Hindawi Publishing Corporation
Journal of Robotics
Volume 2012, Article ID 376293, 9 pages
doi:10.1155/2012/376293

Research Article

Fuzzy Interpolation and Other Interpolation Methods Used in Robot Calibrations

Ying Bai,[1] Nailong Guo,[2] and Gerald Agbegha[3]

[1] *Johnson C. Smith University, Charlotte, NC 28216, USA*
[2] *Benedict College, Columbia, SC 29204, USA*
[3] *Georgia Gwinnett College, Atlanta, GA 30043, USA*

Correspondence should be addressed to Ying Bai, ybai@jcsu.edu

Received 8 September 2011; Revised 4 February 2012; Accepted 6 February 2012

Academic Editor: G. Muscato

A novel interpolation algorithm, fuzzy interpolation, is presented and compared with other popular interpolation methods widely implemented in industrial robots calibrations and manufacturing applications. Different interpolation algorithms have been developed, reported, and implemented in many industrial robot calibrations and manufacturing processes 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.

1. 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 that was interpolated as the interpolation data. They developed a new rational interpolation with a biquadratic denominator to create a space surface using only values of the interpolated function and designed a bivariate rational Hermite interpolation to create a space surface using both function values and the first-order partial derivatives of the function 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^μ-rational spline function classes over triangles and quadrilaterals were investigated [12]. Hu and Tan 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 and Hussein 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 semicardinal 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 Noether-type theorem of piecewise algebraic curves on cross-cut partitions and used interpolation along a piecewise algebraic curve [19]. Maleknejad and Derili used box spline quasiinterpolants 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 [21–26].

Both linear and cubic spline interpolation methods can achieve satisfactory interpolation results for a common measurement and calibration process [27]. 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 [28]. 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 [28, 29]. 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 [30–42]. Different strategies of fuzzy interpolation have been developed and applied in real applications. Triantafilis et al. and Dragicevic et al. reported approaches of using fuzzy interpolation methods to estimate the soil layer and geographical distributions for GIS database [43, 44]. Song et al. described a fuzzy logic methodology for four-dimensional (4D) systems with optimal global performance using enhanced cell state space [45]. Li et al. reported a multidimensional fuzzy interpolation neural network to perform the fuzzy interpolations for a multidimensional system [46]. Chang et al. reported to use fuzzy interpolation methods to obtain the trajectory data for a multijoint animation robot [47]. Bai et al. developed a robot calibration algorithm to calibrate parallel machine tools using fuzzy interpolation techniques [48]. 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 [49]. Bai et al. developed a mind assistant system using fuzzy interpolation technique to support the judgment of emotion state for elderly living alone [48]. A supervisory semiactive, nonlinear control system combined with a fuzzy interpolation algorithm is reported by Kim et al. to improve the controllability of a MISO controller [50].

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.

2. 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.

2.1. 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 [29, 30]. The whole process can be simplified to perform three consecutive linear interpolations along three coordinate axes: x, y, and z, respectively.

Refer to Figures 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 by C_{000}, C_{100}, $C_{010}, \dots C_{111}$. Let x_d, y_d, and z_d be the differences between the target of x, y, z and the smaller coordinate related to the cubic lattice $[x]$, $[y]$, and $[z]$, the error values at 8 corners of the cubic lattice are $V_{000}, V_{100}, V_{010}, \dots$ and so forth $\dots V_{111}$, which is

$$x_d = x - [x], \qquad y_d = y - [y], \qquad z_d = z - [z]. \tag{1}$$

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

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

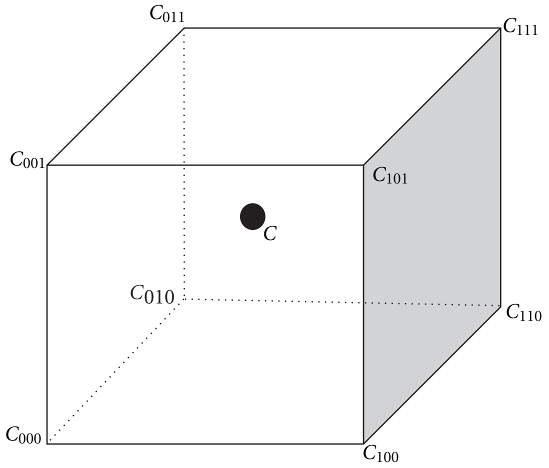


Figure 1: Eight corner points.

