



# Implementation of a Variable D-H Parameter Model for Robot Calibration Using an FCMAC Learning Algorithm <sup>\*</sup>

KUU-YOUNG YOUNG and JIN-JOU CHEN

*Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan*

(Received: 28 August 1997; accepted: 27 October 1998)

**Abstract.** Current robot calibration schemes usually employ calibration models with constant error parameters. Consequently, they are inevitably subject to a certain degree of locality, i.e., the calibrated error parameters (CEPs) will produce the desired accuracy only in certain regions of the robot workspace. To deal with the locality phenomenon, CEPs that vary in different regions of the robot workspace may be more appropriate. Hence, we propose a variable D-H (Denavit and Hartenberg) parameter model to formulate variations of CEPs. An FCMAC (Fuzzy Cerebellar Model Articulation Controller) learning algorithm is used to implement the proposed variable D-H parameter model. Simulations and experiments that verify the effectiveness of the proposed calibration scheme based on the variable D-H parameter model are described.

**Key words:** robot calibration, variable D-H parameter model, FCMAC learning algorithm.

## 1. Introduction

In research on the integration of CAD systems with robot manipulators, a well-recognized problem is that a robot manipulator is inevitably subject to a certain level of imprecision. In general, robot manipulators offer high repeatability, but their accuracy is relatively low [24]. This high repeatability makes teach-and-playback a feasible approach. However, the relatively low accuracy means that a planned path from a CAD system cannot be sent directly for execution, because the robot manipulator will be unable to locate the end-effector as accurately as required by the CAD system. Therefore a proper calibration scheme is imperative to provide the demanded accuracy, so that equivalence of coordinates between the CAD system and robot system can be achieved. The error sources that result in imprecision can be categorized as *geometric errors*, such as imprecise link lengths, angular offsets due to axis misalignment, and imperfections in the machine axes (i.e., the motions of prismatic and revolute joints are not exclusive from each other), and *non-geometric errors*, such as varying joint compliance due to dif-

---

<sup>\*</sup> This work was supported in part by the National Science Council, Taiwan, under grant NSC 82-0422-E-009-403.

fering payloads and configurations, backlash, gear transmission errors, base tilt, feedback quantization errors, cross coupling of joint rotations, and temperature variations [1, 4, 7, 18, 35]. To identify the error parameters, various calibration schemes have been proposed [7, 11, 12, 14, 16, 18, 20, 27, 29–35, 37]. Among them, Ahmad [1] proposed a mathematical model of the kinematic nonlinear drive train error. Azadivar [4] evaluated the effect of joint position errors of industrial robots using a stochastic model. Chen and Chao [7] examined error sources that contribute to inaccuracy for robots with rotary joints. Everett et al. [11] examined various models proposed for robot kinematic calibration and discussed them from the standpoints of completeness, equivalence, and proportionality. Judd and Knasinski [18] developed models for inaccuracy resulting from errors in the link and joint parameters, imperfections in the main spur and encoder pinion gears, and structural deformations. Wu [35] presented a linear model to represent positioning errors as a function of the errors in link parameters.

Once calibrated error parameters (CEPs) have been determined, they can be incorporated into the formulation, and the resulting Cartesian errors can be obtained by computing the forward kinematics [34]. Compensation schemes can then be applied to derive a modified set of joint solutions to compensate for the Cartesian errors, so that the robot manipulator will move to the specified location accurately [11, 13, 15, 30, 33, 36]. Thus, the compensation issue is basically that of tackling the Jacobian relationship between the Cartesian errors and the equivalent joint variable changes, and this Jacobian relationship has been dealt with in either Cartesian space [33] or joint space [15], and by using various methods, such as the numerical approach [13] and the approach involving the screw theory [36]. On the other hand, Shamma and Whitney [30] proposed the approximation function method to find the desired joint variables in a specific region, which avoids the complicated modeling process of identifying the parameter errors and the corresponding calibration problem.

Intuitively, an appropriate calibration scheme should be capable of identifying the CEPs for a given robot manipulator uniquely. This is because that the error parameters, such as link lengths and angular offsets, should remain constant for relatively long periods of time. However, existing calibration models fail to meet this claim, because they are formulated on the basis of certain dominating factors that affect robot accuracy. Consequently, the effects of minor factors, unformulated or unknown factors, and measurement errors are incorporated into the identified CEPs via imperfect models. In addition, CEPs are strongly affected by the selection of the measured data used for calibration: CEPs are different when they are derived using measured data from different parts of the workspace under different configurations, dynamic effects, payloads, etc. This characteristic demonstrates that the single sets of CEPs identified in current calibration schemes possess locality. Therefore, a single set of CEPs may perform well in certain parts of the workspace, but may not be appropriate for other paths that pass through different regions of the workspace. An approach to resolve this problem might be to

- formulate a complete calibration model that includes all error parameters;
- provide a huge amount of measured data distributed throughout the entire workspace to use in identifying an appropriate set of CEPs;
- perform calibration by collecting measured data around planned paths;
- derive different sets of CEPs corresponding to different parts of the workspace and different conditions by using current calibration schemes.

The first approach is by no means an easy task, considering the complexity of formulating and identifying all possible error parameters in a calibration model. In the second approach, the average accuracy may increase because of the large amount of measured data at the expense of time; however, the demanded path accuracy specified by a given task may still not be achieved. The main problem with the third approach is that it requires that the calibration process be repeated each time a new task is encountered. Last, the feasibility of the fourth approach depends on whether the number of sets of CEPs is reasonable and whether these sets of CEPs can be organized properly. Like the concept mentioned in the fourth approach, in the next section, we propose a new parameter model and a corresponding calibration scheme that provides appropriate sets of CEPs for various planned paths needed in industrial applications.

## 2. Proposed Parameter Model and Calibration Scheme

From the preceding discussions, we may conclude that CEPs that vary in different regions of the robot workspace are better able to deal with the locality phenomenon. Accordingly, we propose a variable D-H (Denavit and Hartenberg [8]) parameter model, that can properly describe the concepts of having different sets of CEPs for different conditions and different regions in the robot workspace, along with a calibration scheme based on it.

### 2.1. VARIABLE D-H PARAMETER MODEL

The variable D-H parameter model is intended to formulate variations of CEPs; the characteristics of the CEPs are evaluated as follows:

- CEP derivation is influenced by data measurements obtained from various locations and configurations in the workspace.
- CEP locality results from unmodeled factors and modeled factors that are localized with respect to different kinematic and dynamic conditions.
- Two sets of CEPs are close in value if they are derived for neighboring areas or similar conditions.
- Robot accuracy after compensation is determined by the selected CEPs.

The local CEP characteristics described above imply that there should be a certain degree of continuity in CEP values among different regions in the workspace. Thus a generalization to the entire workspace based on appropriate finite sets of CEPs may be feasible. In other words, implementation of this variable D-H parameter model can be achieved by using functions capable of generalization, such as learning algorithms, spline functions, and other appropriate methods.

## 2.2. PROPOSED CALIBRATION SCHEME

In the proposed calibration scheme, we use an FCMAC (Fuzzy CMAC) learning algorithm to implement the variable D-H parameter model. The FCMAC learning algorithm is based on the Cerebellar Model Articulation Controller (CMAC) proposed by Albus [2, 3] and on fuzzy logic. Because the purpose of the learning algorithm is for generalization, obviously other neural networks or spline functions are also potential candidates for use in our scheme. One reason we have chosen to use a CMAC-type neural network is based on measurement considerations. Because of the structure and kinematic constraints on robot manipulators and measurement devices, certain locations in the robot workspace may not be accessible for measurement. Consequently, data needed for generalization may be sparse or even missing altogether in some regions. The CMAC-type neural network is considered less sensitive to data paucity, i.e., the data used for training in certain parts of the network does not affect the neighboring regions too much. This property of the CMAC-type neural network makes it well-suited to our needs; in addition, this type of network has a strong generalization capability and a simple structure.

The proposed calibration scheme includes a measurement space analysis module, conventional calibration and compensation schemes, and an FCMAC learning algorithm, as shown in Figure 1. In addition, we also include an automatic measurement process to collect the measurement data needed for calibration. In Figure 1, the workspace is first divided into a finite number of local regions by the measurement space analysis module. Two main issues are considered in dividing the workspace:

- (1) how to divide it into appropriate local regions, and
- (2) how to choose proper measurement locations in each local region to identify a representative set of CEPs.

We refer to the resolution of these issues as measurement space analysis, and discuss it in Section 4. For each local region, a conventional calibration scheme is employed to derive a set of CEPs using the measurement locations specified by the measurement space analysis module. This scheme is discussed in Section 3. The FCMAC learning algorithm generates appropriate sets of CEPs for the entire workspace based on the finite sets of CEPs derived from the processes described above. The FCMAC learning algorithm is discussed in Section 5. The learning process in this calibration scheme needs to be performed just once; the accuracy achieved is on the order of robot repeatability. After the FCMAC learning

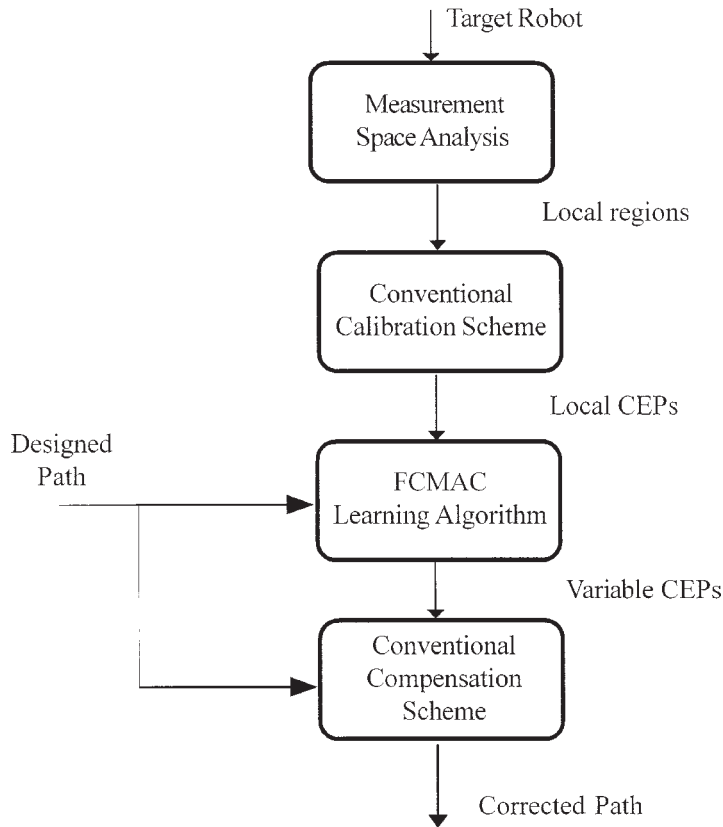


Figure 1. The proposed robot calibration scheme.

algorithm is set up, sets of variable CEPs corresponding to designed paths can be generated. A compensation scheme using the sets of variable CEPs can then be employed to determine corrected sets of joint variables that compensate for inaccuracies. This scheme is discussed in Section 3.

By using the proposed calibration scheme, the generated variable CEPs will be appropriate for achieving the demanded accuracy in each local region. By contrast, conventional calibration schemes in general identify only one set of CEPs from measured data. Consequently, CEPs derived by conventional calibration schemes improve accuracy on average, but are not necessarily appropriate for each local region. The excellence of the proposed calibration scheme is due to its exploitation of the measured data in deriving appropriate sets of CEPs. To demonstrate the feasibility of our scheme, in Sections 6 and 7, simulations and experiments are conducted based on using a Mitsubishi RV-M2 type five-axis industrial robot manipulator. Also in Section 7, the design of an automatic measurement process to collect the measurement data needed for calibration is introduced. Discussion and conclusion are given in Sections 8 and 9, respectively.

### 3. Conventional Calibration and Compensation

We first discuss how to use a conventional calibration scheme to identify the representative set of CEPs for each local region specified by measurement space analysis. As mentioned in the introduction, Cartesian errors result from both geometric and non-geometric errors. Although some of these errors may not be well modeled or properly identified, their effects are all incorporated into the representative set of CEPs. Thus, this set of CEPs is localized to a certain degree and is appropriate for use in compensating for imprecisions in the corresponding local region. Hence, the sets of CEPs for all local regions can serve as local representatives that the learning algorithm can generalize over the entire workspace. A compensation scheme can then be developed to compute the corrected set of joint variables based on the CEPs. The compensation scheme should be independent of the calibration scheme; thus the locality reflected in the set of CEPs for each region will not be affected. The system block diagram of the conventional calibration and compensation schemes is shown in Figure 2 and discussed in the following sections.

