

# Compare Fuzzy Interpolation Algorithm with Other Interpolation Methods Used in Industrial Applications

Gerald Agbegha, Nailong Guo and Ying Bai  
Johnson C. Smith University, Charlotte, North Carolina USA  
[gagbegha@jcsu.edu](mailto:gagbegha@jcsu.edu), [nguo@jcsu.edu](mailto:nguo@jcsu.edu), [ybai@jcsu.edu](mailto:ybai@jcsu.edu)

**Abstract—** A novel interpolation algorithm, fuzzy interpolation, is presented and compared with other popular interpolation methods widely implemented in industrial and manufacturing applications. Different interpolation algorithms have been developed, reported and implemented in many industrial applications in recent years. Most of them are based on looking for the optimal interpolation trajectories based on some known values on given points around a workspace. However, it is rare to build an optimal interpolation results based on some random noises, and this is one of the most popular topics in industrial testing and measurement applications. The fuzzy interpolation algorithm (FIA) reported in this paper provides a convenient and simple way to solve this problem and offers more accurate interpolation results based on given position or orientation errors that are randomly distributed in real time. This method can be implemented in many industrial applications, such as manipulators measurements and calibrations, industrial automations and semiconductor manufacturing processes..

## I. INTRODUCTION

A suitable interpolation method is important to fit the target pose errors based on the pose errors of the neighboring grid points around the target. In recent years, many advanced interpolation algorithms have been designed and developed by different researchers [1-5]. Jakobsson et al developed a technique for interpolation with quotients of two radial basis function expansions to approximate functions with poles [6]. Duan et al constructed a bivariate rational interpolation method using both function values and partial derivatives of the function being interpolated as the interpolation data, developed a new rational interpolation with a bi-quadratic denominator to create a space surface using only values of the function being interpolated, designed a bivariate rational Hermite interpolation to create a space surface using both function values and the first-order partial derivatives of the function being interpolated as the interpolation data, and presented a weighted rational cubic spline interpolation using two kinds of rational cubic splines with quadratic denominator [7-11].

Luo et al developed a range restricted  $C^1$  interpolation local scheme to scattered data.  $C^h$ -rational spline function classes over triangles and quadrilaterals were investigated [12]. Hu et al presented an adaptive osculatory rational interpolation for image processing that preserves the contours or edges [13]. Zhao and Tan introduced block-

based inverse differences to extend the point-based Thiele-type interpolation to the block-based Thiele-like blending rational interpolation. Also, a bivariate analogy and numerical examples were given to show the effectiveness of their method [14]. Sarfraz et al developed a smooth curve interpolation scheme for positive, monotonic, and convex data by using piecewise rational cubic functions [15].

Goodman and Meek presented a planar interpolation method using a pair of rational spirals to solve planar and two-point  $G^2$  Hermite interpolation problem [16]. Hussain and Sarfraz used a  $C^1$  piecewise rational cubic function to visualize the data arranged over a rectangular grid [17]. Bejancu Built a new treatment of univariate semi-cardinal interpolation for natural cubic splines, and the solution was obtained as a Lagrange series with suitable localization and polynomial reproduction properties [18]. Zhu and Wang applied the Nother-type theorem of piecewise algebraic curves on cross-cut partitions, and used interpolation along a piecewise algebraic curve [19]. Lyche et al used box spline quazi-interpolants based on local linear functionals of point evaluator and integral type to reproduce the whole spline space [20].

Among those interpolation methods, two of them are very popular and widely implemented in most industrial and manufacturing processes, trilinear and cubic spline interpolation algorithms.

Both linear and cubic spline interpolation methods can achieve satisfactory interpolation results for a common measurement and calibration process [21]. Generally, the linear interpolation method is based on the assumption that the error distribution is approximately linear, and the interpolated errors are obtained from three plans that are constructed based on 8 neighboring errors on the grid points around the target cubic cell [22]. The cubic spline interpolation technique also assumes that the error of the target pose is located on a cubic curve that is constructed by the pose errors of 8 neighboring grid points around the target [22-23]. In essence, both methods approximate a spatial error surface based on the errors of known points, and assume that the error of the target point is located on that surface. Consequently, the target pose error is estimated by utilizing the equations of the error surface. However, since the actual pose errors are randomly distributed with the time and locations in the measured machine workspace, and therefore it is impossible to pinpoint a pose on the error surface at any given moment accurately, the result is that the

traditional interpolation techniques may not provide an accurate estimation of the pose errors.

The fuzzy error interpolation technique utilizes a fuzzy inference system to estimate machine or manipulator pose errors, which is consistent with the random distributed nature of the pose errors. These pose errors can be considered as a fuzzy set at any given moment of time. The fuzzification process takes into account a range of errors rather than only a crisp error value. Therefore, the fuzzy error interpolation technique has the potential to improve the error estimation and compensation results for the target.

Fuzzy interpolation techniques have been rapidly developed and implemented in many academic and industrial fields in recent years [24-28, 38-42]. Different strategies of fuzzy interpolation have been developed and applied in real applications. Triantafilis and Suzana *et al.* reported approaches of using fuzzy interpolation methods to estimate the soil layer and geographical distributions for GIS database [29-30]. Song *et al.* described a fuzzy logic methodology for four-dimensional (4D) systems with optimal global performance using enhanced cell state space [31]. Li *et al.* reported a multi-dimensional fuzzy interpolation neural network to perform the fuzzy interpolations for a multi-dimension system [32]. Seong *et al.* reported to use fuzzy interpolation methods to obtain the trajectory data for a multi-joint animation robot [33]. Ying Bai *et al.* developed a robot calibration algorithm to calibrate parallel machine tools using fuzzy interpolation techniques [34]. A control algorithm combined with Lagrange fuzzy interpolation, which was reported by Cheng Wang and Shanzhen Xu, can be used to improve control effect and enhance control precision effectively compared with traditional fuzzy control algorithm [35]. Hsu *et al.* developed a mind assistant system using fuzzy interpolation technique to support the judgment of emotion state for elderly living alone [36]. A supervisory semiactive nonlinear control system combined with a fuzzy interpolation algorithm is reported by Yeeseok Kim *et al.* to improve the controllability of a MISO controller [37].