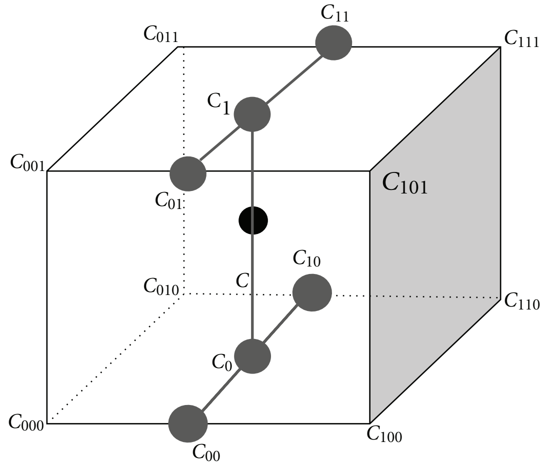


Figure 2: 3D trilinear interpolation.

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

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

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

$$t_p = V_0(1-x_d) + V_1x_d. \tag{4}$$

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.

Combining (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:

$$\begin{aligned} V_{xyz} =& V_{000}(1-x)(1-y)(1-z) + V_{100}x(1-y)(1-z) \\ &+ V_{010}(1-x)y(1-z) + V_{001}(1-x)(1-y)z \\ &+ V_{101}x(1-y)z + V_{011}(1-x)yz \\ &+ V_{110}xy(1-z) + V_{111}xyz. \end{aligned} \tag{5}$$

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 Figures 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.

2.2. 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 1st- and the 2nd-order derivatives ($S'(x, y, z)$ and $S''(x, y, z)$) of the interpolated points are existing, and the function $S''(x,\ y,\ z)$ is a trilinear surface on each cubic cell [43]. 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

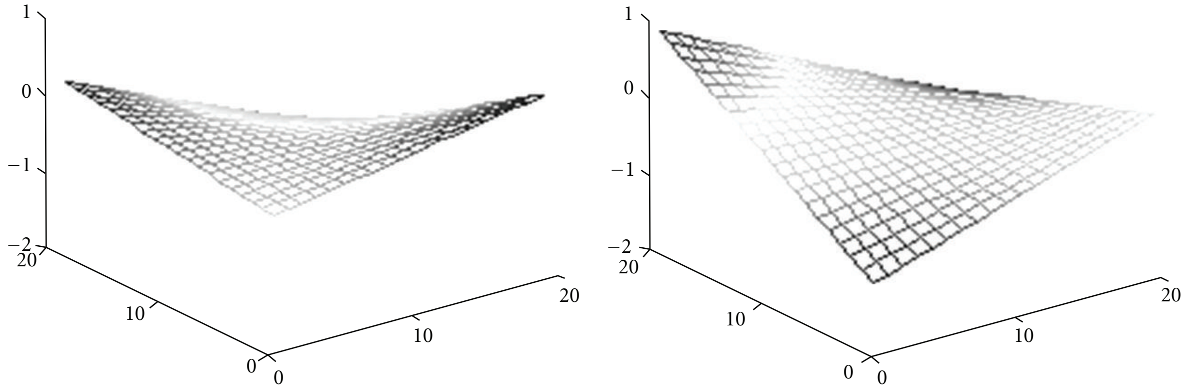
$$M_a = S''(a), \qquad M_b = S''(b). \tag{6}$$

A linear equation can be derived as follows:

$$S''(x) = \frac{(b-x)M_a + (x-a)M_b}{b-a}. \tag{7}$$

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

$$\begin{aligned} S(x) =& \frac{(b-x)^3M_a + (x-a)^3M_b}{6(b-a)} \\ &+ \frac{(b-x)S(a) + (x-a)S(b)}{b-a} \\ &- \frac{(b-a)[(b-x)M_a + (x-a)M_b]}{6}. \end{aligned} \tag{8}$$



(a) in x-direction

(b) in y-direction

Figure 3: Error surfaces in x- and y-directions on a cell.