#### 3.1. CALIBRATION

A calibration model dealing with kinematic error parameters is described by the following equation [33]:

$$POS = BASE \cdot A_1 \cdot \cdots \cdot A_N \cdot FLANGE \cdot TOOL. \quad (1)$$

In the equation above, *BASE* represents the relationship between the world and base coordinates fixed for the robot manipulator;  $A_i$  represents the relationship between joint coordinate frame  $i - 1$  on the joint  $i$  axis and joint coordinate frame  $i$  on the joint  $i + 1$  axis; and *FLANGE* represents a transformation from the  $N$ th joint coordinates to a coordinate frame located on the manipulator's flange used for mounting tools. *TOOL* is the transformation from the mounting point defined by *FLANGE* to the end point of a calibrated mounting tool. To describe the kinematic errors in *BASE*,  $A_i$ , and *FLANGE*, a well-known representation is the homogeneous transformation matrix defined by Denavit and Hartenberg [8]. Their representation uses four kinematic parameters  $\theta$ ,  $d$ ,  $a$ , and  $\alpha$  to formulate the relationship between two consecutive link coordinate frames. In the case of two consecutive parallel joints, in addition to these four Denavit and Hartenberg link parameters, another rotational kinematic error  $\beta$  is introduced to align the two near-parallel joints [33].

Based on the relationship between the correct position  $POS^c$  of some reference point on the manipulator's tool in the presence of parameter errors and the nominal position *POS* of that reference point assuming nominal parameter values, we can formulate the equation for calibration as follows:

$$d\mathbf{x} = [M]CEP, \quad (2)$$

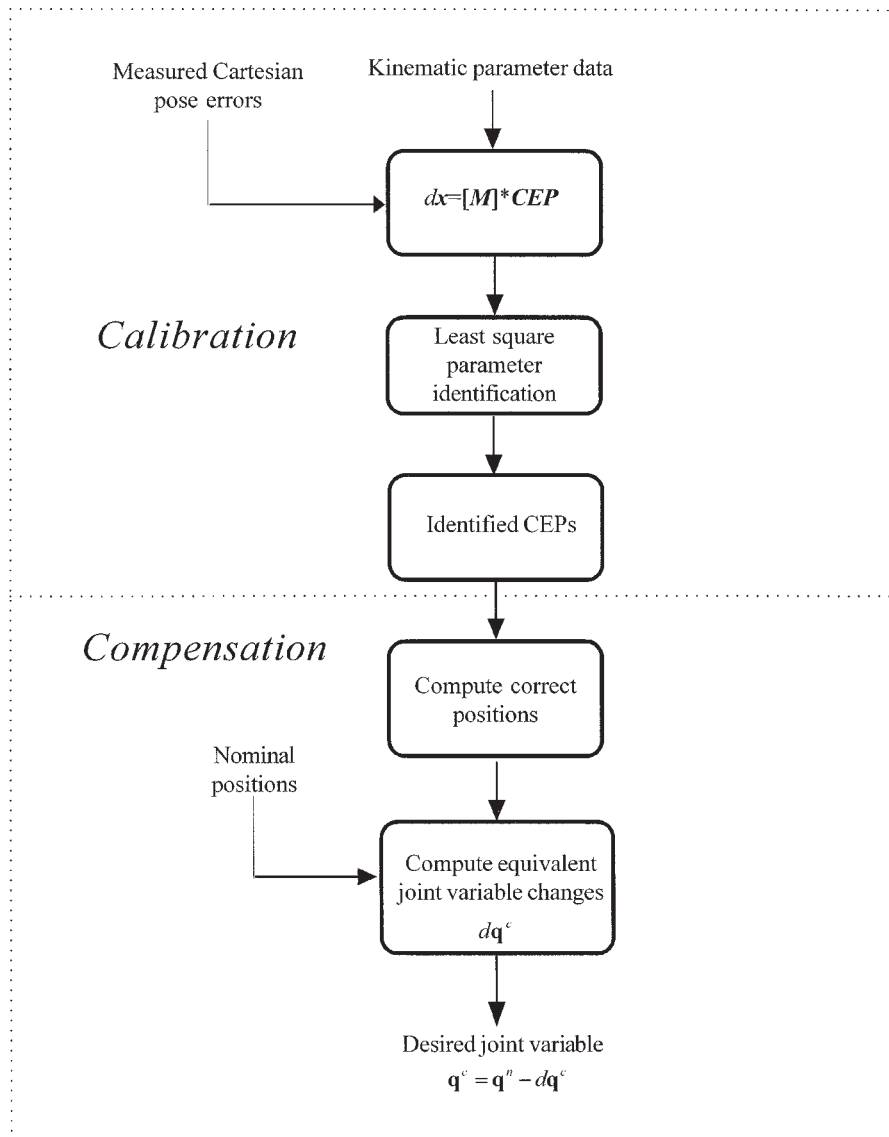


Figure 2. The system block diagram of the conventional calibration and compensation schemes.

where  $\mathbf{dx}$  is the error vector between  $POS$  and  $POS^c$ , CEP stands for the kinematic error parameters, and  $[M]$  is the matrix of the partial derivatives related to  $\mathbf{dx}$  and the CEP. A certain amount of measurement data  $\mathbf{dx}$  are collected from the robot workspace to make the columns of  $[M]$  independent. A set of CEPs is then uniquely determined by means of the least-square-error approach or other numerical method [33].

### 3.2. COMPENSATION

After the calibration process, a compensation scheme is developed to compute a corrected set of joint variables based on a set of CEPs so as to compensate for inaccuracy in the tool location. As the control of industrial robot manipulators is based on exact simplified kinematic equations, the presence of errors will cause the parameter values to deviate from the ideal ones [26]. As a result, the robot manipulator becomes a non-simplified type. Since the inverse solutions for the non-simplified robot manipulator are very complicated to obtain, a feasible approach for finding the desired joint solutions is to continue using the simplified kinematic equations to find the equivalent joint variable changes needed to compensate for the resulting Cartesian errors. The desired joint variables are then derived from the following equation [15, 33]:

$$\mathbf{q}^c = \mathbf{q}^n - d\mathbf{q}^c, \quad (3)$$

where  $d\mathbf{q}^c$  represents the equivalent joint variable changes,  $\mathbf{q}^n$  the nominal solutions derived from the simplified kinematic equations, and  $\mathbf{q}^c$  the joint variables needed to drive the robot manipulator to the desired location. A successful compensation scheme should provide absolute robot accuracy that approaches the repeatability provided by the robot controller [24].

## 4. Measurement Space Analysis

The measurement space analysis module performs two main functions: (1) It divides the workspace into an appropriate number of local regions, and (2) it chooses suitable measurement locations from which to identify representative sets of CEPs for local regions. Because the division of the workspace affects the selection of the measurement locations, these two factors are in fact interrelated.

### 4.1. SELECTION OF MEASUREMENT LOCATIONS

First we deal with the second function in the module: the selection of measurement locations. To accomplish this function, the module needs to find unique CEPs in Equation (2) by choosing finite measurement locations within each local region. For a manipulator with  $N$  degrees of freedom, the problem becomes that of finding a matrix  $[M]$  in Equation (2) with independent columns by moving these  $N$  joints to different locations in a certain region. In addition, the columns of the matrix  $[M]$  are not only expected to be independent, but also to form a well-conditioned matrix for the sake of matrix inverse computation. Another important constraint on choosing measurement locations is that they must be accessible for measurement. This is because due to measurement device limitations not all configurations of a robot manipulator are necessarily measurable.

Several researchers have proposed choosing measurement locations with large observability measure for parameter identification, so that fewer measurement lo-



cations will be required and the CEPs identified will be less affected by measurement noise [5, 6, 9, 12, 19]. Borm and Menq [6] defined an observability index in terms of the singular values of the error propagation matrix, which can be related to the matrix  $[M]$  in Equation (2). They found that as the observability measure of a set of measurement configurations increases, attribution of the Cartesian errors to CEPs becomes dominant, and consequently better estimation of CEPs becomes possible. In addition, they concluded that good selection of points is more important than the number of points in obtaining good identification results. Khalil et al. [19] suggested checking the condition number for observability, which is an index also related to the singular values of the error propagation matrix. Bay [5] proposed an autonomous algorithm for finding a sequence of optimal configurations during calibration so that the estimated error parameters will converge as fast as possible. A cost function based on the Kalman filter is used in searching the configuration space and the direction of steepest descent over all possible directions is chosen.

We also choose measurement locations with large observability for parameter identification. However, instead of the entire workspace, we will select measurement locations within each local region. In choosing appropriate measurement locations for the identification of CEPs, we will adopt the condition number  $\kappa([M])$  as the index, defined as in [19]:

$$\kappa([M]) = \|[M]^t[M]\| \cdot \|[M]^t[M]\|^{-1}, \quad (4)$$

where  $\|[M]\|$  indicates a norm of  $[M]$ . In Equation (4),  $\kappa([M])$  involves the pseudo-inverse, because the equation for calibration in Equation (2) is over-determined. This condition number quantifies the sensitivity of the solution to the noise resulting from measurement and unmodeled errors. A number of measurement data  $d\mathbf{x}$  are then collected from the robot workspace to make the columns of  $[M]$  independent and make Equation (4) yield a small condition number, indicating that the parameter identified is less influenced by measurement and unmodeled errors. One thing to note is that because some parameters can be identified accurately only by excursions of certain distances in the workspace, local regions cannot be too small for measurement location selection: when the local regions are too small, the conditioning of the matrix  $[M]$  in Equation (2) may degrade and make Equation (4) generate large condition numbers, such that accurate identification of some parameters cannot be achieved.

#### 4.2. WORKSPACE DIVISION

Another function to be dealt with in this module is workspace division. Because it is difficult to determine the distributions of the measurement, modeled, and unmodeled errors in the robot workspace, there is no a priori information available for optimal workspace division. Thus, we propose performing workspace division adaptively by evaluating the compensation effect in each local region. The basic concept behind this is that if the derived representative set of CEPs is satisfactory

in achieving the demanded accuracy for the local region, then no further division is needed; otherwise, the local region will be divided further. Intuitively, the dividing procedure should be performed recursively until the representative sets of CEPs for all local regions in the entire workspace are satisfactory. However, as aforementioned in Section 4.1, some parameters can be identified accurately only by excursions of certain distances in the workspace; thus, subdivision cannot be executed without limit. In addition, when there were error sources that varied with high spatial frequencies, a minute subdivision of the workspace might occur. Therefore, the total number of local regions may be set to a pre-determined value to avoid too many subdivisions at the expense of accuracy.

The proposed workspace division strategy will be performed based on mainly the major joints. In general, the length of the tool in a robot manipulator is small compared with those of the links governed by the major joints; consequently, the area swung by the tool, which is governed by the minor joints, is also small. Therefore, the effect of workspace division based on the minor joints is insignificant, and for most cases this division is in fact unnecessary. Simulations have been conducted on this issue and the results verified the statement above. By employing the proposed strategy, for workspace division in the joint space, the complexity in dealing with general non-redundant robot manipulators will be similar to that of a robot manipulator with three degrees of freedom; for workspace division in Cartesian space, the complexity involved is also alike, when only the position, but not the orientation, is of concern. Thus, the proposed calibration scheme can be applied to general multi-joint robot manipulators without excessive workspace division.

One point remains to be discussed is the configuration consideration for workspace division in Cartesian space. When the robot manipulator is with degrees of freedom more than three, it may reach the same 3-dimensional Cartesian position with different configurations. According to the experimental results in [33], it shows that the CEPs derived varied significantly when different configurations were used for calibration. Our experimental results also lead to the same conclusion. Thus, we consider that the configuration should be taken as another parameter for calibration performed in Cartesian space. Therefore, for workspace division in Cartesian space, we propose that one specific configuration be chosen first, and then the calibration be performed based on the same configuration.

Note that in applying the proposed division strategy, the entire workspace is treated as a local region in the beginning. Consequently, like conventional calibration schemes, only one representative set of CEPs is derived, when this representative set of CEPs is satisfactory for the first local region (i.e., the entire workspace). In other words, the conventional constant parameter model can be viewed as a special case of the proposed variable parameter model when no workspace division is conducted. Thus, the performance of the proposed calibration scheme is guaranteed to be better than that of the conventional one. The famous cell decomposition approach is adopted for workspace partitioning [21]. In the cell decomposition approach, the workspace is represented as a collection of cells with a simple pre-

specified shape, e.g., rectangloid shape; these cells can then be organized into a tree-like structure. The algorithm for recursive workspace division is formulated as follows:

**Recursive Workspace Division Algorithm.** Recursively divide the workspace to obtain satisfactory representative sets of CEPs for all local regions in the entire workspace under the constraint of a limited number of local regions.