A comparison between trilinear, cubic spline and fuzzy interpolation methods used in accurate measurements and compensations for machine or manipulator calibration are discussed in this paper. The simulation results show that the fuzzy interpolation outperform other interpolation methods.

The remainder of the paper is organized into the following four sections. The principles of the two popular traditional interpolation techniques, trilinear and cubic spline, are outlined in Section 2. Section 3 discusses the fuzzy error interpolation method. Results from a simulation study are given in Section 4 to illustrate the effectiveness of the fuzzy error interpolation technique. The conclusion is provided in section 5.

## II. TRILINEAR AND CUBIC SPLINE METHODS

The trilinear and cubic spline interpolation methods are designed to construct a surface based on the known errors of neighboring points. The target pose error is then derived by

using an error surface equation. The operation principles of the trilinear and spline interpolation methods are discussed in this section.

### A. Trilinear Interpolation

Trilinear interpolation is a computational process of linearly interpolating points within a 3D box given values at the vertices of the box, and it is the most common application in interpolating within cells of a volumetric dataset [23-24]. The whole process can be simplified to perform three consecutive linear interpolations along three coordinate axes,  $x$ ,  $y$  and  $z$ , respectively.

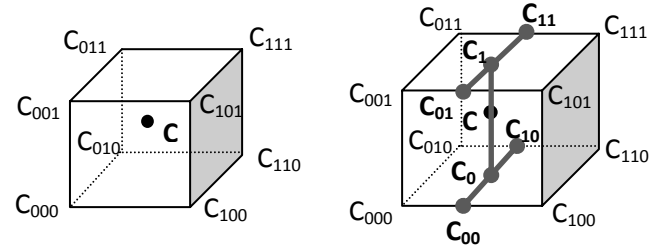


Fig. 1. Eight corner points

Fig.2 3D trilinear interpolation

Refer to Figs.1 and 2, and assume that it is a unit cube with the lower-left-base vertex at the origin. The coordinate values at each vertex will be denoted  $C_{000}$ ,  $C_{100}$ ,  $C_{010}$ , .....  $C_{111}$ . Let  $x_d$ ,  $y_d$ , and  $z_d$  be the differences between the target of  $x$ ,  $y$ ,  $z$  and the smaller coordinate related at the cubic lattice  $[x]$ ,  $[y]$  and  $[z]$ , the error values at 8 corners of the cubic lattice are  $V_{000}$ ,  $V_{100}$ ,  $V_{010}$ , ....etc.... $V_{111}$ , which is:

$$\begin{aligned} x_d &= x - [x] \\ y_d &= y - [y] \\ z_d &= z - [z] \end{aligned} \quad (1)$$

We can first perform the linear interpolation along the  $z$  axis (pushing the front face of the cube to the back), which is:

$$\begin{aligned} V_{00} &= V_{000}(1 - z_d) + V_{100}z_d \\ V_{10} &= V_{010}(1 - z_d) + V_{110}z_d \\ V_{01} &= V_{001}(1 - z_d) + V_{101}z_d \\ V_{11} &= V_{011}(1 - z_d) + V_{111}z_d \end{aligned} \quad (2)$$

Then we interpolate these values along  $y$  axis, as we were pushing the top edge to the bottom, giving:

$$\begin{aligned} V_0 &= V_{00}(1 - y_d) + V_{10}y_d \\ V_1 &= V_{01}(1 - y_d) + V_{11}y_d \end{aligned} \quad (3)$$

Finally we interpolate these values along  $x$  axis (walking through a line) and this gives us a predicted error value for the target point.

$$t_p = V_0(1 - x_d) + V_1x_d \quad (4)$$

The above operations can be illustrated by the following sequence: first we perform linear interpolation between  $C_{000}$  and  $C_{100}$  to find  $V_{00}$ ,  $C_{001}$  and  $C_{101}$  to find  $V_{01}$ ,  $C_{011}$  and  $C_{111}$  to find  $V_{11}$ ,  $C_{010}$  and  $C_{110}$  to find  $V_{10}$ . Then we do interpolation between  $C_{00}$  and  $C_{10}$  to find  $V_0$ ,  $C_{01}$  and  $C_{11}$  to find  $V_1$ . Finally, we calculate the error value  $C$  via linear interpolation of  $C_0$  and  $C_1$ . In practice, a trilinear interpolation is identical to three successive linear interpolations, or two bilinear interpolations combined with a linear interpolation.

Combined Eqs (1) through (4), we can obtain the following equation to interpolate the error value  $V_{xyz}$  at the target position  $[x, y, z]$  assumed that the cubic lattice is a unit one:

$$V_{xyz} = V_{000}(1-x)(1-y)(1-z) + V_{100}x(1-y)(1-z) + V_{010}(1-x)y(1-z) + V_{001}(1-x)(1-y)z + V_{101}x(1-y)z + V_{011}(1-x)yz + V_{110}xy(1-z) + V_{111}xyz$$

In general the box will not be of unit size nor will it be aligned at the origin. Simple translation and scaling (possibly of each axis independently) can be used to transform into then out of this simplified situation.