By using the tridiagonal matrix, M_a and M_b can be derived. By substituting these 2nd-order derivatives back to (8), 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 Figure 3. The error surfaces are estimated based on neighboring grid position errors around the target position using the cubic spline technique.

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.

3. 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 $\mathbf{P}(\mathbf{x},\ \mathbf{y},\ \mathbf{z})$ 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.

In order to solve this problem and to improve the measurement and compensation accuracy, a dynamic online fuzzy interpolation method is introduced. The traditional fuzzy inference system uses predefined 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 offline fuzzy inference system because all inputs and outputs have been defined prior to the application process. This offline 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 offline 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 offline 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 × 40 × 40 small cubic cells, and each cell is $20 \times 20 \times 20$ mm^3. Assume that one lookup table is for one cubic cell, and this needs about 64000 lookup tables! By using an online 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 online fuzzy system is implemented. Therefore, we do not need any offline 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 online fuzzy inference algorithm is shown in Figure 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 Figure 4(a).

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_α, ee_β, and ee_γ, which are shown in Figure 4(b).

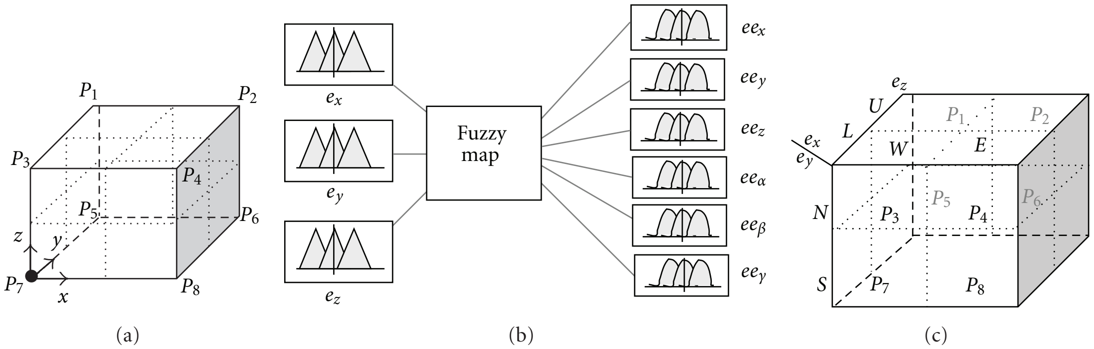


Figure 4: Definition of the fuzzy interpolation inference system.

The control rule is shown in Figure 4(c), 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^3 space. This is a typical workspace of most popular manipulators implemented in semiconductor manufacturing operations [46, 50, 51]. The input membership functions for x-, y-, and z-directions, and the predefined output membership functions are shown in Figure 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 5(b), 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 Figure 5(a), 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 [43, 44]. 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 online or dynamically. Figure 5(b) 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 ith 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 close 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 online 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 Figure 4(c) can be interpreted as follows after the output membership functions are determined:

(i) 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_α is α_1, ee_β is β_1 and ee_γ is γ_1.(P$_1$)