**Step 1:** Take the entire workspace as a local region.

**Step 2:** Derive the representative set of CEPs for each local region by using the conventional calibration scheme described in Section 3.1 and choosing suitable measurement locations following the procedure described in Section 4.1. When small condition numbers are very difficult to find in identifying a representative set of CEPs for a local region, the division for that local region will terminate.

**Step 3:** Randomly select a number of test points for each local region. For each test point, compute the error between  $POS$  in Equation (1) using the model with nominal D-H parameters and the estimated  $POS^c$  using the model with nominal D-H parameters plus the representative set of CEPs. Then apply the compensation scheme described in Section 3.2 to compensate for the error.

**Step 4:** If the demanded accuracy for all test points in the local region is achieved after compensation or the preset total number of local regions is reached, no further division is performed; otherwise, divide the local region into two equal regions, then go to Step 2.

In general, after the recursive division process is completed, the entire workspace will be divided into a number of local regions of different dimensions according to variations in CEPs and the demanded accuracy. Figure 3 shows an example of recursive workspace division. In Figure 3, a tree-like structure is used to organize local regions of different dimensions. A node in the tree stands for a local region and the fraction inside indicates the proportion of its volume compared with that of the entire workspace. A child node is a portion of a parent node and they are connected by a link. A leaf stands for a local region whose representative set of CEPs are satisfactory in achieving the demanded accuracy. For instance, in Figure 3 eight leaves, A-H, were generated in workspace division.

This recursive workspace division algorithm can be applied to both Cartesian and joint spaces. Let us take the Mitsubishi RV-M2 type five-axis robot manipulator shown in Figure 4(a), for example. This robot manipulator is with all revolute joints and its kinematic data are listed in Table I. Figure 4(b) shows an example of workspace division in the joint space, and the division is based on the major joints only, i.e., the first three joints. This example also corresponds to the recursive division illustrated in Figure 3, and the rectangloids of different sizes in Figure 4(b) correspond to the eight leaves in Figure 3. Figure 4(c) shows an example of workspace division in Cartesian space, and the division is imposed upon

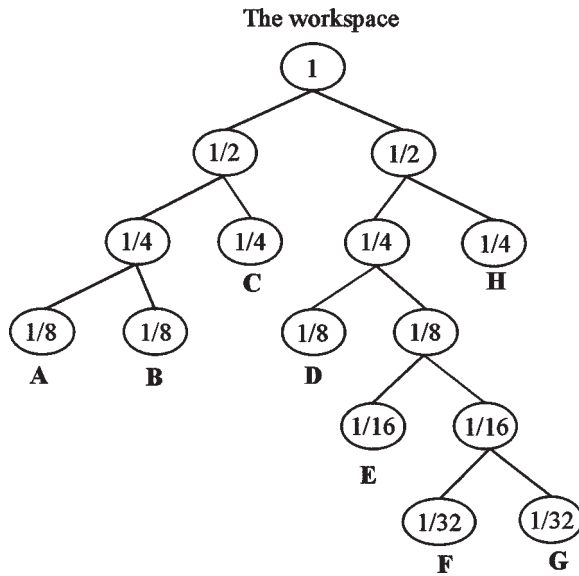


Figure 3. An example of recursive workspace division.

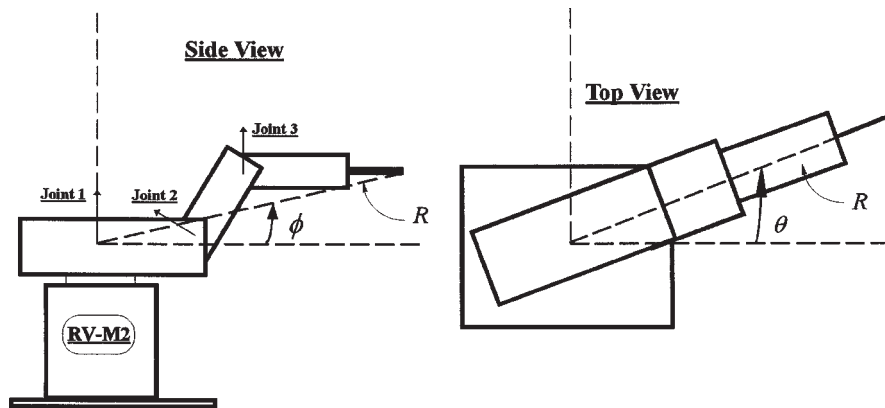


Figure 4(a). A Mitsubishi RV-M2 type five-axis robot manipulator.

the position only, but not the orientation. Considering that all the manipulator's joints are revolute, in Figure 4(c) the workspace is described using a spherical coordinate system and indexed by the spherical coordinates ( $R$ : 500 mm  $\sim$  805 mm,  $\theta$ :  $-45^\circ \sim 45^\circ$ ,  $\phi$ :  $-150^\circ \sim 150^\circ$ ). Obviously, other coordinate systems can also be applied, such as Cartesian and cylindrical coordinate systems. The example shown in Figure 4(c) also corresponds to that in Figure 3, and only region A is depicted as an illustration.

The representative set of CEPs should be satisfactory in achieving the demanded accuracy, if the recursive workspace division algorithm is not terminated due to the preset total number of local regions is reached or due to the local regions are

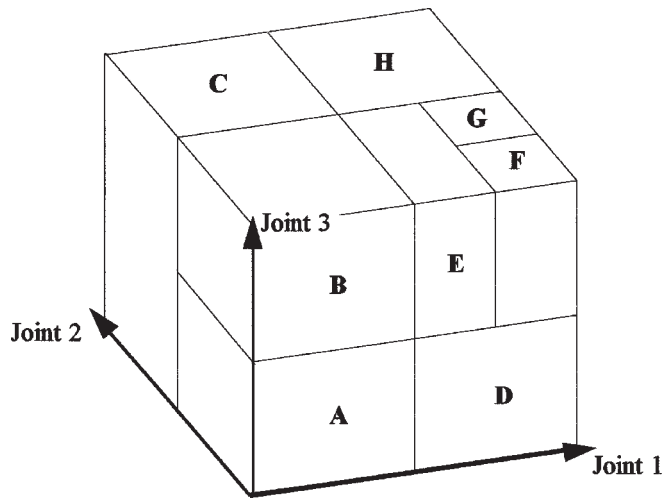
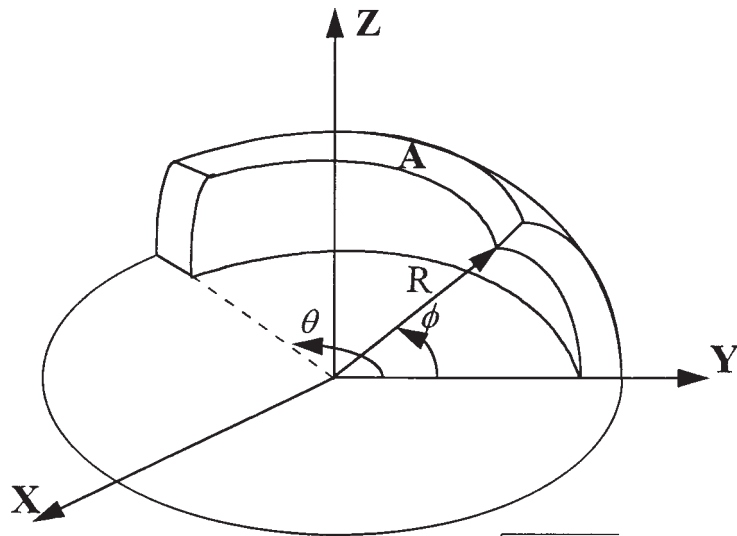


Figure 4(b). Workspace division in the joint space.



$$\begin{aligned}
 x &= R \cos \phi \sin \theta \\
 y &= R \cos \phi \cos \theta \\
 z &= R \sin \phi
 \end{aligned}$$

$$R = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \tan^{-1} \frac{x}{y}$$

$$\phi = \tan^{-1} \frac{z}{\sqrt{x^2 + y^2}}$$

Figure 4(c). Workspace division in Cartesian space.

Table I. Kinematic table for the Mitsubishi RV-M2 type five-axis robot manipulator

	$\theta_i$	$d_i$	$a_i$	$\alpha_i$	$\beta_i$
World-J1	$90^\circ$	400 mm	-80 mm	$180^\circ$	$0^\circ$
J1-J2	$\theta_1$	0 mm	120 mm	$-90^\circ$	$0^\circ$
J2-J3	$\theta_2$	0 mm	250 mm	$0^\circ$	$0^\circ$
J3-J4	$\theta_3$	0 mm	265 mm	$-90^\circ$	$0^\circ$
J4-J5	$\theta_4$	0 mm	0 mm	$90^\circ$	$0^\circ$
J5-Flange	$\theta_5$	0 mm	0 mm	$0^\circ$	$0^\circ$
Flange-Tool	$0^\circ$	0 mm	170 mm	$0^\circ$	$0^\circ$

$$\theta_1 : -150^\circ \sim 150^\circ$$

$$\theta_2 : -30^\circ \sim 100^\circ$$

$$\theta_3 : -120^\circ \sim 0^\circ$$

$$\theta_4 : -110^\circ \sim 110^\circ$$

$$\theta_5 : -180^\circ \sim 180^\circ$$

too small for accurate parameter identification, and if the set of test points chosen to ensure accuracy in the local region is adequate. In the meantime, because the representative set of CEPs are constant over the entire local region, it is possible that discontinuities are present between neighboring regions. Consequently, the accuracy after compensation may vary evidently around boundary areas. This is due to that different levels of accuracy are reached in adjacent local regions, when their representative sets of CEPs are all satisfactory in achieving the demanded accuracy. Because continuity of CEPs between neighboring local regions is expected as discussed in Section 2.1, we suggest that discontinuities in representative sets of CEPs between neighboring local regions should be smoothed out. This is another reason why we use the FCMAC learning algorithm to implement the variable D-H parameter model. Later, in Sections 6 and 7, simulation and experimental results show that continuous CEPs generated by the FCMAC learning algorithm perform better than discontinuous CEPs when a path crosses a boundary area.

## 5. The FCMAC Learning Algorithm

The FCMAC learning algorithm module is developed to generate appropriate sets of CEPs for the entire workspace based on the finite sets of CEPs for the local regions derived by the measurement space analysis module. In addition, it can also smooth out CEP discontinuities between neighboring local regions resulting from the recursive workspace division described above. As mentioned in Section 2.2, the FCMAC is not the only possible choice for this purpose. The reasons for adopting this algorithm are its simple structure, ability to generalize, and insensitivity to

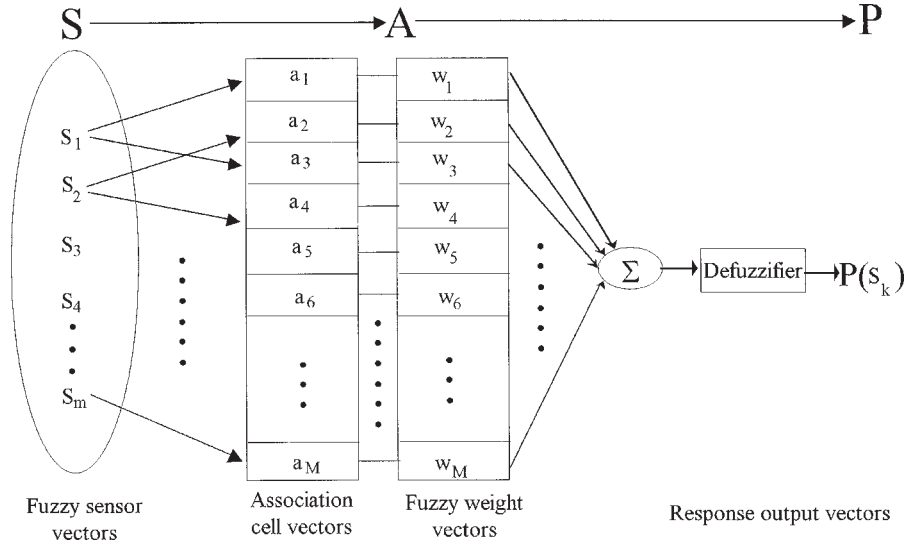


Figure 5. Basic FCMAC mappings.

sparse data or missing data. The FCMAC is basically a CMAC-type neural network in which fuzzy sets are stored and manipulated using fuzzy logic. The inputs and outputs of the CMAC are basically discrete, because continuous data are quantized in the sensor layer; by contrast, the inputs and outputs of the FCMAC are continuous, because of fuzzy representation and manipulation. This feature makes the FCMAC more suitable than the CMAC for storing continuous CEPs [17, 25]. The basic mappings in the FCMAC are as shown in Figure 5. Details of the FCMAC learning algorithm can be found in the appendix, and a brief summary of the operations in the FCMAC learning algorithm is described below [2, 3, 23]:

**Step 1:** Index the input vector  $S$  in fuzzy representation.

**Step 2:** Map it into the corresponding association cell vectors  $A$ .

**Step 3:** Update the fuzzy weights according to the difference (*DIF*) between the actual and desired outputs.

**Step 4:** If the *DIF* in Step 3 is within the tolerance, set up the response output vector  $P$ ; otherwise, go to Step 3.

