Journal of Intelligent and Fuzzy Systems 1 (2019) 1–15
IOS Press

Developing a Computationally Effective Interval Type-2 TSK Fuzzy Logic Controller

Abel Hailemichael [a], Syed Moshfeq Salaken [b], Ali Karimoddini [a,*], Abdollah Homaifar [a], Abbas Khosravi [b] and Saeid Nahavandi [b]

[a] *Department of Electrical and Computer Engineering, North Carolina Agricultural and Technical State University, Greensboro, NC 27411 USA* [**]

[b] *Institute of Intelligent Systems Research and Innovation, Deakin University, Victoria, Australia*

Abstract. Type-2 fuzzy logic controllers are capable of handling different types of uncertainties that naturally exist in most practical situations. However, the high computation cost of type-2 fuzzy logic controllers is a bottleneck for practically applying them to real-world applications. This paper introduces a novel approach for designing a computationally effective type-2 fuzzy logic controller. For this purpose, on the antecedent side, interval type-2 fuzzy sets are employed to capture the signal readings, which significantly reduce the computation costs while preserving the major advantages of general type-2 fuzzy logic systems. On the consequent side, however, the Takagi-Sugeno-Kang (TSK) technique is integrated with the proposed controller to render the control outputs in a parallel way. To further reduce the computation cost, the theory of uncertainty bounds is employed for the output processing of the proposed controller. To develop this control structure, a decomposition technique is integrated to break down the original type-2 fuzzy processes into type-1 and take advantage of type-1 fuzzy techniques, followed by an aggregation mechanism to calculate the collective output. The approach is applied to the control of an inverted pendulum and cart model. The simulation results of the developed interval type-2 fuzzy logic controller is compared with a type-1 TSK fuzzy logic controller and a classical proportional derivative (PD) controller. From the results, we have found a 16.6% and 23.3% improvement in Root Mean Square (RMS) error compared to type-1 TSK fuzzy logic controller and classical PD controller, respectively.

Keywords: Interval Type-2 Fuzzy Logic Controller, TSK Fuzzy Logic Controller, Uncertainty Bounds Output Processing.

1. INTRODUCTION

When practically implementing a controller, real-world uncertainties challenge the modelling, analysis, and performance of controlled systems. The uncertainties may rise from several sources such as noise, precision of sensors and actuators as well as environmental conditions. Type-2 fuzzy logic systems (FLSs), initially developed by Zadeh [1, 2], are the extended versions of type-1 FLSs and are unique in their ability to model and handle uncertainties [3–5], while being able to deal with complex control structures and linguistic variables.

Nevertheless, despite the capabilities of a type-2 FLSs, efforts to use them for real-time control applications is relatively sparse and just recent [6–13]. The main challenge which hinders the application of type-2 FLSs for control applications is their large computation cost [14–20]. This computational cost can be best explained by the "curse of dimensionality", the term

[*]Corresponding author: A. Karimoddini, Tel: +13362853313, akarimod@ncat.edu.

[**]This work is supported by Air Force Research Laboratory and Office of the Secretary of Defense under agreement number FA8750-15-2-0116, and US ARMY Research Office under agreement number W911NF-16-1-0489.

used for the explosion of combinatorial calculations within the controller [21].

To facilitate the use of type-2 FLSs for control applications and reduce their computation cost, in this paper, we propose an interval type-2 Takagi-Sugeno-Kang (TSK) fuzzy logic controller (FLC) [22] incorporated with an aggregation approach inspired by the *uncertainty bounds* method. Interval type-2 FLSs [2, 21, 23–26] reduce the computational complexity of the general type-2 FLSs, while preserving the major advantages of general type-2 FLSs, particularly, the ability of handling uncertainty [27–29]. To further reduce the computation cost, the proposed controller is integrated with the TSK technique [30–32]. By expressing fuzzy rule outputs as a function of the inputs, the TSK technique improves the processing time of FLSs and enables the parallel processing of the fuzzy rule outputs [30, 33]. Additional reduction in computation cost will be achieved by proposing an aggregation mechanism, which adopts the uncertainty bounds technique [34, 35]. The proposed algorithm ultimately requires little information (only the upper and lower bounds for the input membership functions and rule output constants) which significantly reduces the computation cost for calculating the output control signal.

Thus, the contributions of this paper are the development of a computationally effective interval type-2 fuzzy logic control structure that is capable of capturing input uncertainties and the development of an algorithm that can be used for the implementation of the proposed control structure. To capture input uncertainties, the proposed control structure utilizes interval type-2 fuzzy sets for describing the input space. Additionally, the control structure integrates interval type-2 antecedents with TSK rule outputs to enable parallel processing of rule outputs. Unlike many other methods, the proposed technique does not require type reduction of the output fuzzy set since it directly generates a type-1 output fuzzy set. By using only the upper and lower bounds of the firing levels and rule outputs coefficients, the proposed technique is able to calculate the output control signals. Compared to our preliminary results in [26], the proposed algorithm in this paper is more efficient as, instead of searching over all combination of lower and upper membership values to find the bounds of the output interval, we have integrated the proposed method with the uncertainty bounds aggregation method. Furthermore, to evaluate the performance of the proposed controller and its ability to capture input uncertainties, the fuzzy type-2 controller is applied to the position control of an inverted pendulum on a moving cart in a noisy environment. The simulation results of the proposed IT2 TSK FLC are then compared with a type-1 TSK FLC and a classical proportional derivative (PD) controller.

The rest of this paper is organized as follows. Section 2 briefly discusses preliminaries and notations. Section 3 details the development of the proposed fuzzy type-2 controller and discusses its different parts including the input processing, rule sets, output processing, and uncertainty bounds aggregation. Section 4 demonstrates the simulation of an inverted pendulum on a moving using the proposed IT2 TSK FLC, a type-1 TSK FLC, and a proportional derivative (PD) controller. The paper is concluded in Section 5.

2. PRELIMINARIES

The contribution of this paper is two fold. One, the paper designs an IT2 TSK FLC that has a satisfactory performance compared to the baseline controllers; and second, the paper presents a new decomposition method to easily incorporate an IT2 TSK FLC into the overall control design methodology. The proposed decomposition method also helps with explaining the operations of a complex fuzzy model. Before, discussing this decomposition technique, we first briefly reviews different types of fuzzy sets (FSs), particularly Interval Type-2 fuzzy sets, which are used throughout this paper. Some of the notations are borrowed from [36] and [23].

Definition 1. *Type-1 Fuzzy Set*

A type-1 fuzzy set is composed of pairs of $(x, \mu_A(x))$*, in which for each member of domain,* $x \in X$*, a membership value* $\mu_A(x) \in [0, 1]$ *can be defined as fol-*

lows:

$$A = \{(x, \mu_A(x)) | \forall x \in X, \mu_A(x) \in [0,1]\} = \sum_{x \in X} (x, \mu_A(x)) \quad (1)$$

Here, X is the universe of discourse and $\sum$ is the collection of elements of the set.

Even though type-1 fuzzy sets provide a degree of membership for all their elements, they are not capable of quantifying the level of uncertainty in the degree of membership [21]. The uncertainty in the degree of membership can be quantified using type-2 fuzzy sets [37–42]. Type-2 fuzzy sets can be defined as follows:

Definition 2. *Type-2 Fuzzy Set*

A type-2 fuzzy set is composed of triples $((x,u),\ \mu_{\tilde{A}}(x,u))$ in which for each member of domain $x \in X$, there exists a primary membership value, $u \in J_x$ (J_x is the range of primary membership for a given x) and a secondary membership, $\mu_{\tilde{A}}(x,u)$. Mathematically, a type-2 fuzzy set can be defined as follows:

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \subseteq [0,1], \mu_{\tilde{A}}(x,u) \in [0,1]\} = \sum_{u \in J_x} \sum_{x \in X} ((x,u), \mu_{\tilde{A}}(x,u)) \quad (2)$$

In general, using type-2 FLSs is computationally expensive. For reducing this computational burden, it is common to use interval type-2 FLSs. Interval type-2 FLSs significantly reduce the computation costs while maintaining major advantages of type-2 FLSs [27,29].