As illustrated in Figs. 1 and 2, the trilinear interpolation technique is based on two assumptions. First, the pose error of the target  $e_p$  must be located on three error surfaces, which is built based on errors of 8 neighboring grid points around a cubic cell. Secondly, the error surface has to be constructed prior to the application of the trilinear interpolation technique. However, these assumptions have their drawbacks. Pose errors on each cell are randomly distributed and the error curving surfaces,  $e_x$ ,  $e_y$  and  $e_z$ , are also randomly distributed at any given moment. One can consider the  $e_x(x, y, z, \alpha, \beta, \gamma)$  as a fourth dimensional function value based on the pose  $[x, y, z, \alpha, \beta, \gamma]$  inside each cell. The same consideration is applied to  $e_y(x, y, z, \alpha, \beta, \gamma)$  and  $e_z(x, y, z, \alpha, \beta, \gamma)$ . Therefore, the compensation accuracy of bilinear interpolation is limited by these assumptions.

### B. Cubic Spline Interpolation

The cubic spline method is to estimate a cubic surface  $s(x, y, z)$  based on the position errors of the neighboring grid points around the target. This method assumes that both the 1<sup>st</sup> and the 2<sup>nd</sup> order derivatives ( $s'(x, y, z)$  and  $s''(x, y, z)$ ) of the interpolated points are existed, and the function  $s''(x, y, z)$  is a trilinear surface on each cubic cell [29]. To simplify our discussion, consider the one-dimensional situation. Since the function  $s''(x)$  is a linear function at the interval of each cell in the  $x$  direction, the error function  $s(x)$  should be a cubic curve. If  $a$  and  $b$  are two neighboring points in the  $x$  direction, we define two values  $M_a$  and  $M_b$  as,

$$M_a = s''(a), \text{ and } M_b = s''(b) \quad (5)$$

A linear equation can be derived as follows:

$$s''(x) = \frac{(b-x)M_a + (x-a)M_b}{b-a} \quad (6)$$

After quite a bit of manipulation, this result is in the cubic polynomial [11]:

$$s(x) = \frac{(b-x)^3 M_a + (x-a)^3 M_b}{6(b-a)} + \frac{(b-x)s(a) + (x-a)s(b)}{b-a} - \frac{(b-a)[(b-x)M_a + (x-a)M_b]}{6} \quad (7)$$

By using the tridiagonal matrix,  $M_a$  and  $M_b$  can be derived. By substituting these 2<sup>nd</sup> order derivatives back to (7), any point's error in  $x$  direction can be interpolated.

An example of a 2D error surfaces in  $x$  direction,  $e_x(x, y)$ , and in  $y$  direction,  $e_y(x, y)$ , for a cell is shown in Fig.3.

The error surfaces are estimated based on neighboring grid position errors around the target position using the cubic spline technique.

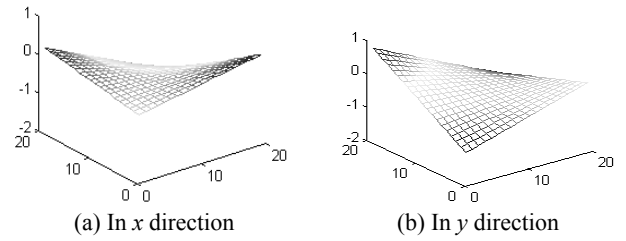


Fig. 3. Error surfaces in  $x$  and  $y$  directions on a cell

Compared with trilinear interpolation method, the cubic spline method uses a more arbitrary shaped surface to approximate the error. Therefore, it provides more accurate position compensation results for known positions around a target. For unknown interpolated data, such as random noises, the interpolation results may not as good as desired since the interpolated data are randomly distributed noises.

### III. FUZZY ERROR INTERPOLATION METHOD

From the structure of the trilinear interpolation technique, it can be observed that the method assumes that the position error on the target point  $\mathbf{P}(x, y, z)$  must be located on the intersection of the three plans that are built based on errors of the 8 neighbouring grid points  $\mathbf{P}_1$ -  $\mathbf{P}_8$ . However in the real world, this assumption may not hold. The compensation accuracy of using this interpolation technique is limited by this assumption. For the cubic spline interpolation method, the assumption is that all interpolated data should be definite or with little degree of uncertainty. However, this assumption cannot be satisfied when the interpolated data are random noises, and therefore the interpolation results may not be as good as desired.

