

Using a 3D Interval Type-2 Fuzzy Interpolation System to Improve Modeless Robots Measurement and Calibration Accuracy

Ying Bai¹ and Dali Wang²

1. Department of Computer Science and Engineering, Johnson C. Smith University, Charlotte 28216, USA

2. Department of Physics and Computer Science, Christopher Newport University, Newport News 23606, USA

Abstract: This paper is an extended research for a novel technique used in the pose error compensations of the robot and manipulator calibration process based on an IT2FEI (interval type-2 fuzzy error interpolation) method. Robot calibrations can be classified into model-based and modeless methods. A model-based calibration method normally requires that the practitioners understand the kinematics of the robot therefore may pose a challenger for field engineers. An alternative yet effective means for robot calibration is to use a modeless method; however with such a method there is a conflict between the calibration accuracy of the robot and the number of grid points used in the calibration task. In this paper, an interval type-2 fuzzy interpolation system is applied to improve the compensation accuracy of the robot in its 3D workspace. An on-line type-2 fuzzy inference system is implemented to meet the needs of on-line robot trajectory planning and control. The simulated results given in this paper show that not only robot compensation accuracy can be greatly improved, but also the calibration process can be significantly simplified, and it is more suitable for practical applications.

Key words: Type-2 fuzzy interpolations, interval type-2 fuzzy logic, modeless robotic calibrations, robot kinematics, type-2 fuzzy control.

1. Introduction

A model-based method for robot calibration involves setting up a kinematic model for the robot, measuring positions and orientations of the robot end-effector, identifying its kinematic parameters and compensating its pose errors by modifying its joint angles [1]. Most of published research results for robot calibration belong to this category. The advantage of a model-based calibration method is that a large workspace can be calibrated accurately and all pose errors within the calibrated workspace can be compensated by joint angles. Its disadvantage lies in the fact that the understanding of kinematic modeling and identification processes needs advanced knowledge in robot kinematics, which may pose a

challenge to field engineers.

On the other hand, a modeless method does not go through any kinematic modeling and identification steps. In a pose measuring process, a robot workspace is divided into a sequence of small squares in a 2D case, or cubes in a 3D case with nominal grid points around each cell are assumed known. All position errors on the grid points are measured and recorded by moving the robot through all the grid points. These position errors are stored in memory for future usage. With a modeless method, simple error compensation for a target position can be realized by interpolating errors from its neighboring grid points [2]. Its disadvantage, however, is conflict between calibration accuracy and number of grid points. In spite of this, because of its simplicity and effectiveness, the modeless calibration technique is widely adopted in industrial applications.

Corresponding author: Ying Bai, Ph.D., professor, research fields: intelligent and fuzzy controls, robots calibrations.

There are also several different alternative approaches to the model-based and modeless robot calibration methods discussed at the above. Whitney and Shamma [3, 4] reported a non-parametric accuracy compensation method using polynomial approximating functions; however a nominal inverse kinematic model of the robot to be calibrated is still needed to calculate the joint position vector corresponding to the desired position of the robot end-effector. Another non-parametric compensation approach is to divide the workspace of the robot into a sequence of discrete areas or cubic cells and then use a numerical procedure to determine the inverse kinematic solution for each area or cubic cell. The problem of implementing that approach is that a huge size of memory space is needed to store those inverse kinematic solutions. To solve this problem, James Albus [5, 6] developed a so-called CMAC (cerebellar model articulation controller) to reduce the memory size. By using this non-parametric compensation method, the workspace of the compensated robot is about a quarter of the normal robot workspace and the accuracy is around 0.12 mm [3].

Most traditional modeless calibration methods use some interpolation techniques, such as linear interpolation, bilinear interpolation and cubic spline interpolation, which are widely utilized in industrial settings. Because actual error distribution of a robot may not satisfy linear or cubic models, the estimated errors would not be as accurate as desired. To improve numerical interpolation accuracy, different approaches have been attempted by researchers. Zhuang and Wu [7] reported a histogram method for estimating the optimal membership function distribution in order to improve the accuracy of measured positions. Carlson and Looney reported a new image interpolation method that refined the bilinear and fuzzy interpolation techniques [8]. Song, Smith and Rizk provided an optimized fuzzy logic controller for the generation of optimal trajectories based on OCT (optimal control table) [9]. Bai and Wang developed a

type-1 fuzzy interpolation technique to improve the accuracy of robots calibration in the 2D and 3D spaces [10-12].

In the research reported in this paper, a modeless on-line interval type-2 fuzzy interpolation method is developed and implemented to improve the calibration accuracy of the robot in its 3D workspace. A comparison between the compensation results of using the type-1 and the type-2 fuzzy interpolation techniques is made through simulation studies. Three error models, sinusoidal waveform, normal and uniform distributed errors, are tested. Because the actual neighboring errors on each grid point are random distributed, and the error surfaces on each cell are also random functions at a certain moment, the traditional membership functions are not suitable for our study (each cell is a cubic that is surrounded by 8 grid points). Moreover, the lookup table may not be calculated in advance until the output membership functions that are associated with the actual position errors on grid points are determined during the error measurement process. This means that one cannot use the traditional off-line fuzzy technology to obtain the lookup table for fuzzy error interpolation. The crisp output must be estimated on-line based on the errors at the neighboring grid points, which means that the output membership functions must be dynamic functions of the neighboring errors.

The remainder of the paper is organized as follows: The type-1 fuzzy interpolation technique tailored to robot calibration is discussed in Section 2. Section 3 provides an introduction to interval type-2 fuzzy interpolation technique, and the simulation results to illustrate the feasibility and merits of using the type-2 fuzzy interpolation over the type-1 fuzzy interpolation method is given in Section 4. Concluding remarks are given in Section 5.