Definition 3. *Interval Type-2 Fuzzy Set (IT2 FS)*

An interval type-2 fuzzy set is a type-2 fuzzy set in which the secondary membership values of it's elements are always unity and is defined as:

$$\tilde{A} = \{((x,u), 1) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\} = \sum_{u \in J_x} \sum_{x \in X} ((x,u), 1) \quad (3)$$

IT2 FSs can be further simplified by employing embedded IT2 FSs and embedded type-1 FSs, which next will be defined.

Definition 4. *Embedded Interval Type-2 FS (EIT2 FS)*

An embedded interval type-2 FS is an interval type-2 FS in which for all $x \in X$, the primary membership is a single value:

$$\tilde{A}_e = \{((x,u_x), 1) | \forall x \in X, \exists! u_x \in J_x \subseteq [0,1]\} = \sum_{x \in X} ((x,u_x), 1) \quad (4)$$

The following lemma describes that an IT2 FS can be expressed as the union of EIT2 FSs.

Lemma 1. *(IT2 to EIT2 FSs)*

Any IT2 FS can be described by the collection of (infinite) EIT2 FSs as $\tilde{A} = \sum_j \tilde{A}_e^j$ if for any $((x,u_x),1) \in \tilde{A}$, there exists a j, such that $((x,u_x),1) \in \tilde{A}_e^j$.

Proof. Since any member of $\tilde{A}$ has been included at least in one of $\tilde{A}_e^j$, it follows that the collection of $\tilde{A}_e^j$ results in $\tilde{A}$. □

We can further simplify EIT2 FSs by employing type-1 FSs, which can be defined as follows:

Definition 5. *Embedded Type-1 FSs (ET1 FSs)*

An embedded type-1 FS is an embedded interval type-2 FS in which the secondary membership values are dropped.

$$A_e = \{(x,u_x) | \forall x \in X, \exists! u_x \in J_x \subseteq [0,1]\} = \sum_{x \in X} (x,u_x) \quad (5)$$

Lemma 2. *(EIT2 to ET1 FSs)*

Any EIT2 FS can be expressed using its corresponding ET1 FS as $\tilde{A}_e = A_e \times 1$, where $\times$ denotes the cartesian product.

Combining Lemmas 1 and 2, the following theorem describes IT2 FSs based on ET1 FSs:

Theorem 1. *(Decomposition of IT2 into ET1 FSs)*

Any IT2 FS can be described by the collection of *(infinite) ET1 FSs as* $\tilde{A} = \sum_j A_e^j \times 1$ *if for any* $((x, u_x), 1) \in \tilde{A}$*, there exists a* j*, such that* $(x, u_x) \in A_e^j$.

Proof. From Lemma 1, we know that $\tilde{A}$ can be described by a collection of (infinite) EIT2 FSs as $\tilde{A} = \sum_j \tilde{A}_e^j$. Based on Lemma 2, each EIT2 FSs can be expressed using its corresponding ET1 FSs as $\tilde{A}_e^j = A_e^j \times 1$. Substituting this value of $\tilde{A}_e^j$, we obtain $\tilde{A} = \sum_j \tilde{A}_e^j = \sum_j A_e^j \times 1$. □

By using the above theorem, we can decompose an IT2 FS into type-1 FSs.

3. DEVELOPING THE PROPOSED IT2 TSK FUZZY CONTROLLER

This section, discusses the internal structure and block diagrams of the proposed controller in detail. Figure 1 shows a typical feedback control loop in which we have used our developed IT2 TSK fuzzy logic controller.

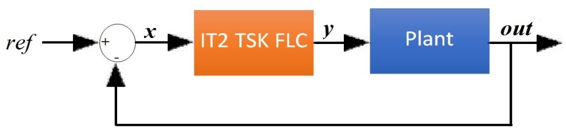


Fig. 1. Feedback control using IT2 TSK FLC.

3.1. Interval Type-2 TSK FLC Structure

The basic idea for developing the proposed IT2 TSK FLC is to use Theorem 1 to decompose interval type-2 fuzzy sets into type-1 fuzzy sets for which we can employ well-matured control techniques such as TSK FLCs to design the controller.

The block diagram of the proposed type-2 TSK FLC is shown in Fig. 2. For any set of crisp inputs, the *Fuzzifier* block converts the crisp inputs into fuzzy inputs. The *Rule Base* block contains a set of rules in the form of fuzzy *If-Then* statements, relating the inputs and outputs. Once the inputs are fuzzified, the *Firing Level* block calculates the firing levels based on the predefined input interval type-2 membership functions and the rule base. In parallel, the *Rule Output* block takes the crisp inputs and calculates rule outputs, based on the output membership functions and the rule base. The *Aggregator* block combines the firing levels and rule outputs into an aggregated type-1 fuzzy set. Finally, the *Defuzzifier* block converts the obtained type-1 fuzzy set into a crisp output value. This control process is detailed in the following sections.

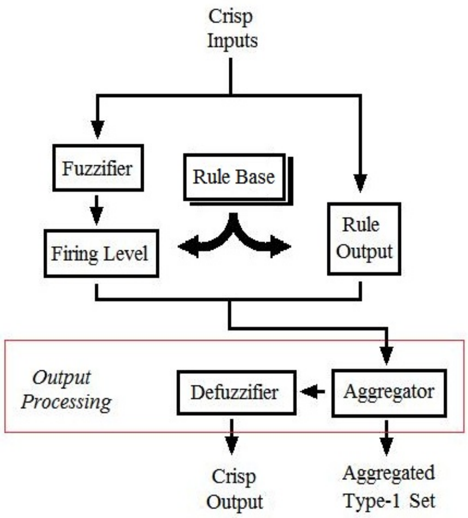


Fig. 2. The proposed IT2 TSK FLC structure.

3.2. Interval Type-2 Fuzzification

3.2.1. Membership Decomposition

The proposed IT2 FLC perceives the environment from a fuzzy perspective. The *Fuzzifier* block converts the control inputs (sensor readings), which are crisp values, to fuzzy values using the predefined fuzzy type-2 memberships. Here, we use IT2 fuzzy sets to describe the input space. Each input channel, X_i, can be captured by n_i membership functions (MFs) as follows:

$$F_i = \sum_{j=1}^{n_i} F_{i,j} \tag{6}$$

where F_i represents all interval type-2 membership functions related to input channel X_i, including $F_{i,1}$,

$F_{i,2}$,..., F_{i,n_i}. The subscript j in $F_{i,j}$ is used to indicate the jth individual MF in F_i.

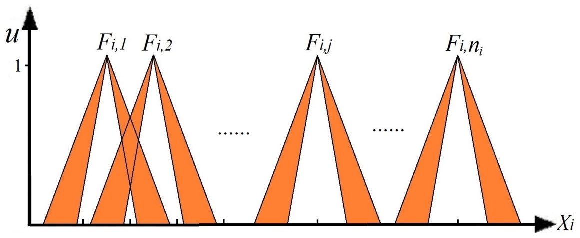


Fig. 3. IT2 MFs for input channel x_i.

For simplicity, the secondary membership values are not shown as they are always unity. The distance between the upper and lower MFs represents the measurement uncertainty of the inputs and is referred as Footprint of Uncertainty (FOU).

