

Developing an Interval Type-2 TSK Fuzzy Logic Controller

Nnamdi Enyinna, Ali Karimodini, Daniel Opoku, Abdollah Homaifar, Shannon Arnold

Abstract—Type-2 Fuzzy Logic Controllers offer great capabilities in modeling and handling the effects of real world uncertainties from sensors, actuators and the environment. Nevertheless, the general Type-2 Fuzzy Logic Controllers enormously suffer from high computation cost. To overcome this problem, in this paper, we present a computationally effective Type-2 Fuzzy Logic Controller which uses Interval Type-2 fuzzy sets to capture the control inputs and utilizes the Takagi-Sugeno-Kang technique to render the control outputs. It is shown that the proposed technique greatly reduces the computation cost since only the lower and upper bounds of the input fuzzy sets suffice to calculate the control output.

I. INTRODUCTION

The Type-2 fuzzy logic system (T-2 FLS), initially developed by Zahdeh [12], is unique in its ability to model and handle uncertainty while also being able to handle complex control structures and linguistic variables. Over the years, a great deal of efforts have been put into reducing the computation cost, and thus, improving the performance of the Type-2 fuzzy controllers. The focus of these efforts has been mainly on the type reduction of fuzzy type-2 sets in the last stage of control computation process in a general T-2 FLC (e.g., [11], [1], [7], and [2]). In comparison, in this paper, we propose a new technique for designing a T-2 FLC which reduces the computation cost both in input-fuzzification and output signal processing. For this purpose, we use Interval Type-2 fuzzy sets to capture the input space. The Interval Type-2 fuzzy sets [12], [5], [4], were initially introduced to reduce the computational complexity of the general T-2 FLSs. Interval T-2 FLSs have become even more popular over the recent years because they can be extended to general T-2 FLSs [2], [6], while still preserving the major properties of general T-2 FLSs. On the other hand, to reduce the computation cost in the output processing part of the FLCs, we employ the Takagi-Sugeno-Kang (TSK) controller. The TSK controller or Sugeno controller was originally introduced by Sugeno et al [9]. The TSK controller allows for the modeling of more complex output functions while greatly reducing the computation cost by parallel processing of the

inputs and outputs. While some work has been done on Type-2 TSK controller design (e.g., [10], [8] and [3]), this paper provides a unique theoretical approach and theorem which together, simplifies the understanding of the Type-2 TSK controller design. The proposed algorithm ultimately requires little information (the upper and lower bounds for the firing levels and output constants) which significantly reduces the computation cost for calculating the output control signal.

Thus, the contribution of this paper is the development of a computationally effective Type-2 fuzzy logic controller by using the Interval Type-2 fuzzy sets to capture the input space, and by employing the TSK Fuzzy Logic Controller for the output processing. An illustrative example and the simulation results for a cart-centering problem are provided which show the satisfactory performance of the developed controller. Integrating the interval type-2 FLC with Takagi-Sugeno rules makes it possible to use parallel processing of input firing levels and rule outputs which the Mamdani approach is not able to do. This simplifies the control structure so that only the upper and lower bounds of the firing levels and rule outputs suffices to calculate the controller output. Unlike many other methods, this technique directly generates a fuzzy type-1 output which can be simply defuzzified and hence, there is no need for type reduction.

The rest of this paper is organized as follows. In Section II, we briefly discuss the necessary preliminaries and Fuzzy sets background. In Section III, we detail the development of the proposed T-2 FLC and discuss its different parts including the input processing, rule sets and output processing. Section IV simulates the developed controller. The paper is concluded in Section V.

II. PRELIMINARIES

In this section, we briefly review the necessary background on Fuzzy Set Theory which is essential to explain the proposed Type-2 fuzzy controller. Some of the notations are borrowed from Mendel et al [5]. We also will develop a useful theorem that simplifies the presentation of Interval Type-2 fuzzy sets.