To demonstrate how the sets of CEPs for each local region are stored in the FCMAC, let us again take the Mitsubishi RV-M2 type robot manipulator in Figure 4(a) as an example and use the Cartesian space division in Figure 4(c). The input vector  $S_k$  in  $S$  will contain three elements ( $R, \theta, \phi$ ) (indicated in Figure 4(c)) for indexing various locations in the workspace:

$$S_k = (R, \theta, \phi). \quad (5)$$

$(R, \theta, \phi)$  will then be described by fuzzy sensor vectors. Via the network mappings, the fuzzy sensor vectors should correspond to output responses of desired sets of CEPs, when appropriate weights for the active association cells in  $\mathbf{A}$  are assigned. Thus, after successful learning by utilizing the representative sets of CEPs as training patterns, the FCMAC learning algorithm module will generate sets of CEPs in  $\mathbf{P}$  for input vectors indexing arbitrary locations in the workspace. The module can then generate appropriate sets of CEPs to provide high accuracy for different paths corresponding to various robot tasks. The accuracy that the learning algorithm can provide will depend on the resolution of input vectors  $\mathbf{S}_k$ , which results from the workspace division and the selection of measurement locations in the measurement space analysis module. Since this FCMAC learning algorithm involves mainly summations of weights after an off-line learning process, it is computationally efficient and can be used for on-line implementation.

## 6. Simulation

To verify the proposed calibration scheme, simulations were performed for cases involving geometric errors, non-geometric errors, and random measurement errors using a Mitsubishi RV-M2 type five-axis robot manipulator, as shown in Figure 4(a). The kinematic data for this robot manipulator are listed in Table I. The workspace division was based on the major joints, i.e., the first three joints. Thus, to simplify the analysis, the angles of joints four and five were fixed at zero. Fifteen error parameters ( $\Delta\theta_0, \Delta a_0, \Delta\alpha_0, \Delta\beta_0, \Delta\theta_1, \Delta d_1, \Delta a_1, \Delta\alpha_1, \Delta\theta_2, \Delta a_2, \Delta\alpha_2, \Delta\beta_2, \Delta\theta_3, \Delta d_3, \Delta a_3$ ) out of the twenty-five (five error parameters ( $\Delta\theta, \Delta d, \Delta a, \Delta\alpha, \Delta\beta$ ) for each of the five transformations, *Base*,  $A_1$ – $A_3$ , and *Flange* in Equation (1)) were found to be observable. Therefore, we need at least five measurement locations to make the columns of the  $3 \times 15$  matrix  $[M]$  in Equation (2) independent. Considering that non-geometric errors and other uncertainties may affect measurement data, we chose twenty measurement locations in each local region to average out their influences. The selection criteria were that the twenty measurement locations had to be accessible for measurement and their corresponding condition number had to be smaller than a certain value [19]. The use of twenty measurement locations was found to be adequate for identifying the representative set of CEPs for each local region in our simulations.

In the first set of simulations, the workspace was divided into local regions in the joint space. By applying the recursive workspace division algorithm described in Section 4.2, the workspace was divided into 110 local regions of differing dimensions. For each local region, the demanded accuracy after compensation was within 0.1 mm for the Cartesian position error. After the division and calibration processes, the representative set of CEPs was derived for each local region and the CEPs were constant over the entire local region. The representative sets of CEPs for these local regions were then used as training patterns to set up the FCMAC to generate smooth CEPs between neighboring local regions and for unmeasurable



regions. During the training of the representative sets of CEPs for the local regions, several locations within one local region were chosen along with the corresponding representative set of CEPs to serve as the input-output pairs. In the FCMAC, the sensory layer for input vectors and the association layer for association cell vectors included 39 and 2057 cells, respectively. The number of units of the fuzzy sensors with the largest membership grades used for representing the input vector was chosen to be  $\mathbf{c} = (3, 3, 3)$  for  $(\theta_1, \theta_2, \theta_3)$ . Thus each output vector in the response layer connected  $27 = 3 \times 3 \times 3$  cells. The hash-coding technique was not used in the mapping.

The robot manipulator was simulated to follow a straight-line path in the joint space from  $(60^\circ, 60^\circ, -35^\circ)$  to  $(125^\circ, 90^\circ, -10^\circ)$ , which passed through several local regions. Kinematic errors were assigned to be 0.04 cm or 0.05 degree for each of the fifteen observable error parameters. The non-geometric errors were formulated based on a model similar to that in [18], which describes non-geometric errors  $e_n$  as combinations of sin and cos functions:

$$e_n = f_1(\cos \theta_1, \sin \theta_1) + f_2(\cos \theta_2, \sin \theta_2) + f_3(\cos \theta_3, \sin \theta_3), \quad (6)$$

where  $f_1$ ,  $f_2$  and  $f_3$  are linear functions. The non-geometric errors were assigned to contribute about 8–10% of the total error. In this set of simulations, we did not include random measurement errors. Figure 6(a) shows the Cartesian position errors without calibration. The errors ranged between 0.4 and 1.2 mm. For comparison, three kinds of calibration were performed to compensate for the errors:

- (i) the conventional calibration scheme,
- (ii) the proposed calibration scheme without the FCMAC, and
- (iii) the proposed calibration scheme with the FCMAC.

The same set (consequently the same number:  $2200 = 20$  measurements  $\times$  110 (regions)) of measurement data were used to derive the CEPs for these three kinds of calibration. Thus, the comparison was made on an equal basis. Simulation results are shown in Figures 6(b)–(d), and the corresponding CEPs  $\Delta\theta_1$ ,  $\Delta d_1$ ,  $\Delta a_1$ ,  $\Delta\alpha_1$  for these three cases in Figures 6(e)–(g), respectively. In Figure 6(b), the conventional calibration scheme reduced the errors to less than 0.2 mm. The corresponding CEPs were constant along the simulated path, as shown in Figure 6(e). The CEPs values deviated from the assigned kinematic errors due to the presence of non-geometric errors. Figures 6(c)–(d) show that the proposed calibration scheme, with and without, the FCMAC yielded better results than the conventional one (below 0.1 mm); but the residual errors after compensation varied evidently around boundary areas using the proposed calibration scheme without the FCMAC. Figures 6(f)–(g) show the corresponding CEPs for the calibration schemes without and with the FCMAC, respectively. Note that in Figure 6(f) each strip stands for the portion of a local region that a path passed through, not necessarily equal to the dimension of the entire local region. The CEPs are constant within each local region in Figure 6(f); while in Figure 6(g) smooth curves across neighboring regions replace abrupt jumps between straight lines in Figure 6(f). This indicates that

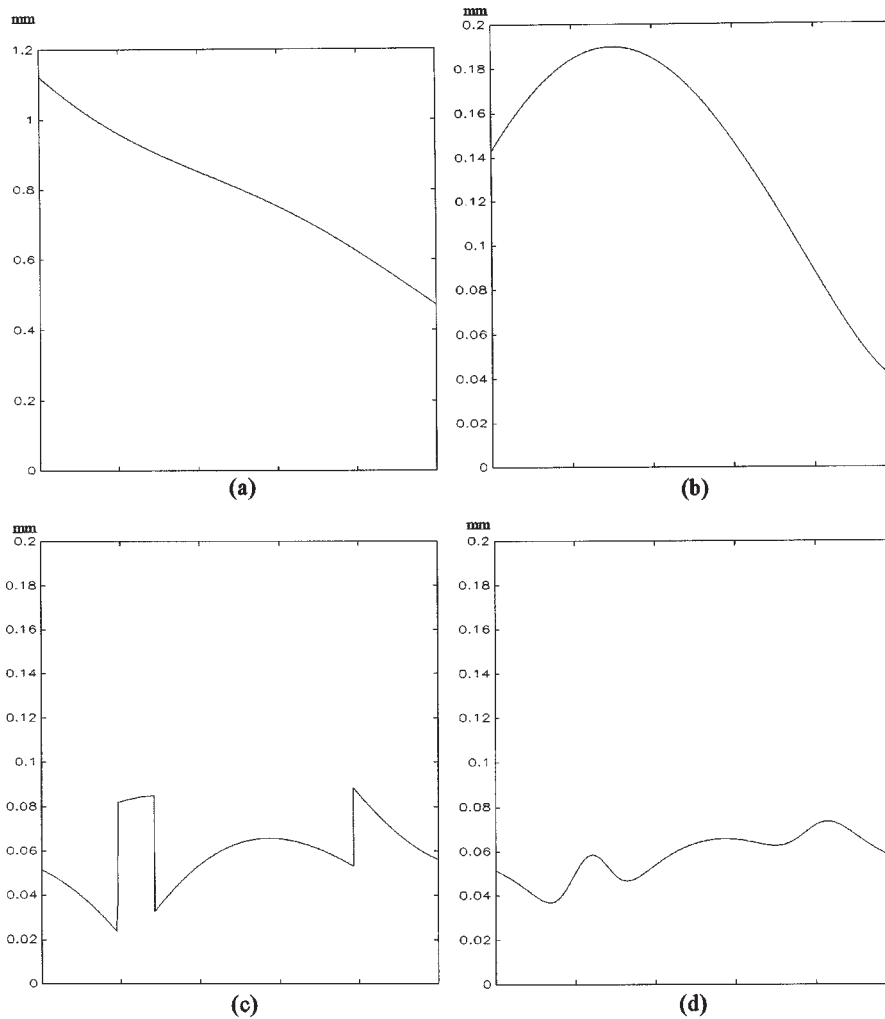


Figure 6. Simulation results with workspace division in the joint space. Position errors: (a) without calibration, (b) the conventional calibration scheme, (c) the proposed calibration scheme without the FCMAC, and (d) the proposed calibration scheme with the FCMAC. Note that the  $x$  axis: points on a straight-line joint path,  $(60^\circ, 60^\circ, -35^\circ)$  to  $(125^\circ, 90^\circ, -10^\circ)$ .

the FCMAC learning algorithm smoothed out the CEPs, and consequently more homogeneous compensation effects were obtained across regional boundaries.