Using lemma 1, each of the input interval type-2 MFs, $F_{i,j}$, can be broken down into a collection of EIT2 membership sets $F_{i,j}^k$ as follows:

$$F_{i,j} = \sum_{k=1}^{n_{i,j}} F_{i,j}^k \tag{7}$$

where $F_{i,j}^k$ represents a particular EIT2 MF in $F_{i,j}$, and $n_{i,j}$ represents the number of EIT2 MFs in $F_{i,j}$, which can be infinitely large in the case of connected membership functions. A typical decomposition of an IT2 FS to EIT2 FSs for an input X_i is shown in Fig. 4. Now, following Theorem 1, for each membership function, $F_{i,j}$, we arbitrarily pick one of EIT2 MFs, $F_{i,j}^k$ and will denote its corresponding EIT1 membership function as $F'_{i,j}$ (Fig. 5). We can repeat the same procedure for other inputs. This results in a set of type-1 membership functions for all input variables for which we can apply a method similar to conventional TSK fuzzy control techniques. Then, we will aggregate the results for all possible choices of decomposed ET1 MFs.

3.2.2. *Fuzzifying the input variables*

The purpose of the *Fuzzifier* block is to map the crisp input variables to fuzzy values with memberships ranging from 0 to 1. An illustrative example shown in Fig. 6, which describes how crisp values $x_1 \in X_1$ and $x_2 \in X_2$ can be mapped to fuzzy values using the selected ET1 MFs, is discussed in Section 3.2.1. In Fig. 6a, the crisp value x_1 belongs to $F'_{1,1}$, $F'_{1,2}$, ... $F'_{1,j}$, ..., F'_{1,n_1} with the membership values of $F'_{1,1}(x_1) = 0.8$, $F'_{1,2}(x_1) = 0.2$, ... $F_{1,j} = 0$, ..., $F_{1,n_1} = 0$. These degrees of membership are called fuzzified values of x_1. A similar fuzzification process is performed for x_2 which is shown in Fig. 6b.

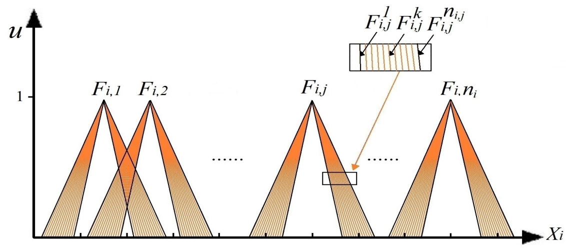


Fig. 4. Decomposed IT2 MFs to EIT2 MF.

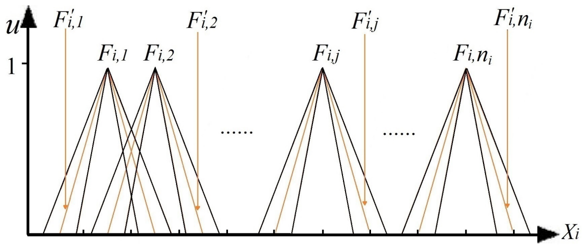


Fig. 5. A set of arbitrary choices of ET1 MFs.

3.3. *Rule Base*

Fuzzy rule base is a collection of conditional statements defining the outputs which have to be fired based on the fuzzified inputs. The rules may be constructed manually by an expert or by using data-driven machine learning algorithms [43], [44]. When creating the rules, specially using manual techniques, it is essential to have a previous knowledge about the system and its reactions to different sets of inputs. Rules of TSK FLSs have antecedents which are fuzzy sets and consequents which are functions of the inputs [30, 45]. In the proposed control technique, to capture input/output uncertainty, we use a first-order TSK FLS for which we represent antecedents using interval type-2 fuzzy sets and consequents using linear first-order polynomial functions of the inputs. A general multi-input-single-output type-2 TSK FLS rule with M rules, each having p antecedents, can be expressed

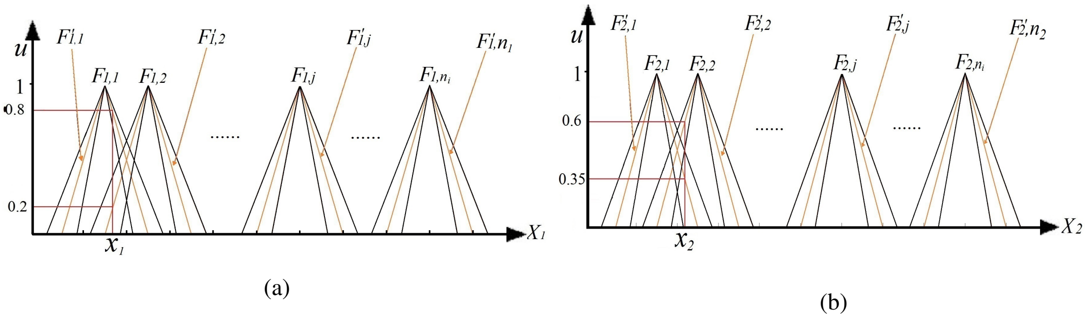


Fig. 6. (a) Fuzzification of x_1, (b) Fuzzification of x_2.

as:

$$R^{\ell}\text{: } \textbf{\textit{IF}}\ x_1 \text{ is } {F_1}^{\ell}, \text{ and } x_2 \text{ is } {F_2}^{\ell} \text{ and} \ldots \text{and } x_p \text{ is } {F_p}^{\ell}$$
$$\textbf{\textit{THEN}}\ y^{\ell} = c_0^{\ell} + c_1^{\ell} x_{\ell} + \cdots + c_p^{\ell} x_p \tag{8}$$

where R^{ℓ} is the ℓth rule and ${F_p}^{\ell}$ is the activated anticident interval type-2 fuzzy set for input channel x_p, which could be one of $F_{p,j}$, j=$1, 2, \ldots n_p$. In order to capture uncertainties in the outputs of of the FLC, the coefficients are interval type-1 fuzzy sets, bounded by $\underline{c_i^{\ell}}$ and $\overline{c_i^{\ell}}$, as shown in Fig. 7. With this setup, the rule output, y^{ℓ}, clearly is a type-1 fuzzy set.

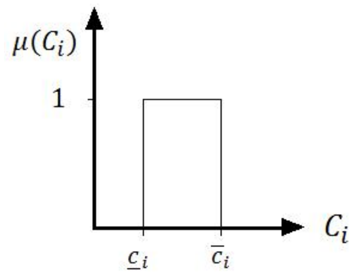


Fig. 7. Rule output function constants for the ℓ_{th} rule.

Interpreting these rules over IT2 MFs would not be easy. Therefore, instead of deriving the controller for these interval type-2 TSK fuzzy rules, we decompose the input membership functions into ET1 fuzzy sets and the output coefficients into crisp values, and interpret these rules over all possible choices of input ET1 fuzzy sets and crisp output coefficients, as described in the following sections.

3.4. Firing Level

For each rule, we should calculate the firing level. The firing level can be explained as the strength of a fuzzy rule or its influence over all rules based on the given inputs. For each rule, R^{ℓ}, its corresponding firing level is calculated in the *Firing Level* block. As mentioned, we interpret these rules over decomposed ET1 fuzzy sets. For this purpose, the crisp inputs have to be fuzzified in the form of $F'_{i,j}(x_i)$ as discussed in Section 3.3. Then, for any multi-input-single-output system with p inputs, we can calculate the firing level of the ℓth rule using the t-norm operator as:

$$f^{\ell'} = {F_1}^{\ell'}(x_1) * {F_2}^{\ell'}(x_2) * \cdots * {F_p}^{\ell'}(x_p) \tag{9}$$

For example, consider an FLS whose first rule, R^1, is

$$R^1\text{: } \textbf{\textit{IF}}\ x_1 \text{ is } F_{1,1} \text{ and } x_2 \text{ is } F_{2,2}$$
$$\textbf{\textit{THEN}}\ y^1 = c_0^1 + c_1^1 x_1 + c_2^1 x_2$$

Also, consider the MFs of this FLS as Fig. 6. Now, for the selected ET1 MFs shown in Fig. 6a, the firing level of R^1 can be calculated as

$$f^{1'} = F'_{1,2}(x_1) * F'_{2,2}(x_2) = 0.8 * 0.6 = 0.48$$

Note that the firing levels are calculated using the antecedent part of the conditional statements in each rule. Next, we show how the consequent parts of the rules are used for calculating the rule outputs.

