Speeding Up Interval Type 2 Fuzzy Logic Computation Using Fuzzy Bilinear Interpolation Look-Up Table

Wakhyu Dwiono, Arif Johar Taufiq, and M. Taufiq Tamam
Department of Electrical Engineering, Universitas Muhammadiyah Purwokerto
Jl Raya Dukuhwaluh Purwokerto 53182, Indonesia.
wakhyudwiono@ump.ac.id

***Abstract*—Interval type 2 fuzzy logic system (IT2FLS) is one of the most straightforward approaches of type 2 fuzzy logic system that is relatively easy to implement. In line with the enhanced capability of the microprocessor, the interval type 2 fuzzy logic system becomes one of the solutions to overcome any problems associated with uncertainty. However, due to computation speed and limited resources, a real-time problem occurs when the IT2FLS is implemented to the low-cost microcontrollers. Based on this problem, it is essential to speed up the computation of IT2FLS. This paper proposes a method named Fuzzy Bilinear Interpolation-Look Up Table algorithm that uses saved information and fuzzy interpolation to obtain the output based on neighboring data to get higher data accuracy. From the simulation results, it was found that Fuzzy Bilinear Interpolation-LUT IT2FLS provides the same performance and computes 14 times faster than IT2FLS.**

***Index Terms*—Fuzzy Bilinear Interpolation; Interval Type 2 Fuzzy; IT2FLS; Speed Up IT2FLS.**

I. INTRODUCTION

IT2FLS on embedded systems using low-cost microcontrollers needs an appropriate method due to the limited resources support. However, it is different if it is applied to a personal computer. IT2FLS requires complex computation from fuzzification to defuzzification [1-3]. The famous defuzzification process is obtained using the KM algorithm that may need a repeating calculation [2-3].

As type 1 fuzzy logic system (T1FLS) has been implemented as controllers [4-13], some of them are embedded systems [4-6]. IT2FLS, which is a simplification of Type 2 Fuzzy Logic System (T2FLS) has better performance than type 1 fuzzy logic systems because of its ability to handle uncertainty [2]. IT2FLS has better performance because it has a Footprint of Uncertainty (FOU) on its membership function [14]. As a controller, IT2FLS is applied as either in direct form [15-19], or in combination with other methods, such as combining with PID controller [20], functioning as an observer on the robust controller [21], model reference adaptive [22], sliding mode [23], and optimal hierarchy [24].

As mentioned above, most of the IT2FLS are used as Interval Type 2 Fuzzy Logic Controllers (IT2FLC), in which some of them are simulated using computer software [15-18], [20-25]. IT2FLS has also been applied as an embedded controller [19]. In some cases, the IT2FLC has been applied to the PIC 18F4685 with inverted pendulum on cart plant simulated by using MATLAB.

How to accelerate fuzzy inference computation have been done by many researchers both for T1FLS and IT2FLS. The methods for this are Look-Up Table (LUT) [12], [13], [25], [27], FPGA [10] and by applying the Graphics Processing Unit (GPU) [26]. However, in the LUT method and GPU usage there are still limitations because the input discretization for data table is the membership function [12]. In this case, when the membership function is many, it will slow down the computation because of the large number of data that must be accessed. The simpler LUT method is established by making the input into some regions, with each region has its fuzzy rule [25]. However, this will decrease its accuracy because if the input range is zoomed, a lot of different information that is considered the same will exist. This problem can be fixed through Fuzzy LUT by adding the interpolation process [27]. In this case, the table is arranged based on the bit data resolution, and it uses 6 bits data that is interpolated to 8 bits of data to save the memory storage.

The purpose of this paper is to develop a slightly different method from [27] by using a fuzzy bilinear interpolation method. The data will be interpolated with discretized source data in different ranges. The rest of this paper is organized as follows. Section 2 presents the KM Algorithm of IT2FLS type reducer, while Section 3 and 4 present the description of fuzzy bilinear interpolation and simulation results from the two-mass spring's x2 position control using Fuzzy Bilinear Interpolation-LUT IT2FLS.

II. KM TYPE REDUCER ALGORITHM

A detailed description of IT2FLS is found in [2]. The fuzzy interval of type 2 is shaped like a ribbon bounded by the upper membership function and the lower membership function. The area is called FOU. The degree of secondary membership is 1. If $\mu_{\tilde{A}_k^i(x_k)}$ is the set of interval type 2 fuzzy, then:

$$\mu_{\tilde{A}_k^i(x_k)} = \int_{w^i} \in \left[\underline{\mu}_{\tilde{A}_k^i(x_k)}, \overline{\mu}_{\tilde{A}_k^i(x_k)}\right] 1/w^i \quad (1)$$

If there are n inputs, then the i_{th} fuzzy rule is $R^i = IF\ x_1\ is\ \tilde{A}_1^i, x_2\ is\ \tilde{A}_2^i, \ldots, x_n\ is\ \tilde{A}_n^i, THEN\ y\ is\ \tilde{B}^i$ and if the fuzzification is singleton, then the fired strength formulation of each rule is:

$$\underline{f}^i = \underline{\mu}_{\tilde{A}_1^i(x_1)} * \ldots * \underline{\mu}_{\tilde{A}_n^i(x_n)} \quad (2)$$

$$\bar{f}^i = \bar{\mu}_{\tilde{A}_1^i(x_1)} * \dots * \bar{\mu}_{\tilde{A}_n^i(x_n)} \quad (3)$$

If the fuzzy rules R = 1,2,3,...,J, then each rule has a fired strength value that lies between $\underline{f}^1, \underline{f}^2, \underline{f}^3, \dots, \underline{f}^J$ and $\bar{f}^1, \bar{f}^2, \bar{f}^3, \dots, \bar{f}^J$ then the output of each rule extends from $y_l^1, y_l^2, y_l^3, \dots, y_l^J$ to $y_r^1, y_r^2, y_r^3, \dots, y_r^J$ which is assumed to be arranged from the smallest to the largest, then y_l and y_r can be calculated using KM-algorithm [2] as follows:

1. Calculate the initial value of y_l using the following formula:

$$y_l = \sum_{i=1}^{J} f_l^i y_l^i / \sum_{i=1}^{J} f_l^i \quad (4)$$

With $f_l^i = \left(\underline{f}^i + \bar{f}^i\right)/2$, then set $y_l' = y_l$.
2. Find $L(1 \le L \le J-1)$, by condition $y_l^L \le y_l' \le y_l^{L+1}$
3. Calculate the value of y_l using (4) with $f_l^i = \bar{f}^i$ for $i \le L$, and $f_l^i = \underline{f}^i$ for $i > L$, then set $y_l'' = y_l$,
4. If $y_l'' = y_l'$ then calculation is completed so $y_l = y_l''$.
5. If $y_l'' \ne y_l'$, set y_l'' equal to y_l', then go back to step 2.