Another set of simulations was performed to investigate division in Cartesian space. The workspace was divided into local regions in Cartesian space, and the spherical coordinate system shown in Figure 4(c) was used. By applying the recursive workspace division algorithm, the workspace was divided into 130 local regions of differing dimensions. For each local region, the demanded accuracy after compensation was within 0.1 mm for the Cartesian position error. The specifications for the FCMAC learning algorithm were similar to those used for division in

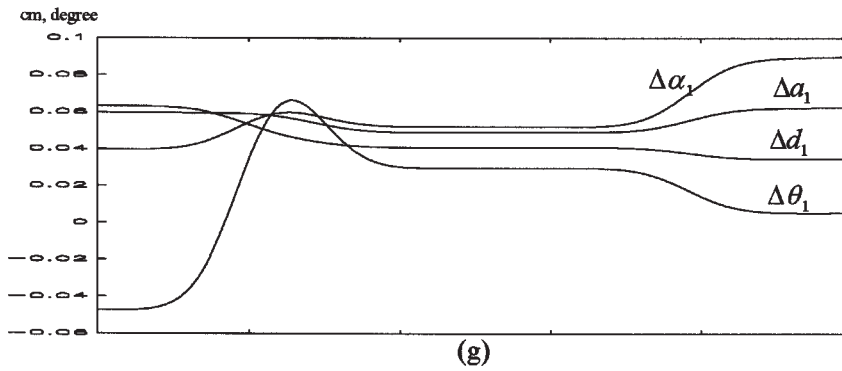
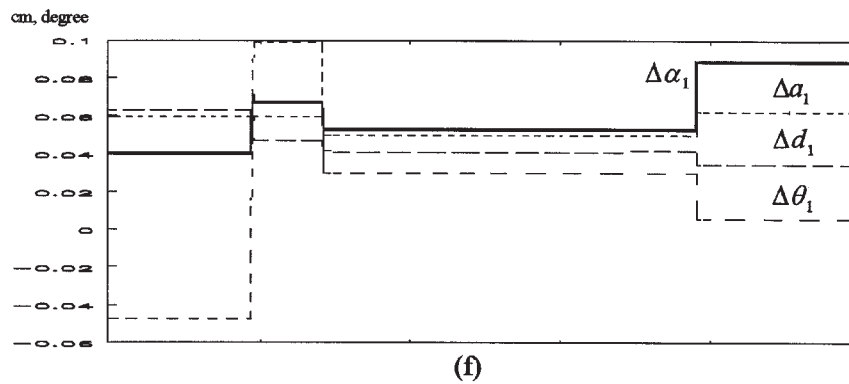
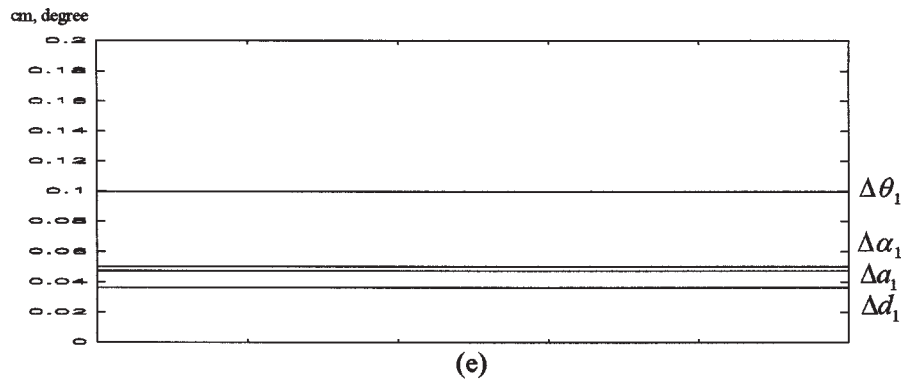


Figure 6 (Continued). Corresponding CEPs  $\Delta\theta_1$ ,  $\Delta d_1$ ,  $\Delta a_1$ ,  $\Delta\alpha_1$ : (e) for case (b), (f) for case (c), and (g) for case (d).

the joint space. The robot manipulator was simulated to follow a straight-line path described in the spherical coordinate system from (640 mm, 20°, 0°) to (740 mm, 125°, -20°), which passed through several local regions. Results similar to those for division in the joint space were obtained, as shown in Figure 7. Note that for the simulations shown in Figures 6 and 7, the CEPs derived by the conventional

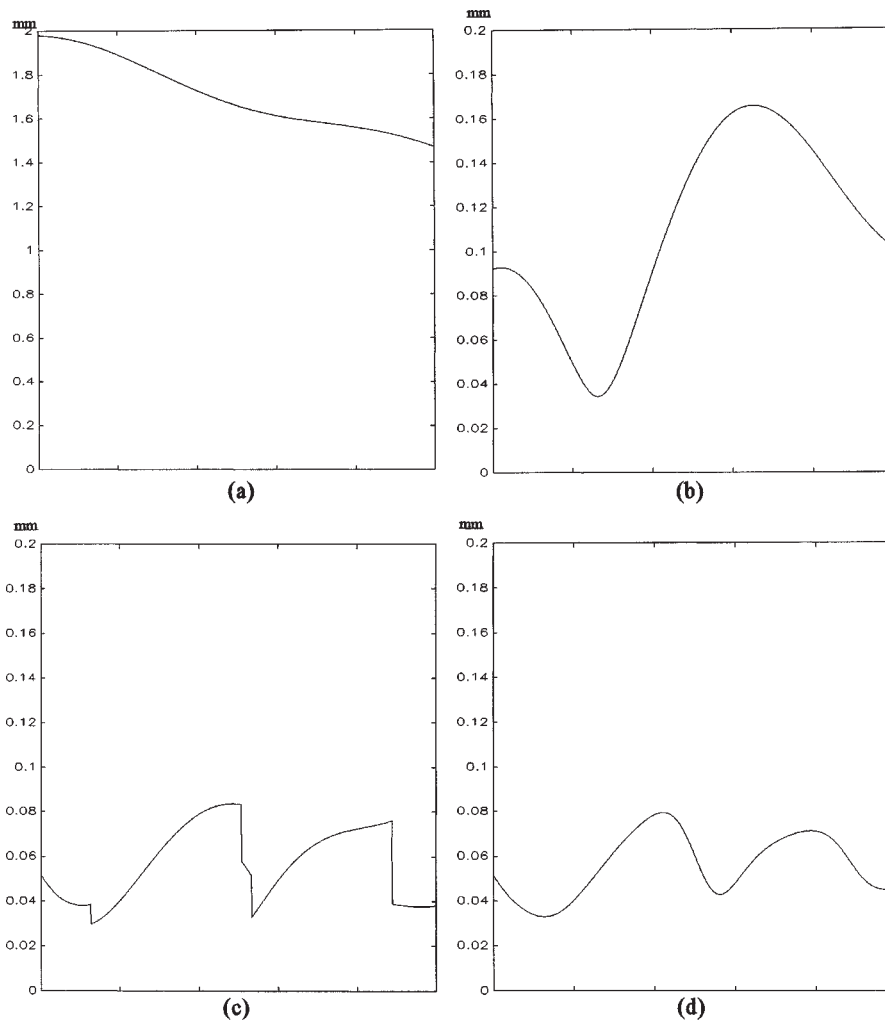


Figure 7. Simulation results with workspace division in Cartesian space. Position errors: (a) without calibration, (b) the conventional calibration scheme, (c) the proposed calibration scheme without the FCMAC, and (d) the proposed calibration scheme with the FCMAC. Note that the  $x$  axis: points on a straight-line Cartesian path (spherical coordinates),  $(640 \text{ mm}, 20^\circ, 0^\circ)$  to  $(740 \text{ mm}, 125^\circ, -20^\circ)$ .

calibration scheme are different, because different sets of measurement data were used. Different compensation effects were also observed. On the other hand, the proposed calibration scheme demonstrated similar performances using these two different sets of measurement data.

To test the ability of the proposed calibration scheme to generalize CEPs, simulations were performed for cases in which measurement data were not available for certain local regions. The same robot path as that shown in Figure 6 was simulated and workspace division was in the joint space. We assumed that one of the local

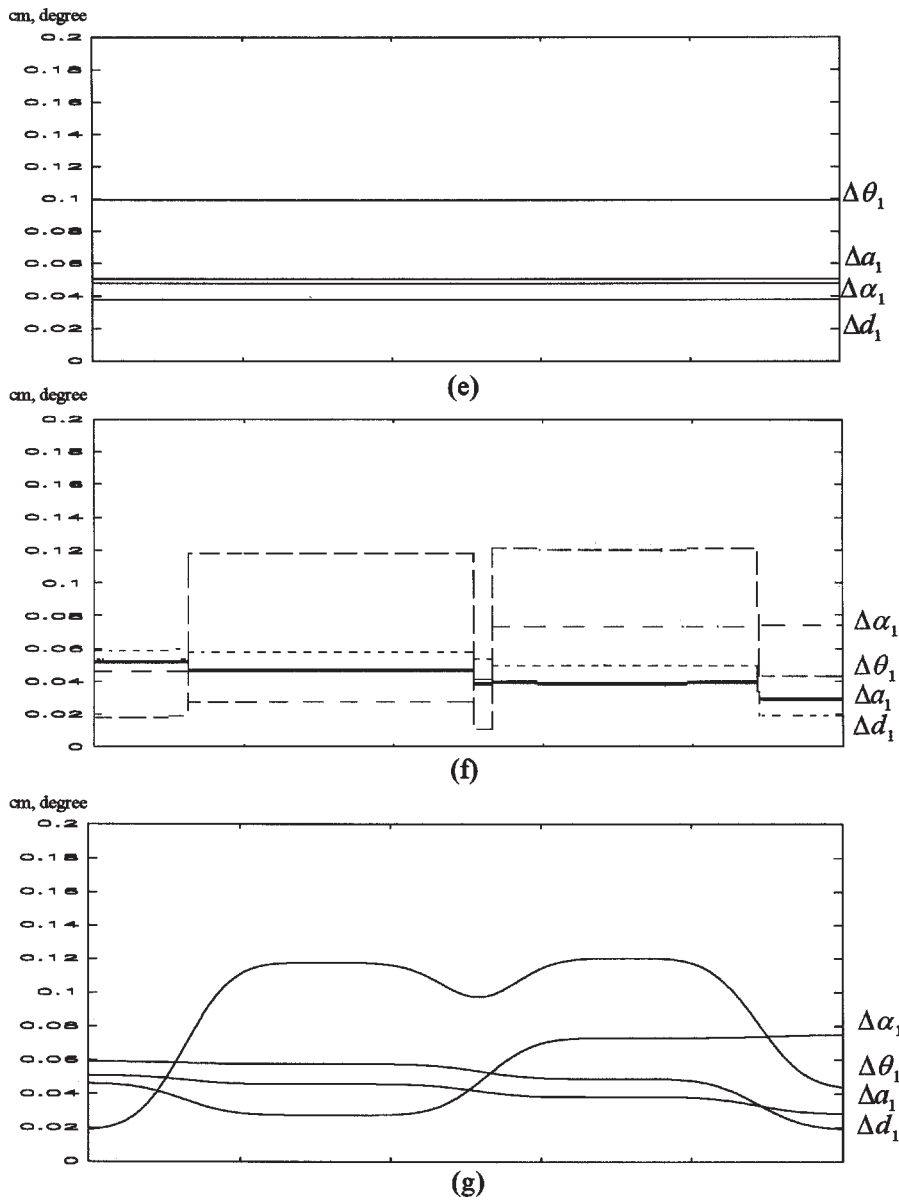


Figure 7 (Continued). Corresponding CEPs  $\Delta\theta_1$ ,  $\Delta d_1$ ,  $\Delta a_1$ ,  $\Delta\alpha_1$ : (e) for case (b), (f) for case (c), and (g) for case (d).

regions the path passed through was not measurable. We also used the three kinds of calibration mentioned previously to generate CEPs for the unmeasurable local region. The conventional calibration scheme used the same set of CEPs for both the measurable and unmeasurable local regions. The proposed calibration scheme without the FCMAC used average CEPs derived from CEPs for neighboring local