3.5. Interval Type-2 Output Processing

Similar to decomposing input memberships, we decompose type-1 output fuzzy sets into crisp values as shown in Fig. 8. For each coefficient c_i^ℓ, we pick a single crisp value $c_i^{\ell'}$ to be used in the calculation of the rule output as shown in Fig. 8.

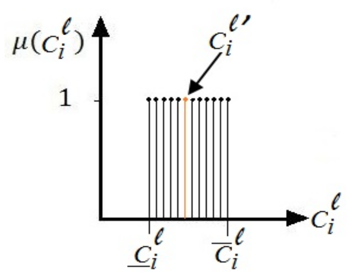


Fig. 8. Decomposing the type-1 FSs of the output coefficients into crisp numbers.

Following this procedure, for an arbitrary selection of crisp coefficients $c_0^{\ell'}$, $c_1^{\ell'}$,...,$c_p^{\ell'}$, the output for the ℓth rule is:

$$y^{\ell'} = c_0^{\ell'} + c_1^{\ell'} x_1 + \cdots + c_p^{\ell'} x_p \quad (10)$$

Performing the above procedure for all rules will result in crisp rule outputs $y^{1'}, y^{2'}, \ldots, y^{M'}$ for rules R^ℓ = 1, 2, ..., M, for the chosen set of crisp output coefficients.

The $OutputProcessing$ block takes the rule outputs and their respective firing levels and calculates the weighted average output as follows:

$$Y'_{TSK} = \frac{\sum_{i=1}^{M} f^{i'} y^{i'}}{\sum_{i=1}^{M} f^{i'}} \quad (11)$$

where, $f^{i'}$ is the firing level which was calculated in Section 3.4 and $y^{i'}$ is the crisp rule output for the chosen crisp coefficients. Using Equation 11, it is possible to obtain a crisp output for a particular set of ET1 MFs and crisp output coefficients values. However, different choices of ET1 MFs and output coefficients results in different crisp outputs. Hence, the aggregated output will be the collection of all weighted outputs for all possible combinations of ET1 MFs and output coefficients. All input fuzzy sets and output coefficients are connected sets. Therefore, the collection of outputs for different choices of ET1 MFs and output coefficients form a connected IT1 fuzzy set, will be a connected set, bounded by y_l and y_r, as shown in Fig. 9.

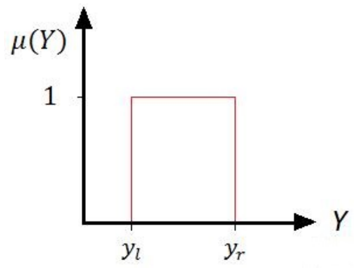


Fig. 9. The aggregated output for all possible choices of ET1 input MFs and output coefficients.

The lower bound y_l can be calculated as follows:

$$y_l = \min_{\underline{f^i}, \overline{f^i}} \left\{ Y'_{TSK} = \frac{\sum_{i=1}^{M} f^{i'} y_l^i}{\sum_{i=1}^{M} f^{i'}} \right\} \quad (12)$$

where y_l^i are the minimum values for $y^{i'}$, as calculated in Equation 16, and $\underline{f^i}$ and $\overline{f^i}$ are the minimum and maximum values of $f^{i'}$, $i = 1, \ldots, M$. Similarly, the upper bound can be found as follows:

$$y_r = \max_{\underline{f^i}, \overline{f^i}} \left\{ Y'_{TSK} = \frac{\sum_{i=1}^{M} f^{i'} y_r^i}{\sum_{i=1}^{M} f^{i'}} \right\} \quad (13)$$

where y_r^i denotes the maximum values for $y^{i'}$.

$$\underline{f^{\ell'}} = \underline{F_1^{\ell'}}(x_1) * \underline{F_2^{\ell'}}(x_2) * \cdots * \underline{F_p^{\ell'}}(x_p) \quad (14)$$

$$\overline{f^{\ell'}} = \overline{F_1^{\ell'}}(x_1) * \overline{F_2^{\ell'}}(x_2) * \cdots * \overline{F_p^{\ell'}}(x_p) \quad (15)$$

$$y_l^i = \underline{c_0^\ell} + \underline{c_1^\ell} x_1 + \underline{c_2^\ell} x_2 + \cdots + \underline{c_p^\ell} x_p \quad (16)$$

$$y_r^i = \overline{c_0^\ell} + \overline{c_1^\ell} x_1 + \overline{c_2^\ell} x_2 + \cdots + \overline{c_p^\ell} x_p \quad (17)$$

In the above equations, $\underline{F_p^{\ell'}}$ and $\overline{F_p^{\ell'}}$ represent the lower and upper ET1 MFs of the activated antecedent interval type-2 fuzzy sets in the ℓth rule of the p_{th} input channel, which are the bounds of their corresponding input IT2 MFs, $F_{p,j}$, as described in Section 3.3.

The question here is, what combination of minimum and maximum values of $f^{i'}$ should be used to calculate y_l and y_r. One way to answer this question is to find all possible combinations of minimum and maximum values of $f^{i'}$ (total of 2^M choices) to calculate y_l and y_r. This would require too much computation particularly for a large number of rules. Alternatively, we can find approximate values of y_l and y_r using the method of uncertainty bounds, which will be discussed in the next section.

3.6. Applying Uncertainty Bounds

To approximate the upper and lower bounds of weighted outputs, y_l and y_r, we integrate the uncertainty bounds technique introduced in [34]. The proposed control algorithm estimates y_l and y_r by calculating and averaging their respective upper and lower bounds values, $\underline{y_l}$, $\overline{y_l}$, $\underline{y_r}$, and $\overline{y_r}$, as shown in Fig. 10.

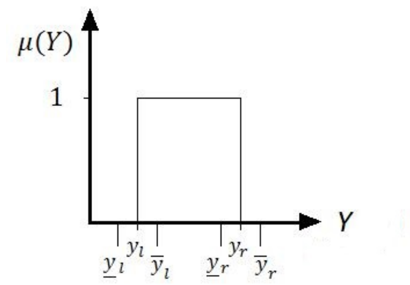


Fig. 10. Visual overview of uncertainty bounds.