Definition 1: Type-1 Fuzzy Sets (T-1 FSs)

A Type-1 Fuzzy Set [4] 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 follows:

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

N. Enyinna, A. Karimodini and A. Homaifar are with the Department of Electrical and Computer Engineering, North Carolina Agricultural and Technical State University, Greensboro, NC 27411 USA; D. Opoku is with Navigation Systems Engr, Mercury Data Systems, Lexington, KY, USA, and S. Arnold is with Unmanned and Autonomous Systems Test, Lead Test Resource Management Center, 4800 Mark Center Blvd, Suite 07J22 Alexandria VA 22350.

Corresponding author: A. Karimodini, Tel: +13362853313, akarimod@ncat.edu.

Here \sum denotes the collection of elements of a set. Extending from Type-1 fuzzy sets, Type-2 fuzzy sets allow for capturing more degrees of uncertainty which can be defined as follows:

Definition 2: Type-2 Fuzzy Sets (T-2 FSs)

A Type-2 Fuzzy Set [4] is composed of triples $((x, u), \mu_A(x, u))$ in which for each member of domain $x \in X$, a primary membership value, $u \in J_x$ (J_x is the range of primary membership for a given x) and a secondary membership, $\mu_{\tilde{A}}(x, u)$ can be defined as follows:

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

Fig. 1a shows a simple Type-2 fuzzy set for a case that X and J_x are connected sets, and $\mu_{\tilde{A}}$ is a continuous function.

Type-2 fuzzy sets can be seen as a set of weighted Type-1 fuzzy sets, and can model uncertainty in memberships due to imprecise measurements. Despite the capabilities of this set up, particularly in capturing uncertainties, it is found to be computationally expensive. Alternatively we can use Interval Type-2 fuzzy sets, which significantly reduce the computation costs while maintaining major advantages of Type-2 fuzzy sets.

Definition 3: Interval Type-2 Fuzzy Sets (IT2 FSs)

An Interval Type-2 Fuzzy Set [5] is a Type-2 Fuzzy Set in which the secondary grade values are always unity:

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

Fig. 1b gives an example of an Interval Type-2 fuzzy set. Next, we simplify the expression of IT2 FSs using Embedded IT2 FSs and Embedded Type-1 FSs.

Definition 4: Embedded IT2 FSs (EIT2 FSs)

An Embedded Interval T-2 FS [5] is an Interval T-2 FS in which for all $x \in X$, the primary membership is a single value:

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

In the case that X is a connected set and u is a continuous function, an Embedded Interval T-2 FSs can be seen as a wavy slice shown in Fig. 1c.

Following the preceding definition, the following lemma shows that IT2 FSs can be expressed in terms 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 now define Embedded T-1 FSs and show how EIT2 FSs can be further simplified.

Definition 5: Embedded T1 FSs (ET1 FSs)

An ET1 FS is a T-1 FS with the same x and primary membership values in its corresponding EIT2 FS but with the secondary membership (unity) ignored.

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

Fig. 1d shows an Embedded T-1 FS with connected X and continuous u_x .

Lemma 2: (EIT2 to ET1 FSs)

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

Combining Lemmas 1 and 2, we can now present the following theorem which describes IT2 FSs based on ET1 FSs:

Theorem 1: (IT2 to 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 exist 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 $\tilde{A} = \sum_j \tilde{A}_e^j = \sum_j A_e^j \times 1$. ■

This theorem allows us to represent the 3-dimensional IT2 FSs as a set of 2-dimensional T-1 FSs. This theorem will be used throughout the paper to simplify the explanation of the developed Type-2 TSK FLC using Type-1 FSs.

III. DEVELOPING THE CONTROLLER

The basic idea for developing the proposed IT2 TSK FLC is to use Theorem 1 to decompose Type-2 fuzzy sets into Type-1 fuzzy sets for which we can employ well-matured control techniques such as TSK to design the controller. Fig. 2 shows the structure of the proposed T2 TSK FLC. The blocks of this control structure are detailed in the following sections.