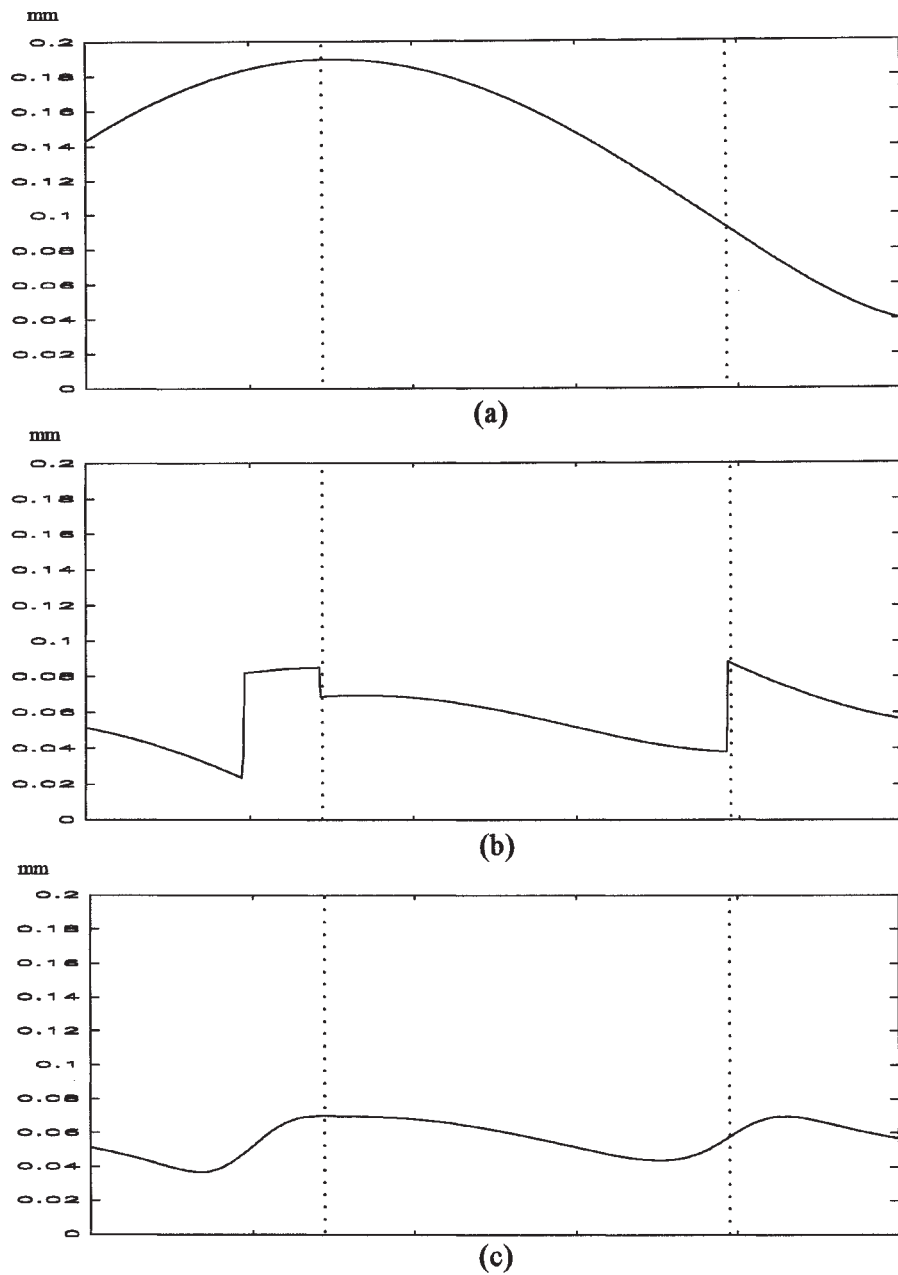


Figure 8. Simulation results with one local region without measurement data. Position errors: (a) the conventional calibration scheme, (b) the proposed calibration scheme without the FCMAC, and (c) the proposed calibration scheme with the FCMAC. Note that the  $x$  axis: points on a straight-line joint path,  $(60^\circ, 60^\circ, -35^\circ)$  to  $(125^\circ, 90^\circ, -10^\circ)$ .

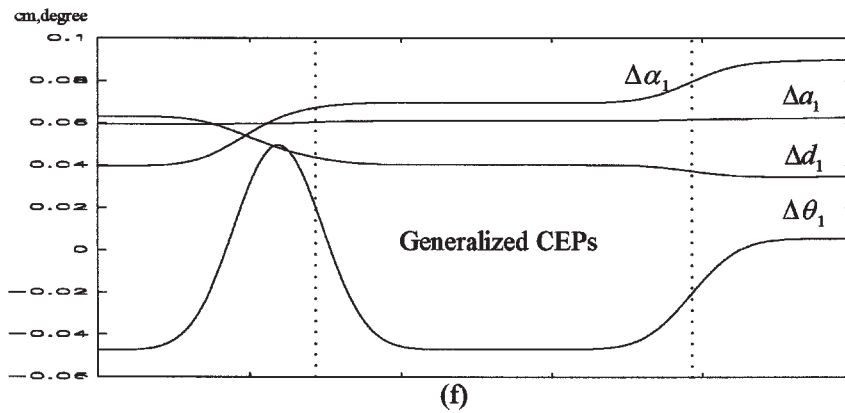
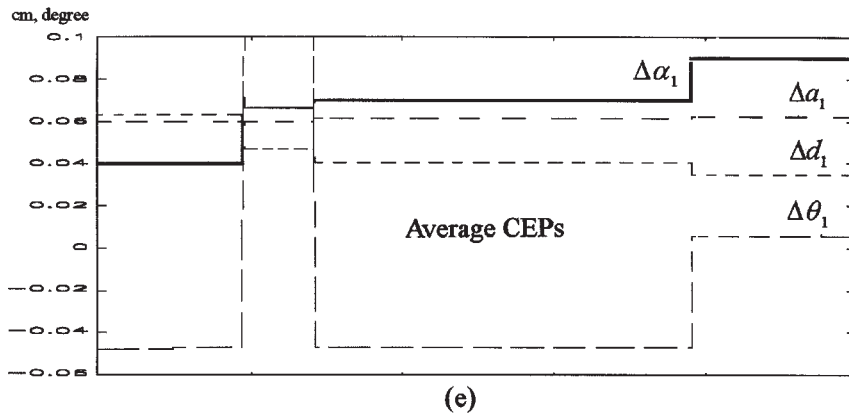
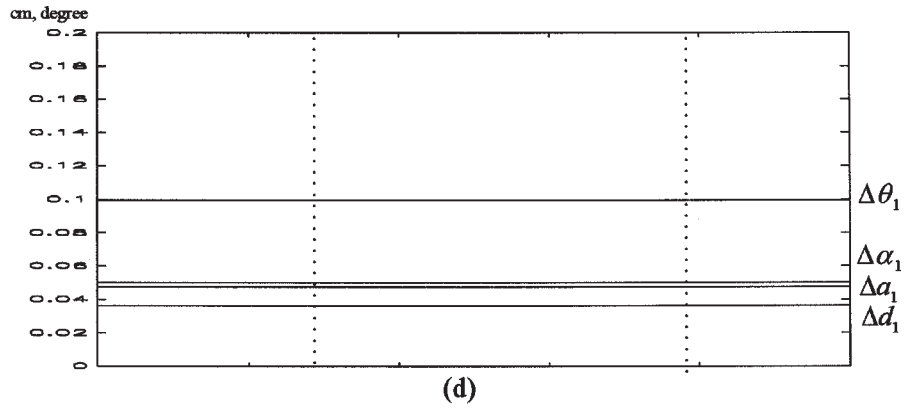


Figure 8 (Continued). Corresponding CEPs  $\Delta\theta_1$ ,  $\Delta d_1$ ,  $\Delta a_1$ ,  $\Delta\alpha_1$ : (d) for case (a), (e) for case (b), and (f) for case (c).

regions. And the proposed calibration scheme with the FCMAC used generalized CEPs generated by the FCMAC. Simulation results obtained using these three kinds of CEPs for calibration are shown in Figures 8(a)–(c), and the corresponding CEPs  $\Delta\theta_1$ ,  $\Delta d_1$ ,  $\Delta a_1$ ,  $\Delta\alpha_1$  in Figures 8(d)–(f), respectively. In Figure 8, the region without measurement data is bounded by two dotted straight lines. Satisfactory performances were obtained using the proposed calibration scheme with and without the FCMAC. This demonstrates that the proposed calibration scheme is capable of generalizing CEPs from neighboring regions when regional measurement data are not available.

Simulations were also performed for several different paths. It was found that the proposed calibration scheme provided more homogeneous compensation, yielding consistently good performance for different simulated paths. On the other hand, the conventional calibration scheme demonstrated different compensation effects for different paths. The results indicate that the FCMAC learning algorithm organizes the sets of CEPs properly, and consequently it effectively exploits the measured data acquired from different regions for compensation. In addition, we also performed simulations including random measurement errors (about 3% of the total error). Results similar to those in Figures 6–8 were obtained except that small random variations were present in the error profile. Finally, simulations were executed to compare the performances of the CMAC and FCMAC learning algorithms in generating smooth CEPs. It was observed that the CEPs generated by the CMAC learning algorithm varied more abruptly when the path crossed regional boundaries. This phenomenon may occur because the data and mapping in the CMAC are quantized. By contrast, the difference is smoothed out by the fuzzy representation in the FCMAC. The results demonstrate that the FCMAC is more suitable than the CMAC for generating smooth CEPs.

## 7. Experiment

Experiments were conducted also using the Mitsubishi RV-M2 type five-axis industrial robot manipulator. Its repeatability is on the order of 0.1 millimeter. A three-axis measurement device (an  $x$ - $y$ - $z$  table) was used to provide accurate Cartesian positions in the  $x$ ,  $y$  and  $z$  directions. The  $x$ - $y$ - $z$  table has three prismatic joints driven by DC servo motors. The movements of these three joints are along the  $x$ ,  $y$  and  $z$  directions with measurement ranges of 800 mm, 560 mm, and 580 mm, respectively. The accuracy for each direction is on the order of 0.02 mm. An analog type touch panel is attached to the end of the  $z$  axis arm of the  $x$ - $y$ - $z$  table. The upper layer of the touch panel is covered by transparent electrodes and the bottom layer is by electrode strips. Its function is to automatically record the coordinates of the contact point with respect to the  $x$ - $y$ - $z$  table, when the manipulator touches the effective area on the touch panel. The effective area of the touch panel is 133 mm  $\times$  80 mm with a resolution of 512  $\times$  256. With this resolution, the accuracy provided by the touch panel is approximately 0.1 mm. Figure 9(a) shows



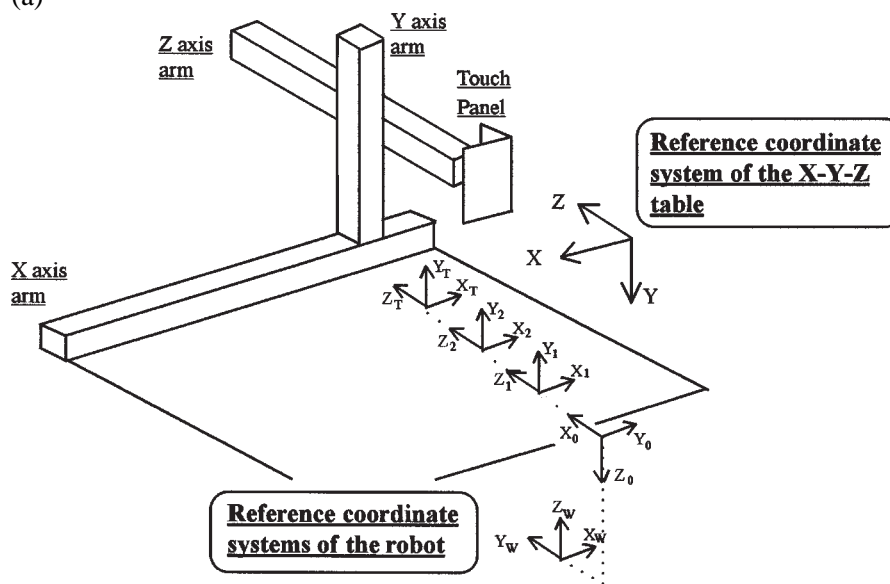
the experimental setup with a Mitsubishi RV-M2 type robot manipulator, an  $x$ - $y$ - $z$  table, and a touch panel. Their corresponding coordinate frame systems are shown in Figure 9(b). These three devices are all connected to a PC-486, which serves as a control unit for coordination. This experimental setup was intended to measure Cartesian position only and not for orientation.

To avoid the tedious and time-consuming procedures involved in collecting a large number of measurement data for calibration, an automatic measurement process was designed. In this design, we first needed to find a feasible and time-efficient measurement sequence. In Figure 9, to avoid contact between the links of the robot manipulator and the arms of the  $x$ - $y$ - $z$  table from the side, the  $z$  axis arm of the  $x$ - $y$ - $z$  table should move from a position far from the pin mounted on the robot manipulator, and then approach the robot manipulator gradually. Therefore, for those measurement locations specified by the measurement space analysis module, the measurement sequence was initially arranged in an ascending order of  $z$  coordinates. Then, to reduce the time needed to reach all the locations in the measurement sequence, the measurement locations were further ordered according to their  $x$  and  $y$  coordinates. In the process of automatic measurement, both the  $x$ - $y$ - $z$  table and the robot manipulator executed the measurement sequence simultaneously via coordination by the PC-486. Because inconsistency was present between the coordinate systems of the  $x$ - $y$ - $z$  table and the robot manipulator before calibration, the robot manipulator could not reach the precise measurement location specified by the  $x$ - $y$ - $z$  table. That is why we needed a touch panel, that provided a small area for deviation. Note that the angle between the touch panel and the pin on the manipulator when in contact had to be greater than a certain figure ( $15^\circ$  in this experiment) for an effective measurement by the touch panel. Consequently, this constraint limited the configurations of the robot manipulator in reaching a specified measurement location.