The inner upper and lower bounds ($\overline{y_l}$ and $\underline{y_r}$), can be calculated as follows:

$$\overline{y_l} = \min\{y_{ll}, y_{ul}\} \tag{18}$$

$$\underline{y_r} = \max\{y_{lr}, y_{ur}\} \tag{19}$$

where,

$$y_{ll} = \frac{\underline{f}^i y_l^1 + \cdots + \underline{f}^M y_l^M}{\underline{f}^i + \cdots + \underline{f}^M} \tag{20}$$

$$y_{ul} = \frac{\overline{f^i} y_l^1 + \cdots + \overline{f^M} y_l^M}{\overline{f^i} + \cdots + \overline{f^M}} \tag{21}$$

$$y_{lr} = \frac{\underline{f}^i y_r^1 + \cdots + \underline{f}^M y_r^M}{\underline{f}^i + \cdots + \underline{f}^M} \tag{22}$$

$$y_{ur} = \frac{\overline{f^i} y_r^1 + \cdots + \overline{f^M} y_r^M}{\overline{f^i} + \cdots + \overline{f^M}} \tag{23}$$

With the inner bounds calculated, the outer bounds, $\underline{y_l}$ and $\overline{y_r}$, can be calculated as follows:

$$\underline{y_l} = \overline{y_l} - \left[\frac{\sum_{i=1}^{M} (\overline{f^i} - \underline{f}^i)}{\sum_{i=1}^{M} \overline{f^i} \times \sum_{i=1}^{M} \underline{f}^i} \times \frac{\sum_{i=1}^{M} \underline{f}^i (y_l^i - y_l^1) \times \sum_{i=1}^{M} \overline{f^i} (y_l^M - y_l^i)}{\sum_{i=1}^{M} \underline{f}^i (y_l^i - y_l^1) + \sum_{i=1}^{M} \overline{f^i} (y_l^M - y_l^i)} \right] \tag{24}$$

$$\overline{y_r} = \underline{y_r} + \left[\frac{\sum_{i=1}^{M} (\overline{f^i} - \underline{f}^i)}{\sum_{i=1}^{M} \overline{f^i} \times \sum_{i=1}^{M} \underline{f}^i} \times \frac{\sum_{i=1}^{M} \overline{f^i} (y_r^i - y_r^1) \times \sum_{i=1}^{M} \underline{f}^i (y_r^M - y_r^i)}{\sum_{i=1}^{M} \overline{f^i} (y_r^i - y_r^1) + \sum_{i=1}^{M} \underline{f}^i (y_r^M - y_r^i)} \right] \tag{25}$$

Now, the lower and upper bounds can be estimated by taking the average of their respective lower and upper bounds as follows:

$$y_l = \frac{\underline{y_l} + \overline{y_l}}{2} \tag{26}$$

$$y_r = \frac{\underline{y_r} + \overline{y_r}}{2} \tag{27}$$

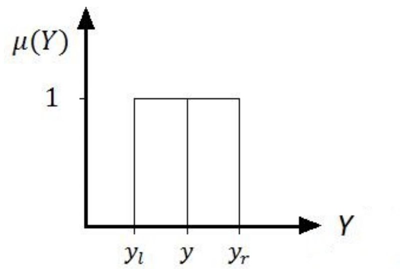


Fig. 11. Aggregated type-1 output fuzzy set and the defuzzified output, y.

3.7. Defuzzification

The final step of developing the controller is to defuzzifiy the aggregated type-1 fuzzy set, which is shown in Fig. 11. The defuzzified crisp output is easily computed by taking the average of the upper and lower bounds, y_r and y_l, which were obtained in Equations 26 and 27, as follows:

$$y = \frac{y_l + y_r}{2} \tag{28}$$

The defuzzified crisp output, y, is then applied to the plant as a control input.

3.8. Summerizing the Control Design Process

Computing the crisp output for all (infinite) possible combinations of the decomposed ET1 input MFs and output coefficients is computationally expensive. However, as it was presented in Section 3.6 and Section 3.7, only the upper and lower bounds of input membership functions and the coefficients of output rules are required for calculating the crisp control output. This significantly reduces the computation cost and the processing time. Algorithm 1 describes an efficient way for generating the control output using the proposed control structure.

Algorithm 1 Interval type-2 TSK fuzzy control

Input: crisp input variables
Output: crisp output control signal and uncertainty range

Begin Procedure

For the given set of control inputs, compute lower and upper bounds for the firing level of each rule ($\underline{f^i}$ *and* $\overline{f^i}$), $i = 1, \ldots, M$, *using* (14) *and* (15).

For the given set of control inputs, compute lower and upper bounds for the rule outputs (y_l^i *and* y_r^i), $i = 1, ..., M$, *using* (16) *and* (17).

Compute the inner uncertainty bounds ($\overline{y_l}$ *and* $\underline{y_r}$) *using* (18) *and* (19).

Compute the outer uncertainty bounds ($\underline{y_l}$ *and* $\overline{y_r}$) *using* (24) *and* (25).

Compute the lower and upper bounds for the output signal.
(y_l *and* y_r) *using* (26) *and* (27).

Compute the output control signal, y*, using* (28).

Return output control signal, y.

Return uncertainty range, [y_l, y_r].

End Procedure

4. SIMULATION RESULTS

In this section, we evaluate the performance of the developed IT2 TSK FLC by implementing it for the control of an Inverted Pendulum and a Cart System. The objective of the controller is to balance the inverted pendulum in a noisy uncertain environment by applying a control force to a moving cart that the pendulum is attached to. The inverted pendulum will fall down if the cart does not move appropriately to balance it. We have also compared the results of the developed IT2 TSK FLC with a type-1 TSK FLC and a PD controller.

An inverted pendulum on a cart, shown in Fig. 12, can be modeled as:

$$[\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+hl^2)b}{I(H+h)+Hhl^2} & \frac{h^2gl^2}{I(H+h)+Hhl^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-hlb}{I(H+h)+Hhl^2} & \frac{hgl(H+h)}{I(H+h)+Hhl^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+hl^2}{I(H+h)+Hhl^2} \\ 0 \\ \frac{hl}{I(H+h)+Hhl^2} \end{bmatrix} F \tag{29}$$

where, ϕ is the angle between the pendulum and the system's equilibrium position, x is the cart's position, F is the control force, H is mass of the cart, h is mass of the pendulum, b is the coefficient of friction for the cart, l is length of the pendulum and I is the moment of inertia for the pendulum. The constant physical properties of the inverted pendulum and the cart are stated in Table 1.

Table 1
Distance (x) traveled by the cart.

Physical property	Value
H	0.5 Kg
h	0.2 Kg
b	0.1 N/m/sec
l	0.3 m
I	0.006 Kg.m

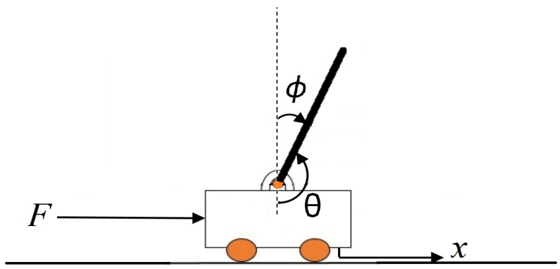


Fig. 12. An inverted pendulum on a moving cart.

From this state-space model of the system, one can find the system's eigenvalues as:

$$\lambda = (0, -6.1301, -0.2498, 6.0067) \quad (30)$$

Which shows that the open-loop system is unstable due to an eigenvalue in the right-half of the S-plane.

In order to simulate the real-world uncertainties, we have introduced noise, disturbance and unbalanced initial condition to the system. The noise introduced to the system is a uniform random noise added to the signal readings of ϕ and the actuator signal, F. The range and maximum amplitude of the noise injected to the system is stated in Table 2. The disturbance is applied to the actuator signal during the simulation time. Furthermore, the initial state of the inverted pendulum is deviated by $\phi(0) = 0.25\ rad$ from its equilibrium position. The applied disturbance is shown in Fig. 13. The initial conditions of the remaining state variables were set at, $\dot{\phi} = 0\ rad/s$, $x = 0\ m$ and $\dot{x} = 0\ m/s$.

Table 2

Uncertainty ranges due to noise

Variable	Range
ϕ	$\pm 0.01\ rad$
F	$\pm 1\ N$

4.1. Simulating the IT2 TSK FLC

The developed IT2 FLC was applied to the control of the inverted pendulum and cart system described in the previous section. The general control structure is shown in Fig. 14. The controller's inputs are ϕ, $\dot{\phi}$, and $\dot{x}$, which are signals read from the system. The controller's output is the force, F, which is applied to the cart.