The method to calculate y_r is using the same steps as calculating y_l only by replacing variable y_l^L with y_r^R in step 2, $R(1 \le R \le J-1)$ so that $y_r^R \le y_r' \le y_r^{R+1}$, in step 3 with $f_r^i = \underline{f}^i$ for $i \le R$ and $f_r^i = \bar{f}^i$ for $i > R$.

III. FUZZY BILINEAR INTERPOLATION ALGORITHM

Although the KM type reducer algorithm is simplifying the IT2FLS computation, it is still substantial enough for the computer system, when it is applied to the low-cost microcontroller. Therefore, the idea appears to utilize the fuzzy input-output mapping table. This paper discusses two inputs and one output mapping table. Input 1 and 2 are discretized as much as n pieces (n is an odd number) so that two matrices (1 x n) are formed. Input 1 and input 2 has a range of values between -A and A. To create an input-output map, each component of inputs matrices is combined to calculate its IT2FLS output using the KM type reducer Algorithm. It will form an IT2FLS output matrix with size (n x n). Due to the limitation of resolution in this discretization process, it is necessary to interpolate to calculate IT2FLS output for the instantaneous inputs that are unavailable in the table, which is done by Fuzzy Bilinear Interpolation-LUT Algorithm based on the continuous contour surface of the input-output map. Figure 1 shows the control surface of the IT2FLS I/O map that has four different spacing, 5x5, 9x9, 15x15 and 19x19 of a data table. This difference spacing data table can be used in the interpolation process.

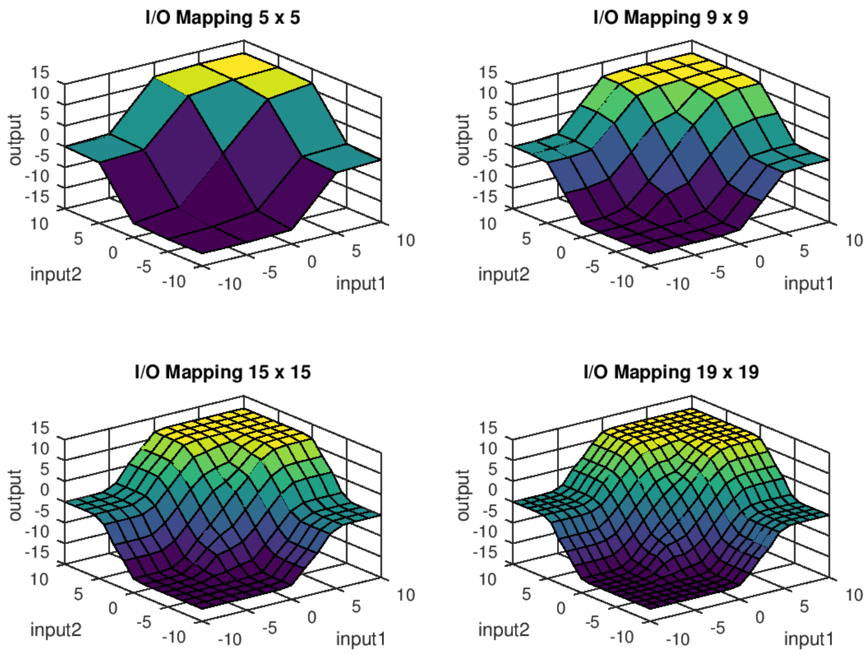

Figure 1: The continuous control surface of the input-output map for different resolution discretization inputs IT2FLS

The interpolation process to obtain the output is shown using the illustration in Figure 2 and 3. Input 1 and input 2 stored data index are symbolized as *x1(k)*, *x1(k+1)*, *x2(k)* and *x2(k+1)* that being formed in the integer value. Then *O(x1(k),x2(k))*, *O(x1(k+1),x2(k))*, *O(x1(k),x2(k+1))* and *O(x1(k+1), x2(k+1))* are the outputs value corresponding to input 1 and input 2 stored data index. Those data are stored in the table. At a time, t is obtained data *e(t)* and *de(t)*, then the data is matched with the table. The position of *e(t)* lies between *x1(k)* and *x1(k+1)*, position of *de(t)* lies between *x2(k)* and *x2(k+1)*. Then the output value of the predicted result is $\hat{O}\left(e(t), de(t)\right)$.

To calculate the IT2FLS output, the four neighboring output values are interpolated using fuzzy bilinear interpolation to get the value of $h1$, $h2$ and $\hat{O}\left(e(t), de(t)\right)$. Each side of the rectangular area has two low membership functions (e_{dn} and de_{dn}) and two high membership functions (e_{up} and de_{up}), as shown in Figure 4 and 5.