It can be seen that for the experimental setup in Figure 9, some locations in the robot workspace were not accessible for measurement, i.e., the measurement space was limited due to the structures and kinematic constraints on the robot manipulator and the  $x$ - $y$ - $z$  table. This is why we adopted the FCMAC-type learning algorithm: because it provided proper sets of CEPs for those regions without appropriate measurable locations by using data from neighboring regions. One possible resolution of this measurement problem would be to use laser, infrared rays, ultrasound, or visible-light measuring devices, and place them at locations far from the robot manipulator [28]. The loss of measurement space due to possible collision with the robot manipulator would then be avoided. However, in general it is not easy to design a feasible experimental setup using this type of measurement device for technical and economic reasons, when the demanded accuracy is on the order of 0.1 mm and the measurement space is three-dimensional [10, 27]. We considered that the proposed experimental setup to be simple, inexpensive, and appropriate to this study, although it may not be suitable for general calibration purposes. By using this setup, the measurement data needed for the experiments



(a)



(b)

Figure 9. (a) The experimental setup with a Mitsubishi RV-M2 type robot manipulator, an  $x$ - $y$ - $z$  table, and a touch panel. (b) Corresponding coordinate frame systems for the Mitsubishi RV-M2 type robot manipulator and the  $x$ - $y$ - $z$  table.

were collected in about a couple of hours (approximately five seconds for one accurate measurement).

As in the simulations in Section 6, the angles of joints four and five were fixed at zero. The workspace was divided into 85 local regions of differing dimensions in Cartesian space using the recursive workspace division algorithm. The demanded accuracy after compensation for each local region was within 0.5 mm for the Cartesian position error. As described in Section 6, there were fifteen independent CEPs for calibration from the base to the end-effector. Before execution of the automatic measurement process, the relative distance and orientation between the base coordinate frames of the  $x$ - $y$ - $z$  table and the robot manipulator had to be identified. Because the robot manipulator was fastened to a fixture, it was very difficult to locate the origin of its base coordinate frame precisely. Therefore, we estimated the relationship between the  $x$ - $y$ - $z$  table and the robot manipulator by moving them to a number of randomly chosen locations for measurement. The residual imprecision after estimation was then taken as another error source and would be incorporated into the representative set of CEPs after calibration. Twenty measurement locations in each local region were chosen to identify the representative set of CEPs. The specifications for the FCMAC learning algorithm were similar to those for the simulations in Section 6.

The robot manipulator was commanded to follow a straight-line path from  $(-300 \text{ mm}, 520 \text{ mm}, 150 \text{ mm})$  to  $(350 \text{ mm}, 690 \text{ mm}, 600 \text{ mm})$  in Cartesian space. Figure 10(a) shows the position errors without calibration. The errors ranged between 0.3 and 2.5 mm. The three kinds of calibration for the simulations in Section 6 were performed to compensate for the errors. The same set of measurement data were used to derive the CEPs in these calibrations. Experimental results are shown in Figures 10(b)–(d), and the corresponding CEPs  $\Delta\theta_1$ ,  $\Delta d_1$ ,  $\Delta a_1$ ,  $\Delta\alpha_1$  for these three cases in Figures 10(e)–(g), respectively. In Figure 10(b), the conventional calibration scheme reduced the errors to less than 1.2 mm. Figures 10(c)–(d) show that the proposed calibration scheme, with and without, the FCMAC reduced the errors approximately 50% more than those by the conventional one (below 0.5 mm). Similar to the simulation results in Section 6, the residual errors after compensation varied more evidently around boundary areas using the proposed calibration scheme without the FCMAC. Experiments were also performed for several different paths. The proposed calibration scheme, but not the conventional one, demonstrates consistently good performance.

To summarize, simulation and experimental results under different conditions are listed below.

- The proposed calibration scheme using a variable parameter model performs better than conventional calibration schemes with constant parameter models. Especially the latter can be viewed as a special case of the former, when the entire workspace is taken as a local region.
- The performances of the proposed calibration schemes with and without the FCMAC learning algorithm are similar, except that the FCMAC learning algo-

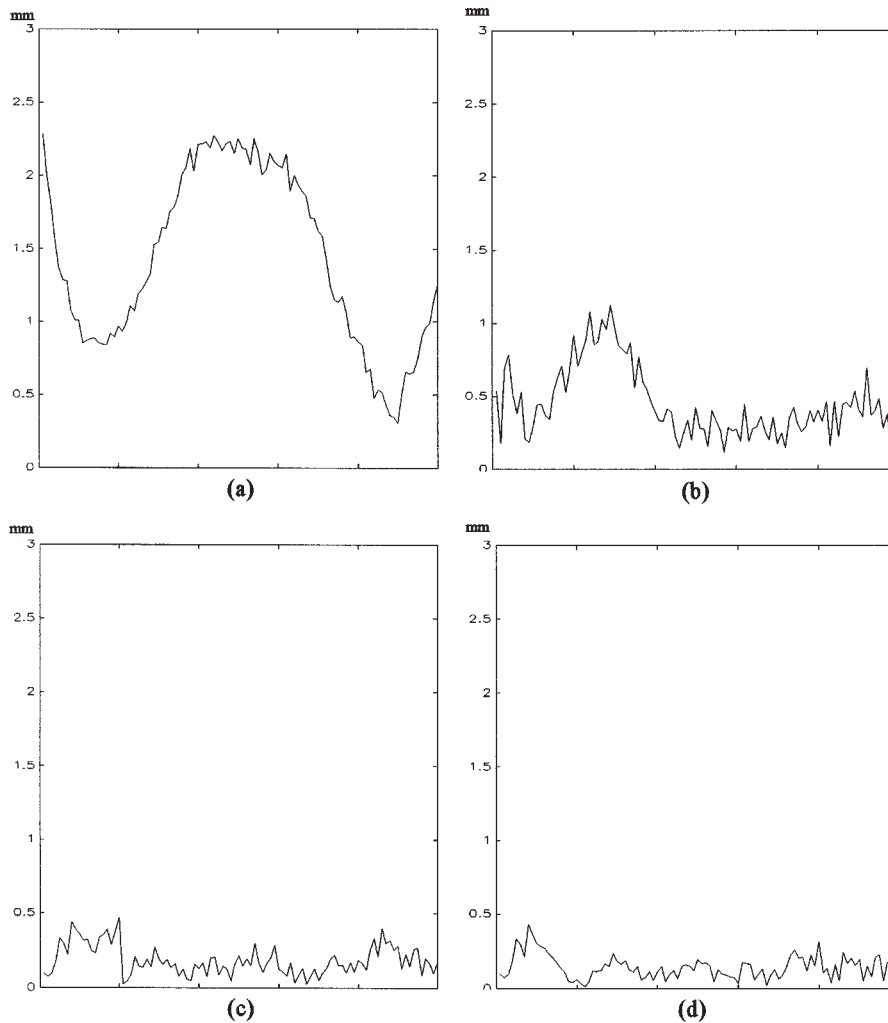


Figure 10. Experimental results. Position errors: (a) without calibration, (b) the conventional calibration scheme, (c) the proposed calibration scheme without the FCMAC, and (d) the proposed calibration scheme with the FCMAC. Note that the  $x$  axis: points on a straight-line Cartesian path,  $(-300 \text{ mm}, 520 \text{ mm}, 150 \text{ mm})$  to  $(350 \text{ mm}, 690 \text{ mm}, 600 \text{ mm})$ .

rithm can smooth out CEPs, consequently more homogeneous compensation effects are obtained across regional boundaries.

- When measurement data are not available for certain local regions, the proposed calibration scheme can derive CEPs from CEPs for neighboring local regions. Satisfactory performances are obtained using the generalized CEPs.
- The proposed calibration scheme demonstrates similar performances for workspace division in either Cartesian or joint space.

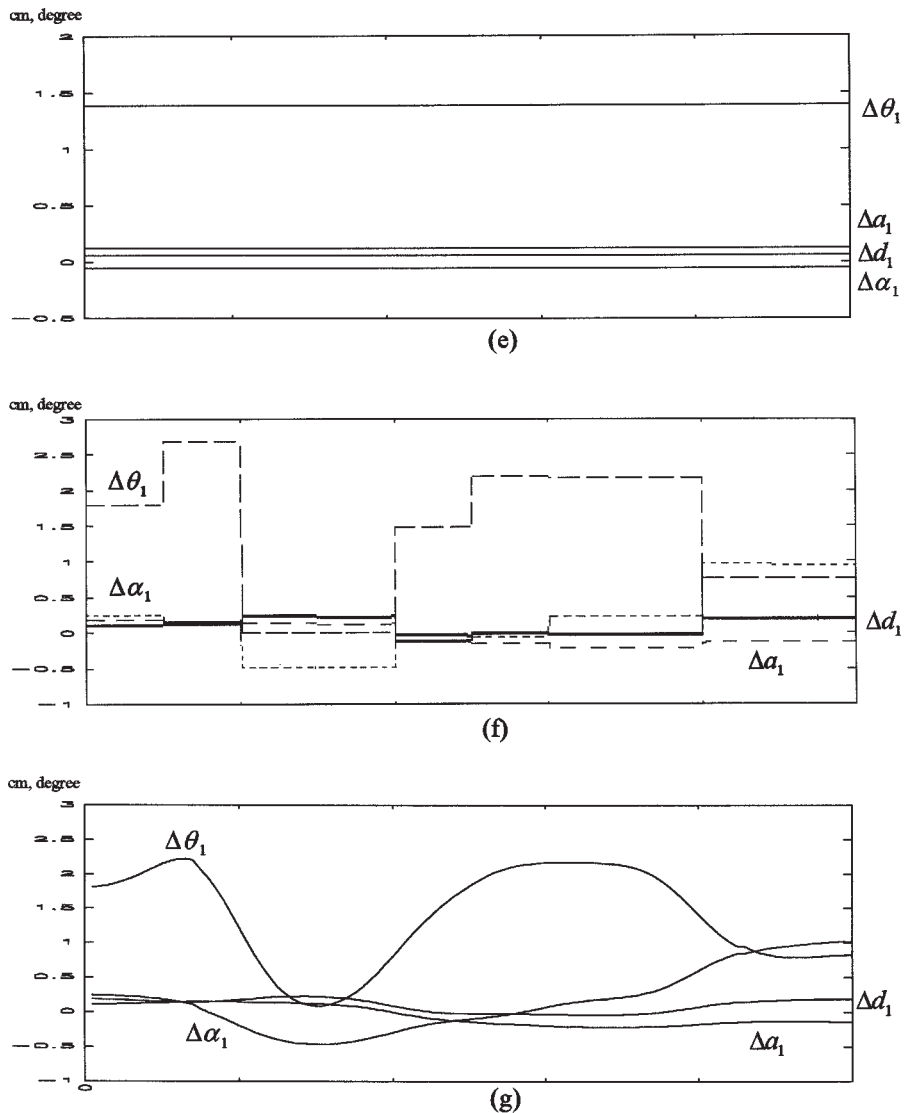


Figure 10 (Continued). Corresponding CEPs  $\Delta\theta_1$ ,  $\Delta d_1$ ,  $\Delta a_1$ ,  $\Delta\alpha_1$ : (e) for case (b), (f) for case (c), and (g) for case (d).

- The proposed calibration scheme demonstrates consistently good performance for different paths; by contrast, the conventional calibration scheme exhibits different compensation effects for different paths.
- The FCMAC is more suitable than the CMAC for generating smooth CEPs due to the smoothing effect of its fuzzy representation.

## 8. Discussion

By employing the proposed calibration scheme in this paper, although the number of local regions seems to be many in both simulations and experiments, the demanded accuracy is achieved in each local region; by contrast, when only one set of CEPs is derived from the measurement data, accuracy is improved on average, but not necessarily appropriate for each local region. It is because that the proposed calibration scheme selects the measurement locations properly and exploits the measurement data well. In some sense, dividing the workspace into local regions can be viewed as a particular way for the arrangement of the measurement data. Then, the number of measurement data and how the CEPs are derived from the data and their effects may be more significant than the number of local regions.

A point that also deserves discussion is the convergence and memory management of the FCMAC network. We believe that the convergence of the FCMAC network should not be a problem in implementing the variable D-H parameter model, because the nonlinearity to be dealt with in learning CEPs is mainly kinematic and consequently not very complicated. Especially only CEP variations between adjacent regions and CEP generalization for unmeasurable regions are of concern. The memory size of the network depends on both the degrees of freedom of the robot manipulator and the measurement space division. Because the proposed workspace division strategy is performed based on mainly the major joints, for most cases the proposed calibration scheme can be applied to general multi-joint robot manipulators without excessive workspace division. Nevertheless, if the memory requirements become too large for some cases or for workspace division including the minor joints when long tools are involved, the hash-coding technique can then be employed at the expense of the network resolution [3]. Or the total number of local regions can be set to a pre-determined value to avoid too many subdivisions at the expense of accuracy.