A. Interval Type-2 Fuzzification

1) *Membership Decomposition:* 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 as follows:

$$F_i = \sum_{j=1}^{n_i} F_{i,j} \quad (6)$$

Where F_i represents all MFs related to input channel X_i , including $F_{i,1}, F_{i,2}, \dots, F_{i,n_i}$. The subscript j in $F_{i,j}$ is used

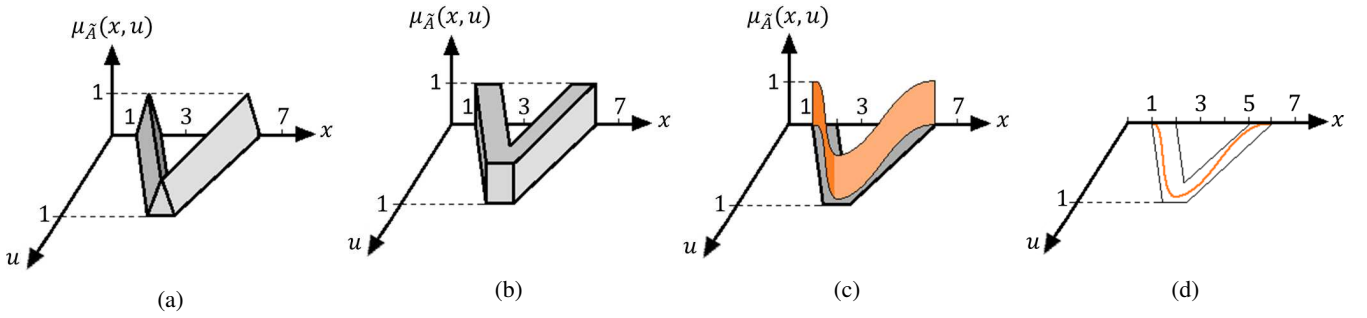


Fig. 1: (a) A Type-2 FS. (b) An Interval Type-2 FS. (c) An Embedded IT2 FS. (d) An Embedded T1 FS.

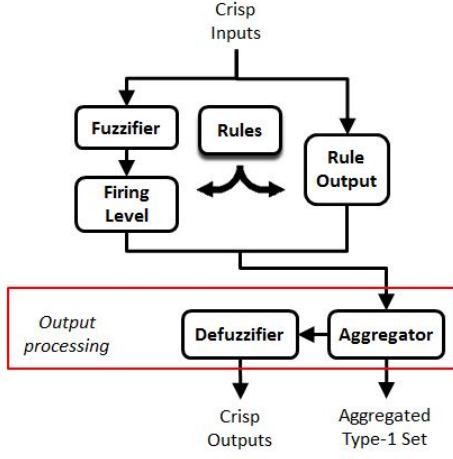


Fig. 2: T2 TSK FLC structure

to indicate the j th individual MF in F_i . Fig. 3a shows the IT2 input membership function for the input channel of a system with three IT2 membership sets. For simplicity, the secondary grade values are not shown as they are always unity.

Using Theorem 1, each of the input Interval Type-2 MFs, $F_{i,j}$ can be broken down into a collection of ET1 membership sets $F_{i,j}^k$ as follows:

$$F_{i,j} = \sum_{k=1}^{n_{i,j}} F_{i,j}^k \quad (7)$$

Where $F_{i,j}^k$ represents a particular ET1 MF in $F_{i,j}$, and $n_{i,j}$ represents the maximum number of ET1 MFs in $F_{i,j}$. Fig. 3b, shows a typical decomposition for input X_i . Now, for each Membership Function, $F_{i,j}$, we arbitrarily pick one of ET1 MFs, $F_{i,j}^k$ and will denote it $F'_{i,j}$. Fig. 3c shows this new set of ET1 MFs for input X_i . We repeat the same procedure for all input variables. 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.