2. On-Line 3D Type-1 Fuzzy Interpolation

As has been mentioned, a modeless compensation method involves only two steps: robot pose measuring

and error compensation. Compared with the model-based compensation, the kinematic modeling and identification steps are no longer needed for the modeless compensation method. This can significantly save workload in a robot calibration. It starts with a measurement process, in which position errors of the robot end-effector are measured at all specified grid points on a sequence of pre-determined cubic cell within the robot workspace. A suitable interpolation technique is then applied to fit the target position error based on the position errors of neighboring grid points around the target position. Among several interpolation methods, the linear interpolation technique is a simple and popular one, and is widely adopted by modeless robot calibration.

When using the modeless method to calibrate robots, it is necessary to have some proper measurement device with sufficient accuracy, such as a CMM (coordinate measurement machine), a set of cameras (a stereo camera system) or a laser tracking system, which is used to gather robot position errors as the robot moves its end-effector to all grid points on the pre-determined cubic cell.

Fig. 1 shows a configuration setup for modeless robots calibration process.

The robot workspace is divided into a sequence of

small cubic cells and each small cell is surrounded by 8 neighboring grid points, which is shown in Fig. 2. At each grid point, the laser tracker is used to check the position errors of the end-effector of the robot. In Fig. 2, the desired position of the grid point i is (x_i, y_i, z_i) , and the actual position of the robot end-effector is (x'_i, y'_i, z'_i) . The position errors for this grid point are $e_x = x_i - x'_i$, $e_y = y_i - y'_i$ and $e_z = z_i - z'_i$. The robot will be moved to all grid points on the robot workspace, and all position errors on these grid points will be measured and stored in the memory for future usage. During the compensation process, as the robot is moved to a certain target point on the workspace, an interpolation technique, say the linear interpolation method, is used to estimate the target position error based on the errors of the neighboring grid points around the target point, and finally these errors are added into the target position to obtain the compensated position, with which, the robot is commended to move to that position.

Basically, the linear interpolation method is based on the assumption that errors in a cubic cell are distributed uniformly, and the interpolated error is obtained from three plans that are constructed based on 8 neighboring errors on the grid points around the target cubic cell. However in the real world, this

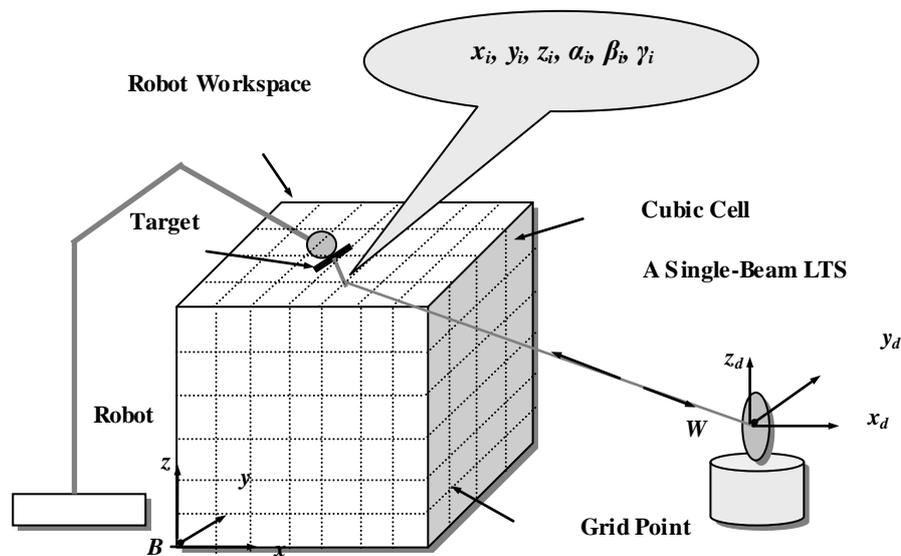


Fig. 1 A setup for modeless robot calibrations.

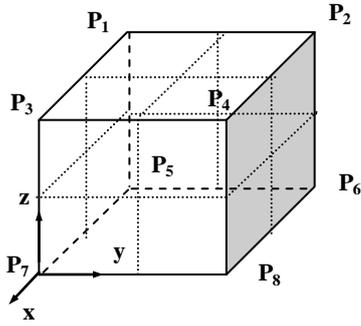


Fig. 2 A cubic cell in robot workspace.

assumption may not be valid. Position errors on each cell may not be distributed uniformly. We can consider the $e_x(x', y', z')$, as the fourth dimension function based on the position x', y' and z' inside each cubic cell. Same consideration is held for $e_y(x', y', z')$ and $e_z(x', y', z')$. The compensation accuracy of using either the linear or bilinear interpolation technique is limited by this assumption.

In order to solve this problem and to improve the compensation accuracy, an on-line fuzzy interpolation method is introduced. The traditional fuzzy inference system uses pre-defined membership functions and control rules to construct lookup tables, and then pick up the associated control output from the lookup table as the fuzzy inference system works in an application. This type of system is called off-line fuzzy inference system because all inputs and outputs have been defined prior to the application process. In this study, this off-line fuzzy system cannot meet the task requirement based on the following reasons: first, the position error of the target point is estimated based on 8 errors of the neighboring grid points, and these 8 neighboring errors are random distributed. The off-line fuzzy output membership functions are defined based on the errors range, in here, the neighboring errors' range. However this range estimation is not as good as the actual errors obtained on the grid points. Second, since one cell needs one lookup table for the off-line fuzzy system, it requires huge memory space to save a large number of lookup tables, which is both space and time consuming and

therefore is not suitable for industrial applications. For example, if the robot workspace is divided into $40 \times 40 \times 40$ small cubic cells, one needs to have 64,000 lookup tables! By using an on-line dynamic fuzzy inference system, one can estimate the target position error on-line by combining the output membership functions that are obtained from the actual errors on the neighboring grid points. With this approach, one does not need off-line lookup tables at all. This means that the system determines the output membership functions only after the fuzzy inference system is applied to an actual process, using real errors on the grid points, not a range.