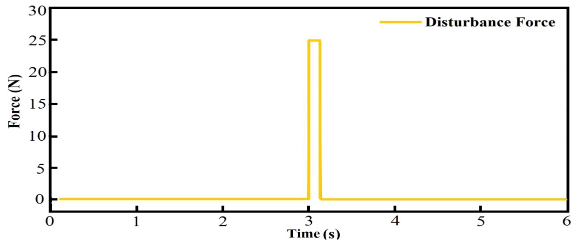


Fig. 13. Disturbance force applied to the cart.

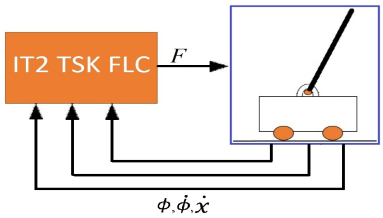


Fig. 14. Interval type-2 TSK FLC.

In our design, as shown in Figures 15a and 15c, the memberships of ϕ and $\dot{x}$ are composed of five fuzzy sets: negative large (NL), negative small (NS), zero (O), positive small (PS) and positive large (PL). On the other hand, as shown in Fig. 15b, the memberships of $\dot{\phi}$ are composed of three fuzzy sets: negative (N), zero (O) and positive (P). For all inputs, an optimized FOU for the membership functions is is obtained using genetic algorithms.

The rule outputs are designed as:

$$y^{\ell} = c_1^Q \phi + c_2^Q \dot{\phi} \quad (31)$$

where ℓ represents the number of rules, $\ell = 1, 2, \ldots, 75$, and Q represents the label for output membership functions including VS, S, M, L and VL. Note that each rule output resembles a PD controller with c_1 being the the proportional gain and c_2 being the derivative gain. The upper and lower bounds for the coefficients of the rule outputs are obtained through optimization using genetic algorithms and are presented in Table 3.

The rule base for controlling the inverted pendulum and cart system using the developed IT2 TSk FLC is presented in Table 4.

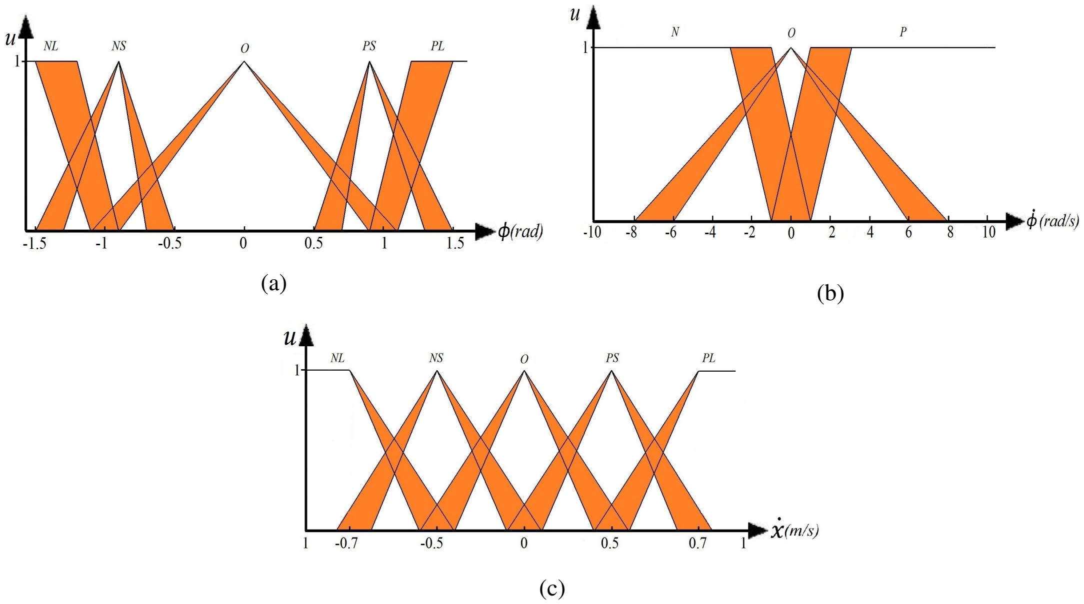


Fig. 15. (a) IT2 MFs for ϕ, (b) IT2 MFs for $\dot{\phi}$ and (c) IT2 MFs for $\dot{x}$.

Table 3
IT2 TSK rule output coefficient bounds

		Rule Coefficients			
		$\underline{c}_1^Q$	$\bar{c}_1^Q$	$\underline{c}_2^Q$	$\bar{c}_2^Q$
Rule Labels (Q)	VS	67.6	77.6	3	4
	S	91.3	101.3	4.4	5.4
	M	113.1	123.1	5.1	6.1
	L	125.6	135.6	6.5	7.5
	VL	183.8	193.8	9.3	10.3

4.2. Simulating the Type-1 TSK FLC

Here, we compare the performance of the developed IT2 TSK FLC structure, with a type-1 TSK FLC. To make the comparison fair, the type-1 membership functions of the type-1 TSK FLC are set at the centers of the footprints of the IT2 MFs of the IT2 TSK FLC described in Section 4.1. These type-1 membership functions are shown in Fig. 16.

The rule outputs for the type-1 TSK FLC are expressed in the form:

$$y^\ell = c_1^Q \phi + c_2^Q \dot{\phi} \tag{32}$$

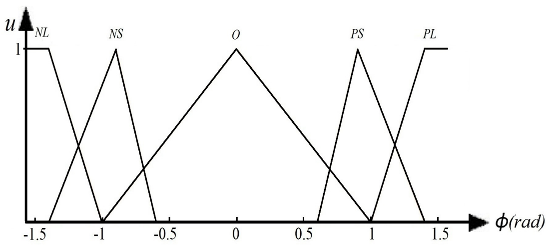
where ℓ represents the number of rules, $\ell = 1, 2, \ldots, 75$. However, unlike the developed interval type-2 TSK FLC, the output coefficients of a type-1 TSK FLC are crisp values. Again, to make the comparison fair, the output coefficients c_1^Q and c_2^Q are selected to be at the centers of their corresponding rule output coefficients presented in Table 3. These type-1 TSK rule output coefficients are presented in Table 5.

The rule base which is used for the type-1 TSK FLC is the same rule base used for the IT2 TSK FLC, and is presented in Table 4. For obtaining the final crisp control output, weighted average defuzzification method is used.

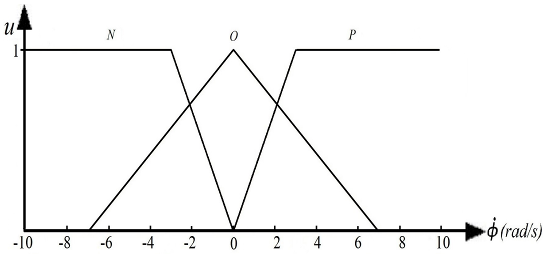
4.3. Simulating the Classical Proportional Derivative (PD) Controller

The rule outputs of the IT2 TSK FLC are a function of ϕ and $\dot{\phi}$. This resembles the output of a classical PD controller. To evaluate the performance of the developed IT2 TSK FLC compared to a PD controller, we also designed a PD controller for the inverted pendulum and cart system. To make the comparison fair, the gains for the proportional and derivative gains were set at the average of the centers of the designed IT2 TSK rule output coefficient bounds. The designed PD controller, therefore, has the following structure:

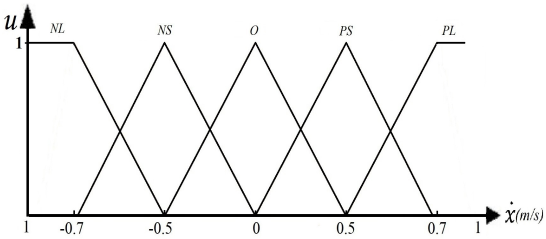
$$F = 121.28\phi + 6.16\dot{\phi} \tag{33}$$



(a) Type-1 MF for ϕ.