2) *Fuzzifying the input variables*: The Fuzzifier block maps the crisp input variables to fuzzy membership values between 0 and 1. Fig. 4 illustrates 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 discussed in Section III-A.1. In this figure,

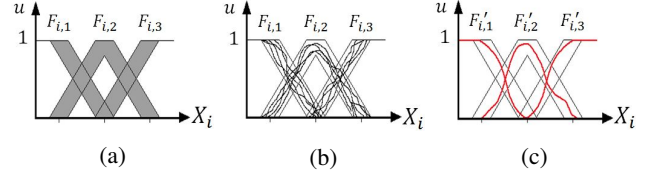


Fig. 3: (a) IT2 Membership Functions for input X_i . (b) Decomposed IT2 MFs for input X_i . (c) A set of arbitrary choices of ET1 MFs for input X_i .

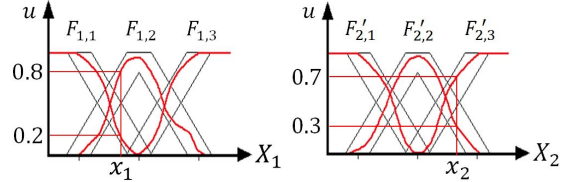


Fig. 4: Fuzzification of x_1 and x_2

the crisp value x_1 belongs to $F'_{1,1}$, $F'_{1,2}$, and $F'_{1,3}$, with the membership values of $F'_{1,1}(x_1) = 0.2$, $F'_{1,2}(x_1) = 0.8$, and $F'_{1,3}(x_1) = 0$, which are known as the fuzzified value of x_1 .

B. Rule Sets

The Rule Sets are a predetermined set of linguistic rules indicating which fuzzy outputs are created from different fuzzy input combinations. As an example, in a system with 2 inputs and one output, one of the rules could be:

$$R^1: \text{IF } x_1 \text{ is } F'_{1,2}, \text{ and } x_2 \text{ is } F'_{2,3} \\ \text{THEN } y^1 = c_0^1 + c_1^1 x_1 + c_2^1 x_2 \quad (8)$$

In general, for any multi-input-single-output system, the l th rule, R^l , $l = 1, \dots, m$, will be as follows:

$$R^l: \text{IF } x_1 \text{ is } F_{1,1}^l, \text{ and } \dots \text{ and } x_p \text{ is } F_{p,1}^l \\ \text{THEN } y^l = c_0^l + c_1^l x_1 + \dots + c_p^l x_p \quad (9)$$

Where $F_{p,j}^l$ represents the chosen ET1 MF of the corresponding $F_{p,j}$, and y^l is the output value, which is a linear combination of the inputs. With the defined Rule Set, we now proceed to calculate Firing Levels and Rule Outputs for each rule.

C. Rule Firing Levels

In the Firing Level block, we calculate the firing level for each rule in the Rule Set. The Firing Level can be seen as a

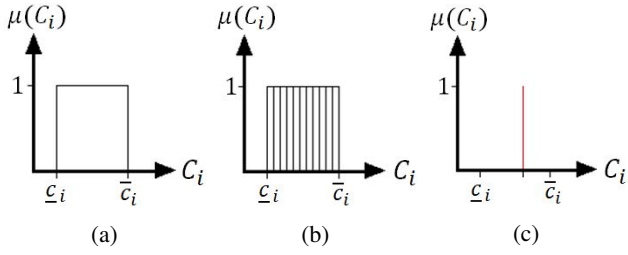


Fig. 5: (a) Rule Output function constants. (b) Discretization of the output function constants. (c) Selected crisp value c_i^l' .