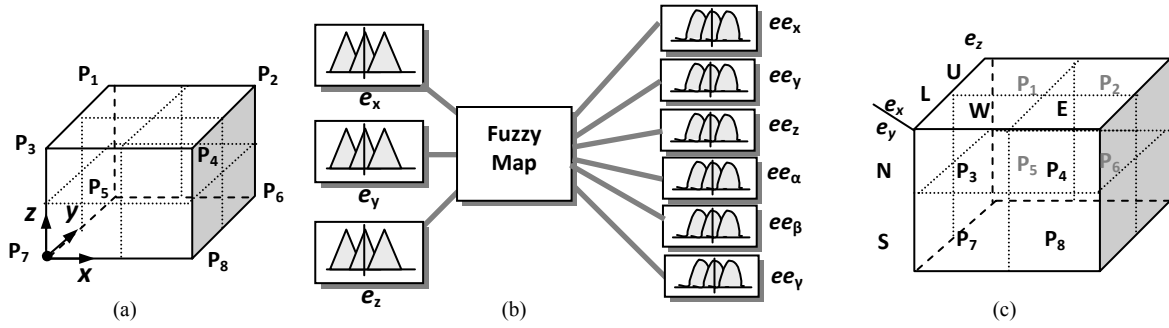


Fig. 4. Definition of the Fuzzy Interpolation Inference System

In order to solve this problem and to improve the measurement and compensation accuracy, a dynamic 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 picks up the associated control output from the lookup table as the fuzzy inference system works in an application. This kind of system is often called an off-line fuzzy inference system because all inputs and outputs have been defined prior to the application process. This off-line fuzzy system may not meet accuracy requirements in certain applications based on the following reasons: First, the pose error of the target is estimated based on errors of 8 neighbouring grid points, and these neighbouring errors are randomly distributed. The off-line fuzzy output membership functions are defined based on the errors range, say the neighbouring errors' range. However this range estimation is not as good as the one deduced from the actual errors obtained on 8 grid points. Second, since each cell needs one lookup table for the off-line fuzzy system, it needs a large memory space to save a great number of lookup tables, which is both space and time consuming, and therefore not suitable for real time processing. For example, in our study, the robot workspace is divided into  $40 \times 40 \times 40$  small cubic

cells and each cell is  $20 \times 20 \times 20 \text{ mm}^3$ . Assume that one lookup table is for one cubic cell and this needs about 64000 lookup tables! By using an on-line dynamic fuzzy inference system, one can estimate the target pose error by combining the output membership functions, which are obtained from real errors on the neighbouring grid points, with the control rules in real time after this on-line fuzzy system is implemented. Therefore, we do not need any off-line lookup tables at all. This means that one cannot determine the output membership functions until the fuzzy inference system is applied to a real process, and this is based on the real errors on the grid points, not a range.

The definition of this dynamic on-line fuzzy inference algorithm is shown in Fig. 4. Each small cube, which is surrounded by 8 neighboring grid points, is defined as a cubic cell. Furthermore this cubic cell is divided into 8 equal smaller cubic cells, which are also shown in Fig. 4a.

The pose 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 fuzzy inference system, the interpolation method is divided into three dimensions separately, so the inputs to the fuzzy inference system are  $e_x, e_y$  and  $e_z$ . The outputs are  $ee_x, ee_y, ee_z, ee_\alpha, ee_\beta$  and  $ee_\gamma$ , which are shown in Fig. 4b.

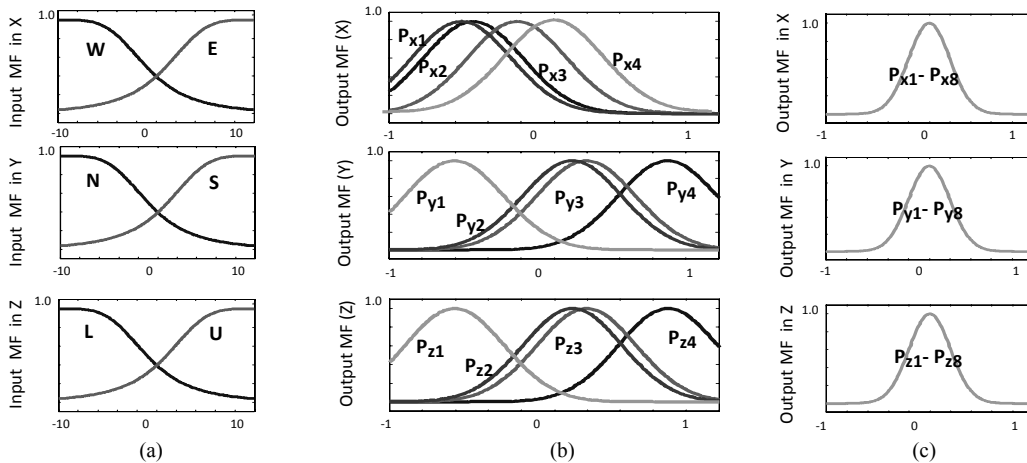


Fig. 5. Input and output membership functions

The control rule is shown in Fig. 4c, which is straightforward and based on the human being knowledge. It is worth to note that each  $P_i$  should be considered as a combination of three position and three orientation error components on each grid point.

The distance between the neighboring grid points of each cell on the workspace is 20 mm in  $x$ ,  $y$  and  $z$  directions for our current study, which is a standard interval for a small-size calibration workspace. Totally the workspace includes 20 by 20 by 20 cells, which is equivalent to a 400 by 400 by 400 mm<sup>3</sup> space. This is a typical workspace of most popular manipulators implemented in semiconductor manufacturing operations [32,36,37]. The input membership functions for  $x$ ,  $y$  and  $z$  directions and the predefined output membership functions are shown in Fig. 5.

The predefined output membership functions are used as default ones, and the actual output membership function will be obtained by shifting the default one based on 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 Figure 5b, only 4 position output membership functions are shown here because of the space limitation. In a real application, total 8 orientation and 8 position 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. 5a, the ranges of inputs are between -10 and 10 mm (20 mm interval on grid points). The reason for this selection is that the Gaussian-bell waveform has a smooth curve and therefore can make measurements more accurate [35-36].  $W$  and  $E$  represent the inputs located at different areas in the  $x$  direction,  $N$  and  $S$  represent those in the  $y$  direction, and  $L$  and  $U$  represent those 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 interpolation to compensate the pose errors. So the output membership functions will be determined during the application of the fuzzy inference system on-line or dynamically. Fig. 5b 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  $P_{zi}$  corresponds to the pose error at the  $i$ th grid point, respectively. During the design stage, all output membership functions should be initialized to a gaussian waveform with a mean of 0 and a range that is closed to the actual possible output range which can be estimated based on the different manipulators for the different applications. 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 during the compensation process.