The fuzzy membership functions for input 1 and input 2 are identical. Membership function $\mu\, e_{dn}$ and $\mu\, de_{dn}$ are:

$$\mu\, e_{dn} = \frac{x1(k+1) - e(t)}{x1(k+1) - x1(k)} \quad (5)$$

$$\mu\, de_{dn} = \frac{x2(k+1) - de(t)}{x2(k+1) - x2(k)} \quad (6)$$

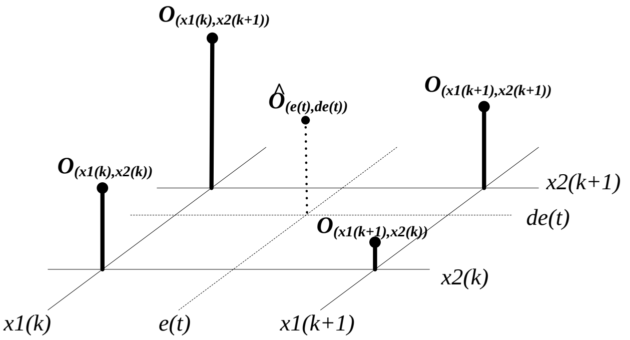

Figure 2: IT2FLS output calculation for the instantaneous inputs *e(t)* and *de(t)* that is unavailable in the table

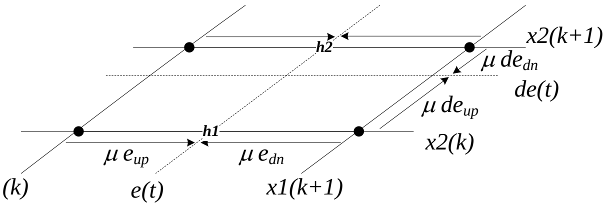

Figure 3: Two sides of rectangular interpolated value based on four neighbouring output values

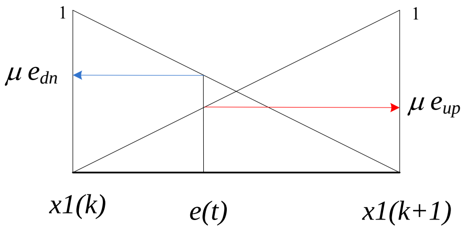

Figure 4: Calculating μe_{dn} and μe_{up} of input 1

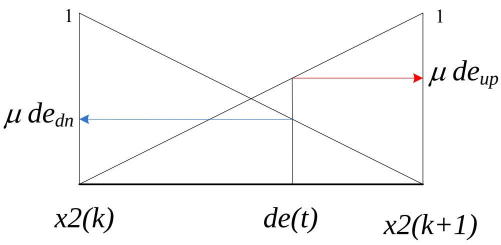

Figure 5: Calculating μde_{dn} and μde_{up} of input 2

Membership function $\mu\, e_{up}$ and $\mu\, de_{up}$ are:

$$\mu\, e_{up} = (e(t) - x1(k))/(x1(k+1) - x1(k)) \quad (7)$$

$$\mu\, de_{up} = (de(t) - x2(k))/(x2(k+1) - x2(k)) \quad (8)$$

Table data indices are integer $(x1(k+1) - x1(k) = x2(k+1) - x2(k) = 1)$, then Equations 5 to 8 become:

$$\mu\, e_{dn} = x1(k+1) - e(t) \quad (9)$$

$$\mu\, de_{dn} = x2(k+1) - de(t) \quad (10)$$

$$\mu\, e_{up} = e(t) - x1(k) \quad (11)$$

$$\mu\, de_{up} = de(t) - x2(k) \quad (12)$$

To process *e(t)* and *de(t)*, it must be normalized against the index and the number of discretization of a data table, using the following equation:

$$id_e = \left(\frac{(e(t))}{\left(\frac{A}{B}\right)}\right) + (B+1) \quad (13)$$

$$id_{de} = \left(\frac{(de(t))}{\left(\frac{A}{B}\right)}\right) + (B+1) \quad (14)$$

With id_e and id_de are normalized input data against the index and number of discretization of a data table, A is the maximum value of the input data range. $B = \left(\frac{n+1}{2}\right) - 1$, n is the number of discretization of a data table, so that the Equations 9 to 12 become:

$$\mu\, e_{dn} = x1(k+1) - id\_e \quad (15)$$

$$\mu\, de_{dn} = x2(k+1) - id\_de \quad (16)$$

$$\mu\, e_{up} = id\_e - x1(k) \quad (17)$$

$$\mu\, de_{up} = id\_de - x2(k) \quad (18)$$

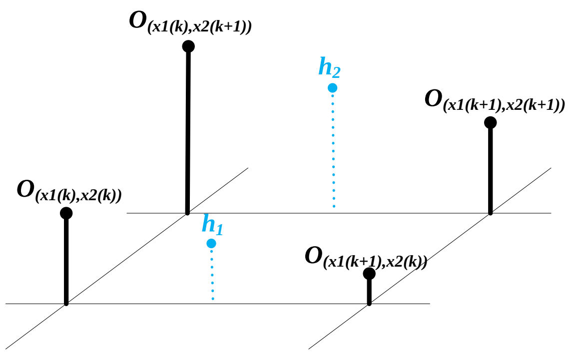

Figure 6: Calculating *h1* and *h2* using linear interpolation

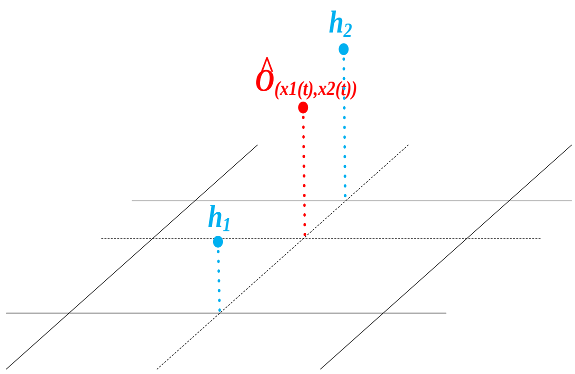

Figure 7: Illustration of $\hat{O}(e(t), de(t))$ using linear interpolation

Value of *h1*, *h2* and $\hat{O}(e(t), de(t))$ (as described in Figures 6 and 7) are calculated by using fuzzy bilinear interpolation formula as follows:

$$h1 = (((\mu\_edn * O1) + (\mu\_eup * O2)))/\,(\mu\_edn + \mu\_eup) \quad (19)$$

$$h2 = ((\mu_{edn} * O3) + (\mu\_eup * O4))/(\mu\_edn + \mu\_eup) \quad (20)$$

$$\hat{O}(e(t), de(t)) = ((\mu\_dedn * h1) + (\mu\_deup * h2))/\,(\mu\_dedn + \mu\_deup) \quad (21)$$

where: $O1 = O\big(x1(k), x2(k)\big)$
$O2 = O\big(x1(k+1), x2(k)\big)$
$O3 = O\big(x1(k), x2(k+1)\big)$
$O4 = O\big(x1(k+1), x2(k+1)\big)$