Fig. 3 shows conceptually the definition of the fuzzy error interpolation inference system. Each small cubic cell that is surrounded by 8 neighboring grid points is defined as a cell, and furthermore this cell is further divided equally into 8 smaller cells, which are shown in Fig. 3a. The position error at each grid point is defined as $P_1, P_2, P_3, P_4, P_5, P_6, P_7$ and P_8 .

For the type-1 fuzzy inference system, we apply the interpolations in three dimensions separately, so the inputs to the fuzzy inference system are e_x, e_y and e_z . The outputs are ee_x, ee_y and ee_z , which are shown in Fig. 3b.

The control rule is shown in Fig. 3c, which is straightforward and based on the human knowledge. The only point to be emphasized is that each P_i should be considered as a combination of three error components on each grid point, P_{xi}, P_{yi} and P_{zi} , which are responding to errors in x, y and z directions.

The distances between the neighboring grid points of each cell on the workspace are 20 mm in x, y and z directions for this study, which is a standard interval for a mid-size calibration workspace. Totally the workspace includes $40 \times 40 \times 40$ cells, which is equivalent to an $800 \times 800 \times 800 \text{ mm}^3$ space. The input membership functions for x, y and z directions and the predefined output membership functions are shown in Fig. 4.

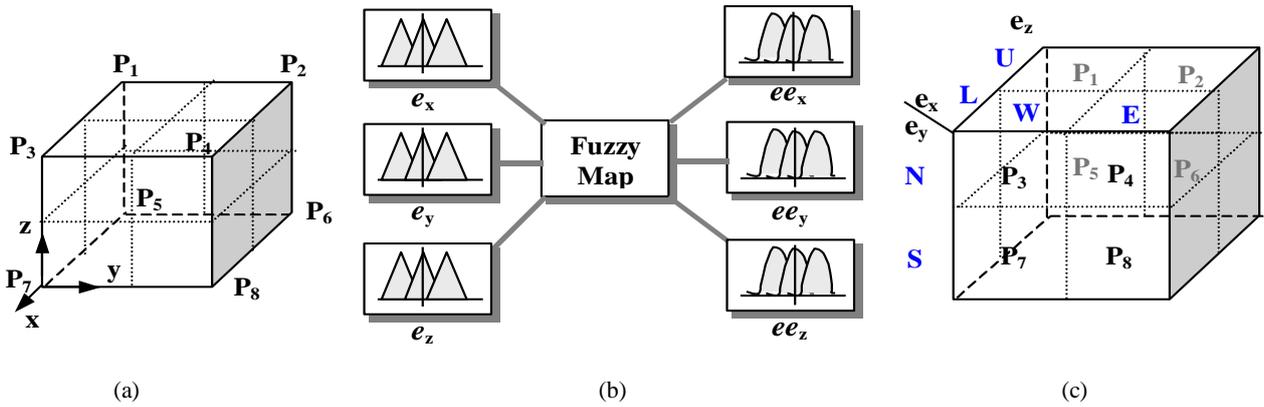


Fig. 3 Definition of the fuzzy interpolation inference system.