The control rules shown in Fig. 4c can be interpreted as follows after the output membership functions are determined:

- If  $e_x$  is  $W$ ,  $e_y$  is  $N$  and  $e_z$  is  $U$ , then  $ee_x$  is  $P_{x1}$ ,  $ee_y$  is  $P_{y1}$  and  $ee_z$  is  $P_{z1}$ , and  $ee_\alpha$  is  $\alpha_1$ ,  $ee_\beta$  is  $\beta_1$  and  $ee_\gamma$  is  $\gamma_1$ . ( $P_1$ )
- If  $e_x$  is  $W$ ,  $e_y$  is  $N$  and  $e_z$  is  $L$ , then  $ee_x$  is  $P_{x3}$ ,  $ee_y$  is  $P_{y3}$  and  $ee_z$  is  $P_{z3}$ , and  $ee_\alpha$  is  $\alpha_3$ ,  $ee_\beta$  is  $\beta_3$  and  $ee_\gamma$  is  $\gamma_3$ . ( $P_3$ )
- If  $e_x$  is  $W$ ,  $e_y$  is  $S$  and  $e_z$  is  $U$ , then  $ee_x$  is  $P_{x5}$ ,  $ee_y$  is  $P_{y5}$  and  $ee_z$  is  $P_{z5}$ , and  $ee_\alpha$  is  $\alpha_5$ ,  $ee_\beta$  is  $\beta_5$  and  $ee_\gamma$  is  $\gamma_5$ . ( $P_5$ )
- If  $e_x$  is  $W$ ,  $e_y$  is  $S$  and  $e_z$  is  $L$ , then  $ee_x$  is  $P_{x7}$ ,  $ee_y$  is  $P_{y7}$  and  $ee_z$  is  $P_{z7}$ , and  $ee_\alpha$  is  $\alpha_7$ ,  $ee_\beta$  is  $\beta_7$  and  $ee_\gamma$  is  $\gamma_7$ . ( $P_7$ )
- If  $e_x$  is  $E$ ,  $e_y$  is  $N$  and  $e_z$  is  $U$ , then  $ee_x$  is  $P_{x2}$ ,  $ee_y$  is  $P_{y2}$  and  $ee_z$  is  $P_{z2}$ , and  $ee_\alpha$  is  $\alpha_2$ ,  $ee_\beta$  is  $\beta_2$  and  $ee_\gamma$  is  $\gamma_2$ . ( $P_2$ )
- If  $e_x$  is  $E$ ,  $e_y$  is  $N$  and  $e_z$  is  $L$ , then  $ee_x$  is  $P_{x4}$ ,  $ee_y$  is  $P_{y4}$  and  $ee_z$  is  $P_{z4}$ , and  $ee_\alpha$  is  $\alpha_4$ ,  $ee_\beta$  is  $\beta_4$  and  $ee_\gamma$  is  $\gamma_4$ . ( $P_4$ )
- If  $e_x$  is  $E$ ,  $e_y$  is  $S$  and  $e_z$  is  $U$ , then  $ee_x$  is  $P_{x6}$ ,  $ee_y$  is  $P_{y6}$  and  $ee_z$  is  $P_{z6}$ , and  $ee_\alpha$  is  $\alpha_6$ ,  $ee_\beta$  is  $\beta_6$  and  $ee_\gamma$  is  $\gamma_6$ . ( $P_6$ )
- If  $e_x$  is  $E$ ,  $e_y$  is  $S$  and  $e_z$  is  $L$ , then  $ee_x$  is  $P_{x8}$ ,  $ee_y$  is  $P_{y8}$  and  $ee_z$  is  $P_{z8}$ , and  $ee_\alpha$  is  $\alpha_8$ ,  $ee_\beta$  is  $\beta_8$  and  $ee_\gamma$  is  $\gamma_8$ . ( $P_8$ ) (8)

The control rules are straightforward and they are based on the human knowledge. The error on  $P_1$  grid point should carry larger weight if the target position (input) is located inside the NWU area on a cell. Similar consideration should be given for errors on all other grid points.

The input error variables can be expressed as a label set  $L$ , with  $E$  being a linguistic input variable:

$$L(E) = \{NWU, NWL, NEU, NEL, SWU, SWL, SEU, SEL\} \quad (9)$$

Assume that  $u_i$  is the membership function,  $U_i$  the universe of discourse and  $m$  the number of contributions, the traditional output of the fuzzy inference system can be represented as:

$$u = \frac{\sum_{i=1}^m (u_i \times U_i)}{\sum_{i=1}^m u_i} \quad (10)$$

where  $u$  is the current crisp output of the fuzzy inference system and (10) is obtained by using the Center-Of-Gravity method (COG). In this study, both  $u_i$  and  $U_i$  in the output membership functions are randomly distributed variables and the actual values of these variables depend upon the position errors of 8 neighboring grid points around the target position. These relationships can be expressed as:

$$u_{xi} = F_{xi}(P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8}) \quad (11)$$

$$U_{xi} = Q_{xi}(P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8}) \quad (12)$$

where  $F_{xi}$  is the membership function of the input pose in the  $x$  direction and it is a predetermined membership function as shown in Fig. 5a.  $Q_{xi}$  is the real error output membership function, which is a randomly distributed function, and it gives the error output contributions in the  $x$  direction. This membership function is determined by the real pose errors at the 8-neighboring grid points in the  $x$  direction,  $P_{x1} \sim P_{x8}$ . This membership function determines the degree to which the current pose input belongs to each

different real error output based on the 8 control rules defined in (8) in the  $x$  direction, and it is equivalent to the universe of discourse or a weighing factor. Substituting (11) and (12) into (10), one obtains:

$$u_i = \frac{\sum_{i=1}^m F_{xi}(P_{x1}, P_{x2}, \dots, P_{x8}) \times Q_{xi}(P_{x1}, P_{x2}, \dots, P_{x8})}{\sum_{i=1}^m F_{xi}(P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8})} \quad (13)$$