Low index for input 1 *(x1(k) = iddn_e)* and input 2 *(x2(k) = iddn_de)* were obtained by using Equations 22 and 23.

$$\mathrm{iddn\_e} = \mathrm{floor(id\_e)} \quad (22)$$

$$iddn\_de = floor(id\_de) \quad (23)$$

High index for input 1 (*x1(k) = iddn_e*) and input 2 (*x2(k) = iddn_de*) were obtained by using Equations 24 and 25.

$$idup\_e = ceil(id\_e) \quad (24)$$

$$idup\_de = ceil(id\_de) \quad (25)$$

A. Output Calculation Steps

Output ($\hat{O}\big(e(t), de(t)\big)$) calculation in detail is described in the following steps:

1) Obtain input 1 ($e(t)$) and input 2 ($de(t)$).
2) Set $e(t)$ and $de(t)$, if exceed the minimum and maximum bound of the input range to the minimum and maximum values. If $e(t) < -A$ then $e(t) = -A$, if $e(t) > A$ then $e(t) = A$. If $de(t) < -A$ then $de(t) = -A$, if $de(t) > A$ then $e(t) = A$.
3) Set the value $e(t)$ as id_e and $de(t)$ as id_de that lies between two neighboring index, using Equations 13 and 14.
4) Compute low index for input 1, $iddn\_e = floor(id\_e)$, and high index for input 1, $idup\_e = ceil(id\_e)$.
5) Compute low index for input 2, $iddn\_de = floor(id\_de)$, and high index for input 2, $idup\_de = ceil(id\_de)$.
6) Get 4 neighboring outputs from output table that their index computed at step 4 and 5.

$$O1 = O\big(x1(k), x2(k)\big) = output(iddn_e, iddn_{de})$$
$$O2 = O\big(x1(k+1), x2(k)\big) = output(idup_e, iddn_{de})$$
$$O3 = O\big(x1(k), x2(k+1)\big) = output(iddn_e, idup_{de})$$
$$O4 = O\big(x1(k+1), x2(k+1)\big) = output(idup_e, idup_{de})$$

7) Compute the membership degree of instantenous $e(t)$ $de(t)$ on two sides of rectangular using Equations 15 to 18.
8) Compute maximum and minimum value of *O1*, *O2, O3* and *O4.*
9) If maximum value equals to the minimum than $\hat{O}\big(e(t), de(t)\big)$ = maximum or minimum value. If not, $\hat{O}\big(e(t), de(t)\big)$ is calculated using Equations 19 to 21.

B. Illustrative Example

Assuming that the table of IO map with two inputs *e(t)* and *de(t)* with each of input lies between -10 to 10 and is discretized as 9x9; therefore, A = 10, n = 9, B = 4 as shown in Table 1.

Table 1
Two Inputs One Output IO Map Data Table

e(t)	*de(t)*	Original Index	-10 1	-7.5 2	-5 3	-2.5 4	0 5	2.5 6	5 7	7.5 8	10 9
original	**index**										
-10	**1**		-12	-12	-12	-12	-11.6	-6.06	-0.45	0	0
-7.5	**2**		-12	-12	-12	-11.9	-11.6	-6.06	-0.45	0	0
-5	**3**		-12	-12	-11.7	-11.4	-10.4	-5.1	0	0.4	0.4
-2.5	**4**		-12	-11.9	-11.4	-6.9	-5.4	0	5.2	6.1	6.1
0	**5**		-11.6	-11.6	-10.4	-5.5	0	5.5	10.4	11.6	11.6
2.5	**6**		-6.1	-6.1	-5.2	0	5.4	6.9	11.4	11.9	12
5	**7**		0.4	0.4	0	5.1	10.4	11.4	11.7	12	12
7.5	**8**		0	0	0.45	6.06	11.6	11.9	12	12	12
10	**9**		0	0	0.45	6.06	11.6	12	12	12	12

At a time, the value of *e(t)* is 2.3 whereas *de(t)* is 4.5, then *id_e* and *id_de* are calculated using Equations 13 and 14.

$$id_e = \left(\frac{2.3}{\left(\frac{10}{4}\right)}\right) + 5 = 5.92$$

Low index for *e(t)=x1(k)= iddn_e*=floor(5.92)=5
High index for *e(t) =x1(k+1)=idup_e*=ceil(5.92)=6

$$id\_de = \left(\frac{4.5}{\left(\frac{10}{4}\right)}\right) + 5 = 6.8$$

Low index for *de(t)=x2(k)= iddn_de*=floor(6.8)=6
High index for *de(t)=x2(k+1)= idup_de*=ceil(6.8)=7

Based on Table 1 (shaded space), the value of *O1* = 5.4 is derived from the data table with index (5,6), *O2* = 6.9 is derived from the table data with index (6,6), *O3* = 10.4 is derived from the data table with index (5,7) while *O4* = 11.4 is derived from the data table with the index (6,7).

Membership degree calculation can be obtained by using Equations 15 to 18, as explained below:

$$\mu\, e_{dn} = 6 - 5.92 = 0.08$$
$$\mu\, de_{dn} = 7 - 6.8 = 0.2$$
$$\mu\, e_{up} = 5.92 - 5 = 0.92$$
$$\mu\, de_{up} = 6.8 - 6 = 0.8$$

h1, *h2* and *O(e(t),de(t))* can be calculated by using Equations 19 to 21.

$$h1 = \big((0.08 * 5.4) + (0.92 * 6.9)\big)/\,(0.08 + 0.92) = 6.78$$
$$h2 = \big((0.08 * 10.4) + (0.92 * 11.4)\big)/\,(0.08 + 0.92) = 11.32$$
$$\hat{O}(e(t), de(t)) = ((0.2 * 6.78) + (0.8 * 11.32))/\,(0.2 + 0.8) = 10.41$$

If bilinear interpolation process is described, then the output of interpolation results can be seen in Figure 8.