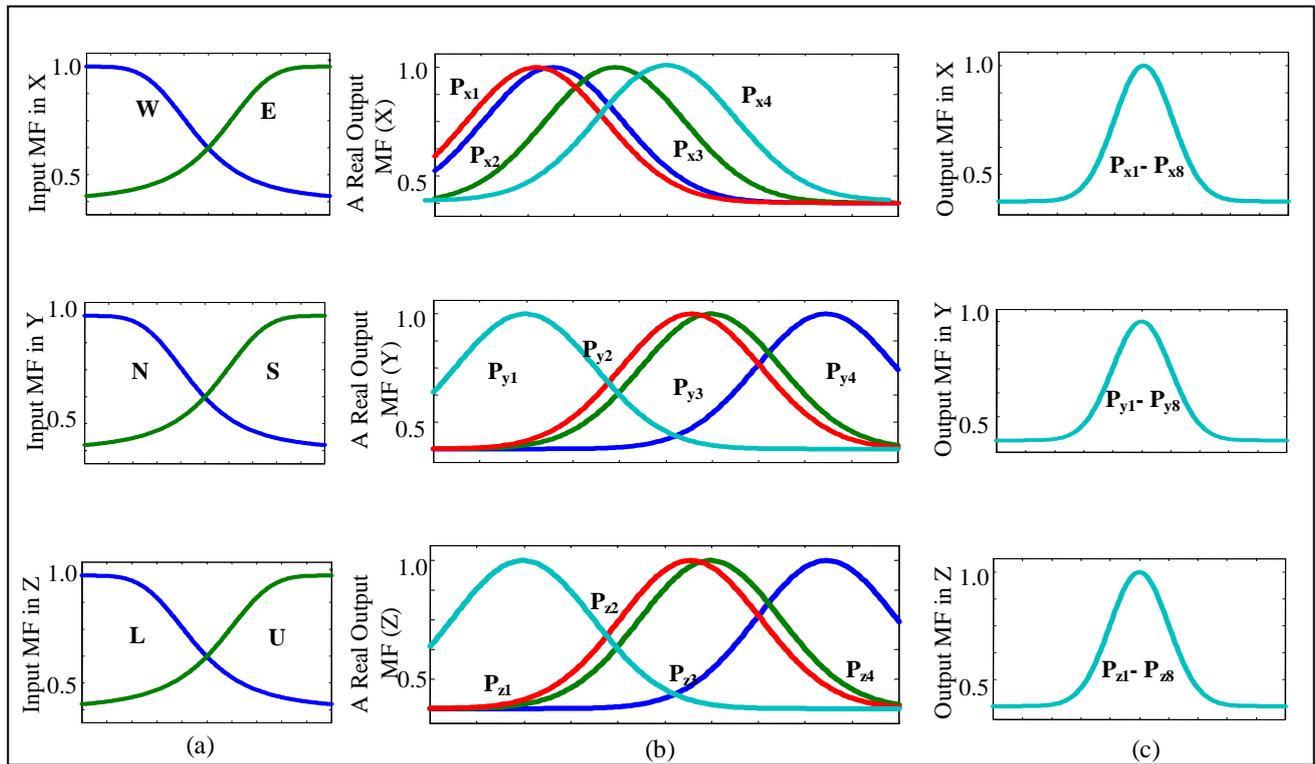


Fig. 4 Input and output membership functions.

The predefined output membership function is used as a default one, and the final output membership function will be obtained by shifting the default one by the actual error values on the grid points. For each cell, 8 output membership functions are implemented and each one is associated with the error at one grid point. In Fig. 4b, only 4 output membership functions are shown here because of the space limitation. In an application, total 8 membership functions should be utilized.

The gaussian-bell waveforms are selected as the shape of the membership functions for three inputs as shown in Fig. 4a, the ranges of inputs are between -10 and 10 mm (20 mm interval on grid points). W and E represent the inputs located at different areas in the x direction, N and S represent the inputs located at different areas in the y direction, and L and U represent the inputs located at different areas in the z direction. Unlike the traditional fuzzy inference system, in which all membership functions

should be determined to produce the lookup table prior to the implementation of the fuzzy system, in this study, the output membership functions will not be defined until the implementation of the fuzzy error mapping to compensate the position errors. So the output membership functions will be determined during the application of the fuzzy inference system on-line or dynamically. This is called dynamic fuzzy system. Fig. 4b shows an example of the output membership functions, which are related to the simulated random errors at neighboring grid points. Each P_{xi} , P_{yi} and P_{zi} responds to the position error at the i th grid point in the x , y and z directions, respectively. During the design stage, all output membership functions should be initialized to a gaussian waveform with a mean of 0 and a range between -0.5 and 0.5 mm, which is a typical error range for this workspace in robotic calibration. These output membership functions will be determined on-line based on the errors of the neighboring grid points around the target point in the workspace as mentioned above.

The control rules shown in Fig. 3c can be interpreted as follows after the output membership functions are determined:

If e_x is W and e_y is N, and e_z is U, then ee_x is P_{x1} , ee_y is P_{y1} and ee_z is P_{z1} . (P_1)

If e_x is W and e_y is N, and e_z is L, then ee_x is P_{x3} , ee_y is P_{y3} and ee_z is P_{z3} . (P_3)

If e_x is W and e_y is S, and e_z is U, then ee_x is P_{x5} , ee_y is P_{y5} and ee_z is P_{z5} . (P_5)

If e_x is W and e_y is S, and e_z is L, then ee_x is P_{x7} , ee_y is P_{y7} and ee_z is P_{z7} . (P_7)

If e_x is E and e_y is N, and e_z is U, then ee_x is P_{x2} , ee_y is P_{y2} and ee_z is P_{z2} . (P_2)

If e_x is E and e_y is N, and e_z is L, then ee_x is P_{x4} , ee_y is P_{y4} and ee_z is P_{z4} . (P_4)

If e_x is E and e_y is S, and e_z is U, then ee_x is P_{x6} , ee_y is P_{y6} and ee_z is P_{z6} . (P_6)

If e_x is E and e_y is S, and e_z is L, then ee_x is P_{x8} , ee_y is P_{y8} and ee_z is P_{z8} . (P_8)

The control rules are straightforward and they are based on the human knowledge. The error on P_1 grid point should be taken more weight if the target position (input) is located inside the NWU area on a cell. Similar conclusion can be derived for errors on all other grid points.

3. Interval Type 2 Fuzzy Interpolation System

3.1 Overview of the Interval Type 2 Fuzzy Interpolation System

Similar to type 1 fuzzy inference system, the type 2 fuzzy inference systems still use the input and output membership functions, combined with the control rules, to derive the outputs [13-21]. However, the fuzzy sets used in the type 2 fuzzy logic or the membership grades involved in each membership function are not crisp values, but another fuzzy sets. This means that the membership degrees for all membership functions used in the type 1 fuzzy system are fixed values and can be determined uniquely before the fuzzy inference system works. But the membership degrees for all membership functions used in the type 2 fuzzy system are fuzzy sets. The difference between the standard type-2 fuzzy system and the so-called interval type-2 fuzzy system is that in the former system, the membership degrees are pure fuzzy sets, but the membership degrees are a set of crisp values with a range of $0 \sim 1$ or an interval for the latter.

Fig. 5 shows the functional block diagram of an Interval Type-2 FLS [22]. It is similar to Typr-1 FLS, but the major difference is that at least one of the fuzzy sets in the rule base is an IT2 fuzzy set. The outputs of the inference engine are IT2 fuzzy sets, and a type-reducer is needed to convert them into a Typr-1 fuzzy set before defuzzification can be started.