Here  $u_x$  represents the final error output of the fuzzy interpolation method in the  $x$  direction. In (13),  $Q_{xi}$  will not be determined until the fuzzy error interpolation technique is applied in an actual compensation process, which means that this fuzzy inference system is an on-line process. The final crisp output of the fuzzy error interpolation system is determined by the neighboring pose errors of 8 grid points. Similar calculations can be implemented for the error outputs in the  $y$  and  $z$  directions as well as three orientations.

The advantage of using the on-line fuzzy inference system is that the control output has the real time control ability, but the drawback is that this type of control has a relative longer response time because of the calculation performed in the fuzzy inference system. This shortcoming becomes of little importance as the availability of high speed CPUs for the controllers.

#### IV. SIMULATION RESULTS

Extensive simulation studies have been performed with a PUMA 560 robot in order to illustrate the effectiveness of the proposed fuzzy error interpolation technique in comparison to the trilinear and cubic spline interpolation methods. The simulated position error is a uniformly distributed random noise  $U[-0.05, 0.05]$  mm, and the simulated orientation error is also a uniformly distributed random noise  $U[-0.001, 0.001]$  radian degrees. Fig. 8a shows a comparison of interpolated position errors using three interpolation techniques, trilinear, cubic spline and fuzzy. Fig. 8b shows a histogram comparison among three interpolation methods. Fig. 6a and 6b show the comparisons of three interpolation techniques for orientation compensation results.

It can be found that the fuzzy interpolation method has more accurate compensation result for both position and orientation errors compared with both trilinear and cubic spline methods. The max position error of the fuzzy interpolation method is about 0.026 mm, which is about 41% smaller compared with the error obtained from the trilinear method (0.044 mm) and 25% smaller with respective to the error interpolated from the cubic spline method (0.033 mm). For the mean position errors, the fuzzy interpolation method also outperforms the other two methods. Similar comparison results can be obtained from the orientation errors shown in Fig. 7.

The numbers in the horizontal axes in Figs 6 and 7 are the number of errors in 20 cubic cells.

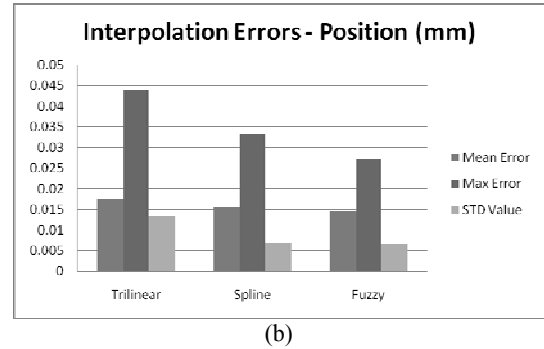
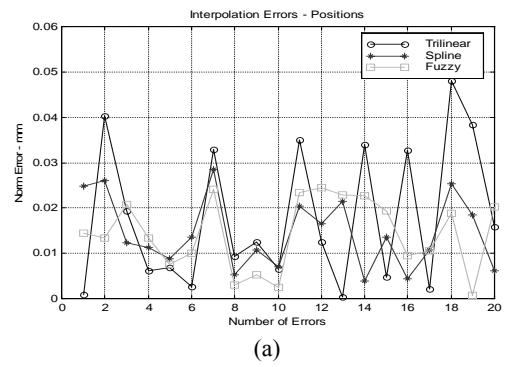


Fig. 6 Simulated interpolation results – position errors

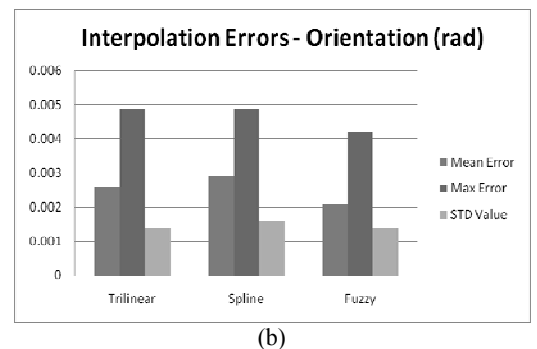
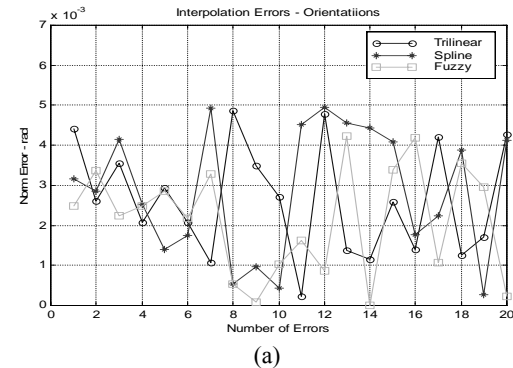


Fig. 7 Simulated interpolation results – orientation errors

#### V. CONCLUSIONS

A comparison of fuzzy error interpolation technique with trilinear and cubic spline interpolation methods used for high accuracy measurement and calibration of robots is discussed and analyzed in this paper. The simulation results show that the measurement and calibration results can be

greatly improved when a fuzzy interpolation method is adopted. By using this fuzzy error interpolation algorithm, both position and orientation errors, especially for the random-distributed errors, can be significantly reduced and suppressed, and therefore the measurement and calibration accuracy can be greatly improved. This algorithm can be conveniently implemented in the real manufacturing process to reduce the production cost and operation times. The key technology used in this algorithm is the dynamic and online process in which the output membership functions are determined online based on the real position and orientation errors of the grid points around the target.