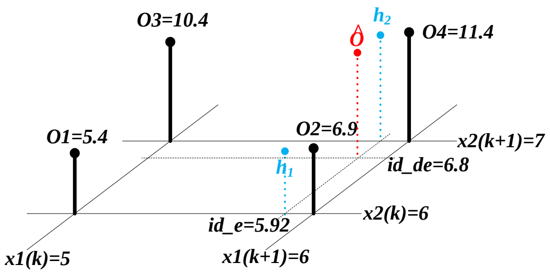

Figure 8: Output of fuzzy bilinear interpolation for *e(t)* =2.3 and *de(t)*=4.5

IV. Simulation Result And Discussion

In this simulation, the Mod-LUT ITFLS was applied as a controller to control the x_2 position of the two mass-spring systems (ACC Benchmark Problems), as shown in Figure 9. This simulation has been run using GNU Octave 4.2.1, with the used membership function, as shown in Figure 4 and controller structure [27], as shown in Figure 10.

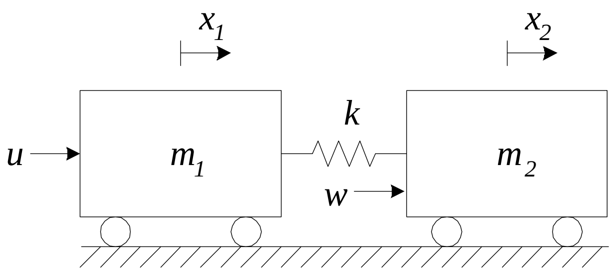

Figure 9: Controlling x2 position for the two mass-spring systems

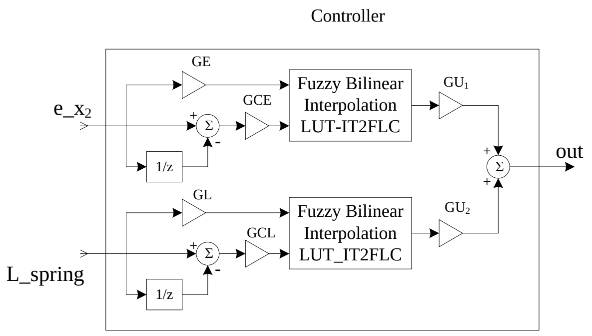

Figure 10: Controller structure that was used to regulate the position of *x2*

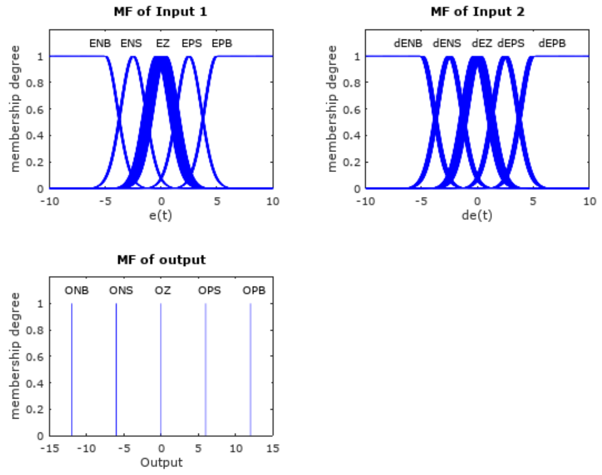

Figure 11: Input and output membership function that was used in the simulation

The state-space equation of the two-mass spring system is:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -{}^{k}/_{m_1} & {}^{k}/_{m_1} & 0 & 0\\ {}^{k}/_{m_2} & -{}^{k}/_{m_2} & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \begin{bmatrix}0\\ 0\\ {}^{1}/_{m_1}\\ 0\end{bmatrix}u + \begin{bmatrix}0\\ 0\\ 0\\ {}^{1}/_{m_2}\end{bmatrix}w \qquad (26)$$

$$y = x_2 + v \qquad (27)$$

A. Surface Control

The control surface of real IT2FLS and Fuzzy Bilinear Interpolated-LUT IT2FLS using data table 5x5, 15x15 and 25x25 are shown in Figure 12. Based on the image, it can be seen that when the input is discretized to many more pieces, the surface control becomes smoother.

B. Simulation Result

The parameters for the plant are k=0.2 N/m, m_1=1kg, m_2=2kg; with gain of observer L_1=0.3783, L_2=1.1493, L_3=0.0933 and L_4 = 0.6605. While the parameters for the mod-LUT IT2FLS controller are *GE*=0.5, *GCE*=100, *L*=0.9, *GCL*=10, GU_1=0.42 and GU_2=0.56.

This simulation was run using a PC with an Intel (R) Core (TM) i3-2330M processor with 4 GB of RAM size. During the simulation, two noise sources of *w* and *v* were added to the two mass-spring systems in the form of normal random with 0 and 1 seeds.

The simulation results were compared with the simulation using IT2FLS to demonstrate the effectiveness of the Mod-LUT algorithm. The simulation results are shown in Table 2. Number 1 is the result of IT2FLS simulation, while number 2 to 8 is the result of simulation using fuzzy bilinear interpolation-LUT with the variation of data I/O mapping table 5x5 to 35x35. IT2FLS takes much longer time, which is about 14 times longer than Mod-LUT IT2FLS. With data I/O mapping table 35x35, the performance of fuzzy bilinear interpolation-LUT IT2FLS is even better than IT2FLS. This can be seen from the smaller amount of *IAE* and *Σ of force*.

This performance can also be seen in the graph of the control system responses that are shown in Figures 13 and 14.

Figure 13 is a graph of x2 position control system response from a two mass-spring system by adding two noise sources *w* and *v*. In the simulation, the reference for x2 is 10 meters with noise (using the randn function) magnitude 1. It is seen in figure 13 that the position of x2 is coincidental, although it used three types of controller, IT2FLS, fuzzy bilinear interpolated-LUT IT2FLS 5x5 and 15x15. Thus it can be concluded that between the bilinear interpolated-LUT IT2FLS controller has the same performance as IT2FLS controller.

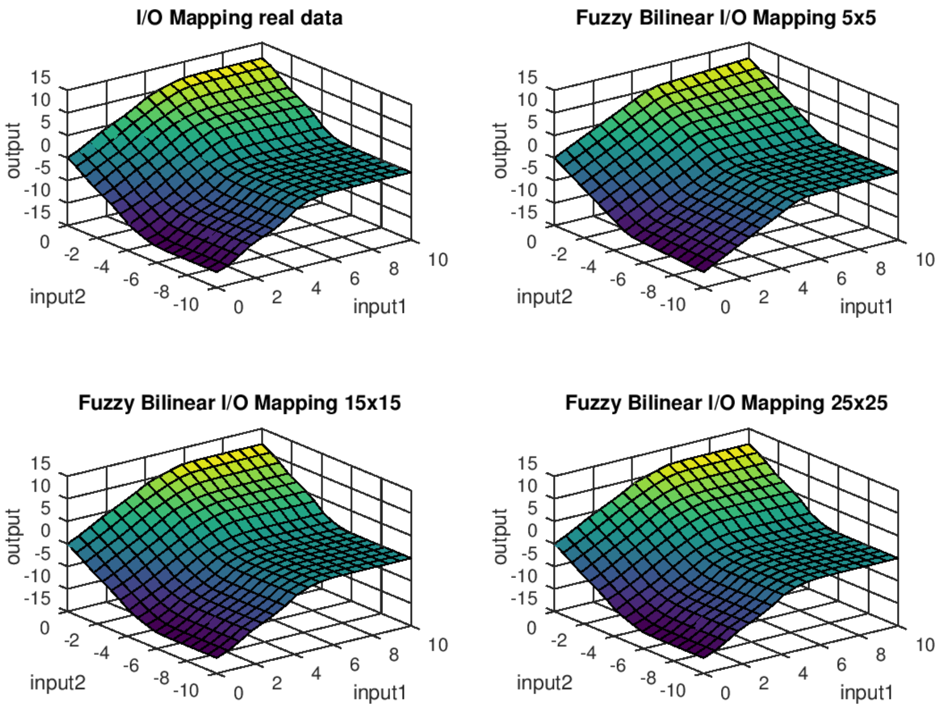

Figure 12: The control surface of real IT2FLS and Fuzzy Bilinear Interpolated-LUT IT2FLS using data table (5x5, 15x15 and 25x25)

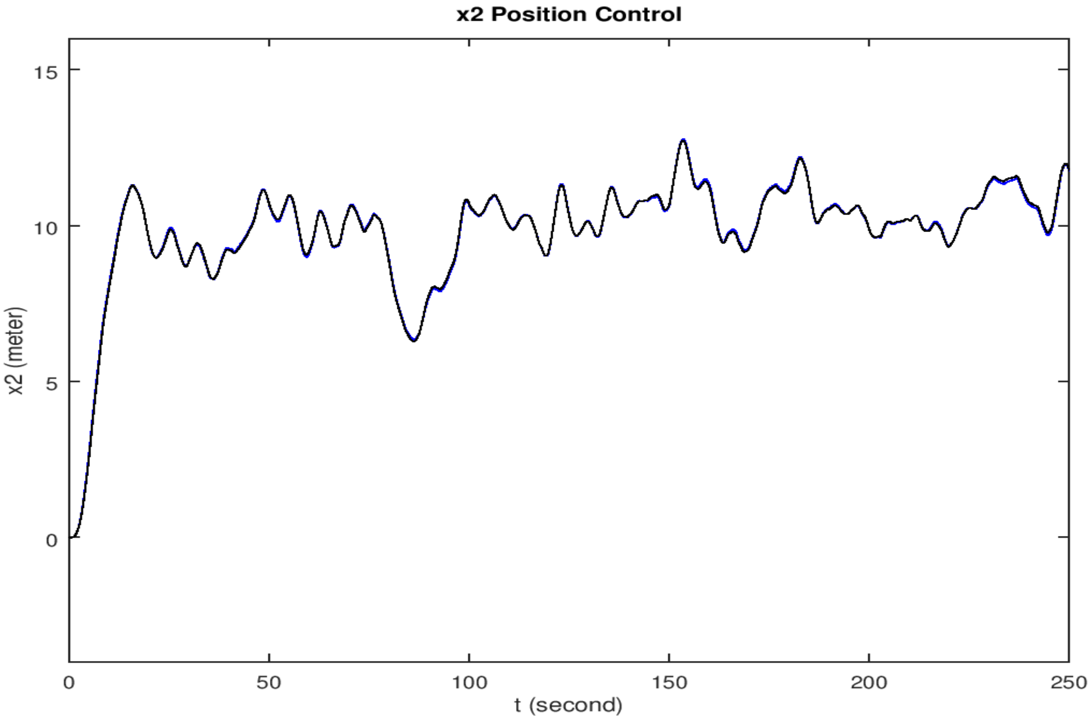

Figure 13: System response for positioning x2 at 10 m

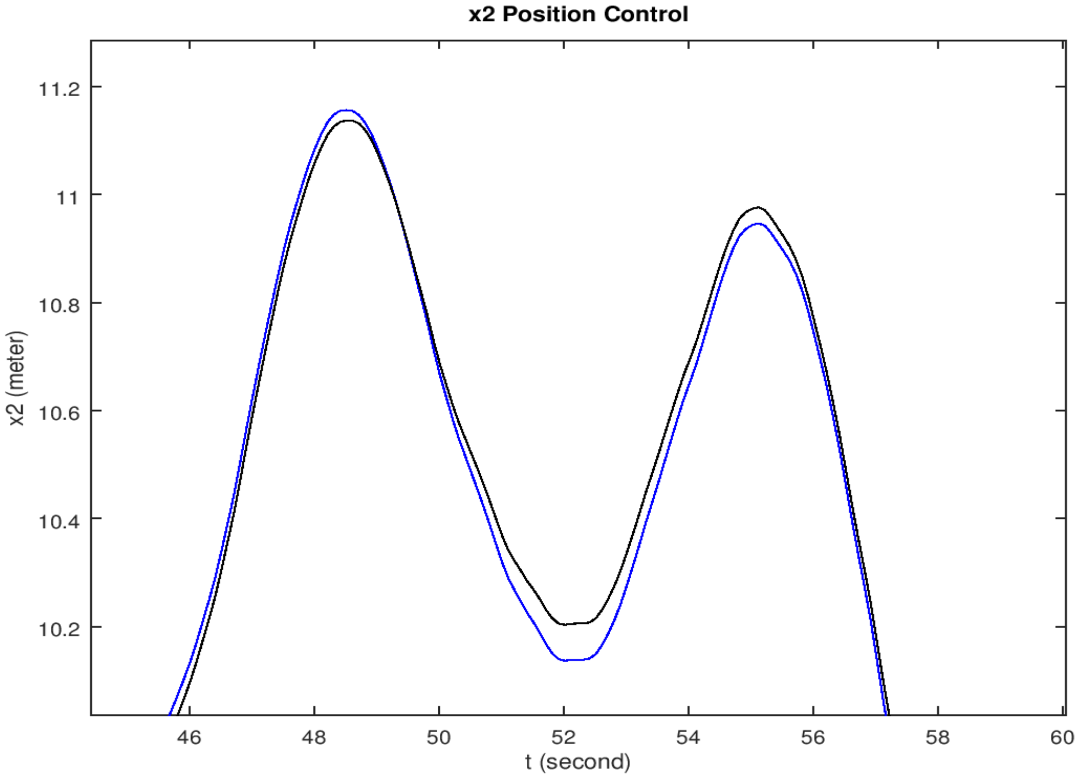

Figure 14: Magnification of figure 8 at the point of 52 seconds. The blue line is ITFLS while the black line is fuzzy bilinear interpolation-LUT IT2FLS for 15x15 discretization

Table 2
Comparison of IT2FLS and Fuzzy Bilinear Interpolation-LUT IT2FLS

No	discretization	Elapsed Time	IAE	Σ of force
1	none	40.8 sec	260.1	453.8
2	5x5	2.83 sec	365.7	413.11
3	9x9	2.83 sec	260.9	433.6
4	15x15	2.85 sec	261.9	441.6
5	19x19	2.95 sec	261	444.8
6	25x25	2.82 sec	260.3	447.8
7	29x29	2.84 sec	260.5	448.7
8	35x35	2.84 sec	260.1	450.8

Figure 14 is an enlarged image of Figure 13 at 52nd seconds. This magnification was done to show the performance difference of each controller. It appears that the fluctuation value between the black lines (the response of the control system using fuzzy bilinear interpolation-LUT) is slightly smaller than the blue line (response of the ITFLS control system). This emphasizes the evidence that is presented in Table 2, that is the total force being delivered by the fuzzy bilinear interpolation-LUT controller is smaller than ITFLC. Control system responses using IT2FLS (blue line) and fuzzy bilinear interpolation-LUT IT2FLS (black line) appear to coincide, indicating that their performance is equal (just differ slightly).

C. Implementation on a Low-cost Microcontroller

Table 3 shows information about implementation the IT2FLS and fuzzy bilinear interpolation-LUT IT2FLS on a low-cost microcontroller, PIC16F877A. The IT2FLS was unable to implement on PIC16F877A because more space of RAM were needed; hence, an error message “Not Enough Space of RAM” appeared when the program is compiled. On the other hand, fuzzy bilinear interpolation-LUT IT2FLS can be implemented, in which it has the same fixed computation time.

Table 3
Comparison of Realized IT2FLS and Fuzzy Bilinear Interpolation-LUT IT2FLS

FLC Controller Design	Hex File Size	Used Space of RAM	Computation Time
Three triangles MFs IT2FLS+LCD	Cannot be compiled because of not enough space of RAM	Greater than its RAM capacity	-
fuzzy bilinear interpolation-LUT IT2FLS 9x9 data table+LCD	3019 words (37% of capacity)	97 byte (28% of capacity)	45 ms
fuzzy bilinear interpolation-LUT IT2FLS 15 x15 data table+LCD	3381 words (41% of capacity)	97 byte(28% of capacity)	45 ms

V. CONCLUSION

Based on the simulation results, it has been proven that the Mod-LUT IT2FLS algorithm can speed up IT2FLS computing process that is about 14 times faster. By increasing the number of input discretization, the performance of the fuzzy bilinear-LUT IT2FLS will be the same with IT2FLS. When they were implemented on the low-cost microcontroller (PIC16F877A), The IT2FLS could not be implemented because of its RAM space is too small, while the fuzzy bilinear-LUT IT2FLS could be implemented with same space of RAM, although the data table size was increased from 9x9 to 15x15. Further, the computation time is the same too. Both of these need 45ms (using 20MHz crystal) for computation time. This fact will make it easier to apply to low-cost microcontroller.

REFERENCES

[1] Jerry M. Mendel, Robert I. John, and Feilong Liu, "*Interval Type-2 Fuzzy Logic Systems Made Simple*", IEEE Trans. On Fuzzy Sist, Vol. 14, No. 6, pp. 808-821, December 2006.

[2] Qilian Liang and Jerry M. Mendel, "*Interval Type-2 Fuzzy Logic Systems: Theory and Design*", IEEE Trans. On Fuzzy Syst, vol 8, No 5, pp. 535-550, October 2000.

[3] N. N. Karnik, J. M. Mendel, "*Centroid of a type-2 fuzzy set*," Inform. Sci., vol 132, pp 195-220, 2001.

[4] Iluminada Baturone, Francisco J. Moreno-Velo, Víctor Blanco, and Joaquín Ferruz , "*Design of Embedded DSP-based Fuzzy Controllers for Autonomous Mobile Robots*", IEEE Trans. On Industrial Electronics, Vol 55, No 2, pp 928-936, February 2008.

[5] Ferenc Farkas, Sándor Halász, "*Embedded Fuzzy Controller for Industrial Applications* ", Acta Polytechnica Hungarica, Vol. 3, No. 2, pp 41-63, 2006.

[6] Hajer Omrane, Mohamed Slim Masmoudi, and Mohamed Masmoudi, "*Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation*," Computational Intelligence and Neuroscience, Vol. 2016, Article ID 9548482, 10 pages, 2016. DOI:10.1155/2016/9548482.

[7] Henry N., Afiq A. Dahlan, Affandi M. Nasib, Azhar A. Aziz, and Sumeru, "*Indoor Temperature Control and Energy Saving Potential of Split Unit Air Conditioning System using Fuzzy Logic Controller*," IAENG International Journal of Computer Science, Vol. 43, No.4, pp.402-405, 2016.

[8] Abdel-Hamid Attia, Sohair F. Rezeka, Ahmed M. Saleh, "*Fuzzy logic control of the air-conditioning system in residential buildings*", Alexandria Engineering Journal, Vol. 54, No. 3, pp.395-403 September 2015.

[9] Ahmed M. El-Garhy, Gamal A. El-Sheikh, Mohamed H. El-Saify, "*Fuzzy Life-Extending Control of Anti-Lock Braking System*", Ain Shams Engineering Journal, Vol. 4, No. 4, pp. 735-751 December 2013.

[10] Zeyad Assi Obaid, Nasri Sulaiman, M. H. Marhaban, M. N. Hamidon, "*Analysis and Performance Evaluation of PD-like Fuzzy Logic Controller Design Based on Matlab and FPGA*", IAENG International Journal of Computer Science, Vol. 37, No. 2, May 2010.

[11] Lorenzo Dambrosio, "*Data-based Fuzzy Logic Control Technique Applied to a Wind System*", Energy Procedia, Vol 126, pp 690-697, September 2017.

[12] Jinwook Kim, Yoon-Gu Kim, Jinung An, "*A Fuzzy Obstacle Avoidance Controller Using a Lookup-Table Sharing Method and Its Applications for Mobile Robots*", Int J Adv Robotic Sy, Vol. 8, No. 5, pp. 39-48, 2011.

[13] Farzin Piltan, M.A. Dialame, A. Zare, A. Badri, "*Design Novel Lookup Table Changed Auto Tuning FSMC: Applied to Robot Manipulator*", International Journal of Engineering (IJE), Vol 6, No 1, pp.25-41, 2012.

[14] Jerry M. Mendel, Robert I. John, "*Footprint of uncertainty and its importance to type-2 fuzzy sets*," in Proc. 6th IASTED Int. Conf. Artificial Intelligence and Soft Computing, Banff, Canada, pp. 587–592, Jul. 2002.

[15] Lazim Abdullah, C.W.R. Adawiyah, C.W. Kamal, "A, "*Decision Making Method Based on Interval Type-2 Fuzzy Sets: An Approach For Ambulance Location Preference*", Applied Computing and Informatics, Vol 14, No 1, pp. 65-72, January 2018.

[16] Ahmad M. El-Nagar, Mohammad El-Bardini, Nabila M. EL-Rabaie, "*Intelligent Control for Nonlinear Inverted Pendulum Based on Interval Type-2 Fuzzy PD Controller", Alexandria Engineering Journal*, Vol 53, No 1, pp. 23-32, March 2014.

[17] Dipak Kumar Jana, Oscar Castillo, Sutapa Pramanik, Manoranjan Maiti, "*Application of Interval Type-2 Fuzzy Logic to Polypropylene Business Policy in A Petrochemical Plant in India*", Journal of the Saudi Society of Agricultural Sciences, Vol 17, No 1, pp 24-42 January 2018.

[18] Amir Gheibi, S.M.A. Mohammadi, M. maghfoori, "*Maximum Power Point Tracking of Photovoltaic Generation Based on Type 2 Fuzzy Logic Control Method*", Energy Procedia, Vol 12, pp 538-546, 2011.

[19] Ahmad M. El-Nagar, Mohammad El-Bardini, "*Practical Implementation for The Interval Type-2 Fuzzy PID Controller Using A Low-Cost Microcontroller*", Ain Shams Engineering Journal, Vol 5, No 2, pp 475-487, June 2014.

[20] P.N.Q. Nhon, I. Elamvazuthi, H.M. Fayek, S. Parasuraman, M.K.A. Ahamed Khan, "*Intelligent Control of Rehabilitation Robot: Auto Tuning PID Controller with Interval Type 2 Fuzzy for DC Servomotor*", Procedia Computer Science, Vol 42, pp 183-190, 2014.

[21] Mahmoodian Hamid, Lotfizadeh Dehkordi Elaheh, "*Designing a Fuzzy Type-2 Model-based Robust Controller for Ball and Beam System*", Procedia Computer Science, Vol 105, pp 125-130, 2017.

[22] Mohammad El-Bardini, Ahmad M. El-Nagar, "*Direct Adaptive Interval Type-2 Fuzzy Logic Controller for The Multivariable Anaesthesia System*", Ain Shams Engineering Journal, Vol 2, No 3–4, pp 149-160 September–December 2011.

[23] Ji-Hwan Hwang, Young-Chang Kang, Jong-Wook Park, Dong W. Kim, "*Advanced Interval Type-2 Fuzzy Sliding Mode Control for Robot Manipulator*," Computational Intelligence and Neuroscience, Vol. 2017, Article ID 9640849, 11 pages, 2017. DOI:10.1155/2017/9640849.

[24] Mohsen Ebrahimian Baydokhty, Assef Zare, Saeed Balochian, "*Performance of Optimal Hierarchical Type 2 Fuzzy Controller for Load–Frequency System with Production Rate Limitation and Governor Dead Band*", Alexandria Engineering Journal, Vol 55, No 1, pp 379-397, March 2016.

[25] Khatoon Shahida, Ibraheem, Moinuddin, M Farooq," *A Table Lookup Scheme for Fuzzy Logic Based Model Identification Applied to Time Series Prediction", Proceedings of the 6th International Conference on Information Fusion*, pp. 1449-1456, 2003.

[26] Long Thanh Ngo, Dzung Dinh Nguyen, Long The Pham, and Cuong Manh Luong, "*Speedup of Interval Type 2 Fuzzy Logic Systems Based on GPU for Robot Navigation*," Advances in Fuzzy Systems, vol. 2012, Article ID 698062, 11 pages, 2012. DOI:10.1155/2012/698062.

[27] István Kecskés, László Székács, Péter Odry, "*Lookup table-based fuzzy controller implementation in low-power microcontrollers of hexapod robot Szabad(ka)-II*", 3rd International Conference & Workshop Mechatronics in Practice and Education – MECHEDU 2015.