Some fundamental operations in the type-2 fuzzy system are union (1), intersection (2) and complement (3) [23]. The union for interval type-2 fuzzy sets \tilde{A} and \tilde{B} is:

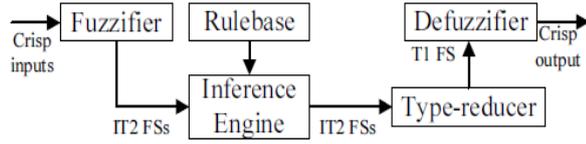


Fig. 5 A functional block diagram of the interval type-2 fuzzy system.

$$\begin{aligned} \tilde{A} \cup \tilde{B} = & \left\{ \int_{x \in X} \mu_{\tilde{A}}(x) \cup \mu_{\tilde{B}}(x) / x \right\} = \\ & \left\{ \int_{x \in X} \left[\int_{\alpha \in [\underline{\mu}_{\tilde{A}}(x), \underline{\mu}_{\tilde{B}}(x), \overline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{B}}(x)]} 1/\alpha \right] / x \right\} \end{aligned} \quad (1)$$

The intersection for interval type-2 fuzzy sets \tilde{A} and \tilde{B} is:

$$\begin{aligned} \tilde{A} \cap \tilde{B} = & \left\{ \int_{x \in X} \mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x) / x \right\} = \\ & \left\{ \int_{x \in X} \left[\int_{\alpha \in [\underline{\mu}_{\tilde{A}}(x) \wedge \underline{\mu}_{\tilde{B}}(x), \overline{\mu}_{\tilde{A}}(x) \wedge \overline{\mu}_{\tilde{B}}(x)]} 1/\alpha \right] / x \right\} \end{aligned} \quad (2)$$

The complement for interval type-2 fuzzy sets \tilde{A} and \tilde{B} is:

$$-\tilde{A} = \left\{ \int_{x \in X} \mu_{-\tilde{A}}(x) / x \right\} = \int_{x \in X} \left[\int_{\alpha \in [1 - \overline{\mu}_{\tilde{A}}(x), 1 - \underline{\mu}_{\tilde{A}}(x)]} 1/\alpha \right] / x \quad (3),$$

In practice the computations in an IT2 FLS can be significantly simplified. Consider the rule base of an IT2 FLS consisting of N rules assuming the following form [22]:

R^n : if x_1 is X_1^n and ... and x_i is X_i^n , then y is

Y^n ; $n = 1, 2, \dots, N$.

where X_i^n ($i = 1 \sim I$) are IT2 fuzzy sets, and $Y^n = [y_1^n, y_2^n]$ is an interval, which can be understood as the centroid [24, 25] of a consequent Interval Type-2 fuzzy set, or the simplest TSK model, for its simplicity. In many applications we use $y_1^n = y_2^n$, i.e., each rule consequent is a crisp number.

Assume the input vector is $x' = (x'_1, x'_2, \dots, x'_I)$. Typical computations in an IT2 FLS involve the following steps:

- (1) Compute the membership of x'_i on each X_i^n ;
- (2) Compute the firing interval of the n^{th} rule, $F^n(x')$;
- (3) Perform type-reduction to combine $F^n(x')$ and the corresponding rule consequents with the center-of-sets type-reducer [24]:

$$Y_{\text{cos}}(x') = \bigcup_{\substack{f^n \in F^n(x') \\ y^n \in Y^n}} \frac{\sum_{n=1}^N f^n y^n}{\sum_{n=1}^N f^n} = [y_l, y_r] \quad (4)$$

- (4) Compute the defuzzified output as:

$$y = \frac{y_l + y_r}{2} \quad (5)$$

3.2 Membership Functions

Similar to type-1 fuzzy interpolation system, the input membership functions for x , y and z directions and the predefined output membership functions for IT2 FLS are shown in Fig. 6. The predefined output membership functions are used as default functions, and the final output membership function will be obtained by shifting the default those by the actual error values on the grid points.

We use W and E to represent the location of inputs in x direction, N and S to the location of inputs in y direction, and U and L to the location of the inputs in z direction. For real outputs, 8 membership functions, $p_{x1} \sim p_{x8}$, should be designed for the x direction, and another 8 membership functions, $p_{y1} \sim p_{y8}$, and $p_{z1} \sim p_{z8}$ are to be built for the y and z directions. These output functions should be located at the center position, which are defined as the default location, as the beginning and changed to the real locations based on the actual position errors on each grid point. In Fig. 6, these functions are all displayed but not in the default locations.

As for the control rules, the identical control rules are used for this IT2 FLS, but the fuzzy sets are used as the degrees to replace those crisp values used in the type-1 FLS.

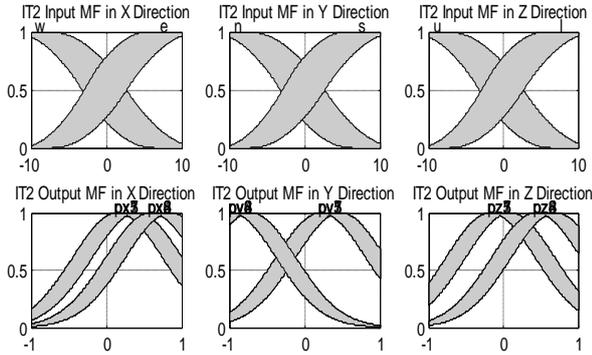


Fig. 6 The input and output membership functions for IT2 FLS.

4. Simulation Result

Extensive simulation for position compensations has been performed in order to illustrate the effectiveness of this IT2 fuzzy error interpolation technique in comparison to the type-1 FLS. Because of the similarity, the angles simulations are not shown in this paper. Due to the random nature of the position errors, three different types of error are simulated in this study. These are:

- Normal distributed random error;
- Uniform distributed random error;
- Sinusoidal waveform error.