## REFERENCES

- [1] Marjeta Krajnc FMF and IMFM, Ljubljana, Slovenia, "Interpolation Scheme for Planar Cubic G2 Spline Curves", *Acta Applicandae Mathematicae: an international survey journal on applying mathematics and mathematical applications Archive*, Vol.113 Issue 2, Feb. 2011, pp. 129-143.
- [2] Annie Cuyt and Wen-shin Lee, "Sparse interpolation of multivariate rational functions", *Theoretical Computer Science*, Vol. 412 Issue 16, Apr., 2011, pp. 1445-1456.
- [3] Renzhong Feng and Xun Zhou, "A kind of multiquadric quasi-interpolation operator satisfying any degree polynomial reproduction property to scattered data", *Journal of Computational and Applied Mathematics*, Vol. 235, Issue 5, Jan. 2011, pp. 1502-1514.
- [4] Giampietro Allasia and Cesare Bracco, "Two interpolation operators on irregularly distributed data in inner product spaces", *Journal of Computational and Applied Mathematics table of contents archive*, Vol. 235, Issue 7, Feb. 2011, pp. 1763-1774.
- [5] Nardo Giménez, Joos Heintz, Guillermo Matera and Pablo Solernó, "Lower complexity bounds for interpolation algorithms", *Journal of Complexity*, Vol. 27 Issue 2, Apr., 2011, pp. 151-187.
- [6] Stefan Jakobsson, Björn Andersson and Fredrik Edelvik, "Rational radial basis function interpolation with applications to antenna design", *Journal of Computational and Applied Mathematics*, 233, 2009, pp. 889 – 904.
- [7] Q. Duan, Y. Zhang and E.H. Twizell, "A bivariate rational interpolation and the properties", *Applied Mathematics and Computation*, 1 August 2006 179 (1): 190-199.
- [8] Q. Duan, H. Zhang, A. Liu and H. Li, "A bivariate rational interpolation with a bi-quadratic denominator", *Journal of Computational and Applied Mathematics*, 15 October 2006 195 (1-2): 24-33.
- [9] Q. Duan, L. Wang and E.H. Twizell, "A new C2 rational interpolation based on function values and constrained control of the interpolant curves", *Applied Mathematics and Computation*, 4 February 2005 161 (1): 311-322.
- [10] Q. Duan, S. Li, F. Bao and E.H. Twizell, "Hermite interpolation by piecewise rational surface", *Applied Mathematics and Computation*, 15 April 2008 198 (1): 59-72.
- [11] Q. Duan, L. Wang and E.H. Twizell, "A new weighted rational cubic interpolation and its approximation", *Applied Mathematics and Computation*, 15 September 2005 168 (2): 990-1003.
- [12] Z. Luo and X. Peng, "A C1-rational spline in range restricted interpolation of scattered data", *Journal of Computational and Applied Mathematics*, 1 October 2006 194 (2): 255-266.
- [13] M. Hu and J. Tan, "Adaptive osculatory rational interpolation for image processing", *Journal of Computational and Applied Mathematics*, 15 October 2006 195 (1-2): 46-53.
- [14] Q.-J. Zhao and J. Tan, "Block-based Thiele-like blending rational interpolation", *Journal of Computational and Applied Mathematics*, 15 October 2006 195 (1-2): 312-325.
- [15] M. Sarfraz and M.Z. Hussain, "Data visualization using rational spline interpolation", *Journal of Computational and Applied Mathematics*, 1 May 2006 189 (1-2): 513-525.
- [16] T.N.T. Goodman and D.S. Meek, "Planar interpolation with a pair of rational spirals", *Journal of Computational and Applied Mathematics*, 1 April 2007 201 (1): 112-127.
- [17] M.Z. Hussain and M. Sarfraz, "Positivity-preserving interpolation of positive data by rational cubics", *Journal of Computational and Applied Mathematics*, 1 September 2008 218 (2): 446-458.
- [18] A. Bejancu, "Semi-cardinal **interpolation** and difference equations: From cubic B-splines to a three-direction box-spline construction", *Journal of Computational and Applied Mathematics*, 1 December 2006 197 (1): 62-77.
- [19] C.-G. Zhu, R.-H. Wang, "Lagrange interpolation by bivariate splines on cross-cut partitions", *Journal of Computational and Applied Mathematics*, 15 October 2006 195 (1-2): 326-340.
- [20] K. Maleknejad and H. Derili, "Numerical solution of Hammerstein integral equations by using combination of spline-collocation method and Lagrange interpolation", *Applied Mathematics and Computation*, 15 July 2007 190 (2): 1557-1562.
- [21] M. A. Penna, "Camera Calibration: A Quick and Easy Way to Determine the Scale Factor", *IEEE Trans. on Pattern Anal. and Machine Intell.* 13, 1991, pp. 1240-1245.
- [22] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design – A Practical Guide*, 3<sup>rd</sup> edition, Academic Press, Inc. 1993.
- [23] I. J. Schoenberg, "Cardinal Spline Interpolation", *Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics*, Philadelphia, PA 1973.
- [24] Zhiheng Huang; Qiang Shen, "Fuzzy Interpolation and Extrapolation: A Practical Approach", *IEEE Trans on Fuzzy Systems*, Vol. 16, Issue 1, Feb. 2008, Page(s):13 – 28.
- [25] Ying Bai; Hanqi Zhuang, "On the comparison of model-based and modelless robotic calibration based on the fuzzy interpolation technique", *2004 IEEE Conference on Robotics, Automation and Mechatronics*, Vol. 2, 1-3 Dec. 2004, Page(s):892 – 897.
- [26] Ying Bai; Dali Wang, "Improve the position measurement accuracy using a dynamic on-line fuzzy interpolation technique", *2003 IEEE Intl Symposium on Computational Intelligence for Measurement Systems and Applications*, 29-31 July 2003, page(s): 227 – 232.
- [27] Bizon, N.; Gabriel, I.; Oproescu, M, "Fuzzy interpolation of the average signal steps", *2009 International Symposium on Signals, Circuits and Systems*, 9-10 July 2009, Iasi, pp. 1 – 4.
- [28] Tang-Kai Yin, "A Characteristic-Point-Based Fuzzy Inference Classifier by a Closeness Matrix", *IEEE Trans on Fuzzy Systems*, Vol. 13, Issue 5, Oct. 2005, Page(s):673 – 687.
- [29] Triantafilis, W. T. Ward, I.O.A Odeh and A.B. McBratney, "Creation and Interpolation of Continuous Soil Layer Classes in the Lower Namoi Valley", *Soil Science Society of America Journal*, Mar/Apr 2001, Vol. 65, Issue 2, pp. 403-414.
- [30] D. Suzana, J.D. Marceau and M. Claude, "Space, time, and dynamics modeling in historical GIS databases: A Fuzzy