(b) Type-1 MF for $\dot{\phi}$.



(c) Type-1 MF for $\dot{x}$.

Fig. 16. (a) Type-1 MF for ϕ, (b) Type-1 MF for $\dot{\phi}$ and (c) Type-1 MF for $\dot{x}$.

4.4. Results

The responses of the system when using the IT2 TSK FLC, type-1 TSK FLC and classical PD controller are presented in Figures 17 and 18. From Fig. 17, we can observe that the IT2 TSK FLC handled uncertainties in ϕ as well as F and controlled the inverted pendulum smoother than the other two types of controllers. Even-though the initial condition of the inverted pendulum was not at the equilibrium position, the IT2 TSK FLC was capable of stabilizing the system faster than the other controllers while maintaining the smallest overshoot and fastest settling time.

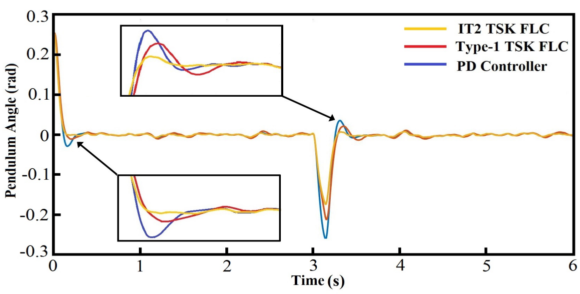


Fig. 17. Pendulum angle (ϕ) vs. Time.

When an external disturbance force shown in Fig. 13 was applied to the system, the results presented in Fig. 17 show that the IT2 TSK FLC returned the inverted pendulum back to its equilibrium position smoothly with the smallest overshoot, fastest rise time and fastest settling time than the other two controllers. Here, it can be observed that, in the presence of a disturbance, the type-1 TSK FLC controlled the system smoother than the classical PD controller.

The position of the cart throughout the simulation time is shown in Fig. 18. It is presented in Table 6 that the distance traveled by the cart for stabilizing the inverted pendulum throughout the simulation time is the shortest when the IT2 TSK FLC is used. In all the simulation results, even-though the IT2 TSK FLC performed better than the other two controllers, the type-1 controller performed better than the PD controller.

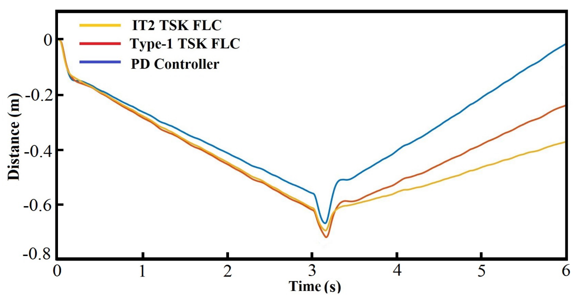


Fig. 18. Cart position (x) vs. Time.

Table 4

Rule base for the inverted pendulum problem

ℓ	ϕ	$\dot{\phi}$	$\dot{x}$	F	ℓ	ϕ	$\dot{\phi}$	$\dot{x}$	F
1	NL	N	NL	VL	39	NS	O	PS	S
2	NL	N	NS	VL	40	NS	O	PL	S
3	NL	N	O	VL	41	NS	P	NL	M
4	NL	N	PS	L	42	NS	P	NS	M
5	NL	N	PL	L	43	NS	P	O	M
6	NL	O	NL	VL	44	NS	P	PS	S
7	NL	O	NS	VL	45	NS	P	PL	S
8	NL	O	O	VL	46	PS	N	NL	S
9	NL	O	PS	L	47	PS	N	NS	S
10	NL	O	PL	L	48	PS	N	O	M
11	NL	P	NL	VL	49	PS	N	PS	M
12	NL	P	NS	VL	50	PS	N	PL	M
13	NL	P	O	VL	51	PS	O	NL	S
14	NL	P	PS	L	52	PS	O	NS	S
15	NL	P	PL	L	53	PS	O	O	M
16	O	N	NL	S	54	PS	O	PS	L
17	O	N	NS	S	55	PS	O	PL	L
18	O	N	O	S	56	PS	P	NL	S
19	O	N	PS	S	57	PS	P	NS	S
20	O	N	PL	VS	58	PS	P	O	M
21	O	O	NL	S	59	PS	P	PS	L
22	O	O	NS	S	60	PS	PS	PL	L
23	O	O	O	VS	61	PL	N	NL	L
24	O	O	PS	S	62	PL	N	NS	L
25	O	O	PL	S	63	PL	N	O	VL
26	O	P	NL	VS	64	PL	N	PS	VL
27	O	P	NS	S	65	PL	N	PL	VL
28	O	P	O	S	66	PL	O	NL	L
29	O	P	PS	S	67	PL	O	NS	L
30	O	P	PL	S	68	PL	O	O	VL
31	NS	N	NL	L	69	PL	O	PS	VL
32	NS	N	NS	L	70	PL	O	PL	VL
33	NS	N	O	M	71	PL	P	NL	L
34	NS	N	PS	S	72	PL	P	NS	L
35	NS	N	PL	S	73	PL	P	O	VL
36	NS	O	NL	L	74	PL	P	PS	VL
37	NS	O	NS	L	75	PL	P	PL	VL
38	NS	O	O	M	–	–	–	–	–

To quantify the performance improvement achieved by using the IT2 TSK FLC, we have calculated the Root Mean Square (RMS) error of the inverted pendulum throughout the simulation time. As presented in Table 7, the proposed IT2 TSK FLC has controlled the system with the least RMS error than type-1 TSK FLC or PD controller. From the simulation results, we have found a 16.6% and 23.3% improvement in RMS error when the proposed control structure is used instead of a type-1 TSK FLC or a PD controller, respectively.

Table 5

Type-1 rule output coefficient bounds

		Rule Coefficients	
		c_1^Q	c_2^Q
Rule Labels (Q)	VS	72.6	3.5
	S	96.3	4.9
	M	118.1	5.6
	L	130.6	7
	VL	188.8	9.8

Table 6

Distance (x) traveled by the cart.

Controller Type	Distance (m)
IT2 TSK FLC	1.02
Type-1 TSK FLC	1.21
PD	1.32

Table 7

RMS error for ϕ.

Controller Type	RMS Error (rad)
IT2 TSK FLC	0.0320
Type-1 TSK FLC	0.0383
PD	0.0418

5. CONCLUSION

In this paper, we developed a novel and computationally effective IT2 TSK FLC structure that is capable of capturing input uncertainties. Additionally, an algorithm that can be used for the implementation of the proposed control structure was presented. The developed control structure and algorithm exploits the ability of IT2 FLSs to measure and quantify uncertainties for generating an improved control output. For capturing uncertainties, the control structure utilizes IT2 fuzzy sets for describing the input space. Additionally, for enabling parallel processing of rule outputs, IT2 antecedents of the rules were integrated with TSK outputs. Furthermore, for making the developed control structure computationally effective, the type reduction process of the FLC was bypassed using the uncertainty bounds output processing technique.