It can be seen that a minute subdivision of the workspace would be induced when there were error sources that varied very fast and no constraint was imposed on the division. Under a limited number of subdivisions, it indicates that the proposed calibration scheme is most effective for error sources that vary slowly. It is because that the proposed calibration scheme intends to organize CEPs for different local regions instead of organizing all the joint solutions over the entire workspace. An analogy in daily life would be choosing political representatives from individual states to attend a national assembly for policy-making instead of inviting all the people in the country to attend. While the latter is almost impossible, the former may result in the ignorance of some small local matters in policy-making. Thus, the precise implementation of the variable D-H parameter model and the proper division of the workspace stand as a tradeoff in applications.

In this paper, although only position was considered for workspace division in Cartesian space, it is straightforward to apply the proposed calibration scheme to deal with both position and orientation. It requires that orientation be properly

parametrized. For example, the orientation can be described in the form of Euler angles [21]. Consequently, three more parameters  $(\phi, \theta, \psi)$ , representing the three Euler angles, would have to be incorporated into the FCMAC network. The tradeoff in incorporating orientation into the formulation is the expense of memory in the network. In addition, a device for providing accurate orientation measurement of the robot manipulator is demanded.

## 9. Conclusion

In this paper, a variable D-H parameter model and a robot calibration scheme based on it have been proposed to resolve the coordinate equivalence problem in integrating CAD systems and robot manipulators by overcoming the locality exhibited by current calibration schemes. The proposed calibration scheme can be implemented on-line and can achieve high accuracy for arbitrarily planned paths while avoiding direct formulations and identifications of all possible error sources resulting in imprecision. Simulations and experiments, which are with complexity similar to that demonstrated by general non-redundant robot manipulators, verify the feasibility of the proposed calibration scheme. The locality problem is not unique to robot calibration, but is also present in other areas, e.g., camera calibration. The proposed variable parameter model and calibration scheme may provide an effective framework for resolving this general phenomenon.

## Appendix. The FCMAC Learning Algorithm

The basic concept behind the CMAC-type neural network can be represented by a pair of mappings, as shown in Figure 5 [2, 3, 23]:

$$\mathbf{f}: \mathbf{S} \rightarrow \mathbf{A}, \quad (\text{A1})$$

$$\mathbf{g}: \mathbf{A} \rightarrow \mathbf{P}, \quad (\text{A2})$$

where  $\mathbf{S}$  stands for the set of fuzzy sensor vectors representing the input data,  $\mathbf{A}$  stands for the set of association cell vectors, and  $\mathbf{P}$  stands for the set of fuzzy response output vectors. The first mapping maps the input data onto a finite set of intermediate states, called association cells. The mapping is generally a fixed relation, since it is a process of indexing the input data. Each input data  $S_i$  in  $\mathbf{S}$  maps to a number of locations in the  $N$ -dimensional  $\mathbf{A}$ , as shown in the simplified one-dimensional illustration in Figure 5. The number of the same excited units of  $\mathbf{A}$  for two different inputs  $S_i$  and  $S_j$  decreases monotonically as the similarity between  $S_i$  and  $S_j$  decreases. This arrangement will produce generalization between nearby input vectors and no generalization between distant input vectors. In order to reduce the memory size of association cells for a practical mapping, Albus [2, 3] also proposed a procedure for hash-coding. The second mapping depends on the values of weights assigned to every association cell, which will be modified during

the training process. These weights can be adjusted according to the difference between the desired output and the produced output. This mapping then sums up the weights attached to the active association cells to produce the output  $\mathbf{P}$ .

In Figure 5, the implementation of the CMAC neural network is in a fuzzy representation. Each unit of the fuzzy sensor carries a continuous excitation between zero and one, and a typical fuzzy sensor vector can be described using the Gaussian function

$$\mu(S_k) = \exp(-|S_k - m|^2/2\sigma^2), \quad (\text{A3})$$

where  $S_k$  stands for an input variable,  $\mu$  stands for the excitation of the fuzzy sensor, and  $m$  and  $\sigma$  are the mean and standard deviation, respectively. Since the Gaussian function in Equation (A3) yields nonzero values expanding over an almost infinite range, an input variable  $S_k$  will correspond to almost infinite units of fuzzy sensors whose membership grades are nonzero. For the sake of computational efficiency, we select the first  $c$  units of fuzzy sensors with the largest membership grades to represent  $S_k$ . The max-min composition is adopted for the fuzzy AND and OR operations used in the mappings specified by Equations (A1), (A2) [22]. Finally, to produce a single numerical output, the centroid defuzzification method is used for defuzzifying the fuzzy response output  $P(S_k)$ , formulated as follows:

$$P(S_k) = \frac{\sum_c \mu w}{\sum_c \mu}, \quad (\text{A4})$$

where  $w$  stands for the weight [22].

A training process is employed to modify the weights through an updating function using the difference between the set of desired CEPs and that generated by the FCMAC for each training pattern. The updating function is as follows:

$$w_{n+1} = w_n + \beta \cdot \mu \cdot (CEP - CEP'), \quad (\text{A5})$$

where  $n$  denotes the stage of the training process,  $\beta$  is the learning rate,  $\mu$  denotes the strength of excitation, and  $CEP$  and  $CEP'$  are the desired and actual outputs for the training pattern, respectively. The learning process will terminate when the differences between the set of desired CEPs and that generated by the network are within a pre-chosen tolerance  $\varepsilon$ .

## References

1. Ahmad, S.: Analysis of robot drive train errors, their static effects, and their compensation, *IEEE J. Robotics and Automation* **4**(2) (1988), 117–128.
2. Albus, J. S.: A new approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC), *ASME J. Dynamic Systems Meas. Control* **97**(3) (1975), 220–227.
3. Albus, J. S.: Data storage in the Cerebellar Model Articulation Controller (CMAC), *ASME J. Dynamic Systems Meas. Control* **97**(3) (1975), 228–233.



4. Azadivar, F.: The effect of joint position errors of industrial robots on their performance in manufacturing operations, *IEEE J. Robotics Automat.* **3**(2) (1987), 109–114.
5. Bay, J. S.: Autonomous parameter identification by optimal learning control, *IEEE Control Systems Magazine* **13**(3) (1993), 56–61.
6. Borm, J. H. and Menq, C. H.: Determination of optimal measurement configurations for robot calibration based on observability measure, *Int. J. Robotics Research* **10**(1) (1991), 51–63.
7. Chen, J. and Chao, L. M.: Positioning error analysis for robot manipulators with all rotary joints, *IEEE J. Robotics Automat.* **3**(6) (1987), 539–545.
8. Denavit, J. and Hartenberg, R. S.: A kinematic notation for low-pair mechanisms based on matrices, *ASME J. Appl. Mechanics* **77** (1955), 215–221.
9. Driels, M. R. and Pathre, U. S.: Significance of observation strategy on the design of robot calibration experiments, *J. Robotic Systems* **7**(2) (1990), 197–223.
10. Driels, M. R. and Swayze, W. E.: Automated partial pose measurement system for manipulator calibration experiments, *IEEE Trans. Robotics Automat.* **10**(4) (1994), 430–440.
11. Everett, L. J., Driels, M. R., and Mooring, B. M.: Kinematic modeling for robot calibration, in: *IEEE Int. Conf. on Robotics and Automation*, 1987, pp. 183–189.
12. Goswami, A., Quaid, A., and Peshkin, M.: Complete parameter identification of a robot from partial pose information, in: *IEEE Int. Conf. on Robotics and Automation*, 1993, pp. 168–173.
13. Gupta, K. C. and Kazerounian, K.: Improved numerical solutions of inverse kinematics of robots, in: *IEEE Int. Conf. on Robotics and Automation*, 1985, pp. 743–748.
14. Hayati, S. A.: Robot arm geometric link parameter estimation, in: *22nd IEEE Conf. on Decision and Control*, 1983, pp. 1477–1483.
15. Hayati, S. A. and Roston, G. P.: Inverse kinematic solution for near-simple robots and its application to robot calibration, in: M. Jamshidi et al. (eds), *Recent Trends in Robotics: Modeling, Control and Education*, Elsevier, Amsterdam, 1986, pp. 41–50.
16. Hollerbach, J. M. and Bennett, D. J.: Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints, *IEEE Trans. Robotics Automat.* **7**(5) (1991), 597–606.
17. Jou, C. C.: A fuzzy cerebellar model articulation controller, in: *IEEE Int. Conf. on Fuzzy Systems*, 1982, pp. 1171–1178.
18. Judd, R. P. and Knasinski, A. B.: A technique to calibrate industrial robots with experimental verification, *IEEE Trans. Robotics Automat.* **6**(1) (1990), 20–30.
19. Khalil, W., Gautier, M., and Enguehard, Ch.: Identifiable parameters and optimum configurations for robot calibration, *Robotica* **9** (1991), 63–70.
20. Kirchner, H. O., Gurumoorthy, B., and Prinz, F. B.: A perturbation approach to robot calibration, *Int. J. Robotics Research* **6**(4) (1987), 47–59.
21. Latombe, J.-C.: *Robot Motion Planning*, Kluwer Academic, Dordrecht, 1991.
22. Lee, C. C.: Fuzzy logic in control systems: Fuzzy logic controller – Parts I and II, *IEEE Trans. Systems Man Cybernet.* **20**(2) (1990), 404–435.
23. Miller, W. T., Glanz, F. H., and Kraft, L. G.: Application of a general learning algorithm to the control of robotic manipulators, *Int. J. Robotics Research* **6**(2) (1987), 84–98.
24. Mooring, B. W. and Pack, T. J.: Determination and specification of robot repeatability, in: *IEEE Int. Conf. on Robotics and Automation*, 1986, pp. 1017–1023.
25. Nie, J. and Linkens, D. A.: A fuzzified CMAC self-learning controller, in: *IEEE Int. Conf. on Fuzzy Systems*, 1993, pp. 500–505.
26. Orin, D. E. and Tsai, Y. T.: A real-time computer architecture for inverse kinematics, in: *IEEE Int. Conf. on Robotics and Automation*, 1986, pp. 843–850.
27. Renders, J.-M., Rossignol, E., Becquet, M., and Hanus, R.: Kinematic calibration and geometrical parameter identification for robots, *IEEE Trans. Robotics Automat.* **7**(6) (1991), 721–732.

28. Rene Mayer, J. R. and Parker, G. A.: A portable instrument for 3-D dynamic robot measurements using triangulation and laser tracking, *IEEE Trans. Robotics Automat.* **10**(4) (1994), 504–516.
29. Roth, Z. S., Mooring, B. W., and Ravani, B.: An overview of robot calibration, *IEEE J. Robotics Automat.* **3**(5) (1987), 377–385.
30. Shamma, J. S. and Whitney, D. E.: A method for inverse robot calibration, *ASME J. Dynamic Systems Meas. Control* **109** (1987), 36–43.
31. Stone, H. W. and Sanderson, A. C.: A prototype arm signature identification system, in: *IEEE Int. Conf. on Robotics and Automation*, 1987, pp. 175–182.
32. Vaishnav, R. N. and Magrab, E. B.: A general procedure to evaluate robot positioning errors, *Int. J. Robotics Research* **6**(1) (1987), 124–129.
33. Veitschegger, W. K. and Wu, C. H.: Robot calibration and compensation, *IEEE J. Robotics Automat.* **4**(6) (1988), 643–656.
34. Whitney, D. E., Lozinski, C. A., and Rourke, J. M.: Industrial robot forward calibration method and results, *ASME J. Dynamic Systems Meas. Control* **108** (1986), 1–8.
35. Wu, C. H.: A kinematic CAD tool for the design and control of a robot manipulator, *Int. J. Robotics Research* **3**(1) (1984), 58–67.
36. Wu, C. H., Ho, J., and Young, K. Y.: Design of robot accuracy compensator after calibration, in: *IEEE Int. Conf. on Robotics and Automation*, 1988, pp. 780–785.
37. Zhuang, H., Roth, Z. S., and Hamano, F.: A complete and parametrically continuous kinematic model for robot manipulators, *IEEE Trans. Robotics Automat.* **8**(4) (1992), 451–463.