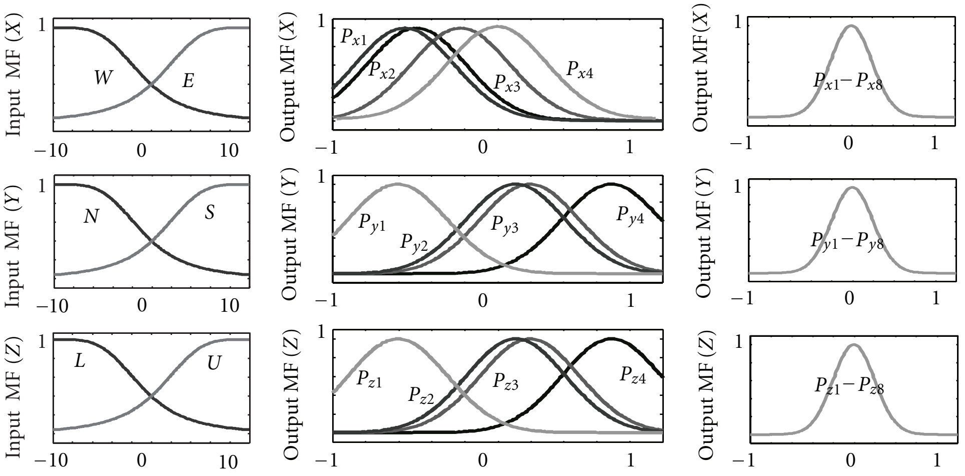
(ii) If e_x is W, e_y is N and e_z is L, then ee_x is P_{x3}, ee_y is P_{y3} and ee_z is P_{z3}, and ee_α is α_3, ee_β is β_3 and ee_γ is γ_3.(P$_3$)

(iii) 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_α is α_5, ee_β is β_5 and ee_γ is γ_5.(P$_5$)

(iv) 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_α is α_7, ee_β is β_7 and ee_γ is γ_7.(P$_7$)

(v) 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_α is α_2, ee_β is β_2 and ee_γ is γ_2.(P$_2$)

(vi) 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_α is α_4, ee_β is β_4 and ee_γ is γ_4.(P$_4$)



(a) (b) (c)

FIGURE 5: Input and output membership functions.

(vii) 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_α is α_6, ee_β is β_6 and ee_γ is γ_6.(P$_6$)

(viii) 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_α is α_8, ee_β is β_8 and ee_γ is γ_8.(P$_8$)

(9)

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:

$$L(E) = \{NWU, NWL, NEU, NEL, SWU, SWL, SEU, SEL\}. \tag{10}$$

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

$$u = \frac{\sum_{i=1}^{m}(u_i \times U_i)}{\sum_{i=1}^{m} u_i}, \tag{11}$$

where u is the current crisp output of the fuzzy inference system, and (11) 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

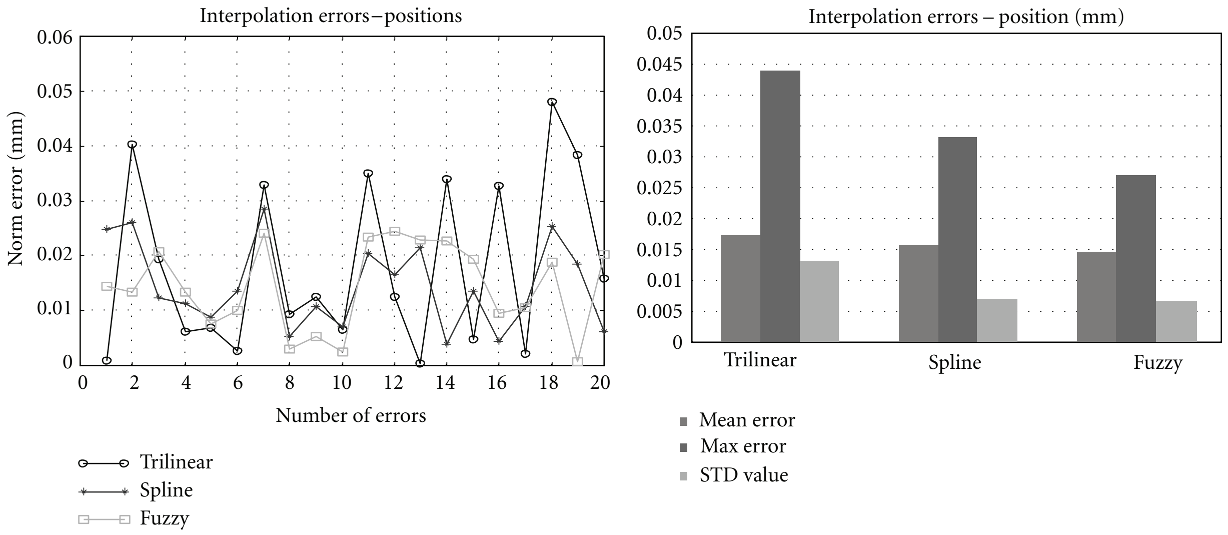
$$\begin{aligned} u_{xi} &= F_{xi}\,(P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8}), \\ U_{xi} &= Q_{xi}(P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8}), \end{aligned} \tag{12}$$