weight or influence of a particular rule on the overall output. The firing levels f^l are calculated using the set of fuzzified inputs, $F'_{i,j}(x_i)$ discussed in Section III-B. The firing level for l th rule is the t-norm of these fuzzy values as indicated by the Rule Set. As an example, for the rule, R^1 , defined in (8), and the fuzzified values shown in Fig. 4, the firing level will be $f^1 = F'_{1,2}(x_1) * F'_{2,3}(x_2) = 0.8 * 0.7 = 0.56$. In general, the firing level of rule l for any Multi-Input-Single-Output system can be given as:

$$f^l = F_1^l(x_1) * F_2^l(x_2) * \dots * F_p^l(x_p) \quad (10)$$

D. Processing the Rule Outputs

Similar to the conventional TSK control technique [9], here we consider the output for each rule as a linear combination of the crisp inputs as follows:

$$y^l = c_0^l + c_1^l x_1 + \dots + c_p^l x_p \quad (11)$$

The difference here is that to capture the uncertainty in the output, the coefficients c_0, c_1, \dots, c_p are Interval Type-1 fuzzy sets, and so is the rule output, y_l . Fig. 5a, shows the coefficient c_i^l which is described by a Type-1 fuzzy set bounded by \underline{c}_i^l and \bar{c}_i^l .

These IT1 constants allow us to express uncertainties in the output measurements similar to how IT2 functions convey uncertainties in the input measurements.

To better express the Rule Outputs, we can break down the IT1 constants into a set of several (infinite) crisp constants similar to the IT2 breakdown in Section III-A.1. For each constant c_0^l , we can pick a single crisp value c_i^l' to be used in the calculation of the Rule Output as shown in Fig. 5b and Fig. 5c.

For a particular choice of $c_0^l, c_1^l, \dots, c_p^l$ the output for the l th rule is:

$$y^l = c_0^l + c_1^l x_1 + \dots + c_p^l x_p \quad (12)$$

This will lead to crisp rule outputs y^1, y^2, \dots, y^m for each rule $l = 1, 2, \dots, m$, for the chosen set of crisp values of output coefficients.

E. Interval Type-2 Output Processing

Recall in Section III-A, we broke down the IT2 memberships into a set of ET1 memberships which led to the calculation of the rule firing levels in Section III-C. Using this method, we can find the firing levels for the rules as f^1, f^2, \dots, f^m for a particular choice of ET1 memberships.

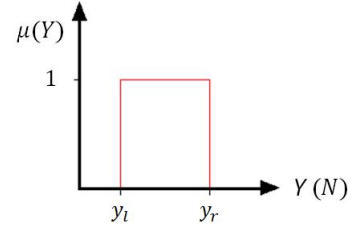


Fig. 6: All Possible TSK Outputs

Also, in Section III-D we broke down the IT1 rule output MFs into a set of crisp constants which led to the calculation of the rule outputs. For a particular choice of ET1 memberships and output coefficients, we will find the outputs y^1, y^2, \dots, y^m .

The Output processing block takes the Rule Outputs and their respective Firing Levels and calculates the weighted average output as follows:

$$Y_{TSK} = \frac{\sum_{i=1}^m f^{i'} y^{i'}}{\sum_{i=1}^m f^{i'}} \quad (13)$$

This leads to a single crisp output calculated from the chosen ET1 MFs and output coefficients. However, different choices of ET1 MFs and output coefficients lead to different output values. Then, the output is the collection of all weighted output calculated for different set of ET1 MFs and output coefficients. Since all input fuzzy sets and output coefficients are assumed to be connected sets, the collection of weighted outputs for different choices of ET1 MFs and output coefficients form a connected IT1 fuzzy set will be a connected interval Type-1 set bounded by y_l and y_r , as shown in Fig. 6. The lower bound can be found as follows:

$$y_l = \min_{\underline{f}^i, \bar{f}^i} \left\{ Y_{TSK} = \frac{\sum_{i=1}^m \underline{f}^i y_l^i}{\sum_{i=1}^m \underline{f}^i} \right\} \quad (14)$$