- Logic Approach”, *Environment & Planning B: Planning & Design*, Jul 2001, Vol. 28 Issue 4, pp. 545-563.
- [31] F. Song, S. M. Smith, and C. G. Rizk, “A Fuzzy Logic Controller Design Methodology for 4D System with Optimal Global Performance Using Enhanced Cell State Space Based Best Estimate Directed Search Method”, *IEEE Int. Conf. On Systems, Man and Cybernetics*, 1999.
- [32] Li, Dayou Yue, Yong Maple, Carsten Schetinin, Vitaly Qiu, Hua “Multi-dimensional fuzzy interpolation neural network”, *2009 IEEE International Conference on Automation and Logistics*, 5-7 Aug. 2009, Shenyang, China, pp. 186 – 190.
- [33] Seong-Ryong Chang, Choong-Ho Lee, and Uk-Youl Huh “The natural motion using fuzzy interpolation”, *2007 International Conference on Control, Automation and Systems*, 17-20 Oct. 2007, Seoul, pp. 425 – 428.
- [34] Ying Bai; Hanqi Zhuang; Dali Wang, “Calibration of parallel machine tools using fuzzy interpolation method”, *2008 IEEE International Conference on Technologies for Practical Robot Applications*, 10-11 Nov. 2008, Woburn, MA, USA, pp. 56 – 61.
- [35] Wang, Cheng and Xu, Shanzhen , “Lagrange Fuzzy Interpolating Controlling and Simulation on Semi-Active Suspension of Vehicle”, *ICICTA '09. Second International Conference on Intelligent Computation Technology and Automation*, Vol.2, 10-11 Oct. 2009, Changsha, Hunan, China, pp. 78 – 81.
- [36] Hsu, Chein-Chang and Chien, Youyin, “An Intelligent Fuzzy Affective Computing System for Elderly Living Alone”, *HIS '09. Fifth International Conference on Hybrid Intelligent Systems*, Vol.1, 12-14 Aug. 2009, Shenyang, China, pp. 293 – 297.
- [37] Yeesock Kim, Langari, R. and Hurlebaus, S. , “Supervisory semiactive nonlinear control of a building-magnetorheological damper system”, *American Control Conference*, 11-13 June 2008, Seattle, WA, USA, pp. 2540 – 2545.
- [38] Bizon, N. Gabriel, I. Oproescu, M., “Fuzzy interpolation of the average signal steps”, *International Symposium on Signals, Circuits and Systems*, 9-10 July, 2009, Iasi, pp. 1-4.
- [39] Dayou Li Yong Yue Maple, C. Schetinin, V. Hua Qiu, “Multi-dimensional fuzzy interpolation neural network”, *IEEE Intl. Conf. on Automation and Logistics*, 5-7 August, 2009, Shenyang, China, pp. 186-190.
- [40] Zhiheng Huang; Qiang Shen, “Preserving Piece-wise Linearity in Fuzzy Interpolation”, *Proceedings of the 18th International Conference on Fuzzy Systems (FUZZ-IEEE'09)*, pp. 575-580, 2009.
- [41] Cheng Wang, Shanzhen Xu, "Lagrange Fuzzy Interpolating Controlling and Simulation on Semi-Active Suspension of Vehicle," *ICICTA, 2009 Second International Conference on Intelligent Computation Technology and Automation*, Vol. 2, 2009, pp.78-81, Changsha, Hunan, China.
- [42] Hsiang-Chieh Chen and Wen-June Wang, “Two-Stage Interpolation Algorithm Based on Fuzzy Logics and Edges Features for Image Zooming”, *EURASIP Journal on Advances in Signal Processing*, Vol. 2009 (2009).
- [43] [http://en.wikipedia.org/wiki/Trilinear\\_interpolation](http://en.wikipedia.org/wiki/Trilinear_interpolation)
- [44] <http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/Interpolation>