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 Figure 5(a). 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}–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 (9) in the x direction, and it is equivalent to the universe of discourse or a weighing factor. Substituting (12) into (11), one obtains:

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

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 online 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.

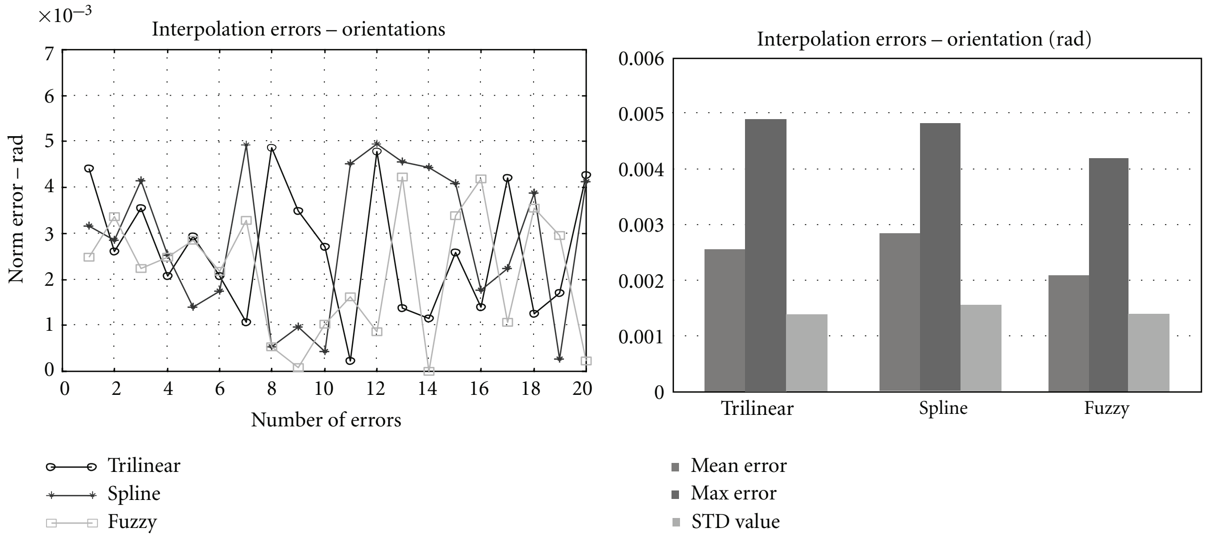
The advantage of using the online fuzzy inference system is that the control output has the real-time control ability, but the drawback is that this type of control has a relative longer response time because of the calculation performed



(a)

(b)

Figure 6: Simulated interpolation results—position errors.



(a)

(b)

Figure 7: Simulated interpolation results—orientation errors.

in the fuzzy inference system. This shortcoming becomes of little importance as the availability of high-speed CPUs for the controllers.

4. 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. Figure 6(a) shows a comparison of interpolated position errors using three interpolation techniques: trilinear, cubic spline, and fuzzy. Figure 6(b) shows a histogram comparison among three interpolation methods. Figures 7(a) and 7(b) 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 respect 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 Figure 7.