Figs. 7-9 show the simulation results of the type-1 and the IT2 fuzzy error interpolation techniques for these three types of error [23].

In these figures, the simulated target (testing) positions on the standard calibration board are spaced from 1 mm to 20 mm within each cell being with a size of 1 mm.

Figs. 10 and 11 show comparisons in mean error, maximum error and STD values between type-1 and IT2 fuzzy error interpolation techniques in the histograms. Only the normal and uniform random errors are shown here since there are not significant differences between the type-1 and type-2 fuzzy interpolations for the sinusoidal error distributions.

It can be seen that both mean errors and maximum errors of the IT2 fuzzy error interpolation technique are smaller than those of type-1 FLS methods. For all three error distribution, the mean errors of the IT2

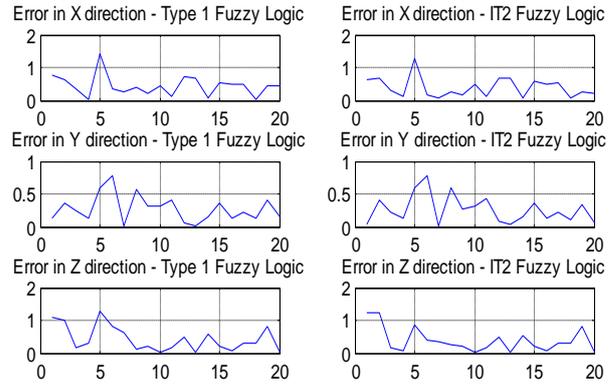


Fig. 7 Interpolation results—normal distributed errors.

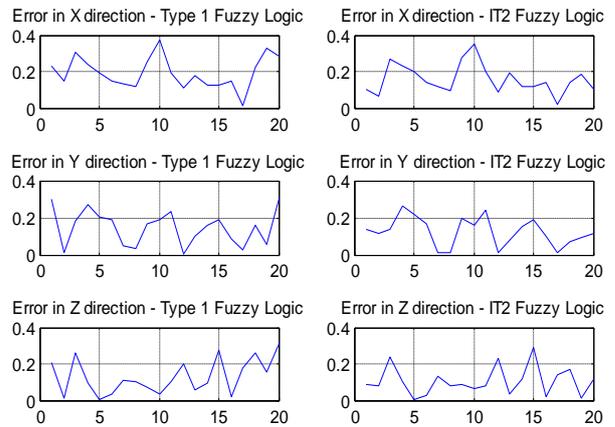


Fig. 8 Interpolation results—uniform distributed errors.

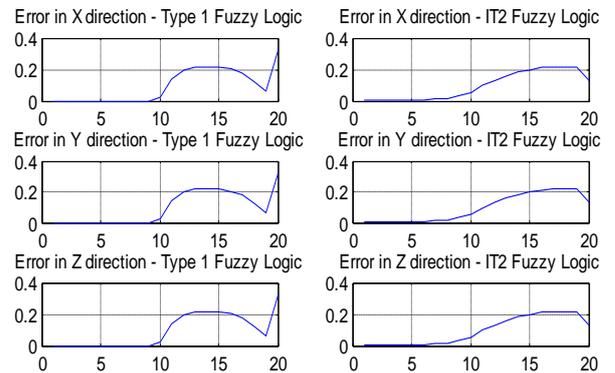


Fig. 9 Interpolation results—sinusoid waveform errors.

fuzzy error interpolation method are approximately 10% to 20% smaller compared with those of type-1 FLS method.

The maximum errors of the IT2 fuzzy error interpolation technique are about 10% to 30% smaller than those of the type-1 FLS method. In one case (normal distribution error in x, y and z direction), the