Where y_l^i are the minimum values for $y^{i'}$; also, \underline{f}^i and \bar{f}^i are the minimum and maximum values of $f^{i'}$, $i = 1, \dots, m$. Similarly, the upper bound can be found as follows:

$$y_r = \max_{\underline{f}^i, \bar{f}^i} \left\{ Y_{TSK} = \frac{\sum_{i=1}^m \bar{f}^i y_r^i}{\sum_{i=1}^m \bar{f}^i} \right\} \quad (15)$$

Where y_r^i denotes the maximum values for $y^{i'}$. The minimum and maximum value of $y^{i'}$ and $f^{i'}$ can be found as follows:

$$\underline{f}^l = \underline{F}_1^l(x_1) * \underline{F}_2^l(x_2) * \dots * \underline{F}_p^l(x_p) \quad (16)$$

$$\bar{f}^l = \bar{F}_1^l(x_1) * \bar{F}_2^l(x_2) * \dots * \bar{F}_p^l(x_p) \quad (17)$$

$$y_l^i = \underline{c}_0^l + \underline{c}_1^l x_1 + \underline{c}_2^l x_2 + \dots + \underline{c}_p^l x_p \quad (18)$$

$$y_r^i = \bar{c}_0^l + \bar{c}_1^l x_1 + \bar{c}_2^l x_2 + \dots + \bar{c}_p^l x_p \quad (19)$$

Where \underline{F}_p^l and \bar{F}_p^l represent the lower and upper ET1 MF, which are the bounds of their corresponding input IT2 MF, $F_{p,j}$, mentioned in Section III-B.

F. Defuzzification

Once the final upper and lower bounds of weighted outputs, y_l and y_r are calculated, we will have the output value as an Interval Type-1 fuzzy set, which was shown in Fig. 6. We now should convert this output fuzzy set to a crisp value as the control output to be applied to the plant. This process, known as defuzzification, can be shown for this type of fuzzy sets by averaging the upper and lower bounds as follows:

$$y = \frac{y_l + y_r}{2} \quad (20)$$

This final crisp output, y represents the final response of the IT2 TSK FLC for the crisp inputs x and v . The interval $[y_l, y_r]$ represents its associate uncertainty range.

G. Refining the algorithm for the Interval Type-2 TSK Fuzzy Controller

Although for the proof of concept, we went through a long procedure including the decomposition of the input fuzzy sets into ET1 fuzzy sets (discussed in Section III-A.1), and the rule outputs into sets of crisp output values (discussed in Section III-D), but at the end, as we presented in Section III-E and III-F, only the upper and lower bounds of input membership functions and coefficients of output rules are enough to calculate the control output. This reduces the computation cost and simplifies the process remarkably. The procedure for processing the output value of the proposed control structure is described in Algorithm 1.

Algorithm 1 Interval Type-2 TSK Fuzzy Control

Input: crisp inputs, rule base & input and output MFs
Output: crisp output control signal & uncertainty range

Begin Procedure

Step 1: For the given set of input signals, compute upper and lower bound for the firing level of each rule (\underline{f}^i and \bar{f}^i), $i = 1, \dots, m$ using (16) and (17).

Step 2: For the given set of input signals, compute upper and lower bound for the rule outputs (y_l^i and y_r^i), $i = 1, \dots, m$, using (18) and (19).

Step 3: Compute the upper and lower bounds for the output signal (y_l and y_r) using (14) and (15).

Step 4: Compute the output control signal y using (20).

Step 5: Return output control signal, y and uncertainty range, $[y_l, y_r]$.

Step 6: **IF** there is a new input signal,
 Go to **Step 1**
Else, wait for new input signal

End Procedure

Remark 1: Note that Step 1 and Step 2 are independent and can be processed in parallel thus contributing to reduced computation time. This is possible through the use of the TSK technique.