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

The FIA method provided in this paper has better performances compared with other interpolation methods. One possible shortcoming of this method is that it may need high-speed computer and large memory space to process and store predefined data on all grid points. However, this disadvantage can be easily overcome by using high-speed CPUs and huge memory spaces in today's computers.

5. 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] M. Krajnc, "Interpolation scheme for planar cubic G2 spline curves," *Acta Applicandae Mathematicae*, vol. 113, no. 2, pp. 129–143, 2011.

[2] A. Cuyt and W.-S. Lee, "Sparse interpolation of multivariate rational functions," *Theoretical Computer Science*, vol. 412, no. 16, pp. 1445–1456, 2011.

[3] R. Feng and X. 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, no. 5, pp. 1502–1514, 2011.

[4] G. Allasia and C. Bracco, "Two interpolation operators on irregularly distributed data in inner product spaces," *Journal of Computational and Applied Mathematics*, vol. 235, no. 7, pp. 1763–1774, 2011.

[5] N. Giménez, J. Heintz, G. Matera, and P. Solernó, "Lower complexity bounds for interpolation algorithms," *Journal of Complexity*, vol. 27, no. 2, pp. 151–187, 2011.

[6] S. Jakobsson, B. Andersson, and F. Edelvik, "Rational radial basis function interpolation with applications to antenna design," *Journal of Computational and Applied Mathematics*, vol. 233, no. 4, pp. 889–904, 2009.

[7] Q. Duan, Y. Zhang, and E. H. Twizell, "A bivariate rational interpolation and the properties," *Applied Mathematics and Computation*, vol. 179, no. 1, pp. 190–199, 2006.

[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*, vol. 195, no. 1-2, pp. 24–33, 2006.

[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*, vol. 161, no. 1, pp. 311–322, 2005.

[10] Q. Duan, S. Li, F. Bao, and E. H. Twizell, "Hermite interpolation by piecewise rational surface," *Applied Mathematics and Computation*, vol. 198, no. 1, pp. 59–72, 2008.

[11] Q. Duan, L. Wang, and E. H. Twizell, "A new weighted rational cubic interpolation and its approximation," *Applied Mathematics and Computation*, vol. 168, no. 2, pp. 990–1003, 2005.

[12] Z. Luo and X. Peng, "A C1-rational spline in range restricted interpolation of scattered data," *Journal of Computational and Applied Mathematics*, vol. 194, no. 2, pp. 255–266, 2006.

[13] M. Hu and J. Tan, "Adaptive osculatory rational interpolation for image processing," *Journal of Computational and Applied Mathematics*, vol. 195, no. 1-2, pp. 46–53, 2006.

[14] Q. J. Zhao and J. Tan, "Block-based Thiele-like blending rational interpolation," *Journal of Computational and Applied Mathematics*, vol. 195, no. 1-2, pp. 312–325, 2006.

[15] M. Sarfraz and M. Z. Hussain, "Data visualization using rational spline interpolation," *Journal of Computational and Applied Mathematics*, vol. 189, no. 1-2, pp. 513–525, 2006.

[16] T. N. T. Goodman and D. S. Meek, "Planar interpolation with a pair of rational spirals," *Journal of Computational and Applied Mathematics*, vol. 201, no. 1, pp. 112–127, 2007.

[17] M. Z. Hussain and M. Sarfraz, "Positivity-preserving interpolation of positive data by rational cubics," *Journal of Computational and Applied Mathematics*, vol. 218, no. 2, pp. 446–458, 2008.

[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*, vol. 197, no. 1, pp. 62–77, 2006.

[19] C. G. Zhu and R. H. Wang, "Lagrange interpolation by bivariate splines on cross-cut partitions," *Journal of Computational and Applied Mathematics*, vol. 195, no. 1-2, pp. 326–340, 2006.

[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*, vol. 190, no. 2, pp. 1557–1562, 2007.

[21] T. M. Lehmann, C. Gönner, and K. Spitzer, "Survey: interpolation methods in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1049–1075, 1999.