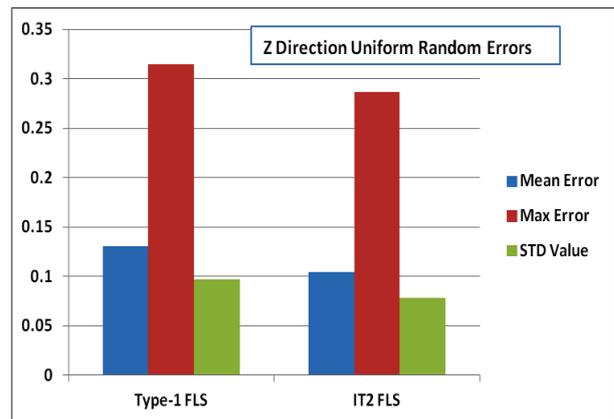
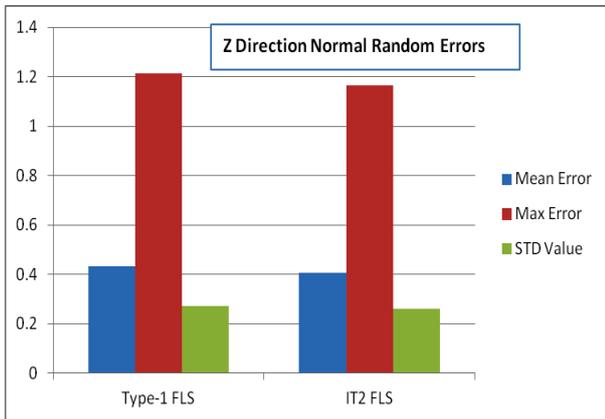
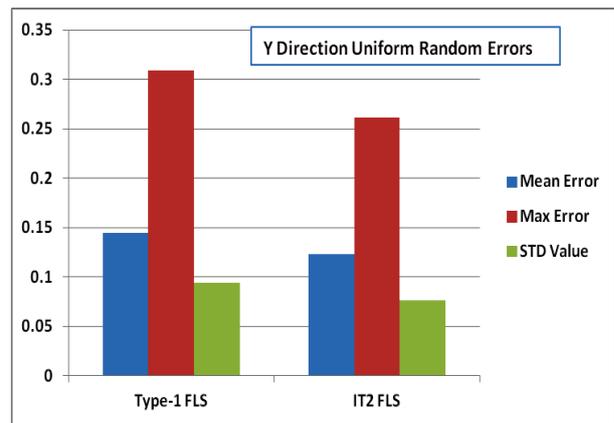
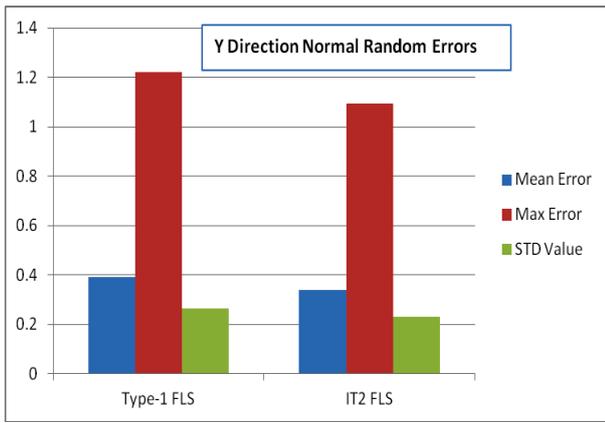
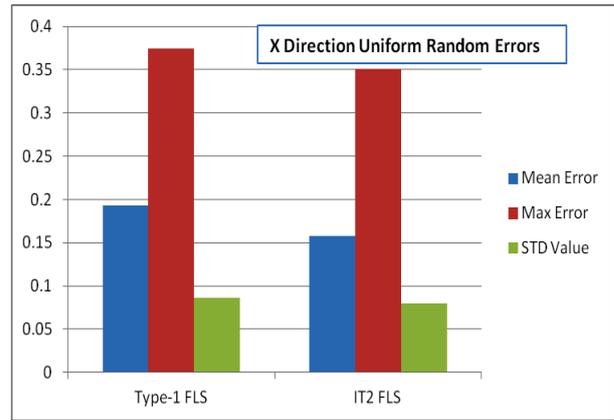
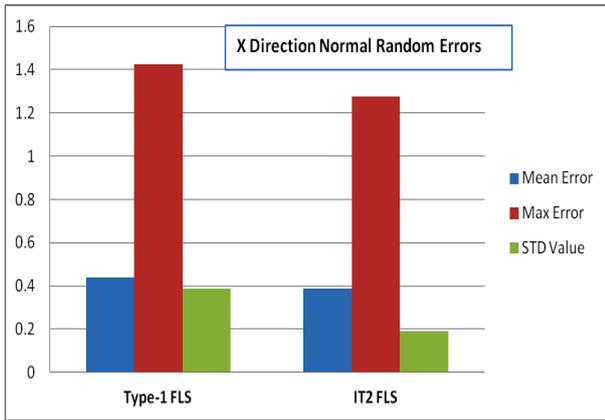


Fig. 10 Normal distributed errors.

Fig. 11 Uniform distributed errors.

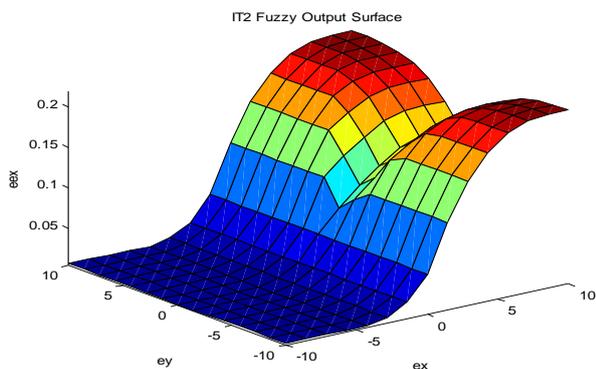


Fig. 12 The IT2 fuzzy interpolation error surface.

maximum errors of the IT2 fuzzy interpolation method are about 4% smaller than those of the type-1 FLS method.

Fig. 12 shows the IT2 fuzzy interpolation error surface.

The simulated results show the effectiveness of the dynamic on-line interval type-2 fuzzy error interpolation technique in reducing the position errors in the modeless robot compensation process.

To implement this interval type-2 fuzzy error interpolation technique as a real time application, an interface between the MATLAB[®] and high level programming languages C/C++ has been developed [26]. Although the most popular real time control programming language is C/C++, the fuzzy error interpolation method is developed and implemented in MATLAB [23]. Using this interface, the measured position errors on the grid points can be passed from C/C++ to MATLAB functions that implement the interval type-2 fuzzy error interpolation functions; and the fuzzy error interpolation results can be sent back to C/C++ for the real time controller to operate on the next target position.

5. Conclusion and Summary

A dynamic on-line interval type-2 fuzzy error interpolation technique is presented in this paper. The compensated position errors in a modeless robot calibration can be greatly reduced by the proposed technique. Simulation results demonstrate the effectiveness of the proposed fuzzy error interpolation

technique. Three typical error models are utilized for comparison and simulation; these include sinusoidal waveform, normally distributed and uniformly distributed errors. This fuzzy error interpolation technique is ideal for the modeless robot position compensation, especially the high accuracy robot calibration process.

Acknowledgment

Special thanks to Dr. Oscar Castillo and his group to provide me the Interval Type-2 Fuzzy Logic Toolbox to enable us to perform our simulations for this paper [27, 28]. We could not develop and share this paper with all of you without his sincere help and support.

