sec.)</td>
<td>1.44</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>$IAE$</td>
<td>0.68807</td>
<td>0.28447</td>
</tr>
<tr>
<td></td>
<td>$ITAE$</td>
<td>0.29912</td>
<td>0.08630</td>
</tr>
<tr>
<td></td>
<td>$ISE$</td>
<td>0.49016</td>
<td>0.19074</td>
</tr>
<tr>
<td></td>
<td>$ITSE$</td>
<td>0.15704</td>
<td>0.02434</td>
</tr>
<tr>
<td>Random noise with a variance of 0.1</td>
<td>$t_r$ (sec.)</td>
<td>1.21</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>$M_p$ %</td>
<td>4.80191</td>
<td>4.28470</td>
</tr>
<tr>
<td></td>
<td>$t_s$ (sec.)</td>
<td>1.42</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>$IAE$</td>
<td>0.68119</td>
<td>0.28314</td>
</tr>
<tr>
<td></td>
<td>$ITAE$</td>
<td>0.30557</td>
<td>0.08655</td>
</tr>
<tr>
<td></td>
<td>$ISE$</td>
<td>0.48081</td>
<td>0.18938</td>
</tr>
<tr>
<td></td>
<td>$ITSE$</td>
<td>0.15218</td>
<td>0.02408</td>
</tr>
<tr>
<td>Unit impulse disturbance applied at 2.5 seconds</td>
<td>$t_r$ (sec.)</td>
<td>1.22</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>$M_p$ %</td>
<td>4.59624</td>
<td>4.29711</td>
</tr>
<tr>
<td></td>
<td>$t_s$ (sec.)</td>
<td>1.44</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>$IAE$</td>
<td>0.72002</td>
<td>0.28640</td>
</tr>
<tr>
<td></td>
<td>$ITAE$</td>
<td>0.38856</td>
<td>0.09130</td>
</tr>
<tr>
<td></td>
<td>$ISE$</td>
<td>0.49220</td>
<td>0.19079</td>
</tr>
<tr>
<td></td>
<td>$ITSE$</td>
<td>0.16255</td>
<td>0.02445</td>
</tr>
</tbody>
</table>

From Table 2, it appears that the proposed IT2 controller outperforms the conventional
IT2 controller for all the performance indices considered here.

6. Conclusion and Future Scope

This paper prescribes a systematic approach to modelling an interval type-2 fuzzy logic controller. It proposes a way to generate primary type-1 membership functions in order to formulate the upper and the lower membership functions for the interval type-2 controller. The number of rules is also reduced from 49 for the conventional IT2 controller to only 9 for the proposed IT2 controller. From the obtained simulation results, it can be seen that the proposed controller outperforms the conventional IT2 controller, and that it also shows robustness to the noise and the applied disturbance.

The present work considers the shape of primary membership functions as being Gaussian. Further research may also be conducted with the purpose of generating an IT2 controller with triangular or trapezoidal membership functions.

REFERENCES