Finally, for evaluating the performance of the developed IT2 TSK FLC, it was applied to the control of an inverted pendulum on a moving cart system in the presence of noise and disturbance. The simulation results were then compared with the results of a type-1 TSK FLC and a classical PD controller. From the simulation results, it was found that the developed IT2 TSK FLC was more stable and robust to external disturbances and noises than a type-1 TSK FLC or a PD controller. Future works includes extending the proposed method to a general type-2 TSK FLC as well as developing software libraries and toolboxes that can ease the development and real-time implementation of the proposed controllers.

References

[1] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-i. *Information sciences*, 8:199–249, 1975.

[2] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-ii. *Information sciences*, 8(4):301–357, 1975.

[3] Nilesh N. Karnik and Jerry M. Mendel. Introduction to type-2 fuzzy logic systems. *IEEE World Congress on Computational Intelligence*, 2:915–920, 1998.

[4] O. Castillo, P. Melin, J. Kacprzyk, and W. Pedrycz. Type-2 fuzzy logic: Theory and applications. pages 145–150, 2007.

[5] Jerry Mendel. Type-2 fuzzy sets: Some questions and answers. pages 10–13. IEEE, 2003.

[6] M. Biglarbegian, W. Melek, and Jerry M. Mendel. Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: Theory and experiments. *IEEE Transactions on Industrial Electronics*, 58(4):1371–1384, 2011.

[7] Yuan S. Chen and L. Yao. Robust type-2 fuzzy control of an automatic guided vehicle for wall-following. In *International Conference of Soft Computing and Pattern Recognition (SOCPAR)*, pages 172–177, 2009.

[8] *Laboration in Automatic Control: Control of an Inverted Pendulum*. Linköping University, 2012.

[9] M. Ri, J. Huang, S. Ri, H. Yun, and C. Kim. Design of interval type-2 fuzzy logic controller for mobile wheeled inverted pendulum. *12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China*, pages 535–540, 2016.

[10] N. Baklouti and A. Alimi. Motion planning in dynamic and unknown environment using an interval type-2 tsk fuzzy logic controller. *IEEE International Fuzzy Systems Conference*, 2007.

[11] S. Zaheer and J. Kim. Type-2 fuzzy airplane altitude control: A comparative study. In *IEEE International Conference on Fuzzy Systems*, pages 2170–2176, 2011.

[12] Qun Ren, Marek Balazinski, and Luc Baron. High-order interval type-2 takagi-sugeno-kang fuzzy logic system and its application in acoustic emission signal modeling in turning process. *The International Journal of Advanced Manufacturing Technology*, 63(9-12):1057–1063, 2012.

[13] Chwan Lu Tseng, Shun Yuan Wang, Shou Chuang Lin, and Yu Yuan Chen. Interval type-2 takagi-sugeno fuzzy controller design for a class of nonlinear singular networked control systems. In *International Conference on Fuzzy Theory and it's Applications (iFUZZY)*, pages 268–272, 2012.

[14] K. Nilesh, Jerry M. Mendel, and Q. Liang. Type-2 fuzzy logic systems. *IEEE transactions on Fuzzy Systems*, 7(6):643–658, 1999.

[15] S. Greenfield. Type-2 fuzzy logic: Circumventing the defuzzification bottleneck. De Montfort University, PhD dissertation, 2012.

[16] Luis A. Lucas, Tania M. Centeno, and Regattieri D. Myriam. General type-2 fuzzy inference systems: Analysis, design and computational aspects. *IEEE World Congress on Computational Intelligence*, pages 1–6, 2007.

[17] S. Greenfield and F. Chiclana. Type-reduction of the discretized interval type-2 fuzzy set: approaching the continuous case through progressively finer discretization. *Journal of Artificial Intelligence and Soft Computing Research*, 1(3):183–193, 2011.

[18] S. Greenfeld, F. Chiclana, and R. John. Type-reduction of the discretised interval type-2 fuzzy set. *IEEE Transactions on Fuzzy Systems*, pages 738–743, 2009.

[19] Nilesh N. Karnik and Jerry Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems*, pages 327–348, 2001.

[20] Luis A. Lucas, Tania M. Centeno, and Myriam R. Delgado. General type-2 fuzzy inference systems: Analysis, design and computational aspects. *SBC - Proceedings of SBGames*, pages 5–8, 2007.

[21] Jerry M Mendel. *Uncertain rule-based fuzzy logic system: introduction and new directions*. 2001.

[22] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):116–132, 1985.

[23] Jerry M. Mendel and Robert I. Bob John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, 2002.

[24] Q. Liang and Jerry M. Mendel. Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy Systems*, 8(5):535–550, 2000.

[25] Dongrui Wu and J. M. Mendel. Designing practical interval type-2 fuzzy logic systems made simple. *IEEE International Conference on Fuzzy Systems*, pages 800–807, 2014.

[26] Daniel Opoku Abdollah Homaifar Shannon Arnold Nnamdi Enyinna, Ali Karimoddini. Developing an interval type-2 tsk fuzzy logic controller. *Fuzzy Information Processing Society NAFIPS*, 5(4):1–6, 2015.

[27] Jerry Mendel and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821, 2006.

[28] H. Hamrawi and S. Coupland. Measures of uncertainty for type-2 fuzzy sets. *Computational Intelligence (UKCI), 2010 UK Workshop*, pages 1–7, 2010.

[29] J. M. Mendel and M. R. Rajati. On computing normalized interval type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 22(5):1335–1340, 2014.

[30] Michio Sugeno. *Industrial Applications of Fuzzy Control*. 1985.

[31] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, MAN and Cybernetics*, 15(1):116–132, 1985.

[32] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and S*, 28(1):15–33, 1988.

[33] Bruno B. Ferreira and Adriano J. O. Cruz. A parallel method for tuning fuzzy tsk systems with cuda. *SBC - Proceedings of SBGames*, pages 5–8, 2012.

[34] Hongwei Wu and Jerry M Mendel. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 10(5):622–639, 2002.

[35] Jerry M. Mendel and Hongwei Wu. Centroid uncertainty bounds for interval type-2 fuzzy sets: Forward and inverse problems. *IEEE International Conference on Fuzzy Systems*, 2:947–952, 2004.

[36] Jerry M. Mendel, Robert I. John, and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821, 2006.

[37] Nilesh N. Kamik and Jerry M. Mendel. Applications of type-2fuzzy logic systems :handling the uncertainty associated with surveys. *IEEE International Fuzzy Systems Conference, Seoul, Korea*, pages 1546–1551, 1999.

[38] F. Doctor, H. Hagras, D. Roberts, and V. Callaghan. A type-2 fuzzy based system for handling the uncertainties in group decisions for ranking job applicants within human resources systems. In *IEEE World Congress on Computational Intelligence*, pages 481–488, 2008.

[39] J. C. Figueroa-Garcia. On a-representation of type-2 fuzzy sets. *Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 1–6, 2016.

[40] Saima Hassan, Abbas Khosravi, and Jafreezal Jaafar. The impact of fou size and number of mfs on the prediction performance of interval type-2 fuzzy logic systems. *International Symposium on Mathematical Sciences and Computing Research (iSMSC)*, pages 104–109, 2015.

[41] A. Yerukayev. An approach for the formation of the type-2 fuzzy sets. *13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, pages 460–462, 2016.

[42] Jerry Mendel. Advances in type-2 fuzzy sets and systems. *Information Sciences*, 177:84–110, 2007.

[43] D. Wu and Woei W. Tan. A type-2 fuzzy logic controller for the liquid-level process. In *IEEE International Conference on Fuzzy Systems*, volume 2, pages 953–958, 2004.

[44] A. Homaifar and E. McCormick. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, 3(2):129–139, 1995.

[45] Q. Liang and Jerry M. Mendel. An introduction to type-2 tsk fuzzy logic systems. *IEEE International Fuzzy Systems Conference Proceedings, Seoul, Korea*, pages 1534–1539, 1999.