IV. SIMULATION RESULTS

To verify the proposed control algorithm, we have applied it to a cart centering problem. In this problem, assume that a cart on a frictionless plane is forced to the center position of the plane. The inputs to the controller are the position, x , and the velocity, v , of the cart at a given time, t , while the control signal is the applied force, F . The cart dynamics can be described as:

$$x(t + \tau) = x(t) + v(t)\tau + \frac{1}{2} \frac{F(t)}{m} \tau^2 \quad (21)$$

$$v(t + \tau) = v(t) + \frac{F(t)}{m} \tau \quad (22)$$

Where τ is the controller sampling time, and m is the mass of the cart. The problem is, having the cart at the initial position, x_0 , with the initial velocity, v_0 , how can we design a T2 FLC to apply the force, F , to bring the cart to the center, $x = 0$, and keep it there with $v = 0$.

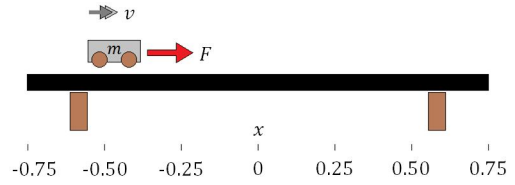


Fig. 7: The Cart Centering Problem

The initial position, x_0 is restricted from $-0.5m$ to $0.5m$. The initial velocity v_0 , is restricted from $-0.5m/s$ to $0.5m/s$. The output force F is restricted from $-0.16N$ to $0.16N$. The sampling time and mass used are $\tau = 0.1s$ and $m = 0.2kg$.

For the cart centering control, we have $X_1 = x$ and $X_2 = v$ as the control inputs with each input membership function composed of 5 fuzzy sets, namely: negative large (NL), negative small (NS), zero (0), positive small (PS) and positive large (PL), shown in Fig. 8a and Fig. 8b.

The distance between the upper and lower MFs represents the uncertainty of the sensor reading for the position and velocity inputs. The ranges of the upper and lower MFs for each of the 5 fuzzy sets are shown in Table I.

TABLE I: Range of values for the Upper and Lower MFs for both the Position and Velocity MFs.

	Upper MF	Lower MF
NL	-0.75 to -0.20	-0.75 to -0.30
NS	-0.55 to 0.05	-0.45 to -0.05
0	-0.30 to 0.30	-0.20 to 0.20
PS	-0.05 to 0.55	0.05 to 0.45
PL	0.20 to 0.75	0.30 to 0.75

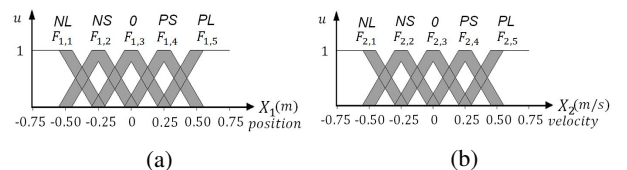


Fig. 8: IT2 Membership Functions for (a) Position, and (b) Velocity, of the cart centering problem.

The rule outputs are in the form of $y^l = c_0^l$, $l = 1, \dots, 5$, where rules $l = 1, 2, \dots, 5$ represent the labels NL, NS, 0, PS and PL. Table II gives the range of values of each Rule Coefficient. The rule base for this cart centering problem is shown in Table III.

TABLE II: Rule Coefficient Center and Range Values

Rule number	Rule label	Center	Range
1	NL	-0.16	-0.18 to -0.14
2	NS	-0.08	-0.10 to -0.06
3	0	0	-0.02 to 0.02
4	PS	0.08	0.06 to 0.10
5	PL	0.16	0.14 to 0.18

TABLE III: Rule Base for the Cart Centering Problem

		x (position)				
		NL	NS	0	PS	PL
v (velocity)	NL	PL	PL	PL	PS	0
	NS	PL	PL	PS	0	NS
	0	PL	PS	0	NS	NL
	PS	PS	0	NS	NL	NL
	PL	0	NS	NL	NL	NL

An example rule taken from the table is:

IF position is NS, and velocity is PL, **THEN** force is NS

To demonstrate the controller's noise handling capabilities, a uniformly distributed two-fold noise is introduced into the system. The noise is introduced into the input signals for position and velocity, before control processing, to simulate sensor noise. Noise is also added to the force output, after control processing, to simulate actuator noise. The noise range used is -0.03 to 0.03 for both position and velocity.