[22] R. P. Woods, *Handbook of Medical Image Processing and Analysis*, Academic Press, New York, NY, USA, 2000, Edited by I. N. Bankman.

[23] http://www.cgg-journal.com/1999-3/02/Cg_97h.htm.

[24] R. Huang, K. L. Ma, P. McCormick, and W. Ward, "Visualizing industrial CT volume data for nondestructive testing applications," in *Proceedings of the 14th IEEE Visualization*, pp. 547–554, Seattle, Wash, USA, October 2003.

[25] http://www.freepatentsonline.com/20030033050.pdf.

[26] http://www.freepatentsonline.com/5726896.pdf.

[27] M. A. Penna, "Camera calibration: a quick and easy way to determine the scale factor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 12, pp. 1240–1245, 1991.

[28] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design—A Practical Guide*, Academic Press, 3rd edition, 1993.

[29] I. J. Schoenberg, "Cardinal spline interpolation," in *Proceedings of the Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics*, Philadelphia, Pa, USA, 1973.

[30] Z. Huang and Q. Shen, "Fuzzy interpolation and extrapolation: a practical approach," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 13–28, 2008.

[31] Y. Bai and H. Zhuang, "On the comparison of model-based and modeless robotic calibration based on the fuzzy interpolation technique," in *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, pp. 892–897, December 2004.

[32] Y. Bai and D. Wang, "Improve the position measurement accuracy using a dynamic on-line fuzzy interpolation technique," in *Proceedings of the IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications*, pp. 227–232, 2003.

[33] N. Bizon, I. Gabriel, and M. Oproescu, "Fuzzy interpolation of the average signal steps," in *Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS '09)*, pp. 1–4, July 2009.

[34] T. K. Yin, "A characteristic-point-based fuzzy inference classifier by a closeness matrix," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 5, pp. 673–687, 2005.

[35] N. Bizon, I. Gabriel, and M. Oproescu, "Fuzzy interpolation of the average signal steps," in *Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS '09)*, pp. 1–4, July 2009.

[36] D. Li, Y. Yue, C. Maple, V. Schetinin, and H. Qiu, "Multidimensional fuzzy interpolation neural network," in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL '09)*, pp. 186–190, Shenyang, China, August 2009.

[37] Z. Huang and Q. Shen, "Preserving piece-wise linearity in fuzzy interpolation," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '09)*, pp. 575–580, August 2009.

[38] C. Wang and S. Xu, "Lagrange fuzzy interpolating controlling and simulation on semi-active suspension of vehicle," in *Proceedings of the 2nd International Conference on Intelligent Computing Technology and Automation (ICICTA '09)*, vol. 2, pp. 78–81, Changsha, Hunan, China, 2009.

[39] H. C. Chen and W. J. Wang, "Two-stage interpolation algorithm based on fuzzy logics and edges features for image zooming," *Eurasip Journal on Advances in Signal Processing*, vol. 2009, Article ID 372180, 2009.

[40] http://en.wikipedia.org/wiki/Trilinear_interpolation..

[41] http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/Interpolation.

[42] G. Agbegha, N. Guo, and Y. Bai, "Compare fuzzy interpolation algorithm with other interpolation methods used in industrial applications," in *Proceedings of the Hawaii University Engineering and Mathematics International Conference*, Honolulu, Hawaii, USA, 2011.

[43] J. 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*, vol. 65, no. 2, pp. 403–413, 2001.

[44] S. Dragicevic, D. J. Marceau, and C. Marois, "Space, time, and dynamics modeling in historical GIS databases: a fuzzy logic approach," *Environment and Planning B*, vol. 28, no. 4, pp. 545–562, 2001.

[45] F. Song, S. M. Smith, and C. G. Rizk, "Fuzzy logic controller design methodology for 4D systems with optimal global performance using enhanced cell state space based best estimate directed search method," in *Proceedings of theIEEE International Conference on Systems, Man, and Cybernetics 'Human Communication and Cybernetics*', October 1999.