References

- [1] Mooring, B. W., Roth, Z. S., and Driels, M. R. 1991. *Fundamentals of Manipulator Calibration*. New York: John Wiley & Sons.
- [2] Zhuang, H., and Roth, Z. S. 1996. *Camera-Aided Robot Calibration*. Florida: CRC Press, Inc.
- [3] Shamma, J. S., and Whitney, D. E. 1987. "A Method for Inverse Robot Calibration." *Journal of Dynamic Systems, Measurement, and Control* 109 (1): 36-43.
- [4] Shamma, J. S. 1985. "A Method for Inverse Robot Calibration." Master's thesis, Massachusetts Institute of Technology.
- [5] Albus, J. S. 1975. "Data Storage in the Cerebella Model Articulation Controller." *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control* 97: 228-33.
- [6] Albus, J. S. 1975. "A New Approach to Manipulator Control: The Cerebella Model Articulation Controller." *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control* 97: 220-7.
- [7] Zhuang, H., and Wu, X. M. 2001. "Membership Function Modification of Fuzzy Logic Controllers with Histogram Equalization." *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 31 (1): 125-31.
- [8] Carlson, B., and Looney, C. 2001. "A New Method of Image Interpolation." In *Proceedings of the ISCA 16th International Conference on Computers and Their Applications*, 25-8.
- [9] Song, F., Smith, S. M., and Rizk, C. G. 1999. "A Fuzzy Logic Controller Design Methodology for 4D System with Optimal Global Performance Using Enhanced Cell State Space Based Best Estimate Directed Search

- Method.” In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 138-43.
- [10] Bai, Y., and Wang, D. L. 2003. “Improve the Position Measurement Accuracy Using a Fuzzy Error Interpolation Technique.” In *Proceedings of the IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications*, 227-32.
- [11] Bai, Y., and Wang, D. L. 2010. “On the Comparison of Trilinear, Cubic Spline and Fuzzy Interpolation Methods in the High Accuracy Measurements.” *IEEE Transactions on Fuzzy Systems* 18 (5): 1016-22.
- [12] Bai, Y. 2007. “On the Comparison of Model-Based and Modeless Robotic Calibration Based on a Fuzzy Interpolation Method.” *International Journal of Advanced Manufacturing Technology* 31 (11-12): 1243-50.
- [13] Kashyap, S. K. 2015. “IR and Color Image Fusion Using Interval Type 2 Fuzzy Logic System.” In *Proceedings of the International Conference on Cognitive Computing and Information Processing*, 1-4.
- [14] Lu, Q., Shi, P., Lam, H. K., and Zhao, Y. 2015. “Interval Type-2 Fuzzy Model Predictive Control of Nonlinear Networked Control Systems.” *IEEE Transactions on Fuzzy Systems* 99: 1-8.
- [15] Wu, D. R., and Mendel, J. M. 2014. “Designing Practical Interval Type-2 Fuzzy Logic Systems Made Simple.” In *Proceedings of the IEEE International Conference on Fuzzy Systems*, 800-7.
- [16] Schrieber, M., and Biglarbegan, M. 2014. “Hardware Implementation of a Novel Inference Engine for Interval Type-2 Fuzzy Control on FPGA.” In *Proceedings of the IEEE International Conference on Fuzzy Systems*, 640-6.
- [17] Tsai, C. C., Kao, J. C., and Chang, C. F. 2014. “Interval Type-2 Fuzzy Gear-Changing Control for Intelligent Bikes.” In *Proceedings of the International Conference on Machine Learning and Cybernetics*, 741-7.
- [18] Nurmaini, S., and Tutuko, B. 2014. “Motion Coordination for Swarm Robots.” In *Proceedings of the International Conference on ICT for Smart Society*, 312-5.
- [19] Kim, C.-J., and Chwa, D. K. 2015. “Obstacle Avoidance Method for Wheeled Mobile Robots Using Interval Type-2 Fuzzy Neural Network.” *IEEE Transactions on Fuzzy Systems* 23 (3): 677-87.
- [20] Allawi, Z. T., and Abdalla, T. Y. 2015. “An Optimal Defuzzification Method for Interval Type-2 Fuzzy Logic Control Scheme.” In *Proceedings of the Science and Information Conference*, 619-27.
- [21] Kumbasar, T., and Hagra, H. 2015. “A Self-tuning Slices-Based General Type-2 Fuzzy PI Controller.” *IEEE Transactions on Fuzzy Systems* 23 (4): 991-1013.
- [22] Wu, D. R. 2012. “A Tutorial on Interval Type-2 Fuzzy Sets and Systems.” University of Southern California.
- [23] ITT/UABC. 2008. *Interval Type-2 Fuzzy Logic Toolbox for Use with MATLAB®*. Tijuana Institute of Technology and Baja California Autonomous University, Tijuana Campus, Mexico.
- [24] Mendel, J. M. 2001. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall PTR.
- [25] Karnik, N. N., and Mendel, J. M. 2001. “Centroid of a Type-2 Fuzzy Set.” *Information Sciences* 132: 195-220.
- [26] Bai, Y. 2003. *Applications Interface Programming Using Multiple Languages*. New Jersey: Prentice Hall PTR.
- [27] Castro, J. R., Castillo, O., Melin, P., Martínez, L. G., Escobar, S., and Camacho, I. 2007. “Building Fuzzy Inference Systems with the Interval Type-2 Fuzzy Logic Toolbox.” *Analysis and Design of Intelligent Systems Using Soft Computing Techniques* 41: 53-62.
- [28] Castro, J. R., Castillo, O., and Melin, P. 2007. “An Interval Type-2 Fuzzy Logic Toolbox for Control Applications.” In *Proceedings of the IEEE International Conference on Fuzzy System*.