Now, for initial conditions $x_0 = 0.5m$ and $v_0 = 0.5m/s$, the position, velocity, and control signal are calculated using the proposed IT2 TSK FLC algorithm. The results are shown in Fig. 9a and Fig. 9b.

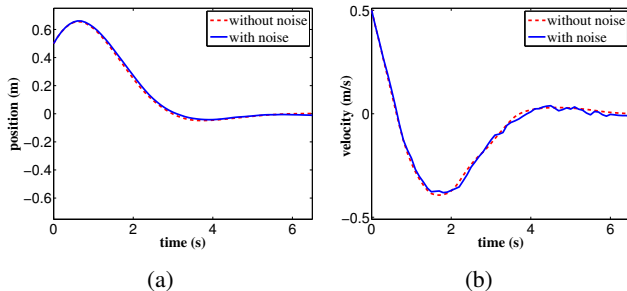


Fig. 9: (a) Cart Position, and (b) Cart Velocity for $x_0 = 0.5m$ and $v_0 = 0.5m/s$.

As can be seen in the figures 9a and 9b, the controller can successfully handle the noise and brings the cart to the center and will keep it there with zero velocity, from its initial point, smoothly and with little oscillation.

V. CONCLUSION AND FUTURE WORK

In this paper, we developed a IT2 TSK FLC algorithm. The proposed control algorithm, on the one hand, takes

advantage of Fuzzy type-2 controller, and on the other hand, significantly reduces the computation cost by describing the inputs using IT2 FSs and integrating the control algorithm with the TSK control technique. The developed controller was applied to a cart centering problem and the simulation results verified the efficacy of the algorithm. As future work, we intend to develop an algorithm for tuning the parameters of the proposed control structure and apply the developed controller to a real robotic platform where we can compare the performance of the system compared to other existing approaches.

ACKNOWLEDGMENT

The second and fourth authors would like to acknowledge the support from Air Force Research Laboratory and OSD for sponsoring this research under agreement number FA8750-15-2-0116. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory, OSD, or the U.S. Government.

REFERENCES

- [1] 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*, vol. 1, no. 3, pp. 183–193, 2011.
- [2] F. Liu, "An efficient centroid type-reduction strategy for general type-2 fuzzy logic system," *Information Sciences*, vol. 178, no. 9, pp. 2224–2236, 2008.
- [3] J. M. Mendel, *Uncertain rule-based fuzzy logic system: introduction and new directions*. Prentice-Hall PTR, 2001.
- [4] J. M. Mendel and R. B. John, "Type-2 fuzzy sets made simple," *Fuzzy Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 117–127, 2002.
- [5] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *Fuzzy Systems, IEEE Transactions on*, vol. 14, no. 6, pp. 808–821, 2006.
- [6] J. Mendel and M. Rajati, "On computing normalized interval type-2 fuzzy sets," *Fuzzy Systems, IEEE Transactions on*, vol. 22, no. 5, pp. 1335–1340, 2014.
- [7] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence)*. IEEE International Conference on. IEEE, 2008, pp. 1425–1432.
- [8] Q. Ren, M. Balazinski, and L. 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*, vol. 63, no. 9-12, pp. 1057–1063, 2012.
- [9] M. Sugeno, *Industrial applications of fuzzy control*. Elsevier Science Inc., 1985.
- [10] C. L. Tseng, S. Y. Wang, S. C. Lin, and Y. Y. Chen, "Interval type-2 takagi-sugeno fuzzy controller design for a class of nonlinear singular networked control systems," in *Fuzzy Theory and its Applications (iFUZZY), 2012 International Conference on*. IEEE, 2012, pp. 268–272.
- [11] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems," in *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2131–2138.
- [12] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning ii," *Information sciences*, vol. 8, no. 4, pp. 301–357, 1975.