[46] D. Li, Y. Yue, C. Maple, V. Schetinin, and H. Qiu, "Multidimensional fuzzy interpolation neural network," in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL '09)*, pp. 186–190, Shenyang, China, August 2009.

[47] S. R. Chang, C. H. Lee, and U. Y. Huh, "The natural motion using fuzzy interpolation," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '07)*, pp. 425–428, Seoul, Republic of Korea, October 2007.

[48] Y. Bai, H. Zhuang, and D. Wang, "Calibration of parallel machine tools using fuzzy interpolation method," in *Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications (TePRA '08)*, pp. 56–61, Woburn, Mass, USA, November 2008.

[49] C. Wang and S. Xu, "Lagrange fuzzy interpolating controlling and simulation on semi-active suspension of vehicle," in *Proceedings of the 2nd International Conference on Intelligent Computing Technology and Automation (ICICTA '09)*, vol. 2, pp. 78–81, Changsha, Hunan, China, 2009.

[50] Y. Kim, R. Langari, and S. Hurlebaus, "Supervisory semiactive nonlinear control of a building-magnetorheological damper system," in *Proceedings of the American Control Conference (ACC '08)*, pp. 2540–2545, Seattle, Wash, USA, June 2008.

[51] C.-C. Hsu and Y. Y. Chien, "An intelligent fuzzy affective computing system for elderly living alone," in *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS '09)*, vol. 1, pp. 293–297, Shenyang, China, 2009.

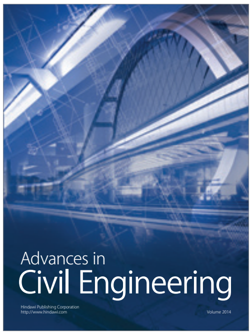
Journal of
Engineering

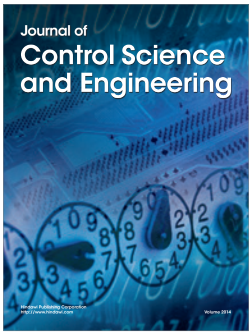
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

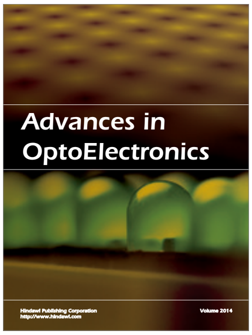
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

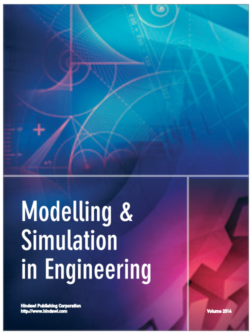
Journal of
Robotics

Journal of
Electrical and Computer Engineering

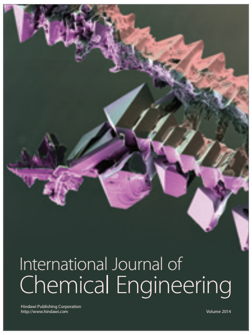

Advances in
OptoElectronics

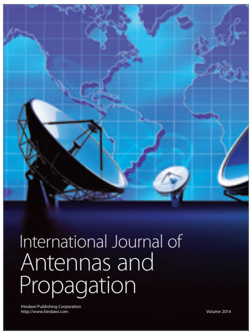
VLSI Design

International Journal of
Navigation and Observation

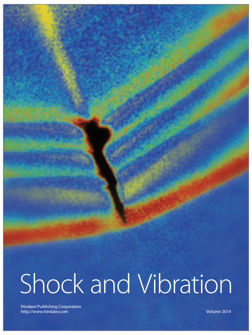

Modelling & Simulation in Engineering

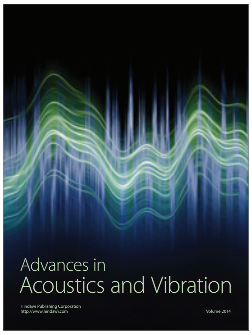
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration