NPL Report CISE 15/98

March 1998

Determining CMM behaviour from measurements of standard artefacts

M G Cox, A B Forbes, P M Harris,
Centre for Information Systems Engineering
and
G N Peggs
Centre for Length Metrology
National Physical Laboratory
Teddington
Middlesex
United Kingdom
TW11 0LW

ABSTRACT

This report is concerned with modelling and estimating the systematic error behaviour of co-ordinate measuring machines (CMMs). We describe two classes of model for these parametric errors, the kinematic models based on an axis-upon-axis build-up of behaviour and more general empirical models that can model any repeatable error behaviour. We then develop a comprehensive mathematical model for determining the parametric errors of a CMM from measurements of a standard artefact such as a ball or hole plate and describe algorithms for determining relatively unbiased and efficient estimates of the parametric errors from such data. These algorithms have been implemented in software, engineered so that they can cater for any parametric error model, ball plate geometry and measurement strategy. The software also calculates standard uncertainties of the fitted parameters and related quantities. We use this software in numerical simulations to analyse the effectiveness of measurement strategies with respect to the statistical accuracy of parametric error estimates. The numerical experiments show that with a well-designed strategy it is possible to provide accurate estimates of the parametric errors using only a minimal amount of calibration information about the ball plate, but that other designs or estimators can give poor and/or biased estimates.

This work was supported by the National Measurement System Policy Unit of the Department of Trade and Industry, UK.

NPL Report CISE 15/98

ISSN 1361-407X

National Physical Laboratory
Teddington, Middlesex, United Kingdom, TW11 0LW

Approved on behalf of Managing Director, NPL
by Dr D Rayner, Head, Centre for Information Systems Engineering

(This page is intentionally blank.)

1 INTRODUCTION

This report is concerned with modelling and estimating the systematic error behaviour of co-ordinate measuring machines (CMMs). In a (conventional) CMM with three, mutually orthogonal linear axes, the position of the probe tip centre is inferred from scale readings on each of the three machine axes. For a CMM with perfect geometry, the scale readings alone are sufficient to provide for accurate co-ordinate measurements. However, in practice, CMMs will have imperfect geometry with respect to the straightness of the axes, the squareness of pairs of axes and also with respect to rotations describing roll, pitch and yaw. These systematic errors - so-called *parametric errors* - have to be taken into account if the accuracy potential of the CMM is to be more fully realised.

In the vast majority of cases, a CMM will undergo a comprehensive and thorough calibration procedure as the final process before delivery to the customer. This procedure will seldom, if ever, be repeated. When a CMM is delivered, an acceptance test is performed by the manufacturer to demonstrate to the user that certain aspects of the CMM's performance are within specification according to a nationally or internationally recognised specification standard (e.g., ISO 10360-2). These specification standards include clauses which require other tests to be performed at regular intervals to ensure that the CMM continues to conform to the specifications. These tests are of two types:

periodic reverification, and
interim checking.

Periodic reverification essentially involves repeating the acceptance test at yearly intervals. The advantage of performing this test is that drift in CMM performance from the initial acceptance testing can be identified, albeit up to a year afterwards. The disadvantage is that the measurement task called for in the standard is generally quite unlike the majority of routine measurements made with a CMM.

Interim checking is carried out between formal periodic reverifications. The purpose of the checks is to provide the user with confidence in the measurements made by the CMM. The complexity of such an interim check will vary greatly between users. In some cases, detailed checking of the CMM will be required, verging on a complete *recalibration* of a CMM, in its fullest sense. In other cases, a very selective test will be required.

There are a number of different levels of checking depending on the degree of confidence with which the CMM needs to be checked (Cox, Forbes and Peggs, 1997). For the purposes of this report, these have been designated as Classes 1 - 4 implying a grading of the comprehensiveness of the tests:

Class 1. Verification of all degrees of freedom of the CMM accounting for any influence of temperature variations;

Class 2. Verification of the entire working volume of the CMM using a standard artefact such as a ball- or hole-plate (a two-dimensional plate with regularly spaced features, usually holes or spheres);

Class 3. Verification of the space envelope of interest using a master artefact representing the workpiece features measured in the user's usual application;

Class 4. Verification of just the working volume of interest using standard artefacts, for example, length bars.

This report focuses on Class 2 checking, i.e., verifying the behaviour of a CMM throughout its entire working volume from measurements of a standard artefact such as a ball- or hole-plate. In Section 2 we discuss models of CMM behaviour, while Sections 3 and 4 describe algorithms and software for determining behaviour from measurements of standard artefacts. In Section 5 we discuss numerical experiments that analyse the behaviour of different error models and in Section 6 we look at the design of Class 2 checking experiments. Section 7 contains our concluding remarks.

2 PARAMETRIC ERROR MODELS

2. EMPIRICAL MODELS OF CMM BEHAVIOUR

At its simplest, the problem is to model the motion of the probe stylus assembly regarded as a rigid body through 3D space, describing its location and orientation as a function of the scale readings $\boldsymbol{x} = (x, y, z)^{\mathrm{T}}$. Our approach is to specify a reference point $\boldsymbol{x}_0$ on the stylus assembly and monitor the location of the point using three functions $e(\boldsymbol{x})$, $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ of the scale readings:

$$\begin{array}{lll} x_0 & x & e(\boldsymbol{x}) \\ y_0 & y & f(\boldsymbol{x}) \\ z_0 & z & g(\boldsymbol{x}) \end{array}$$

Three further functions $u(\boldsymbol{x})$, $v(\boldsymbol{x})$ and $w(\boldsymbol{x})$ are required to monitor the angles of rotation about this point, again as a function of the scale readings, to determine a rotation matrix $R(u, v, w)$. Finally, the true location $\boldsymbol{x}^*$ of a point on the stylus assembly is modelled as

$$\begin{array}{lll} x & x & e(\boldsymbol{x}) \\ y & y & f(\boldsymbol{x}) \\ z & z & g(\boldsymbol{x}) \end{array} + R(u(\boldsymbol{x}), v(\boldsymbol{x}), w(\boldsymbol{x})) \begin{bmatrix} p_x \\ \\ p_z \end{bmatrix}, \qquad (1)$$

where R is a product of three plane rotations

$$R(u,v,w) \quad \begin{bmatrix} \cos w & \sin w & 0 \\ -\sin w & \cos w & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos v & 0 & -\sin v \\ 0 & 1 & 0 \\ \sin v & 0 & \cos v \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos u & \sin u \\ 0 & -\sin u & \cos u \end{bmatrix}, \qquad (2)$$

and $\boldsymbol{p} = (p_x, p_y, p_z)^{\mathrm{T}}$ specifies the location of a specific point on the stylus assembly with respect to the reference point $\boldsymbol{x}_0$. Most often, $\boldsymbol{p}$ will specify the (centre of the) CMM probe stylus tip and will be referred to as the *probe offset*. Different probe assemblies can be accounted for in the model by different probe offsets. We write (1) in briefer notation:

$$\boldsymbol{x}^* = \boldsymbol{x} + \boldsymbol{e}_0 + R\boldsymbol{p},$$

where $\boldsymbol{e}_0$ determines the *positional correction* and $R\boldsymbol{p}$ gives the *rotational correction*. Note that the rotational correction depends explicitly on the probe offset $\boldsymbol{p}$.

From (1), it is seen that the empirical model is defined by six functions: e, f and g specifying the positional correction, and u, v and w specifying the rotational correction, each a mapping from $\mathbb{R}^3 \rightarrow \mathbb{R}$. The model is quite general as it can be used to describe the repeatable behaviour of any rigid body moving in 3D, given three independent sensor readings. However, to make the model totally explicit, the types of function which enter into the model have to be defined.

2.1.1 Basis functions

In the approximation of a function of one variable, $y = h(x)$, a standard approach is to define a set of basis functions $\{\phi_j(x)\}$ and choose coefficients a_j such that

$$\tilde{h} = \sum_j a_j \phi_j(x)$$

is a satisfactory approximation to h. The success of this approach depends on being able to select a set of basis functions which have the flexibility to simulate the behaviour of the function to be approximated, while at the same time giving rise to computational problems that can be solved in a numerically stable manner. In one dimension two types of basis function are frequently used: polynomials, where the basis functions are derived from $\{\phi_j(x) = x^j\}$, and spline functions (see below), where the basis functions are themselves fundamental splines constructed from segments of polynomial curves joined together. Implemented with care, polynomials and splines have proved to be very successful empirical approximating functions.

For the empirical model of CMM behaviour, we would like to find basis functions $\{\phi_i(x)\}$ mapping from $\mathbb{R}^3$ to $\mathbb{R}$ with which to approximate each of the six functions. The common way to define such basis functions is to construct them from univariate basis functions.

Let $\{\phi_i(x)\}$, $\{\psi_j(y)\}$ and $\{\xi_k(z)\}$ be three such sets. Then we can consider:

Sums of basis functions. The model takes the form:

$$h(x, y, z) = \sum_i a_i \phi_i(x) + \sum_j b_j \psi_j(y) + \sum_k c_k \xi_k(z). \qquad (3)$$

The disadvantage of this approach is that the behaviour along any line parallel to a given axis is the same up to a constant, and this limits the behaviour that can be successfully simulated by such functions.

Tensor products. The model takes the form

$$h(x, y, z) = \sum_{i,j,k} a_{i,j,k} \phi_i(x) \psi_j(y) \xi_k(z).$$

The disadvantage of this approach is the large number of parameters required to specify the function; for example, if each set of basis functions has five elements, then 125 parameters are required.

The two methods can be seen as extremes, and an intermediate approach is possible.

Sums of (polynomial-spline) tensor products. The model takes the form:

$$\begin{aligned} h(x, y, z) = & \sum_i (a_{0,i} + a_{1,i} y + a_{2,i} z) \phi_i(x) + \sum_j (b_{0,j} + b_{1,j} x + b_{2,j} z) \psi_j(y) \\ & + \sum_k (c_{0,k} + c_{1,k} x + c_{2,k} y) \xi_k(z), \end{aligned} \qquad (4)$$

where $\{\phi_i(x)\}$, $\{\psi_j(y)\}$ and $\{\xi_k(z)\}$ are sets of basis functions that define, respectively, three (possibly different) classes of univariate function. One choice for these classes of function is a polynomial spline function which we now define more fully.

Polynomial Splines. Suppose $t \in [0, 1]$, and the values λ_i, $i = 1, ..., N$, known as *knots*, satisfying $0 < \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N < 1$, are given. A polynomial spline function $s(t)$ of order n (degree n-1) with these knots may be represented on the interval [0, 1] as a linear combination of so-called *B-spline* basis functions $N_{n,k}(t, \lambda)$, $k = 1, ..., N+n$:

$$s(t) = \sum_{k=1}^{N+n} c_k \, N_{n,k}(t, \lambda).$$

If the knots λ_i are distinct then s and its first $n-2$ derivatives are continuous. If $\lambda_i = \lambda_{i+1}$, then the n-2th derivative can be discontinuous at λ_i. The kth B-spline in this linear combination is itself a spline of order n with knots $\lambda_{k-n}, ..., \lambda_k$, where we define here additional knot values $\lambda_k = 0$, $k < 1$, and $\lambda_k = 1$, $k > N$. Furthermore, it satisfies the important property of *compact support*, *viz* $N_{n,k}(t, \lambda)$ is identically zero outside the interval $[\lambda_{k-n}, \lambda_k]$, a property that we make use of below.

Using B-spline basis functions to define the functions $\{\phi_i(x)\}$, $\{\psi_j(y)\}$ and $\{\xi_k(z)\}$, we obtain the following representation for h:

$$\begin{aligned} h(x, y, z) = & \sum_{i=1}^{N_x+n_x} (a_{0,i} + a_{1,i}y + a_{2,i}z) \, N_{n_x,i}(x, \lambda_x) \\ & + \sum_{j=1}^{N_y+n_y} (b_{0,j} + b_{1,j}x + b_{2,j}z) \, N_{n_y,j}(y, \lambda_y) \\ & + \sum_{k=1}^{N_z+n_z} (c_{0,k} + c_{1,k}x + c_{2,k}y) \, N_{n_z,k}(z, \lambda_z), \end{aligned}$$

where $\phi_i(x)$ is replaced by the ith B-spline basis function of order n_x defined on the N_x knots λ_x, and similarly for $\{\psi_j(y)\}$ and $\{\xi_k(z)\}$.

There are also other (related) ways of representing the polynomial spline $s(t)$. In place of the basis functions

$$N_{n,1}(t), \; N_{n,2}(t), \; ..., \; N_{n,N+n}(t) \; \},$$

we may use, for example,

$$\{ \; 1, \; N_{n,2}(t), \; ..., \; N_{n,N+n}(t) \; \},$$

or, provided $n >$ (as will be the case in our applications below), the functions

$$1, \; t, \; N_{n,2}(t), \; ..., \; N_{n,N+n-1}(t) \; \}.$$

Using this last set of basis functions to define the functions $\{\phi_i(x)\}$, and defining $\{\psi_j(y)\}$ and $\{\xi_k(z)\}$ similarly, it is seen that there is redundancy among the parameters in this representation of h. For example, each of the summations contribute a constant term, and both the first and second summations involve an xy term. It is important when using this type of model to eliminate any such redundancy before applying optimisation software to solve for values of the parameters if problems relating to numerical

stability and uniqueness of solution are to be avoided.

We can do this by isolating those terms from each summation that are linearly dependent, representing h as follows:

$$h(x, y, z) = d_0 + d_1x + d_2y + d_3z + d_4xy + d_5xz + d_6yz$$
$$\sum_{i=2}^{N_x+n_x-1} (a_{0,i} + a_{1,i}y + a_{2,i}z)\, N_{n_x,i}(x, \lambda_x)$$
$$\sum_{j=2}^{N_y+n_y-1} (b_{0,j} + b_{1,j}x + b_{2,j}z)\, N_{n_y,j}(y, \lambda_y)$$
$$\sum_{k=2}^{N_z+n_z-1} (c_{0,k} + c_{1,k}x + c_{2,k}y)\, N_{n_z,k}(z, \lambda_z).$$

Writing h in this way has the further advantage that it is easy to constrain the value of h at certain points. For example, suppose it is required to apply the constraint $h = 0$ at $x = y = z = 0$. By examining all the functions

$$\{1, x, y, z, xy, xz, yz, N_{n_x,i}(x, \lambda_x), N_{n_y,j}(y, \lambda_y), N_{n_z,k}(z, \lambda_z)\},$$

used in the representation of h, and recalling the property of compact support for the B-splines appearing in this list, the unit function is the only basis function that is non-zero at (0, 0, 0). Thus, we set $d_0 = 0$. If, in addition, we wish to apply the constraint $h(1, 0, 0) = 0$, we set $d_1 = 0$. Constraints of this form are important when modelling the components of an error correction field because they correspond to fixing a frame of reference: see Section 2.4.

Radial basis functions. Radial basis functions provide another method of defining basis functions for higher-dimensional approximation. Here the basis functions take the form $\phi_i(\boldsymbol{x}) = \phi(\|\boldsymbol{x} - \boldsymbol{y}_i\|)$ where ϕ is a fixed function from $\mathbb{R}$ to $\mathbb{R}$. A popular choice for ϕ is

$$\phi(x) = \sqrt{(x^2+c^2)}$$

for some constant c. The value of the basis functions thus depends on the distance of the point $\boldsymbol{x}$ to the *centres* $\boldsymbol{y}_i$. Given a suitable distribution of centres it is possible to construct a set of basis functions with satisfactory approximation properties. However, there are at present still some difficulties in implementing numerically stable and efficient approximation algorithms. It should be remembered that early attempts at polynomial and spline approximation were also beset by numerical difficulties until numerically stable algorithms were developed. The simplicity and naturalness of the approach make it likely that models for CMM behaviour using radial basis functions will appear once the numerical problems have been removed.

2.1.3 Stability of the empirical model

The empirical parametric error model is designed to approximate the behaviour of a CMM. We now consider, in general terms, whether the model will lead to numerically stable computations. A model $y = h(\boldsymbol{x}, \boldsymbol{a})$ which gives model values y as a function of variables $\boldsymbol{x}$ and parameters $\boldsymbol{a}$ is a stable model if a small change in the parameter values corresponds to a small change in the model values in the region of interest and *vice versa*. A simple way to examine this type of behaviour is to analyse the rate of change of y_i for a scatter of points $\boldsymbol{x}_i$, $i \in I$, which adequately samples the region. Let J be the associated *Jacobian*

matrix with elements

$$J_{ij} = \frac{\partial h}{\partial a_j}(\boldsymbol{x}_i, \boldsymbol{a}),$$

and let $\boldsymbol{q}$ be a unit vector representing a direction of change in the parameters $\boldsymbol{a}$. If $J\boldsymbol{q}$ is small in norm then changing $\boldsymbol{a}$ in the direction of $\boldsymbol{q}$ will have a correspondingly small effect on the model values y_i. On the other hand, if $J\boldsymbol{q}$ is large (in norm), a change in the direction of $\boldsymbol{q}$ will have a large effect on the model values. Thus, an estimate of

$$\kappa(J) = [\max_{\|\boldsymbol{q}=1\|} \|J\boldsymbol{q}\|]/[\min_{\|\boldsymbol{q}=1\|} \|J\boldsymbol{q}\|]$$

will give a measure of the sensitivity of the model with respect to changes in the model parameters: the closer this value is to 1, the more stable is the parametrization. For stable parametrizations, we expect to be able to calculate model values and find best-fit parameters without any significant loss of accuracy. For unstable parametrizations, significant errors in these calculations are possible. An extreme case occurs when J is not of full rank for in this situation there is a direction of change which will have no effect on the model values and computations involving J can break down.

The quantity κ, defined above for the matrix J, is known as the *condition* of a matrix and can be determined by calculating the singular value decomposition (SVD). For an $m \times n$ matrix A, the SVD is written as

$$\boldsymbol{A} = \boldsymbol{USV}^{\mathrm{T}}$$

where U and V are orthogonal matrices and S is a diagonal matrix with diagonal entries s_i appearing in descending numerical order. The columns $\boldsymbol{v}_j$ of V are known as the right singular vectors and $\boldsymbol{v}_1$ is a direction which will give the maximum change $s_1 = \|J\boldsymbol{v}_1\|$ while $\boldsymbol{v}_n$ is a direction in which there will be minimal change $s_n = \|J\boldsymbol{v}_n\|$. The condition of J is thus given by $\kappa(J) = s_1/s_n$.

For the empirical model (1), we will assume that each of the six components $e(\boldsymbol{x}, \boldsymbol{a})$, $f(\boldsymbol{x}, \boldsymbol{b})$, $g(\boldsymbol{x}, \boldsymbol{c})$, $u(\boldsymbol{x}, \boldsymbol{r})$, $v(\boldsymbol{x}, \boldsymbol{s})$ and $w(\boldsymbol{x}, \boldsymbol{t})$ are determined by linear combinations of basis functions specified by parameters $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{r}$, $\boldsymbol{s}$ and $\boldsymbol{t}$, respectively. We examine the stability of this model by looking at the associated Jacobian matrix for a scatter of points within the working volume of the CMM using three probe offsets $\boldsymbol{p}_l$, $l = 1, 2, 3$. The partial derivatives of $\boldsymbol{x}_i^*$ with respect to the position parameters are straightforward to calculate: they are simply the values of the corresponding basis functions. For the rotational component we note, for example, that

$$\frac{\partial \boldsymbol{x}^*}{\partial r_j} = \begin{bmatrix} \cos w & \sin w & 0 \\ -\sin w & \cos w & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos v & 0 & -\sin v \\ 0 & 1 & 0 \\ \sin v & 0 & \cos v \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\frac{\partial u}{\partial r_j}\sin u & \frac{\partial u}{\partial r_j}\cos u \\ 0 & -\frac{\partial u}{\partial r_j}\cos u & -\frac{\partial u}{\partial r_j}\sin u \end{bmatrix} \boldsymbol{p}$$

and similarly for derivatives with respect to the other rotation parameters. Note that if $u = v = w = 0$, then

$$\frac{\partial x^*}{\partial} = \frac{\partial u}{\partial r_j}\begin{bmatrix} 0 \\ \\ -p_y \end{bmatrix}$$

Let $X_l = \{x_i,\ i \in I_l\}$, $l = 1, 2, 3$, represent the scale readings for the measurements taken with probe offset $\boldsymbol{p}_l$ and let E_l, F_l, etc. be the matrices of basis functions corresponding to e, f, etc., evaluated at these sets of scale readings. Thus, for example,

$$\{E_l\}_{ij} = \frac{\partial e}{\partial a_j}(x_i), \quad i \in I_l$$

Evaluating the Jacobian matrix for the complete set of measurements for the case of all parameters set to zero yields a matrix of the form

$$\begin{bmatrix}
E_1 & 0 & 0 & 0 & -p_{1,z}V_1 & p_{1,y}W_1 \\
0 & F_1 & 0 & p_{1,z}U_1 & 0 & -p_{1,x}W_1 \\
0 & 0 & G_1 & -p_{1,y}U_1 & p_{1,x}V_1 & 0 \\
E_2 & 0 & 0 & 0 & -p_{2,z}V_2 & p_{2,y}W_2 \\
0 & F_2 & 0 & p_{2,z}U_2 & 0 & -p_{2,x}W_2 \\
0 & 0 & G_2 & -p_{2,y}U_2 & p_{2,x}V_2 & 0 \\
E_3 & 0 & 0 & 0 & -p_{3,z}V_3 & p_{3,y}W_3 \\
0 & F_3 & 0 & p_{3,z}U_3 & 0 & -p_{3,x}W_3 \\
0 & 0 & G_3 & -p_{3,y}U_3 & p_{3,x}V_3 & 0
\end{bmatrix} \qquad (5)$$

Here, the columns are ordered to record the derivatives with respect to $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{r}$, $\boldsymbol{s}$ and $\boldsymbol{t}$, while the rows are ordered firstly with respect to probe offset and, within each offset group, with respect to x-, y-, and z-component.

If all six functions are determined from the same set of basis functions and the measurements for each probe offset correspond to exactly the same set of scale readings ($X_1 = X_2 = X_3$), the submatrices are identical, and equal to an $m \times n$ matrix A, say. The Jacobian matrix can then be factored as

$$\begin{bmatrix} I & 0 & 0 & 0 & -p_{1,z}I & p_{1,y}I \\ 0 & I & 0 & p_{1,z}I & 0 & -p_{1,x}I \\ 0 & 0 & I & -p_{1,y}I & p_{1,x}I & 0 \\ I & 0 & 0 & 0 & -p_{2,z}I & p_{2,y}I \\ 0 & I & 0 & p_{2,z}I & 0 & -p_{2,x}I \\ 0 & 0 & I & -p_{2,y}I & p_{2,x}I & 0 \\ I & 0 & 0 & 0 & -p_{3,z}I & p_{3,y}I \\ 0 & I & 0 & p_{3,z}I & 0 & -p_{3,x}I \\ 0 & 0 & I & -p_{3,y}I & p_{3,x}I & 0 \end{bmatrix} \text{diag}\{A\}, \qquad (6)$$

where I is the $m \times m$ identity matrix and $\text{diag}\{A\}$ is a block-diagonal matrix with A repeated along the leading diagonal. This matrix is full rank if and only if i) the matrix A is full rank and ii) the probe offset vectors $\boldsymbol{p}_l$ are not co-linear. The significance of this second condition is that measurements using at least three probe offsets are required if the orientation of the stylus assembly is to be determined and this will only be successful if the centres of the probe stylus tips are distributed so as to uniquely define a frame of reference. If conditions i) and ii) are satisfied, any submatrix of (6) also has full column rank, so that if, for example, the rotation functions are determined by a sub-basis of the position functions then the reduced matrix will still have full rank. The condition of this matrix depends directly on the condition of A and the spatial distribution of the probe offsets. For instance, the condition of the matrix will worsen if the offsets move closer to being co-linear.

The general case (5) involving different sets of basis functions and measured data is more difficult to analyse. However, for simple probe configurations it is easy to derive sufficient conditions for the Jacobian matrix to be full rank. For example, choosing probe offsets $[\lambda\ 0\ 0]^T$, $[0\ \mu\ 0]^T$ and $[0\ 0\ \nu]^T$ produces a matrix of the form

$$\begin{bmatrix} E_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & F_1 & 0 & 0 & 0 & -\lambda W_1 \\ 0 & 0 & G_1 & 0 & \lambda V_1 & 0 \\ E_2 & 0 & 0 & 0 & 0 & \mu W_2 \\ 0 & F_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & G_2 & -\mu U_2 & 0 & 0 \\ E_3 & 0 & 0 & 0 & -\nu V_3 & 0 \\ 0 & F_3 & 0 & \nu U_3 & 0 & 0 \\ 0 & 0 & G_3 & 0 & 0 & 0 \end{bmatrix}.$$

If at least two of λ, μ and ν are non-zero and each submatrix is full rank then the complete matrix is full rank.

The above analysis depends on evaluating the Jacobian matrix with all the error parameters set to zero, corresponding to a CMM with perfect geometry. In practice, the CMM error will not be exactly zero but sufficiently close to zero that the analysis above is a reliable guide to the conditioning of the Jacobian matrices which arise in practice.

In summary, the basic requirements to ensure that the empirical model parameters are well determined by measurement data are that i) the measurements are distributed in such a way that each submatrix of basis function values has full rank, and ii) the probe offsets are not close to being co-linear. In practice this will require that the complete measuring volume will need to be sampled with no large vacant volumes.

2.1.4 Linearisations of the empirical model

If in the definition of the rotation matrix $R(u, v, w)$ in equation (2), we use the approximation $\cos u \doteq 1$ and $\sin u \doteq u$, then R is approximated by

$$\tilde{R} = \begin{bmatrix} 1 & w & -v \\ -w & 1 & u \\ v & -u & 1 \end{bmatrix}, \tag{7}$$

and the location of the probe stylus tip is approximated by

$$\begin{array}{llll} x & x & e(x) & w(x)p_y - v(x)p_z \\ y & y & f(x) & u(x)p_z - w(x)p_x \\ z & z & g(x) & v(x)p_x - u(x)p_y \end{array}$$

We can write this in briefer notation as

$$\boldsymbol{x}^* = \boldsymbol{x} + \boldsymbol{e}(\boldsymbol{x}) + \boldsymbol{p} \times \boldsymbol{u}(\boldsymbol{x}).$$

2.2 KINEMATIC MODELS

The kinematic model of CMM behaviour is based on a superposition of the behaviour of the three axes, each described as a function of the corresponding scale reading. Holding, for example, the y and z scale readings fixed, the behaviour of the stylus assembly can be described by six functions of x, three giving the location of the fixed point on the housing and three specifying the rotation angles that describe the *roll*, *pitch* and *yaw* along the x-axis. The kinematic model describes the behaviour of the CMM in term of these six functions for each of the three axis.

Suppose the position along the x axis is described by

$$\begin{bmatrix} x_x & x + \delta_{xx}(x, \boldsymbol{a}) \\ y_x & \delta_{xy}(x, \boldsymbol{a}) \\ z_x & \delta_{xz}(x, \boldsymbol{a}) \end{bmatrix}$$

and the orientation is described by a rotation matrix $R_x(x, \boldsymbol{r})$ defined as a product of plane rotations as in (2) . The δ_{xx} term is sometimes described as the x-axis *scale error* while δ_{xy} and δ_{xz} give the x-axis *straightness errors*. Similarly the location and orientation along the y-axis is described by

$$\begin{bmatrix} x_y & \delta_{yx}(y, \boldsymbol{b}) \\ y_y & y + \delta_{yy}(y, \boldsymbol{b}) \\ z_y & \delta_{yz}(y, \boldsymbol{b}) \end{bmatrix}$$

and $R_y(y, \boldsymbol{s})$. This information can be combined to provide an estimate of location as a function of x and y:

$$\boldsymbol{x}_{xy} = \boldsymbol{x}_x(x, \boldsymbol{a}) + R_x(x, \boldsymbol{r})\boldsymbol{x}_y(y, \boldsymbol{b})$$

This formulation supposes that the y-motion depends on (or is carried by) the x-motion and moreover that at $x = y = 0$ the axes are orthogonal and the two rotation matrices R_x and R_y are aligned. Adding the z-axis (assumed to be carried by the y-axis) and the probe offset under similar assumptions, the model takes the form

$$\boldsymbol{x} = \boldsymbol{x}_x(x, \boldsymbol{a}) + R_x(x, \boldsymbol{r})\boldsymbol{x}_y(y, \boldsymbol{b}) + R_x(x, \boldsymbol{r})R_y(y, \boldsymbol{s})\boldsymbol{x}_z(z, \boldsymbol{c}) + R_x(x, \boldsymbol{r})R_y(y, \boldsymbol{s})R_z(z, \boldsymbol{t})\boldsymbol{p}, \qquad (8)$$

or, showing the axis-upon-axis build-up,

$$\boldsymbol{x} = \boldsymbol{x} + R_x(\boldsymbol{x}_y + R_y(\boldsymbol{x}_z + R_z\boldsymbol{p}))$$

Thus the kinematic model is specified in terms of 18 individual error functions, three positional and three rotational for each of the three axes. The basic components are therefore functions of a single variable and these can be specified using polynomials or splines, for example. The correct formulation depends on the architecture of the CMM, but the common CMM designs are covered by the model; see, for example, Zhang *et al.* (1988).

If all three rotation matrices are set to the identity matrix, then x^* is given by

$$x = x + p_x + [\delta_{xx}(x) + \delta_{yx}(y) + \delta_{zx}(z)],$$

and similarly for y^* and z^*, showing that the positional correction for x includes a sum of functions of x, y and z. If each of these functions includes a constant term, there will be degrees of freedom in the model which cannot be determined from the data. The simplest way to resolve this is to constrain $\delta_{yx}(0) = \delta_{zx}(0) = 0$. In fact similar constraints have to be placed on all six types of component as will become clear when we consider a linearised version of the kinematic model below.

2.2.1 *Stability of the kinematic model*

If we evaluate the kinematic model at points along each axis we obtain:

$$\begin{aligned} \boldsymbol{x}^*([x,0,0]^T) &= \boldsymbol{x}_x(x,\boldsymbol{a}) + R_x(x,\boldsymbol{r})\,\boldsymbol{p}, \\ \boldsymbol{x}^*([0,y,0]^T) &= \boldsymbol{x}_y(y,\boldsymbol{b}) + R_y(y,\boldsymbol{s})\,\boldsymbol{p}, \\ \boldsymbol{x}^*([0,0,z]^T) &= \boldsymbol{x}_z(z,\boldsymbol{c}) + R_z(z,\boldsymbol{t})\,\boldsymbol{p}, \end{aligned}$$

showing that each of the three sets of parameters can be determined by measurements along the three axes. Thus, a measurement strategy which includes measurements along or parallel to each of the three axes using three or more probe offsets will be sufficient to determine the model parameters, assuming that there are enough measurements along each axis.

2.2.2 *Linearisations of the kinematic model*

Often the kinematic model is used not as given above but in a simpler, linearised form. Employing the same type of linearisation as indicated in Section 2.1.4 we have, for example:

$$\tilde{R}_{xyz} = R_x R_y R_z = \begin{bmatrix} 1 & (w_x+w_y+w_z) & -(v_x+v_y+v_z) \\ -(w_x+w_y+w_z) & 1 & (u_x+u_y+u_z) \\ (v_x+v_y+v_z) & -(u_x+u_y+u_z) & \end{bmatrix} \tag{9}$$

Continuing in this way:

$$R_x \boldsymbol{x}_y = \begin{bmatrix} \delta_{yx}(y)+w_x(x)y \\ \delta_{yy}(y)+y \\ \delta_{yz}(y)-u_x(x)y \end{bmatrix},$$

$$R_x R_y \boldsymbol{x}_z = \begin{bmatrix} \delta_{zx}(z)-(v_x(x)+v_y(y))z \\ \delta_{zy}(z)+(u_x(x)+u_y(y))z \\ \delta_{zz}(y)+z \end{bmatrix}$$

With these linearisations, the expressions for the location of the probe stylus tip (8) become:

$$\begin{aligned} x &= x + p_x + [\delta_{xx}(x)+\delta_{yx}(y)+\delta_{zx}(z)] + w_x(x)y - (v_x(x)+v_y(y))z \\ &\quad + (w_x(x)+w_y(y)+w_z(z))p_y - (v_x(x)+v_y(y)+v_z(z))p_z, \\ y &= y + p_y + [\delta_{xy}(x)+\delta_{yy}(y)+\delta_{zy}(z)] + (u_x(x)+u_y(y))z \\ &\quad - (w_x(x)+w_y(y)+w_z(z))p_x + (u_x(x)+u_y(y)+u_z(z))p_z, \\ z &= z + p_z + [\delta_{xz}(x)+\delta_{yz}(y)+\delta_{zz}(z)] - u_x(x)y \\ &\quad + (v_x(x)+v_y(y)+v_z(z))p_x - (u_x(x)+u_y(y)+u_z(z))p_y. \end{aligned} \tag{10}$$

This formulation is based on particular dependencies of the axes. For other arrangements, similar expressions can be derived (see, for example, Zhang *et al.*, 1988).

Examining the simplified rotation matrix (9), we see that the non-constant terms are sums of three functions and it is necessary to ensure that the constant terms do not appear in these sums more than once. For example, we can constrain $u_y = u_z = v_x = v_z = w_x = w_y = 0$ at 0. These types of constraint have also to be applied in the general kinematic model; see Section 2.4.

2.3 RELATIONSHIP BETWEEN THE EMPIRICAL AND KINEMATIC MODELS

In some sense, the kinematic model is a special case of the empirical model. Writing the kinematic model $\boldsymbol{x}_K^*$ as

$$\boldsymbol{x}_K^* = [\boldsymbol{x}_x(x, \boldsymbol{a}) + R_x(x, \boldsymbol{r})\,\boldsymbol{x}_y(y, \boldsymbol{b}) + R_x(x, \boldsymbol{r})R_y(y, \boldsymbol{s})\boldsymbol{x}_z(z, \boldsymbol{c})] + R_x(x, \boldsymbol{r})R_y(y, \boldsymbol{s})R_z(z, \boldsymbol{t})\,\boldsymbol{p},$$

the kinematic model can be seen to involve a positional correction depending on all three scale variables and a rotational correction, also depending on the three scale variables. Given choices for the 18 kinematic error components it is possible to determine expressions for the six empirical functions which would provide exactly the same model values. However, these six functions would be complicated in that each component would depend, in general, on all the model parameters in a way considerably more involved than the empirical models we have considered so far (in which each component is a linear combination of basis functions).

Another approach is to compare the linearised empirical model with the linearised kinematic model. If we compare the two rotational components $\tilde{R}$ and $\tilde{R}_{xyz}$ given by equations (7) and (10), respectively, we note that each non-constant element of the matrix $\tilde{R}_{xyz}$ is a sum of functions in each of the three scale variables. Thus, the empirical model can have qualitatively the same behaviour as the kinematic model if the component functions u, v and w are drawn from sums of basis functions as in (4). Considering now the positional component, the positional correction for the linearised kinematic model is expressed in terms of sums of functions of each of the three scale variables or products of such functions with one of x, y and z. Thus, the empirical model can have qualitatively the same behaviour as the kinematic model if the component functions e, f and g are drawn from sums of tensor products where one factor is given by a linear polynomial as considered above in (4). Taking these two features together, we can say that given any choice of basis functions for the 18 kinematic error functions, there is an associated set of basis functions for the six empirical error functions which will give the empirical model essentially (i.e to first order) the same behaviour as the kinematic model. In this more usable sense, the kinematic model is a special case of the empirical model.

Given an empirical model, we can determine function values for a kinematic model by evaluating along the three scale axes. Thus, we can easily give values for a kinematic model which agrees with the empirical model along the axes (or on any three lines parallel to the axes). There are, however, important qualitative differences between the empirical and kinematic models. For example, for a kinematic model, roll, pitch and yaw have a direct effect on the positional component of the error function while for an empirical model this need not be so. Consider the linearised empirical model with zero positional component and rotational component signifying linear roll along each axis defined by

$$R(x,y,z) = \begin{bmatrix} 1 & \lambda z & -\lambda y \\ -\lambda z & 1 & \lambda x \\ \lambda y & -\lambda x & 1 \end{bmatrix},$$

with $|\lambda| \ll 1$ For this type of error function $\boldsymbol{x}' = \boldsymbol{x} + \lambda\boldsymbol{x} \times \boldsymbol{p}$. Since $\boldsymbol{x} \times \boldsymbol{p}$ is orthogonal to $\boldsymbol{x}$, we find that to first order,

$$\|\boldsymbol{x}_i' - \boldsymbol{x}_j'\| = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|,$$

showing that this type of error function preserves distances (to first order). This means, for instance, that it is impossible to determine all the parameters of an empirical error model from distance information alone (involving the same probe offset). The situation is different for the kinematic model since the axis roll also affects the positional component of the error. However, there is still a requirement to use more than one probe offset. For example, for the kinematic model described above measurements with a single probe offset parallel to the z-axis will not be able to determine the z-axis roll.

2.4 APPLYING CONSTRAINTS

2.4.1 Frame of reference constraints

So far we have not attempted to specify a frame of reference for the corrected point $\boldsymbol{x}^*$. The natural way to do this is to align the corrected points with the frame of reference used to express the scale readings $\boldsymbol{x}$. For the empirical model we can apply the following six constraints:

$$\begin{aligned} e(0,0,0,\boldsymbol{a}) &= 0, & f(1,0,0,\boldsymbol{b}) &= 0, \\ f(0,0,0,\boldsymbol{b}) &= 0, & g(1,0,0,\boldsymbol{c}) &= 0, \\ g(0,0,0,\boldsymbol{c}) &= 0, & g(0,1,0,\boldsymbol{c}) &= 0. \end{aligned}$$

These constraints force the two x-axes and the two xy-planes to be aligned. For the kinematic model we can apply the same type of constraint with some slight variations. The first three constraints can be implemented by setting $\delta_{xx}(0) = \delta_{yy}(0) = \delta_{zz}(0) = 0$ (i.e. at one end of the axis) so that all nine δ functions are zero at zero. The implementation of the remaining three constraints depends on how the orthogonality of the axes is treated. We have seen that the kinematic model is built up from the behaviour along the three axes. One approach requires the straightness functions δ_{xy}, δ_{xz}, δ_{yx}, δ_{yz}, δ_{zx} and δ_{zy} to be zero also at the other end of the axis (say at 1). This imposes six further constraints. The angles between the three axes are then described by three additional *squareness* parameters adding three degrees of freedom so that the total number of degrees of freedom has been reduced by three. (The eighteen error functions along with the three squareness parameters are often referred to as the 21 parameter model of kinematic behaviour). The other approach is to constrain, for example,

$$\delta_{xy}(1) = \delta_{xz}(1) = \delta_{yz}(1) = 0,$$

and leave the other three straightness functions unconstrained at 1. In this situation the three corresponding squareness parameters are the angles between the following three pairs of vectors:

$$(1,\ 0,\ 0),\quad (\delta_{yx},1,\ 0),$$
$$(\delta_{zx},\delta_{zy},1),\quad (1,\ 0,\ 0),$$
$$(\delta_{zx},\delta_{zy},1),\quad (0,\ 1,\ 0),$$

with all functions evaluated at

Orientation constraints and probe qualification

The constraints discussed in the previous section were concerned with aligning the axes used to record the scale readings and those used to record the location of the CMM probe stylus tip. One further set of constraints has to be considered in order to fix an initial orientation of the stylus assembly. Recall that the rotational correction is given by $R(u,v,w)\boldsymbol{p}$ for a given probe offset $\boldsymbol{p}$. If Q is another rotation matrix then

$$R\boldsymbol{p} = (RQ)(Q^{\mathrm{T}}\boldsymbol{p}),$$

which shows that there is a need to establish either an initial orientation for the rotation matrix or a fixed frame of reference to describe the probe offsets. We will consider two approaches which are used in practice.

Probe qualification using the CMM

In this approach the rotation matrix is constrained to be the identity matrix at the origin (or some other fixed point) of the frame of reference of the scale readings, and the probe offsets are related to measurements of the centre co-ordinates of a small sphere fixed in the CMM's working volume. Suppose that the true location of the sphere centre is given by $\boldsymbol{z}^*$ and the unknown probe offsets are $\boldsymbol{p}_l$. Then:

$$\boldsymbol{z} = \boldsymbol{x}_l + \boldsymbol{e}_0(\boldsymbol{x}_l) + R(\boldsymbol{x}_l)\boldsymbol{p}_l$$

This shows that if the error components are known then the probe offsets are related to the corresponding scale readings. However, in general $\boldsymbol{z}^*$ is unknown and so there are three degrees of freedom in the system of equations. This is generally resolved by assigning one of the probe offsets, $\boldsymbol{p}_1$ say, (usually a master probe that exists for this very purpose) and then relating the other probe offsets to $\boldsymbol{p}_1$:

$$\boldsymbol{x}_1 + \boldsymbol{e}_0(\boldsymbol{x}_1) + R(\boldsymbol{x}_1)\boldsymbol{p}_1 = \boldsymbol{x}_l + \boldsymbol{e}_0(\boldsymbol{x}_l) + R(\boldsymbol{x}_l)\boldsymbol{p}_l$$

If the error behaviour is known, then the probe offsets can be assigned. If the error behaviour is still to be determined then it is safest to regard the probe offsets as extra parameters and the measurements of the reference sphere as extra observations that can be used to determine the probe offsets along with the other model parameters. The effect of assigning $\boldsymbol{p}_1$ (or $\boldsymbol{z}^*$) is to specify the reference point on the stylus assembly about which the rotations are calculated. Without this type of assignment, there will be three unresolvable degrees of freedom in the model.

Pre-calibrated probe configurations

In some situations it is possible to pre-calibrate the probe offsets in a fixed frame of reference. For example the geometry of a star-probe might be known accurately. In this case it is not appropriate to constrain the rotation matrix at any point as the model would not be able to account properly for any difference in

orientation of the CMM axes and the axes employed to describe the offsets. Suppose that $\boldsymbol{q}_l$ denote the pre-calibrated probe offsets. Then the model has the form:

$$\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{e}_0(\boldsymbol{x}) + R(\boldsymbol{x})(\boldsymbol{q}_l - \boldsymbol{q}_0),$$

where $\boldsymbol{q}_0$ is assigned to specify the point (in the pre-calibration frame of reference) about which the rotation takes place (analogous to assigning the master probe offset).

2.5 SUMMARY OF MODELS

In summary, the main features of the empirical models are:

- They depend on six component functions, each a function of three variables.
- They are capable of describing quite arbitrary behaviour, given suitable basis functions. In particular, they can be used to describe the error behaviour of CMMs that are already (partially) corrected for kinematic errors.
- The same basic model can apply to different designs and are not limited to CMMs.
- They can mimic kinematic models if required.
- Suitable sets of basis functions are not so easy to generate as for the kinematic models.

The main features of the kinematic models are:

- They depend on 18 component functions, each a function of one variable.
- Suitable sets of basis functions can be generated easily.
- They cannot describe truly three-dimensional behaviour.
- They are to some extent design-specific and do not readily apply to other co-ordinate measuring systems.

2.6 NOTES AND REFERENCES

The most obvious basis for polynomials are the monomials $\psi_j(x) = x^j$. However, for polynomial approximation with polynomials of even modest degree, this set of basis functions can lead to severe numerical difficulties, especially if the range of x is far from zero. Many of the difficulties are overcome by choosing other bases in which the basis functions satisfy some orthogonality conditions (hence the term *orthogonal polynomials*). Of these, Chebyshev polynomials have been particularly successful and widely employed. See, for example, Forsythe (1957), Clenshaw and Hayes (1965).

The representation for $s(t)$ in terms of B-splines is described in, for example, Cox (1972, 1982) and de Boor (1972, 1978), where numerically stable algorithms for manipulating splines written in this way are also presented. Algorithms for polynomial and spline approximation are implemented in DASL, NPL's *Data Approximation Subroutine Library* (Anthony and Cox (1987)). The library includes subroutines for evaluating spline functions and corresponding basis (Jacobian) matrices.

For more on radial basis functions see, for example, Buhmann (1993), Powell (1993).

The concepts relating to the numerical stability of parametrizations are set out in Anthony *et al.* (1991), Forbes (1996).

The singular value decomposition is one of the most important elements in numerical linear algebra. See, for example, Golub and Van Loan (1996), Golub (1996).

The kinematic model, usually in a linearised form, has been described by a large number of authors: see for example

Cresto (1994), Kruth *et al.* (1994), Kunzmann and Wäldele (1986), Soons *et al.*, (1992), Trapet and Wäldele (1989), Zhang *et al.*, (1988). The linearised version described above is essentially the same as that given in Cresto (1994). The problem of applying frame of reference constraints and probe qualification is also considered in Balsamo (1995).

3 MATHEMATICAL MODELLING OF A CLASS 2 CHECK

3 MEASUREMENT INFORMATION

For the Class 2 checking of a CMM, a standard artefact, such as a ball plate, is placed in a number of positions within the working volume of the CMM and the scale readings corresponding to measurements using various probes of the balls on the plate are recorded. Suppose x_i are the scale readings corresponding to an estimate of the centre of the jth ball in the kth position of the plate using the lth probe, determined by finding the best-fit sphere to the CMM measurements. This measurement information is described by the (three) model equations of the form

$$x_i + e(x_i, a, p_l) = T(y_j, t_k) + \epsilon_i,$$

where

$\{x_i\}_1^{m_X}$	represent the co-ordinates of the measurements of the ball centres in various positions,
$\{y_j\}_1^{n_Y}$	are the co-ordinates in a fixed frame of reference of the ball centres for some initial positioning of the plate,
$\{t_k\}_1^{n_T}$	are parameters describing the transformations $T(y, t_k)$ which transform points $\{y_j\}$ in the initial position of the plate to the kth position,
$\{p_l\}_1^{n_P}$	are the probe offsets,
$e(x, a, p)$	is the error correction, depending on scale readings x, parameters a and the probe offset p, and
$\{\epsilon_i\}_1^{m_X}$	are the random errors $\epsilon_{i,1}$, $\epsilon_{i,2}$ and $\epsilon_{i,3}$ corresponding to the measurement of each co-ordinate of the centre of the ball.

Associated to the ith measurement x_i is a triple $(j(i), k(i), l(i))$ indicating that the measurement relates to the $j(i)$th ball in the $k(i)$th position of the ball-plate using the $l(i)$th probe.

3.2 CALIBRATION INFORMATION

There are two types of calibration information for the Class 2 checking of a CMM that are commonly used. The first relates to the ball plate used to obtain the measurement information. The calibration information involves specifying either the *distances* between the centres of the balls on the plate, or the *locations* of the ball centres, or a combination of these. The second type of calibration information relates to the use of a different artefact, such as a length bar, that is measured separately using the CMM. In either case, the specified calibration information may be regarded either as *exact* or as additional *measurements* that are subject to measurement error.

3.2.1 Calibration information relating to the ball plate

The calibration information is described by model equations that involve the co-ordinates in a fixed frame

of reference of the centres of the balls for some initial positioning of the plate. We describe below the form of these equations when (a) the distances between the centres of the balls on the plate are specified, and (b) the locations of the ball centres are given.

Distance calibration information. Here, the distances between the centres of pairs of balls on the plate are specified. If $y_i = (y_{i,1}, y_{i,2}, y_{i,3})^T$ denotes the co-ordinates in a fixed frame of reference of the centre of the ith ball for some initial positioning of the plate, and D_{ij} is the calibrated distance between the centres of the ith and jth balls, this information is represented by the model equation

$$d_3(y_i, y_j) = D_{ij} + \delta_{ij},$$

where $d_3(y_i, y_j)$ denotes the 3-dimensional Euclidean distance

$$\sqrt{(y_{i,1} - y_{j,1})^2 + (y_{i,2} - y_{j,2})^2 + (y_{i,3} - y_{j,3})^2}$$

between y_i and y_j, and δ_{ij} denotes measurement error for the calibrated distance. Sometimes, the ball plate may be calibrated by measuring the distances between the ball centres in a plane parallel to that of the plate. In this case, if the ball plate is parallel to the xy-plane in its initial position, the calibration information is represented by the model equation

$$d_2(y_i, y_j) = D_{ij} + \delta_{ij},$$

where $d_2(y_i, y_j)$ denotes the 2-dimensional Euclidean distance

$$\sqrt{(y_{i,1} - y_{j,1})^2 + (y_{i,2} - y_{j,2})^2}.$$

In practice, there are limitations on the amount of distance information that is available: it may not be feasible to prescribe the distances between the centres of *all* pairs of balls, and there may be little benefit in doing so. An important aspect of the design of an experiment for the Class 2 checking of a CMM is to decide the amount of calibration information that is necessary to determine accurately the errors associated with the CMM.

Co-ordinate calibration information. Here, the co-ordinates in a fixed frame of reference of the centres of the balls are specified in some initial frame of reference. If $\boldsymbol{Y}_i$ is the calibrated location of the centre of the ith ball, this information is represented by the (three) model equations

$$\boldsymbol{y}_i = \boldsymbol{Y}_i + \boldsymbol{\gamma}_i,$$

where $\boldsymbol{\gamma}_i$ are the measurement errors associated with the calibrated co-ordinate values. Again, the artefact may sometimes be calibrated as a 2-dimensional artefact, in which case only $Y_{i,1}$ and $Y_{i,2}$ are prescribed. Notice that each calibrated ball gives rise to two or three equations corresponding to the calibration of two or three co-ordinates of the ball centre. By associating suitable weights to each equation, reflecting the relative sizes of the components $\gamma_{i,1}$, $\gamma_{i,2}$ and $\gamma_{i,3}$ of $\boldsymbol{\gamma}_i$, we can take account of the variation in accuracy of the calibration information for each co-ordinate of each calibrated ball. This is important where, for example, the calibration of $Y_{i,1}$ and $Y_{i,2}$ is performed to a higher accuracy than that of $Y_{i,3}$.

3.2.2 Calibration information relating to a different artefact

Here, calibration information is provided about a standard artefact that is different from the ball plate used in the Class 2 experiment, and this artefact is then measured by the CMM. For example, suppose z_i and

it is not possible to derive reliable estimates of the standard uncertainties of the fitted parameters or any related quantities. In our approach, the observation equations are uncorrelated (according to the model) and standard uncertainties can be produced in accordance with the ISO Guide (ISO, 1993).

The models involving distances alone use only the fact that the distances between pairs of points are constant in all positions of the ball plate. Our approach uses the fact that the relative location of the points is constant in all positions, generally a more powerful statement. For example, for a nominally planar ball plate, specifying the distances between balls does not lead to a rigid body: an internal ball can move out of the plane without changing the distances to first order. (There is an argument that relying on distances alone means that the model can cope better with changes in ball plate geometry due to deflection under gravity. We believe that such effects are better dealt with by other means.)

Related to the previous point, the optimisation process provides a calibration of the ball plate only in terms of the inter-ball distances. Our approach directly determines a calibration of the ball plate as a three-dimensional artefact.

3.5 ALGORITHM DESIGN

3.5.1 Gauss-Newton algorithm

We give here a brief description of the Gauss-Newton algorithm for solving a non-linear least-squares problem. Suppose we wish to minimise the function

$$E(\boldsymbol{a}) = \sum_{i=1}^{m} d_i^2(\boldsymbol{a}),$$

with respect to the n parameters $a_1, a_2, \ldots, a_n$, where $m \geq n$. Let $\boldsymbol{a}^e$ be an estimate of the solution $\boldsymbol{a}^*$. First, we solve the *linear* least-squares problem

$$\boldsymbol{J}\boldsymbol{p} = -\boldsymbol{d},$$

where J is the $m \times n$ *Jacobian* matrix whose ith row is the gradient of d_i with respect to the parameters $\boldsymbol{a}$ evaluated at $\boldsymbol{a}^e$, i.e.

$$J_{ij} = \frac{\partial d_i}{\partial a_j}(\boldsymbol{a}^e),$$

and $\boldsymbol{d}$ is a vector whose ith component is $d_i(\boldsymbol{a}^e)$. Then, we update our estimate of $\boldsymbol{a}^*$ using

$$\boldsymbol{a}^e := \boldsymbol{a}^e + \boldsymbol{p}$$

These steps are repeated until the estimates are judged to have converged. For convergence, we require that the following quantities are sufficiently small: (a) the change in the sum of squares E, (b) the size of the update $\boldsymbol{p}$ measured, for example, by its Euclidean length $\|\boldsymbol{p}\|$, and (c) the partial derivatives of E with respect to the parameters $\boldsymbol{a}$ measured, for example, by $\|\boldsymbol{g}\|$, where $\boldsymbol{g} = J^{\mathrm{T}}\boldsymbol{d}$. Convergence is not guaranteed: for example, if $\boldsymbol{a}^e$ is a poor estimate of the solution, the iterates generated by the algorithm may diverge. Furthermore, convergence may be slow, as can happen when the quantities $d_i(\boldsymbol{a}^e)$ are large or if J is poorly conditioned. For this reason, the Gauss-Newton algorithm is modified to improve its performance in these situations.

3.4 OTHER ESTIMATORS

Cresto (1994) proposes the following computational problem for the determination of the parameters $\boldsymbol{a}$ that define the error correction function for a CMM. He makes use of the fact that the distance between the centres of a pair of balls is invariant with respect to the positioning of the ball-plate within the working volume of the CMM. If we use $\boldsymbol{x}_{i,k}$ to denote the scale readings recorded by the machine for its measurement of the ith ball centre in the kth position of the plate, this invariance property gives the condition

$$d_3(\boldsymbol{x}^*_{i,k}, \boldsymbol{x}^*_{j,k}) - d_3(\boldsymbol{x}^*_{i,l}, \boldsymbol{x}^*_{j,l}) = 0,$$

where

$$\boldsymbol{x}^*_{i,k} = \boldsymbol{x}_{i,k} + \boldsymbol{e}(\boldsymbol{x}_{i,k}, \boldsymbol{a}, \boldsymbol{p})$$

are the true co-ordinates of the centre of the ith ball in the kth position of the plate. These conditions suggest the following estimation problem: find parameters $\boldsymbol{a}$ that minimise

$$\sum \left(d_3(\boldsymbol{x}^*_{i,k}, \boldsymbol{x}^*_{j,k}) - d_3(\boldsymbol{x}^*_{i,l}, \boldsymbol{x}^*_{j,l})\right)^2$$

where the summation is taken over (i, j) that index pairs of balls, and (k, l) that index pairs of positions of the plate. In fact, the estimation problem does not determine the error correction function *absolutely*, but relative only to an unknown global linear scale. To fix this scale it is necessary to prescribe at least one distance on the measured artefact, such as the distance between the centres of two specified balls.

Kruth, Vanherck and de Jonge (1994) pose a similar estimation problem, but in their approach the calibration information is regarded as additional measurements. Thus, parameters $\boldsymbol{a}$ are computed to minimise

$$\sum \left(d_3(\boldsymbol{x}^*_{i,k}, \boldsymbol{x}^*_{j,k}) - d_3(\boldsymbol{x}^*_{i,l}, \boldsymbol{x}^*_{j,l})\right)^2 + \sum \left(d_3(\boldsymbol{x}^*_{i,0}, \boldsymbol{x}^*_{j,0}) - D_{ij}\right)^2,$$

where $\boldsymbol{x}_{i,0}$* are the true co-ordinates of the ith ball centre in some initial position of the plate, and D_{ij} is the prescribed distance between the ith and jth ball centres in this position. A related approach implemented at NPL defines an objective function of the form

$$\sum_{(i,j,k) \in I_M} \left(d_3(\boldsymbol{x}^*_{i,k}, \boldsymbol{x}^*_{j,k}) - d_{ij}\right)^2 + \sum_{(i,j) \in I_C} \left(d_{ij} - D_{ij}\right)^2, \qquad (12)$$

where $\{d_{ij}\}$ are parameters representing the distances between pairs of balls and $\{D_{ij}\}$ are the measured distances between pairs specified by the index set I_C.

An apparent advantage of these approaches over the one we have adopted is that they do not involve the ball centre or transformation parameters, leading to a somewhat simpler optimisation problem. However, there are also disadvantages:

- The number of observation equations that can be included in the optimisation can be very high, since the number of distances between pairs of points in an $n \times n$ grid is $O(n^4)$. Our approach gives rise to systems that are $O(n^2)$. A way of reducing the computational requirement is to incorporate only a subset of the distance observations into the optimisation, but this leads to questions about which subset to choose and further difficulties with different subsets producing different estimates.
- The fact that the observation equations involve distance means that there is significant correlation in these equations that is not taken into account. This means firstly that the estimates produced in general will be biased (see Section 6.2.7) and secondly that without considerable further analysis

solving the system

$$\boldsymbol{RU} = \boldsymbol{I},$$

again using back substitution

3.5.3 Jacobian for Class 2 checking

We have seen that if the calibration information is treated as additional measurements, the computational problem required for the Class 2 checking of a CMM is to minimise the function defined in equation (11) with respect to the error correction parameters $\boldsymbol{a}$, transformation parameters $\{\boldsymbol{t}_k\}$ and ball centre co-ordinates $\{\boldsymbol{y}_j\}$. To apply the Gauss-Newton algorithm to this problem, it is necessary to evaluate for given estimates of these parameters the Jacobian matrix J and residual vector $\boldsymbol{d}$ determined by the above estimator. Then, we have to solve the linear least-squares problem

$$\boldsymbol{Jp} = -\boldsymbol{d}$$

to obtain updated estimates of the parameters. The residual vector $\boldsymbol{d}$ contains the values for the given parameter estimates of the functions $\epsilon_{i,j}$, δ_{ij}, $\gamma_{i,j}$ and ζ_{ij} defined by the measurement and calibration equations. The matrix J contains the derivatives of these functions evaluated for the given estimates. Examination of the measurement and calibration equations shows that J can be ordered so as to have the general form

$$J = \begin{bmatrix} J_y & J_t & J_a \\ K_y & 0 & 0 \\ 0 & 0 & L_a \end{bmatrix}$$

Here, J_y, J_t and J_a are the Jacobian matrices for the functions $\epsilon_{i,j}$ with respect to, respectively, the parameters $\{\boldsymbol{y}_j\}$, $\{\boldsymbol{t}_k\}$ and $\boldsymbol{a}$. Furthermore, K_y is the Jacobian matrix for the functions δ_{ij} and $\gamma_{i,j}$ with respect to $\{\boldsymbol{y}_j\}$ (notice that δ_{ij} and $\gamma_{i,j}$ do not depend on $\{\boldsymbol{t}_k\}$ or $\boldsymbol{a}$), and L_a is the Jacobian matrix for the functions ξ_{ij} (notice that ξ_{ij} depends only on $\boldsymbol{a}$). Since each observation equation associated with the measurement information involves only one $\boldsymbol{y}_j$, the rows and columns of J_y can be arranged so that it takes the form

$$\begin{bmatrix} J_{y,1} & 0 & 0 \\ 0 & J_{y,2} & 0 \\ \vdots & & \\ 0 & 0 & J_{y,n_Y} \end{bmatrix}$$

Thus, the part of the Jacobian matrix associated with the measurement information is block-angular and this structure can be exploited in the orthogonal factorisation within the Gauss-Newton algorithm to improve computation efficiency (Cox, 1990). A closely related example is explained in more detail in Butler, Forbes and Kenward (1998).

An alternative way of exploiting the structure in the Jacobian matrix is to use an iterative approach based

An important application of the Gauss-Newton algorithm is fitting a non-linear function to measured data. Here, $\boldsymbol{a}$ are the parameters defining the function, and the quantity $d_i(\boldsymbol{a}^c)$ represents the discrepancy between the ith measured value and an estimate of the best-fit function given by parameter values $\boldsymbol{a}^c$. In this application, the performance of the algorithm can be poor if the data is *inaccurate* so that the values $d_i(\boldsymbol{a}^c)$ are large, or if the data is *unrepresentative*, in which case J is poorly conditioned.

3.5.2 Statistics

Suppose the data is contaminated with measurement error that are samples from a normal (Gaussian) probability distribution with zero mean and standard deviation σ^*. If the iterates $\boldsymbol{a}^c$ computed by the Gauss-Newton algorithm converge to the approximation $\boldsymbol{a}^\dagger$ of the true solution $\boldsymbol{a}^*$, the quantity

$$\sigma^\dagger = \sqrt{\frac{\sum_{i=1}^{m} d_i^2(\boldsymbol{a}^\dagger)}{m - n}},$$

called the *root mean square residual error*, provides an unbiased estimate of σ^*. Moreover, if J is the Jacobian matrix evaluated at $\boldsymbol{a}^\dagger$, the *variance-covariance matrix* V for the fitted parameters is given by

$$V = (\sigma^\dagger)^2 (J^T J)^{-1}.$$

The diagonal element v_{ii} of this matrix is the variance of the ith parameter a_i: the standard uncertainty $\sqrt{v_{ii}}$ measures the spread of likely values for a_i, and hence how well this parameter is determined by the data. The off-diagonal elements v_{ij} of V contain the covariances for the parameters, and provide measures of the association between each pair of parameters.

A numerically stable approach to solving the linear system

$$\boldsymbol{J}\boldsymbol{p} = -\boldsymbol{d}$$

is to factorize the matrix J using *orthogonal transformations*. Here, we find an orthonormal matrix Q (with $Q^T Q = I$, the identity matrix, I) such that

$$J = Q \begin{bmatrix} R \\ 0 \end{bmatrix},$$

$$\boldsymbol{d} = Q \begin{bmatrix} \theta \\ \theta' \end{bmatrix},$$

and R is an upper triangular matrix. Since the (Euclidean) length of a vector is invariant under multiplication by an orthonormal matrix, the vector $\boldsymbol{p}$ satisfies

$$\boldsymbol{R}\boldsymbol{p} = -\theta,$$

and may be found by backward substitution. Furthermore, since

$$(J^T J)^{-1} = (R^T R)^{-1} = R^{-1} R^{-T} = U U^T,$$

the variance-covariance matrix V for the parameters is readily computed since U may be obtained by

iterations

Correctness. The implementation has been thoroughly tested. In particular, all the derivative calculations have been tested against finite-difference calculations and solution estimates have been validated using linear search methods.

Statistics. Standard uncertainties of the fitted parameters and related quantities such as distances between error-corrected points are calculated according to the ISO Guide (ISO, 1993).

Data generation. In addition to implementing the optimisation algorithms, the package also includes extensive data generation and numerical simulation modules.

4.2 MAIN COMPUTATIONAL MODULES

In this section, we give an overview of the software.

4.2.1 Error correction functions

Modules for evaluating an error correction function $\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{p})$ and, optionally, the Jacobian matrices storing derivatives with respect to $\boldsymbol{a}$ for each component of this function, are required to have the following generic specification:

```
  function [ efg, Je, Jf, Jg ] = fgerrf(aa, XYZdim, X, IX, PO, ...
                                        p1, p2, p3, p4, p5)
% ------------------------------------------------------------------------
% FGERRF.M              Evaluation of error correction function
%                       and, optionally, derivatives with respect
%                       to function parameters.
% ------------------------------------------------------------------------

% Input
%              aa - naa x 1   array of parameter values
%          XYZdim -   2 x 3   array containing scale readings
%                             defining the dimensions of the
%                             volume of the CMM
%               X -   m x 3   array of scale readings for the
%                             given points
%              IX -   m x 1   array of indexing information:
%                             IX(i) is the index of the probe
%                                   offset for the ith point
%              PO -   p x 3   array of probe offsets
%    <p1, ...,p5> -           optional parameters
%
% Output
%             efg -   m x 3   array of correction vectors (ei, fi, gi)
%       <Je,
%        Jf,
%        Jg> -        m x naa arrays of derivatives of ei, fi and gi
%                             with respect to the parameters aa
%
% [ efg <, Je, Jf, Jg> ] = fgerrf(aa, XYZdim, X, IX, PO ...
%                                    <, p1, p2, p3, p4, p5>)
% ------------------------------------------------------------------------
```

on conjugate gradients (Golub and van Loan, 1996) to solve the linear least squares Jacobian system. The LSQR algorithm of Paige and Sanders (1982) requires only matrix-vector multiplications and it is reasonably straightforward to implement these multiplications which take account of the sparsity structure arising in J_y (and also in J_t, J_a, K_y and L_a, if required).

3.6 NOTES AND REFERENCES

Class 2 checking methods have been described by Cresto (1994), Kruth *et al.* (1994), Kunzmann, *et al.* (1990), among others. Balsamo *et al.* (1997) reports on the comparison of methods developed by IMGC, Italy, PTB, Germany and KUL, Belgium. Our approach differs from these approaches in that the ball centre and transformation parameters enter explicitly into the calculations, allowing us to model correctly the sources of random errors and provide standard uncertainties in association with the ISO Guide (ISO, 1993).

4 SOFTWARE FOR ANALYSING DATA FROM CLASS 2 CHECKS

NPL has implemented a comprehensive suite of modules for determining estimates of the parametric errors from measurements of a ball plate. The software is written in Matlab 4 and also runs under Matlab 5 (MathWorks Inc., 1993).

4. GENERAL FEATURES

The main features of the software package are

Computational aim. The software directly addresses the optimisation problem arising from the model, summarised in Section 3.3. No simplifying assumption or linearisation is made and the estimator implemented has favourable properties with respect to bias and statistical efficiency.

Scope. The functionality of the software covers an almost limitless range of Class 2 experiments. In particular it can cope with:

- *Any model of CMM behaviour* through the use of a generic interface (see Section 4.2 below). A range of kinematic and empirical models have been implemented and other models can be incorporated by providing the appropriate module.
- *Any ball plate geometry.* The only constraint is that not all the balls are co-linear.
- *Any measurement strategy.* The software is completely flexible with respect to the positioning of the ball plate, probing strategy, etc. There is no requirement that all the balls are measured in each position or that the balls are measured in a fixed order.
- *A range of calibration information* discussed in Section 3.2.

Automatic initialisation. The software implements an initialisation algorithm (Butler, Forbes and Kenward, 1998) to determine accurate initial estimates of the parameters $\{y_j\}_1^{n_T}$ and $\{t_k\}_1^{n_T}$. Assuming that the error correction parameters $\boldsymbol{a}$ are nominally zero, estimates of all the optimisation parameters can be assigned automatically. At the highest level the software requires as input only the measurement data and the name of the module evaluating the error function (and its derivatives with respect to $\boldsymbol{a}$).

Numerical stability. The software implements stable numerical procedures. In particular, orthogonal factorisation methods are used throughout, and polynomial and spline functions are stably parametrized. The iterative optimisation algorithm has proven convergence characteristics, and for well designed measurement experiments converges in a small number (typically < 5)

```
    function [ fy, Ky ] = fgcalb(Y, b1, b2, b3)
% ------------------------------------------------------------------
% FGCALB.M                  Function and gradient calculation for
%                           calibration information relating to a
%                           ball plate.
% ------------------------------------------------------------------
% Input
%         Y  -     m x 3    array containing estimates of the x, y
%                           and z-co-ordinates of the centres of the
%                           balls on the ball plate
% <b1,b2,b3> -              optional parameters specifying, for example,
%                           the indices of points, calibration values and
%                           weights for the calibration information
% Output
%        fy  - mcal x 1     weighted residuals corresponding to the
%                           calibration information
%       <Ky> - mcal x 3*m   Jacobian matrix of partial derivatives
%                           of fy with respect to the parameters Y
%
% [ fy <, Ky> ] = fgcalb(Y <, b1, b2, b3>)
% ------------------------------------------------------------------
```

For example, FGDE3.M returns, for given pairs of points, the weighted difference of the (3D) Euclidean distance between the two points and a prescribed calibrated value, *viz.*,

$$v_{ij}\delta_{ij} = v_{ij}(d_3(\boldsymbol{y}_i, \boldsymbol{y}_j) - D_{ij}),$$

(see Section 3.2), together with the derivatives of this difference with respect to the parameters $\{\boldsymbol{y}_j\}$. A matrix E, containing a list of index pairs (i, j), is used to specify those pairs of points for which distance calibration information is prescribed, a vector D contains the corresponding calibrated distances $\{D_{ij}\}$, and a vector W is used to specify weights v_{ij}.

FGXYZ.M returns, for given points, the weighted difference of the co-ordinates of the points and prescribed calibrated values, *viz.*,

$$u_{ij}\gamma_{ij} = u_{ij}(y_{ij} - Y_{ij}), \quad j = 1, 2, 3,$$

(see Section 3.2), together with the derivatives of these differences with respect to the parameters $\{\boldsymbol{y}_j\}$. A vector IP, containing a list of indices I, is used to specify those points for which co-ordinate calibration is prescribed, a matrix Y contains the calibration information $\{\boldsymbol{Y}_i\}$, and a matrix W is used to specify weights u_{ij} for each co-ordinate of each calibrated point.

Versions of these modules, called FGDE2.M and FGXY.M, respectively, are available to allow information about a ball-plate that has been calibrated as a 2D artefact to be incorporated into the analysis.

Calibration information relating to a different artefact. Modules included here cover the prescription of calibration information relating to the measurement of standard artefacts other than the ball plate used in the Class 2 experiment. The routines allow individual items of calibration information to be *weighted* so that account may be taken of their respective accuracies. All modules are required to have the following generic specification:

The module returns in the matrix EFG the values of the components of the error correction function for the measured points whose scale readings are held in X. The matrices JE, JF and JG contain the derivatives with respect to the parameters AA of the component functions at the measured points. It is assumed that the error function is parametrized in such a way that

$$e(x,\ 0,\ p)\ =\ p$$

for all x within the working volume of the CMM.

The numerical simulations described in Sections 5 and 6 involve two particular error correction functions. The first, FGEEM.M, is for an empirical parametric error model, whose six functions (e, f, g, u, v, w) used to construct the components of $e(x, a, p)$ are each composed of tensor products of low degree polynomials and polynomial spline functions as described in Section 2.1. The optional parameters P1, P2 and P3 that appear in the generic specification are used in this case to pass parameters defining the spline functions, *viz.*, their order (degree + 1) and knots.

The second, FGKEM.M, specifies a kinematic parametric error model (without linearisations) described in Section 2.2. The components of this error correction function are constructed from eighteen functions, three positional and three rotational for each of the three axes. The nine positional functions are expressed as polynomials of order NDEL, and the nine rotational as polynomials of order NEPS. The optional parameters P1 and P2 are used in this case to pass the parameters NDEL and NEPS.

For each of these parametric error models, the parameters a relate to functions for which the independent variables are normalised to lie in the interval [0, 1]. The user provides in the variable XYZDIM values for the scale readings that define the dimensions of the volume of the CMM, and this information is used to normalise the scale readings X prior to any evaluation.

4.2.2 *Jacobian for measurement information*

Module USBPSC.M computes and factorizes the Jacobian matrix for the measurement information recorded as part of a general ball-plate experiment. USBPSC.M accepts as a parameter the name of a module for evaluating an error correction function that conforms to the generic specification given in Section 4.2.1. In this way, USBPSC.M is not tied to any pre-defined parametric error model for describing the behaviour of a CMM.

The module assembles the Jacobian for the measurement information, and performs a QR-factorization of this matrix (exploiting structure) using Householder transformations and Matlab's in-built QR function. This latter function implements LINPACK routines documented in Bunch *et al.* (1979).

4.2.3 *Jacobian for calibration information*

Several modules are available for assembling Jacobian matrices K_y and L_a associated with prescribed calibration information. We classify these according to those that specify calibration information about (a) the ball plate that is subsequently measured, and (b) a different standard artefact such as a length bar.

Calibration information relating to the ball plate. Modules included here cover the prescription of distance as well as co-ordinate calibration information about the ball plate in both 2D and 3D. The routines allow individual items of calibration information to be weighted so that account may be taken of their respective accuracies. All modules are required to have the following generic specification:

z_j are the scale readings recorded by the CMM using a probe with probe offset $\boldsymbol{p}$ corresponding to points z_i^* and z_j^* on a calibrated artefact. If the calibrated distance between z_i^* and z_j^* is L_{ij}, this information is represented by the model equation

$$d_3(z_i + e(z_i, \boldsymbol{a}, \boldsymbol{p}) + \epsilon_i, z_j + e(z_j, \boldsymbol{a}, \boldsymbol{p}) + \epsilon_j) = L_{ij} + \zeta_{ij},$$

where ζ_{ij} denotes measurement error for the calibrated length. A simpler way of expressing this calibration information is as follows:

$$d_3(z_i + e(z_i, \boldsymbol{a}, \boldsymbol{p}), z_j + e(z_j, \boldsymbol{a}, \boldsymbol{p})) = L_{ij} + \xi_{ij},$$

although the use of this form in the numerical experiments considered in Section 6 may introduce a small statistical bias in the solutions determined.

Alternatively, suppose z_i and z_j correspond to points z_i^* and z_j^* on the end-faces of a length bar of calibrated length L_{ij}. This information is represented by the model equation

$$\boldsymbol{n}_{ij}^{\mathrm{T}}(z_i + e(z_i, \boldsymbol{a}, \boldsymbol{p}) + \epsilon_i - z_j - e(z_j, \boldsymbol{a}, \boldsymbol{p}) - \epsilon_j) = L_{ij} + \zeta_{ij},$$

where $\boldsymbol{n}_{ij}$ is a unit vector pointing along the length bar. Again, a simpler, but statistically biased, way of expressing this calibration information is as follows:

$$\boldsymbol{n}_{ij}^{\mathrm{T}}(z_i + e(z_i, \boldsymbol{a}, \boldsymbol{p}) - z_j - e(z_j, \boldsymbol{a}, \boldsymbol{p})) = L_{ij} + \xi_{ij}.$$

The calibration information provided may relate to a number of measurements of the artefact, or different artefacts, within the working volume of the CMM. Furthermore, each piece of calibration information may be weighted to reflect its relative accuracy.

3.3 A NON-LINEAR LEAST SQUARES ESTIMATOR

The computational problem required for the Class 2 checking of a CMM is to find optimal values for the error correction parameters $\boldsymbol{a}$, transformation parameters $\{\boldsymbol{t}_k\}$ and ball centre co-ordinates $\{\boldsymbol{y}_j\}$ for the measurement and calibration equations. If the components of the errors ϵ_i, δ_{ij}, γ_i and ξ_{ij} associated with the measurement and calibration information are samples of Gaussian distributions, each with mean zero, estimates of the model parameters are obtained by minimising

$$\sum_{i=1}^{m_X} (w_{i,1}^2\epsilon_{i,1}^2 + w_{i,2}^2\epsilon_{i,2}^2 + w_{i,3}^2\epsilon_{i,3}^2) + \sum_{i,j\in I_D} v_{ij}^2\delta_{ij}^2 + \sum_{i\in I_C} (u_{i,1}^2\gamma_{i,1}^2 + u_{i,2}^2\gamma_{i,2}^2 + u_{i,3}^2\gamma_{i,3}^2) + \sum_{i,j\in I_A} r_{ij}^2\zeta_{ij}^2 \quad (11)$$

with respect to the parameters $\boldsymbol{a}$, $\{\boldsymbol{t}_k\}$ and $\{\boldsymbol{y}_j\}$. Here, we suppose that, in total, m_X ball centre measurements have been made, I_D specifies the index pairs (i, j) for the balls for which distance calibration information is known, I_C specifies the indices of the balls for which co-ordinate calibration information is prescribed, and I_A specifies the index pairs (i,j) for the points for which distance calibration information is specified using some other artefact. In practice some or all of these sets may be empty. The quantities $w_{i,k}$, v_{ij}, $u_{i,k}$ and r_{ij} are *weights* that are chosen to be inversely proportional to the standard deviations of the corresponding Gaussian distributions: these quantities reflect the relative accuracies of the measurement and calibration information. The problem is a non-linear least-squares problem that may be solved using the Gauss-Newton algorithm described in, for example, Gill, Murray and Wright (1981) and Fletcher (1980).

```
function [ fa, La ] = fgcalp( aa, XYZdim, Z, IZ, PO, c1, c2, c3, ...
                              fgerrf, p1, p2, p3, p4, p5 )
% ------------------------------------------------------------------
% FGCALP.M                Function and gradient calculation for
%                         calibration information relating to an
%                         artefact other than the ball plate.
% ------------------------------------------------------------------
% Input
%        aa - naa x 1     vector containing values for the parameters
%                         defining the parametric error model
%    XYZdim -   2 x 3     array containing scale readings
%                         defining the dimensions of the
%                         volume of the CMM
%         Z -  mz x 3     array of scale readings for the measurement
%                         of points on the artefact
%        IZ -  mz x 1     vector of indices that specify for each point
%                         in Z the probe offset for the corresponding
%                         measurement:
%                            1 <= IZ(i) <= p,
%                         where p is the number of probe offsets
%        PO -   p x 3     array of probe offsets
%       c1,
%       c2,
%       c3  -             parameters specifying, for example, the
%                         indices of points, calibration values and
%                         weights for the calibration information
%    fgerrf -             string containing the name of a module for
%                         evaluating a parametric error model and its
%                         gradient:
%                           [ efg, Je, Jf, Jg ] = fgerrf( aa, XYZdim, ...
%                                                 X, IX, PO <, p1, ..., p5> )
% <p1, ...> -             optional parameters to be passed to fgerrf
%
% Output
%       fa  - mcal x 1    weighted residuals corresponding to the
%                         calibration information
%      <La> - mcal x naa  Jacobian matrix of partial derivatives
%                         of fa with respect to the parameters aa
%
% [ fa, La ] = fgcalp( aa, XYZdim, Z, IZ, PO, e, d, w, ...
%                      'fgerrf', p1, p2, p3, p4, p5 )
% ------------------------------------------------------------------
```

For example, FGDP3.M returns, for given pairs of points, the weighted difference of the (3D) Euclidean distance between the two points and a prescribed calibrated value, *viz.*,

$$r_{ij}\xi_{ij} = r_{ij}\{d_3(z_i + e(z_i, a, p), z_j + e(z_j, a, p)) - L_{ij}\},$$

(see Section 3.2), together with the derivatives of this difference with respect to the parameters a. A matrix E, containing a list of index pairs (i,j), is used to specify those pairs of points for which distance calibration information is prescribed, a vector D contains the corresponding calibrated distances L_{ij}, and a vector W is used to specify weights r_{ij}.

FGDP3.M accepts as a parameter the name of a module for evaluating an error correction function that conforms to the generic specification given in Section 4.2.1. In this way, FGDP3.M is not tied to any pre-defined parametric error model for describing the behaviour of a CMM.

4.2.4 User level module

The module CLASS2BP.M computes initial estimates for the parameters $\boldsymbol{a}$, $\{t_k\}$ and $\{y_j\}$ and then implements a Gauss-Newton algorithm to determine values of these parameters that solve the estimation problem for Class 2 checking of a CMM described in Section 3.4. Since we expect (a) measurement errors to be small (typically, better than 1 part in 10^4), (b) the Jacobian matrix to be well-conditioned (at least for a well-designed experiment), and (c) to have accurate starting estimates of the solution parameters, a standard (undamped) Gauss-Newton algorithm will perform well, converging in a small number of iterations.

The module uses USBPSC.M to assemble and factorize the Jacobian for the measurement information. CLASS2BP.M also accepts as input parameters the names of modules for evaluating the calibration information (conforming to the generic specifications given in Section 4.2). The calibration information may relate either to the ball-plate used in the Class 2 experiment, or to a different standard artefact that is measured using the CMM. In this way it is easy to analyse ball-plate experiments constructed from various combinations of measurement and calibration information.

5 NUMERICAL SIMULATION WITH EMPIRICAL AND KINEMATIC MODELS

5. DATA GENERATION

For the purposes of the numerical simulations described in this section we have assumed that the working volume of the CMM corresponds to the unit metre cube. We indicate here how we may generate measurement data that relates to a CMM whose systematic error behaviour is described by a given parametric error model. Let $\boldsymbol{x}^*$ be the true co-ordinates of the centre of a ball that is measured using a probe with probe offset $\boldsymbol{p}$. Then, ignoring measurement error, the recorded scale readings $\boldsymbol{x}$ for the measurement satisfy

$$\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{a}^*, \boldsymbol{p}),$$

where $\boldsymbol{e}$ is an error correction function determined by prescribed parameter values $\boldsymbol{a}^*$. This expression defines the scale readings *implicitly*, and we determine $\boldsymbol{x}$ *approximately* by evaluating

$$\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{e}(\boldsymbol{x}^* - \boldsymbol{p}, \boldsymbol{a}^*, \boldsymbol{p})$$

(If required, an exact solution can be determined easily using standard iterative techniques.) The data generated in this way relates to a CMM whose systematic error behaviour is *approximately* described by the error correction function $\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{a}^*, \boldsymbol{p})$. For the simulations described below the error incurred in this approximation, measured as the difference between the norms of $\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{a}^*, \boldsymbol{p})$ and $\boldsymbol{e}(\boldsymbol{x}^* - \boldsymbol{p}, \boldsymbol{a}^*, \boldsymbol{p})$, is of the order of 10^{-9}; for a CMM with working volume 1 m^3, this corresponds to an error of 1 nanometre.

Part of the numerical simulations described below demonstrates how well we are able to recover the error correction function from data generated in this way. For data simulated using a function from a specified class of model (kinematic or empirical), we consider (a) how well we are able to estimate the CMM's behaviour using an approximating function from the same class, and (b) the merits of using an approximating function from a different class. In addition, numerical experiments are used to investigate the effect of using different distributions of measurement information, and different types and amounts of calibration information.

5.2 APPROXIMATING DATA SIMULATED USING A KINEMATIC ERROR CORRECTION FUNCTION

A simulated ball-plate experiment is defined by the following information:

Artefact: the ball-plate is composed of 25 balls arranged on a 5 × 5 square grid, positioned initially to cover the region [0.1, 0.9] × [0.1, 0.9] within the plane $z = 0.1$.

Calibration information: the artefact is calibrated by prescribing the (3D) Euclidean distances between the centres of neighbouring balls on all rows and columns of this grid, giving 40 items of calibration data. We regard the prescribed distances as having half the uncertainty as that of the measurement information (which depends on the *repeatability* of the CMM), and the model equations corresponding to the calibration information are weighted accordingly.

Measurement information: each ball on the plate is measured in the planes $z = 0.1$ and $z = 0.9$ using a single (vertical) probe, and twice in each of the planes $x = 0.2$, 0.4, 0.6 and 0.8 and $y = 0.2$, 0.4, 0.6, a nd 0.8 with different probes (orientated orthogonally to the plane of measurement). This gives 3 (co-ordinates) × 25 (balls) × 18 (plate positions) = 1350 items of measurement information that are each assumed to be measured to an accuracy given by the repeatability of the CMM. The (true) systematic error behaviour of the CMM is described by a *kinematic* parametric error model, as implemented in FGKEM.M, with NDEL = NEPS = 3. The recorded scale readings corresponding to the measurement information are computed in the manner described in Section 5.1.

We use the Matlab function CLASS2BP.M to determine estimates of the centre co-ordinates $\{\boldsymbol{y}_j\}$ of the balls for the initial position of the plate, transformation parameters $\{\boldsymbol{t}_k\}$ defining subsequent positions of the plate, and parameters $\boldsymbol{a}$ that define an error correction function. The systematic error behaviour of the CMM is modelled using (a) a kinematic parametric error model, as implemented in FGKEM.M, with NDEL = NEPS = 5, and (b) an empirical parametric error model, as implemented in FGEEM.M, where the univariate spline functions in this model are cubic (order 4) with a single interior knot at 0.5.

In Table 1 we indicate, for each parametric error model, the progress of the Gauss-Newton algorithm to a solution. We list for each iteration (a) the size of the Gauss-Newton step, *i.e.*, the Euclidean length of the vector $\boldsymbol{p}$, and (b) the weighted residual sum of squares s^2 for the current estimates of the solution parameters. For the type of Class 2 experiment described here, s^2 is given by

$$s^2 = \sum_{i=1}^{m} (w_{i,1}^2 \epsilon_{i,1}^2 + w_{i,2}^2 \epsilon_{i,2}^2 + w_{i,3}^2 \epsilon_{i,3}^2) + \sum_{i,j \in I_d} v_{ij}^2 \delta_{ij}^2.$$

It is clear from the results presented in this table that, for both models, the algorithm has converged: by the third iteration the change in s^2 and the size of the update $\boldsymbol{p}$ are both small. In addition, we observe that the *value* of s^2 is small, indicating that in each case the data is approximated accurately by the fitted function.

	Kinematic parametric error model		Empirical parametric error model	
Iteration	Length of step $\|\boldsymbol{p}\|$	s^2	Length of step $\|\boldsymbol{p}\|$	s^2
0		2.55471e-08		2.55471e-08
1	9.93433e-05	2.52668e-17	1.04044e-04	2.52556e-17
2	1.45208e-08	6.02182e-19	1.13778e-08	1.85857e-19
3	1.55252e-13	6.02182e-19	4.42790e-14	1.85857e-19

Table 1: Progress of the Gauss-Newton algorithm for data generated for a CMM whose systematic error behaviour is described by a kinematic error function. Two cases are considered: the error behaviour is modelled by (a) a kinematic error function, and (b) an empirical error function.

5.3 APPROXIMATING DATA SIMULATED USING AN EMPIRICAL ERROR CORRECTION FUNCTION

A simulated ball-plate experiment is defined as in Section 5.2, except that the (true) systematic error behaviour of the CMM is described by a simple *empirical* parametric error model. In this model, each of the six functions (e, f, g, u, v, w) used to construct the error correction function are expressed in terms of the basis

$$\{\ , x, y, z, x^2, xy, xz, y^2, yz, z^2\},$$

adjusted so that appropriate frame constraints apply (Section 2.4). We use the Matlab function CLASS2BP.M to determine again estimates of the parameters. As before, we model the systematic error behaviour of the CMM using a kinematic model and an empirical model.

In Table 2 we indicate, for each model, the progress of the Gauss-Newton algorithm. The results show that in each case the algorithm converges to a solution. However, by examining the size of s^2 at the solutions, it is evident that the data is approximated less well by the kinematic parametric error model.

	Kinematic parametric error model		Empirical parametric error model	
Iteration	Length of step $\|p\|$	s^2	Length of step $\|p\|$	s^2
0		2.27966e-08		4.16973e-08
1	1.49142e-03	2.69749e-10	9.91807e-05	5.83832e-18
2	9.95597e-08	2.69749e-10	9.31578e-09	9.41295e-20
3	2.16600e-12	2.69749e-10	5.62944e-14	9.41295e-20
4	1.42876e-13	2.69749e-10		

Table 2: Progress of the Gauss-Newton algorithm for data generated for a CMM whose systematic error behaviour is described by an empirical error function. Two cases are considered: the error behaviour is modelled by (a) a kinematic error function, and (b) an empirical error function.

These results illustrate that while the empirical model can successfully model kinematic error behaviour, the converse is not true. This is to be expected as the kinematic model is a special case of the empirical model (see Section 2.3).

One important application of the empirical error function is in validating a kinematic model. If, in a Class 2 checking experiment, the fit obtained using an empirical model is significantly better than that obtained by a kinematic model as measured, for example by the residual errors, this indicates that the CMM's behaviour is not adaquately modelled by the kinematic model. If, on the other hand, the residual errors are of comparable size, there is reason to believe that the kinematic model explains the behaviour.

5.4 COMPARISON OF "SPARSE" AND "DENSE" MEASUREMENT STRATEGIES

A simulated ball-plate experiment is defined by the following information:

Artefact: the ball-plate is composed of 64 balls arranged on an 8 × 8 square grid, positioned initially to cover the region [0.1, 0.9] × [0.1, 0.9] within the plane $z = 0.1$.

Calibration information: the artefact is calibrated by prescribing the (3D) Euclidean distances between the centres of neighbouring balls on all rows and columns of this grid, giving 112 items of calibration data.

Measurement information: each ball on the plate is measured in the planes $z = 0.1$ and $z = 0.9$ using a single (vertical) probe, and twice in each of the planes $x = 0.5$ and $y = 0.5$ with different probes (orientated orthogonally to the plane of measurement). This gives 3 (co-ordinates) × 64 (balls) × 6 (plate positions) = 1152 items of measurement information that are each assumed to be measured to an accuracy given by the repeatability of the CMM. The (true) systematic error behaviour of the CMM is described by a *kinematic* parametric error model, as implemented in FGKEM.M, with NDEL = NEPS = 3. The recorded scale readings corresponding to the measurement information are computed in the manner described in Section 5.1.

The essential difference between this experiment and that used previously is in the way the measurement

data is distributed throughout the volume of the CMM: approximately the same *amount* of data is available, but it is concentrated in the four planes $z = 0.1$, $z = 0.9$, $x = 0.5$ and $y = 0.5$ rather than filling the volume more completely.

Table 3 shows the progress of the Gauss-Newton algorithm when using (a) a kinematic error function, and (b) an empirical error function, to model the data. We see that with an empirical function, the algorithm does not converge to a well-defined solution because, although changes in the weighted residual sum of squares s^2 are small from iteration to iteration, the algorithm continues to make large changes to the parameter values. This implies that the Jacobian matrix is ill-conditioned. On the other hand, with a kinematic function, the algorithm converges within a few iterations, and the value of s^2 indicates that the data is approximated accurately.

	Kinematic parametric error model		Empirical parametric error model	
Iteration	Length of step $\|\|p\|\|$	s^2	Length of step $\|\|p\|\|$	s^2
0		1.39681e-08		9.40208e-09
1	1.54532e-04	5.21513e-18	1.90833e-03	3.61041e-17
2	1.54361e-08	5.70436e-19	1.07792e-03	1.09524e-18
3	1.60042e-13	5.70436e-19	5.96582e-04	1.81031e-19
4			7.43816e-04	3.21578e-19
5			2.84865e-04	3.21578e-19

Table 3: Progress of the Gauss-Newton algorithm for data generated for a CMM whose systematic error behaviour is described by a kinematic error function. The data is distributed "sparsely" throughout the volume of the CMM.

The ball-plate experiment defined above is repeated, but using an *empirical* parametric error model to describe the systematic error behaviour of the CMM. Table 4 shows the progress of the Gauss-Newton algorithm when using a kinematic and an empirical error function as above to model the data. Again, we see that with an empirical function, the algorithm does not converge to a well-defined solution. With a kinematic function, the algorithm does converge, but this time the value of s^2 suggests that the data is not approximated particularly well.

These results illustrate the fact that the empirical model requires a more dense sampling of the working volume in order to determine accurately estimates of the error parameters.

	Kinematic parametric error model		Empirical parametric error model	
Iteration	Length of step $\|\boldsymbol{p}\|$	s^2	Length of step $\|\boldsymbol{p}\|$	s^2
0		2.09027e-08		1.80819e-08
1	1.11319e-03	2.63829e-10	3.57816e-03	1.63780e-16
2	7.54689e-07	2.63829e-10	1.95307e-03	9.01130e-18
3	8.51430e-12	2.63829e-10	9.68312e-04	7.01899e-19
4	8.43218e-14	2.63829e-10	6.06065e-04	1.02614e-19
5			1.94115e-04	6.41134e-20
6			2.54180e-05	6.36180e-20
7			2.72805e-06	6.36164e-20

Table 4: Progress of the Gauss-Newton algorithm for data generated for a CMM whose systematic error behaviour is described by an empirical error function. The data is distributed "sparsely" throughout the volume of the CMM.

6 DESIGN OF BALL PLATE EXPERIMENTS

In this section we look at the design of ball plate experiments in some detail. The goal of a ball plate experiment is to determine accurate estimates of the parametric errors of the CMM at any given point within its working volume. Factors which will affect the accuracy of these estimates include:

the parametric error model
the repeatability of the machine error
the design of the ball plate
the positioning of the ball plate
the probing strategy
the type of calibration information associated with the ball plate
the accuracy of the calibration information
the method of data analysis

The accuracy of the estimates can be gauged by the standard uncertainties

- $U_1 = U(\boldsymbol{y}_j)$ of the sphere centre co-ordinates in a fixed frame of reference
- $U_2 = U(\boldsymbol{t}_k)$ of the transformation parameters
- $U_3 = U(\boldsymbol{a})$ of the parametric error parameters
- $U_4 = U(\boldsymbol{e}(\boldsymbol{x}_q, \boldsymbol{a}, \boldsymbol{p}_q))$ of the error function evaluated at specified points $\{\boldsymbol{x}_q\}$ within the working volume of the CMM.

These quantities are absolute measures of accuracy and do not take into account correlation between the fitted parameters nor reflect that what is usually of interest is the change in relative error within the working volume. For this reason we also record the standard uncertainties

U_5 = $U(d(y_i, y_l))$ of the distances between pairs of sphere centres
U_6 = $U(d(x_q^*, x_r^*))$ of the distances between pairs of points (x_q^*, x_r^*) where

$$x_q^* = x_q + e(x_q, a, p_q),$$

etc.

These latter two quantities are independent of the choice of the frame of reference and can more easily be compared with the CMM accuracy.

The numerical simulations involve a CMM of working volume 1 m^3 with a repeatability error of 0.2 micrometres in each co-ordinate. It is assumed that these errors are uncorrelated and drawn from a normal distribution. This repeatability error gives a natural baseline with which to compare the standard uncertainties for the fitted parameters and related quantities. A second baseline is given by an (optimistic) target length measuring capability of the CMM which we take to be 0.5 + L/1000. Thus, a ball plate experiment that results in standard uncertainties of the distances between points in the working volume that are greater than 1.5 micrometres will be of limited value while one that results in uncertainties close to the repeatability should be satisfactory for most purposes.

From our investigation of parametric error models, it is known that in order to determine information about the orientation of the stylus assembly it is necessary to use at least three probe offsets in measurements. In the experiments below, we simulate measurements using a "star" probe which has a single stylus along the negative z-axis and stylii along the positive and negative x- and y-axes. It is assumed that any measurement error due to inaccuracies in probe function can be modelled adequately as a component of the CMM repeatability error.

The numerical simulations reported on below were designed to calculate the uncertainty of the estimates of the parametric errors and related quantities. These uncertainties are essentially independent of the actual values for the parametric errors and it is sufficient to perform simulations for a CMM with parametric error parameters set to zero. This has the advantage that statements about uncertainties do not have to take into account the validity of the model. However the uncertainty estimates only apply to the extent that the parametric error models adequately reflect actual CMM behaviour.

6. EXPERIMENT MEASUREMENT OF A BALL PLATE IN FOUR POSITIONS

6.1.1 Symmetrical measurement strategy

The first suite of experiments involve the measurement of a ball plate in four positions.

- *Artefact*: the ball-plate is composed of 49 balls arranged on a 7×7 square grid of dimension 0.75×0.75.
- *Measurement information*: each ball on the plate is measured in the planes $z = 0.1$ and $z = 0.9$ using a single (vertical) probe, and twice in each of the planes $x = 0.5$ and $y = 0.5$ with different probes (orientated orthogonally to the plane of measurement). Each measured co-ordinate is exact subject to a random perturbation drawn from a normal distribution with mean zero and standard deviation 2×10^{-7}.
- *Parametric error model*: the systematic error behaviour of the CMM is modelled by a kinematic error model, as implemented in FGKEM.M.
- *Calibration information*: we consider four cases:

A. The Euclidean distances between the centres of neighbouring balls on all rows and columns of the grid are prescribed, giving 84 items of calibration data.

B. The Euclidean distances between the centres of the four balls at the vertices of the grid measured along the rows and columns of the grid are prescribed, giving 4 items of calibration data.

C. The co-ordinates of the centres of the balls in the initial position of the ball plate are prescribed, giving 147 items of calibration data.

D. The co-ordinates of the centres of the balls at the vertices of the grid in the initial position of the ball plate are prescribed, giving 12 items of calibration data.

Note that calibration information provides constraints on the centre co-ordinates $\{y_j\}$ of the balls for the initial position of the plate (Section 2.4). In each case, the calibration information generates model equations that, as in previous simulations, are given a "weighting" w relative to that for the model equations corresponding to the measurement of the balls on the ball-plate. Thus, if $w = 1$, then each calibration model equation is assumed to be subject to random error drawn from the same normal distribution as the measurement information.

We find that for exact data, that is, data to which no random perturbations have been applied, experiments with calibration information B and D are exactly rank deficient by 2 so that no sensible estimates of the parametric errors can be made. For calibration A and C, the system has full rank and estimates of the standard uncertainties for the various quantities, maximum and average, are given in Table 5 for a range of weights $w = 0.01, 0.5, 1, 2$ and 100.

The results show:

As w gets smaller, all the uncertainties get larger. This is to be expected for as $w \to 0$ the system becomes effectively rank deficient.
As w gets larger the uncertainties get smaller but not significantly compared to the case $w = 1$. The exception to this is U_5 for calibration information C which in this case reflects the complete knowledge about the location of the balls in an original frame of reference. Note that knowing the distances specified by calibration information A accurately is insufficient to determine these locations accurately.
Calibration information C gives more accurate results than A.

In summary, for accurate calibration information, the repeatability of the CMM measurements is the limiting factor while for inaccurate calibration information the calibration information is the limiting factor.

6.1.2 Modified experiment to eliminate symmetries

If we examine the nature of the rank deficiency for cases B and D, we find that a symmetrical change of location of the balls on the plate can be compensated by a symmetrical arrangement of scale errors along the CMM axes. A simple way to eliminate any correlation between the machine errors and the ball centre parameters is to introduce new ball plate positions in such a way that all possible symmetries are destroyed. Consider for example the measurement of a ball plate (of a square grid design as in the above example) placed in three positions, the second representing a rotation of the plate from its initial position by 90 degrees so that the grid points are mapped on to the same points in space, and the third representing a translation of the plate from its original position by one grid unit so that all but one column (or row) of grid points are mapped on to the same points in space. It can be shown that if a plate is measured in these three positions then the machine errors are completely separable from the ball locations up to two scale errors, one in the plane of the plate and one orthogonal to the plate. This is true no matter how complicated the

error behaviour of the machine as long as it is repeatable, smoothly varying and reasonably small in magnitude compared to the dimensions of the plate.

If, alternatively, rotations and translations are introduced into the above experiment then rank deficiency disappears and the machine errors can be estimated successfully using calibration information B or D. In the modified experiment, the ball plate in its initial position occupies the South-West corner of the plane $z = 0.1$ while in the second position it is rotated by 90 degrees and occupies the North-East corner of the plane $z = 0.9$; the other positions are unchanged. Table 6 gives the uncertainties, maximum and average, for the modified experiment with calibration information A and B (and $w = 1$). Note that both are competitive with the original experiment using the maximum amount of calibration information C and represent an accurate determination of the machine errors relative to the target accuracy of the CMM. In fact, in the modified experiment the additional calibration information supplied by A over B has only a very modest effect in reducing the uncertainties.

6.2 EXPERIMENT 2: MEASUREMENT OF A BALL PLATE IN SIX POSITIONS

The second suite of experiments involves the measurement of a ball plate in six positions.

Ball plate positions. The ball plate is placed in the planes $z = t_1$, $z = t_2$, $x = t_1$, $x = t_2$, $y = t_1$ and $y = t_2$, where $0 \le t_1, t_2 \le 1$. For each pair of parallel planes, the ball plate occupies one corner of a square at t_1, the opposite corner at t_2 and, in addition, the ball plate is rotated through 90 degrees as it is moved from one level to the other. In this, way rotations and translations are introduced to eliminate symmetry while at the same time maximising the coverage of the CMM's working volume.
Measurement information: each ball on the plate is measured in the planes $z = t_k$ using a single (vertical) probe, and twice in each of the planes $y = t_k$ and $z = t_k$, $k = 1, 2$, with different probes (orientated orthogonally to the plane of measurement). Each measured co-ordinate is exact subject to a random perturbation drawn from a normal distribution with mean zero and standard deviation 2×10^{-7}.
Parametric error model: the systematic error behaviour of the CMM is modelled by a kinematic error model, as implemented in FGKEM.M.
Calibration information: the Euclidean distances between the centres of the four balls at the vertices of the grid measured along the rows and columns of the grid are prescribed, giving 4 items of calibration data with weight $w = 1$ (case B above).

6.2.1 Experiments with a grid-design ball plate

The first set of experiments involves ball plates of an $n \times n$ grid design of dimension 0.75×0.75, $n = 4, 5, \ldots, 9$, with levels $t_1 = 0.25$ and $t_2 = 0.75$.. Table 7 shows the uncertainties, maximum and average, associated with this set of experiments for the case NDEL = NEPS = 5. The results show a steady reduction in the uncertainties as n increases. The experiments with $n = 4, 5$ involve fewer measurements than in Experiment 1 yet produce better uncertainties. This shows that there are advantages in increasing the number of positions over increasing the number of balls on the plate. If we repeat the same experiment except with NDEL = 7, we find that for $n = 4$ the system is effectively rank deficient, showing that there is not enough information to determine all the parameters. Note, however, that the ball plate itself is still accurately calibrated according to the uncertainties U_5 in the distances between ball centres.

6.2.2 *Experiments with a square-design ball plate*

The rank deficiency experienced above can be attributed in part to the fact that for any position we are trying to determine degree 7 polynomials by sampling as few as four points. The kinematic parametric error model is built up as axis-upon-axis behaviour and by samplying along the three axes it is possible to determine the errors at any point. In order to exploit this fact (which does not apply to the more general empirical error models) some ball plate designs place the balls only on the outer edges of a square - we term this a *square* design as opposed to a *grid* design. A potential advantage of this design is that is possible to have more balls along a row or column without increasing drastically the total number of balls. For example, a square with 17 balls on each side has the same number of balls as an 8×8 grid. Table 8 shows the uncertainties, maximum and average associated with experiments for ball plates of square design with $n = 5, 7, 10, 13, 17$ and 21 (with n chosen so that the number of balls correspond to ball plates of grid design with $n = 4, 5, \ldots, 9$). Comparing Tables 7 and 8, we see that the "performances" of the grid and square designs are comparable except that the square design yields a full rank system for the smallest number of balls when NDEL = 7. This is more generally true. For example, if we consider the same experiment only with NDEL = 10, then a 7×7 grid leads to a rank deficient system while a square with 13 balls on each side produces a full rank system even although the number of balls is approximately the same (49 and 48, respectively).

6.2.3 *Experiments with varying dimensions of the ball plate*

So far, the experiments reported on have involved a ball plate of dimensions 0.75×0.75 (compared to the CMM dimensions of $1.00 \times 1.00 \times 1.00$). We now look at the effect of changing the dimensions on the uncertainties generated. Table 9 records the uncertainties, maximum and average, firstly for a 5×5 grid-design ball plate for a kinematic error model with NDEL = NEPS = 5, and secondly for a 7×7 grid-design ball plate for a kinematic error model with NDEL = 7, NEPS = 5 where the length dimensions L range from 0.8 to 0.3. The results show that as the length decreases the uncertainties rise but not dramatically. For example with $L = 0.5$ the average uncertainty in the distances are still of the order of the repeatability of the CMM even though the ball plate occupies only 1/4 of the area in any plane of the working volume of the CMM. This is reassuring since it means that the same ball plate can be used for CMMs with a range of working volumes.

6.2.4 *Experiments with a modified strategy to calibrate the ball plate*

The increase in the uncertainties U_5 relating to the inter-ball distances in experiment 6.2.3 shows that the ball plate is being less well defined. This is to be expected since for small L the measurement strategy no longer incorporates a rotation and translation of the ball plate on to itself. We can modify the measurement strategy to include two further measurements of the ball plate in the plane $z = 0.5$, one a rotation of the other through 90 degrees. The numerical results in Table 10 show that there is no significant increase in the uncertainties U_5 as L decreases confirming that the measurement strategy is sufficient to calibrate the ball plates accurately. The extra rigidity in the system has a beneficial effect on the uncertainties U_6.

6.2.5 *Calibration information supplied by measurements of a different artefact*

We now consider an experiment similar to 6.2.1 except that the calibration information is supplied by the measurement of a different artefact, in this case a 2×2 grid-design ball plate whose sides of length 0.75 are known. This additional artefact is measured in one position in the plane $z = 0.1$ using a single probe. The calibration information is incorporated into the model through modules FGCALP.M and FGDP3.M. The uncertainties, maximum and average, for two cases - $n = 5$ and NDEL = 5, and $n = 7$ and NDEL = 7 - are presented in Table 11. In comparing this table with the appropriate columns in Table 7 we see that this

method of providing calibration information is equally effective for this measurement strategy

Experiments with an empirical error model

We have seen that the experimental strategy employing six positions of the ball plate can give accurate estimates of the kinematic error model parameters. We can also consider the same strategy but with an empirical error model. These experiments involve an $n \times n$ grid design of dimension 0.75×0.75 placed at levels $t_1 = 0.25$ and $t_2 = 0.75$, as in Experiment 6.2.1. Each of the six component functions of the empirical error model is modelled by sums of polynomial-spline tensors products (Section 2.1.1) where the splines are order 3 with no interior knots. The results are compared with that for a kinematic model with NDEL = NEPS = 5 and are presented in Table 12. They show that the empirical error parameters are also determined accurately using the measurement strategy, although not as accurately as those of the kinematic model.

Experiments with a distances-only estimator

We also briefly investigate the behaviour of an estimator based on equation (12) in which the observation equations involve only distances between points and do not involve the parameters $\{y_j\}$ or $\{t_k\}$. The experiments are the same as Experiment 6.2.1, only the estimator is different. The parameter error model is the kinematic model implemented by FGKEM.M with NDEL = NEPS = 5. The distances between balls on the plate that are parametrized by $\{d_{ij}\}$ involve all nearest neighbours along rows, columns and NW-SE diagonals along with the six distances between all four corners. This is more than sufficient to determine the plate as a rigid two-dimensional artefact. As in Experiment 6.2.1, the four side distances are provided as calibration information.

For a given set of data, let $\boldsymbol{a}$ be the parameters determined by the estimator based on (11) and $\boldsymbol{a}_D$ those determined by the estimator associated with (12). In order to compare the two estimates we evaluate

$$\Delta_i = \|\boldsymbol{x}_{i,} + \boldsymbol{e}(\boldsymbol{x}_{i,1}, \boldsymbol{a}, \boldsymbol{p}_{i,1}) - \boldsymbol{x}_{i,2} - \boldsymbol{e}(\boldsymbol{x}_{i,2}, \boldsymbol{a}, \boldsymbol{p}_{i,2})\| \qquad \|\boldsymbol{x}_{i,1} + \boldsymbol{e}(\boldsymbol{x}_{i,1}, \boldsymbol{a}_D, \boldsymbol{p}_{i,1}) - \boldsymbol{x}_{i,2} - \boldsymbol{e}(\boldsymbol{x}_{i,2}, \boldsymbol{a}_D, \boldsymbol{p}_{i,2})\|$$

for 200 pairs of points $\{(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2})\}$ randomly scattered within the working volume. We see that Δ_i measures the difference between the distances between the ith pair of points determined from the two error maps defined by $\boldsymbol{a}$ and $\boldsymbol{a}_D$. We calculate the root-mean-square (RMS) value and maximum absolute value of these differences.

For a 5×5 grid, $\mathrm{RMS}\{\Delta_i\} = 3.4713e{-}07$ m while $\max\{|\Delta_i|\} = 1.3355e{-}06$ m. For a 7×7 grid, $\mathrm{RMS}\{\Delta_i\} = 1.5798e{-}07$ m and $\max\{|\Delta_i|\} = 8.1069e{-}07$ m. These numbers should be compared with the repeatability error of 0.2 micrometres in each coodinate. The results show that the two estimators produce error maps that have differences of the order of a micrometre, of some significance compared to the CMM target accuracy of 1.5 micrometres. Thus, the choice of estimator can have a significant effect on the error map produced, even for a well-designed experiment. The estimator based on (11) takes proper account of the sources of measurement error and produces relatively unbiased estimates of the parametric errors. Estimators based on distances only do not, in general, treat the measurement errors in an unbiased way.

6.3 TABLES OF RESULTS

TABLE 5

Experiment 6.1.1, calibration information A, maximum uncertainties

Max	w = 0.01	w = 0.5	w = 1	w = 2	w = 100
					4329e

Experiment 6.1.1, calibration information A, average uncertainties

Ave	w = 0.01	w = 0.5	w = 1	w = 2	w = 100
				8976e	

Experiment 6.1.1, calibration information C, maximum uncertainties

Max	w = 0.01	w = 0.5	w = 1	w = 2	w = 100
					0552e

Experiment 6.1.1, calibration information C, average uncertainties

Ave	w = 0.01	w = 0.5	w = 1	w = 2	w = 100
U1	5.0460e-06	1.3796e-07	9.6875e-08	7.0210e-08	2.0551e-09
U2	2.1274e-05	8.5309e-07	7.1663e-07	6.3907e-07	5.6739e-07
U3	1.5409e-04	4.5603e-05	4.4167e-05	4.2915e-05	4.1907e-05
U4	3.1097e-05	1.0684e-06	8.6364e-07	7.5344e-07	6.5969e-07
U5	8.1433e-06	2.1882e-07	1.4596e-07	1.0193e-07	2.9063e-09
U6	1.2825e-05	3.6998e-07	2.6834e-07	2.2210e-07	1.9339e-07

TABLE 6

Experiment 6.1.2, modified by rotation and translations, with calibration information A and B

	Maximum	Average
	A	A
U1	4.8347e-07	3.8455e-07
U2	7.8721e-07	4.5475e-07
U3	2.4525e-04	3.4082e-05
U4	6.7060e-07	4.5723e-07
U5	2.5410e-07	1.6867e-07
U6	6.1832e-07	2.2481e-07

TABLE 7

Experiment 6.2.1, varying grid size, NDEL = 5, maximum uncertainties

	n = 4	n = 5	n = 6	n = 7	n = 8	n = 9
U1	3.2399e-07	3.1249e-07	2.7810e-07	2.3464e-07	2.2560e-07	2.1522e-07
U2	5.4884e-07	5.0804e-07	4.3945e-07	3.6257e-07	3.4211e-07	3.2107e-07
U3	1.7405e-04	1.3898e-04	1.1152e-04	8.7842e-05	8.0266e-05	7.3550e-05
U4	3.8064e-07	3.6705e-07	3.2656e-07	2.7514e-07	2.6387e-07	2.5087e-07
U5	1.6025e-07	1.6500e-07	1.5539e-07	1.3771e-07	1.3951e-07	1.3923e-07
U6	2.9427e-07	2.8577e-07	2.5667e-07	2.1831e-07	2.1115e-07	2.0219e-07

Experiment 6.2.1, varying grid size, NDEL = 5, average uncertainties

	n = 4	n = 5	n = 6	n = 7	n = 8	n = 9
U1	2.2044e-07	2.1203e-07	1.8910e-07	1.6034e-07	1.5518e-07	1.4916e-07
U2	3.5304e-07	3.3025e-07	2.8728e-07	2.3790e-07	2.2508e-07	2.1167e-07
U3	2.5913e-05	2.1235e-05	1.7335e-05	1.3832e-05	1.2769e-05	1.1799e-05
U4	2.5825e-07	2.4421e-07	2.1505e-07	1.8009e-07	1.7206e-07	1.6316e-07
U5	1.2581e-07	1.2993e-07	1.2294e-07	1.0975e-07	1.1118e-07	1.1133e-07
U6	1.6174e-07	1.5069e-07	1.3243e-07	1.1121e-07	1.0671e-07	1.0165e-07

Experiment 6.2.1, varying grid size, NDEL = 7, maximum uncertainties

	n = 4	n = 5	n = 6	n = 7	n = 8	n = 9
U1	2.5510e-03	3.3570e-07	2.7954e-07	2.4355e-07		
U2	5.0560e-03					
U3	1.1616e+01					
U4	1.3037e-02					
U5	1.6945e-07					
U6	1.7399e-02					

Experiment 6.2.1, varying grid size, NDEL = 7, average uncertainties

	n = 4	n = 5	n = 6	n = 7	n = 8	n = 9
U1	2.5347e-03	2.2574e-07	1.8901e-07	1.6530e-07	1.5857e-07	1.5380e-07
U2	1.4629e-03	3.4151e-07	2.8589e-07	2.4508e-07	2.3018e-07	2.1861e-07
U3	1.0026e+00	7.8273e-05	4.2239e-05	3.0802e-05	2.6614e-05	2.4041e-05
U4	4.8187e-03	2.7825e-07	2.2150e-07	1.9028e-07	1.8010e-07	1.7248e-07
U5	1.3176e-07	1.3464e-07	1.2336e-07	1.1405e-07	1.1465e-07	1.1585e-07
U6	6.7450e-03	2.1138e-07	1.5035e-07	1.2581e-07	1.1805e-07	1.1278e-07

TABLE 8

Experiment 6.2.2, varying square size, NDEL = 5, maximum uncertainties

	n = 5	n = 7	n = 10	n = 13	n = 17	n = 21
U1	3.4844e-07	2.9278e-07	2.6568e-07	2.3223e-07	2.0859e-07	1.9308e-07
U2	5.9287e-07	4.8501e-07	4.2906e-07	3.6773e-07	3.2254e-07	2.9145e-07
U3	1.9039e-04	1.4578e-04	1.2516e-04	1.0609e-04	9.2427e-05	8.3239e-05
U4	3.9299e-07	3.2905e-07	2.9789e-07	2.5958e-07	2.3154e-07	2.1198e-07
U5	1.6763e-07	1.5246e-07	1.5523e-07	1.4754e-07	1.4345e-07	1.4023e-07
U6	2.9855e-07	2.5434e-07	2.3714e-07	2.1223e-07	1.9484e-07	1.8243e-07

Experiment 6.2.2, varying square size, NDEL = 5, average uncertainties

	n = 5	n = 7	n = 10	n = 13	n = 17	n = 21
U1	2.4107e-07	2.0276e-07	1.8508e-07	1.6297e-07	1.4758e-07	1.3723e-07
U2	3.6828e-07	3.0050e-07	2.6579e-07	2.2812e-07	2.0056e-07	1.8165e-07
U3	2.7969e-05	2.1593e-05	1.8625e-05	1.5816e-05	1.3797e-05	1.2433e-05
U4	2.8202e-07	2.3355e-07	2.0949e-07	1.8160e-07	1.6126e-07	1.4718e-07
U5	1.3249e-07	1.2145e-07	1.2273e-07	1.1666e-07	1.1401e-07	1.1224e-07
U6	1.7132e-07	1.4224e-07	1.3065e-07	1.1606e-07	1.0592e-07	9.8796e-08

Experiment 6.2.2, varying square size, NDEL = 7, maximum uncertainties

	n = 5	n = 7	n = 10	n = 13	n = 17	n = 21
U1	3.2399e-07	3.1249e-07	2.7810e-07	2.3464e-07	2.2560e-07	2.1522e-07
U2	5.4884e-07	5.0804e-07	4.3945e-07	3.6257e-07	3.4211e-07	3.2107e-07
U3	1.7405e-04	1.3898e-04	1.1152e-04	8.7842e-05	8.0266e-05	7.3550e-05
U4	3.8064e-07	3.6705e-07	3.2656e-07	2.7514e-07	2.6387e-07	2.5087e-07
U5	1.6025e-07	1.6500e-07	1.5539e-07	1.3771e-07	1.3951e-07	1.3923e-07
U6	2.9427e-07	2.8577e-07	2.5667e-07	2.1831e-07	2.1115e-07	2.0219e-07

Experiment 6.2.2, varying square size, NDEL = 7, average uncertainties

	n = 5	n = 7	n = 10	n = 13	n = 17	n = 21
U1	2.6070e-07	2.2196e-07	1.8256e-07	1.8021e-07	1.5247e-07	1.5087e-07
U2	3.8678e-07	3.2777e-07	2.6089e-07	2.5071e-07	2.0580e-07	1.9836e-07
U3	1.0532e-04	5.4961e-05	3.9344e-05	3.6858e-05	2.9864e-05	2.8581e-05
U4	3.4177e-07	2.6825e-07	2.1644e-07	2.1099e-07	1.7575e-07	1.7129e-07
U5	1.3640e-07	1.3159e-07	1.1959e-07	1.2714e-07	1.1595e-07	1.2147e-07
U6	2.6175e-07	1.7955e-07	1.4280e-07	1.4031e-07	1.1880e-07	1.1768e-07

TABLE 9

Expt. 6.2.3, varying plate dimensions, 5 x 5, NDEL = 5, maximum uncertainties

	L = 0.8	L = 0.7	L = 0.6	L = 0.5	L = 0.4	L = 0.3
U1	3.1275e-07	3.1907e-07	3.6608e-07	4.8003e-07	8.5047e-07	2.0018e-06
U2	4.6262e-07	5.4254e-07	7.3559e-07	1.1171e-06	2.4537e-06	6.6909e-06
U3	1.3077e-04	1.4033e-04	1.3376e-04	1.1844e-04	1.2570e-04	1.7318e-04
U4	3.5481e-07	3.8354e-07	4.5107e-07	5.2769e-07	9.6392e-07	2.4108e-06
U5	1.6620e-07	1.5943e-07	1.7081e-07	2.1427e-07	3.4916e-07	6.6062e-07
U6	2.6944e-07	2.9885e-07	3.4971e-07	4.6186e-07	8.3232e-07	2.1162e-06

Expt. 6.2.3, varying plate dimensions, 5 x 5, NDEL = 5, average uncertainties

	L = 0.8	L = 0.7	L = 0.6	L = 0.5	L = 0.4	L = 0.3
U1	2.1417e-07	2.1718e-07	2.4432e-07	3.0300e-07	4.9882e-07	1.0663e-06
U2	3.1100e-07	3.4927e-07	4.6834e-07	7.3674e-07	1.4742e-06	3.3698e-06
U3	2.0639e-05	2.2081e-05	2.1838e-05	2.0241e-05	2.1708e-05	2.8987e-05
U4	2.3863e-07	2.5361e-07	2.9418e-07	3.6247e-07	5.8011e-07	1.2403e-06
U5	1.3049e-07	1.2526e-07	1.3024e-07	1.3937e-07	1.7981e-07	2.8702e-07
U5	1.4410e-07	1.5548e-07	1.8402e-07	2.2678e-07	3.3998e-07	6.4167e-07

Expt. 6.2.3, varying plate dimensions, 7 x 7, NDEL = 7, maximum uncertainties

	L = 0.8	L = 0.7	L = 0.6	L = 0.5	L = 0.4	L = 0.3
			3e-07	4.6982e-07	1.0796e-06	
			1e-07			
			5e-05			
			5e-07			
			0e-07			
			9e-07			

Expt. 6.2.3, varying plate dimensions, 7 x 7, NDEL = 7, average uncertainties

	L = 0.8	L = 0.7	L = 0.6	L = 0.5	L = 0.4	L = 0.3
U1	1.7826e-07	1.8239e-07	1.9069e-07	2.9017e-07	5.9852e-07	1.2519e-06
U2	2.4768e-07	2.8137e-07	3.5095e-07	6.8566e-07	1.6005e-06	3.5541e-06
U3	3.4748e-05	3.2074e-05	3.0194e-05	3.1803e-05	3.6103e-05	4.7436e-05
U4	2.0034e-07	2.1224e-07	2.2893e-07	3.3382e-07	6.5070e-07	1.4550e-06
U5	1.2051e-07	1.1826e-07	1.1581e-07	1.4287e-07	2.1630e-07	3.3738e-07
U6	1.3013e-07	1.3842e-07	1.5006e-07	2.1300e-07	4.1653e-07	1.0118e-06

TABLE 10

Experiment 6.2.4, modified to calibrate the plate, maximum uncertainties

	5 x 5, NDEL = 5			7 x 7, NDEL = 7		
	L = 0.5	L = 0.4	L = 0.3	L = 0.5	L = 0.4	L = 0.3
U1	4.0537e-07	5.8521e-07	1.0081e-06	3.3388e-07	4.8673e-07	8.8193e-07
U2	8.6550e-07	1.6955e-06	3.6657e-06	7.2687e-07	1.4387e-06	3.2323e-06
U3	1.2436e-04	1.3417e-04	1.7746e-04	8.9825e-05	9.9648e-05	1.3438e-04
U4	5.3056e-07	7.7100e-07	1.5124e-06	4.3810e-07	6.5070e-07	1.3933e-06
U5	1.5311e-07	1.5471e-07	1.5635e-07	1.4665e-07	1.5156e-07	1.5681e-07
U6	3.9700e-07	5.6051e-07	1.0092e-06	3.4474e-07	5.1126e-07	1.0383e-06

Experiment 6.2.4, modified to calibrate the plate, average uncertainties

	5 x 5, NDEL = 5			7 x 7, NDEL = 7		
	L = 0.5	L = 0.4	L = 0.3	L = 0.5	L = 0.4	L = 0.3
U1	2.7220e-07	3.8178e-07	6.1905e-07	2.2663e-07	3.2254e-07	5.7504e-07
U2	5.8370e-07	1.0229e-06	1.9941e-06	4.7589e-07	8.7098e-07	1.9329e-06
U3	2.0583e-05	2.2233e-05	2.8037e-05	2.8252e-05	3.0048e-05	3.9832e-05
U4	3.4292e-07	4.9057e-07	8.4976e-07	2.8444e-07	4.1917e-07	8.3056e-07
U5	1.1866e-07	1.1919e-07	1.2005e-07	1.1316e-07	1.1526e-07	1.1779e-07
U6	2.1058e-07	2.9071e-07	4.7010e-07	1.8241e-07	2.7006e-07	5.6380e-07

TABLE

Experiment 6.2.5, calibration information from a different artefact

	5 x 5, NDEL = 5		7 x 7, NDEL = 7
	Max	Ave	Ave
U1	2.5717e-07	1.7596e-07	1.6712e-07
U2	3.9365e-07	2.5949e-07	2.4424e-07
U3	9.5904e-05	1.5104e-05	3.0557e-05
U4	3.0404e-07	1.9744e-07	1.9290e-07
U5	1.7544e-07	1.2994e-07	1.2946e-07
U6	2.4355e-07	1.2203e-07	1.3107e-07

TABLE 12

Experiment 6.2.6, comparison of kinematic and empirical models, 5 x 5 plate

	Empirical		Kinematic	
	Max	Ave	Max	Ave
U1	5.1994e-07	3.1721e-07	3.1249e-07	2.1203e-07
U2	6.9526e-07	5.2065e-07	5.0804e-07	3.3025e-07
U3	2.0046e-05	5.2313e-06	1.3898e-04	2.1235e-05
U4	8.7441e-07	4.0099e-07	3.6705e-07	2.4421e-07
U5	1.6234e-07	1.3755e-07	1.6500e-07	1.2993e-07
U6	4.6406e-07	2.0961e-07	2.8577e-07	1.5069e-07

Experment 6.2.6, comparison of kinematic and empirical models, 7 x 7 plate

	Empirical		Kinematic	
	Max	Ave	Max	Ave
U1	3.9924e-07	2.4724e-07	2.3464e-07	1.6034e-07
U2	5.3477e-07	3.9905e-07	3.6257e-07	2.3790e-07
U3	1.5360e-05	4.0087e-06	8.7842e-05	1.3832e-05
U4	6.7806e-07	3.1016e-07	2.7514e-07	1.8009e-07
U5	1.4810e-07	1.2350e-07	1.3771e-07	1.0975e-07
U6	3.6545e-07	1.6361e-07	2.1831e-07	1.0671e-07

6.4 DISCUSSION

The experiments above show that there are considerable advantages in adopting a measurement stratregy that is sufficient to calibrate the ball plate independently from the choice of error model for the CMM; such strategies generally involve at least one rotation and translation of the plate. For such a strategy, correlations between the ball centre locations $\{y_j\}$, the transformation parameters $\{t_k\}$ and the error parameters $\boldsymbol{a}$ are greatly reduced, leading to a more accurate determination of the parametric error function values, the object of primary interest. The need to perform potentially expensive calibrations of the plate is also eliminated; all that is required is additional calibration information to fix the gobal scale. In the numerical experiments above this has been done by supplying the four edge distances associated with the four corner spheres but could also be supplied by measuring a single-axis ball plate, i.e. two spheres with a calibrated inter-sphere distance, for example.

There are also computational gains to be made by eliminating the need to supply a large amount of calibration information for the ball plate. For example, if the calibration information involves only four of the balls, and the parameters $\{y_j\}$ are numbered so that the ones corresponding to these four balls appear rightmost in the column order of the Jacobian matrix (Section 3.5), then the incorporation of the calibration information during the upper-triangularisation of the Jacobian system involves only 12 ball centre parameters along with the transformation and parametric error parameters. By exploiting this fact, we can considerably reduce the computational and memory requirements. For example, for an $n \times n$ grid-design ball plate with calibration information given in terms of the distances between neighbouring ball centres along rows and columns (case A in experiment 6.1.1) the computational and memory requirements for incorporating the calibration information are $O(n^6)$ and $O(n^4)$, respectively. If only four distances are supplied involving four ball centres the computional and memory requirements are both $O(1)$, i.e. depend only on the number of transformation and error parameters. With this and the structure exploiting methods already described in Section 3.5 we can have solution algorithms whose computational requirements depend only on the number of measurements and are effectively independent of the number of parameters $\{y_j\}$.

7 SUMMARY AND CONCLUDING REMARKS

CMM parametric errors can be described successfully by kinematic and empirical models. Kinematic models are restricted to CMMs whose behaviour can be described by an axis-upon-axis build-up. For this class of CMM (the majority), they represent an efficient description of behaviour. Empirical models are capable of describing more general repeatable CMM behaviour and can be used in validating kinematic models.

We have described a comprehensive mathematical model for determining the parametric errors of a CMM from measurements of a ball plate and calibration information.

For the given model, the estimator described in Section 3.3 is a relatively unbiased and efficient estimator that determines the parametric errors $\boldsymbol{a}$ as well as parameters $\{y_j\}$ and $\{t_k\}$ that describe the measurement experiment.

These algorithms have been implemented in software, providing a flexible numerical simulation and model solving tool. The software has been engineered so that it can cater for any parametric error model, measurement strategy and calibration information by allowing the user to specify these in accordance with natural interface specifications. The software calculates standard uncertainties of the fitted parameters and related quantities.

This software has been used in a wide range of numerical simulations; in particular, involving the kinematic error models reported on here.

By incorporating rotations and translations of the ball plate, using multiple probe styli and an adequate sampling of the CMM working volume along with minimal calibration information, it is possible to provide accurate estimates of the CMM parametric errors.

The choice of estimator can have a significant effect on the error map produced. The estimator we have implemented has been designed to deliver unbiased estimates of the error parameters.

ACKNOWLEDGEMENTS

This work was supported by the National Measurement System Policy Unit of the Department of Trade and Industry. The work was carried out by a project team from NPL and collaborating partners. Particular thanks go to A J Abackerli, B P Butler, B J Chorley, N R Cross, D R Flack, O A Garbutt, S A Hannaby and D Rayner.

8 REFERENCES

Anthony, G. T., Anthony, H. M, Cox, M. G. and Forbes, A. B. (1991): The parametrization of geometric form. CEC report EUR 13517 EN, Luxembourg.

Anthony, G. T. and Cox, M. G. (1987): The National Physical Laboratory's Data Approximation Subroutine Library. In *Algorithms for Approximation*, Mason, J. C. and Cox, M. G., Eds., Clarendon Press, Oxford, 669-687.

Balsamo, A. (1995): Effects of arbitrary coefficients of CMM error maps on probe qualification. *Ann. CIRP*, **44**, 475-478.

Balsamo, A., Franke, M., Trapet, E., Wäldele, F., De Jonge, L. and Vankerck, P. (1997). Results of the CIRP-Euromet intercomparison of ball plate-based techniques for determining CMM parametric errors. *Ann. CIRP*, **46**(1), 463-466.

Butler, B. P., Forbes, A. B. and Kenward, P. D. (1998). Position calibration software. *NPL Report CISE 14/98.*

de Boor, C. (1972): On calculating with B-splines. *J. Approx. Theory*, **6**, 50-62.

de Boor, C. (1978): *A Practical Guide to Splines*, Springer-Verlag, New York.

Buhmann, M. D. (1993): New developments in the theory of radial basis function interpolation. Report DAMTP 1993/NA9, University of Cambridge.

Bunch, J., Dongarra, J. J., Moler, C. and Stewart, G. W. (1979): LINPACK user' s guide. SIAM, Philadelphia.

Clenshaw, C. W. and Hayes, J. G. (1965): Curve and surface fitting. *J. Inst. Math. App.* **1**, 164-183.

Cox, M. G. (1972): The numerical evaluation of B-splines. *J. Inst. Math. App.* **8**, 36-52.

Cox, M. G. (1982): Practical spline approximation. In *Lecture Notes in Mathematics* **965**: *Topics in Numerical Analysis*, Turner, P. R. Ed., Springer-Verlag, Berlin, 135-143.

Cox, M. G. (1990). Linear algebra support modules for approximation and other software. In Scientific Software Systems, J. C. Mason and M. G. Cox, Eds., Chapman & Hall, London, 1990, 21-29.

Cox, M. G., Forbes, A. B. and Peggs, G. N. (1997). CMM verification and grading. *NPL Report CLM 1.*

Cresto, P. C. (1994). Self-calibration with application to CMMs geometry error correction. In *Advanced Mathematical Tools in Metrology* (P. Ciarlini, M.G. Cox, R. Monaco and F. Pavese, Eds.), World Scientific, Singapore, 167-174.

Forbes, A. B. (1992). Position calibration software for the form assessment of complex workpieces. *NPL Report DITC 203/92.*

Forbes, A. B. (1996). Model parametrization. In *Advanced Mathematical Tools in Metrology II*, (P. Ciarlini, M.G. Cox, F. Pavese and D. Richter, Eds.), World Scientific, Singapore, 28-37.

Fletcher, R. (1980). *Practical Methods of Optimization, Volume 1, Unconstrained Optimization*, John Wiley and Sons, New York.

Forsythe, G. E. (1957): Generation and use of orthogonal polynomials for data fitting with a digital computer. *J. Soc. Indust. Appl. Math.*, **5**, 74-88.

Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical Optimization*, Academic Press, London.

Golub, G. H. (1996). The Singular Value Decomposition with applications. In *Advanced Mathematical Tools in Metrology II*, (P. Ciarlini, M.G. Cox, F. Pavese and D. Richter, Eds.), World Scientific, Singapore, 48-55.

Golub, G. H., and Van Loan. C. F. (1996). *Matrix Computations*. John Hopkins University Press, Baltimore. Third Edition.

ISO Standard, ISO 10360-2 *Performance assessment of co-ordinate measuring machines.*

ISO (1993). *Guide to the expression of uncertainties in measurement.* ISO, Geneva.

Kruth, J. P., Vanherk, P., and de Jonge, L. (1994). Self-calibration method and software error correction for three-dimensional co-ordinate measuring machines using artefact measurements. *Measurement*, December 1-11.

Kunzmann, H., Trapet, E. and Wäldele, F. (1990): A uniform concept for calibration, acceptance test and periodic inspection of co-ordinate measuring machines using reference objects. *Ann. CIRP*, 39(1), 561-564.

Kunzmann, H. and Wäldele, F. (1986): Two numerical error-correction methods for co-ordinate measuring machines. In *Software for Co-ordinate Measuring Machines*, Cox, M. G. and Peggs, G. N., Eds., National Physical Laboratory, Teddington, 8-14.

MathWorks, Inc., (1993). *Matlab: High-Performance Numeric Computation and Visualization Software, User's Guide*. Natick, Mass.

Paige, C. C. and Saunders, M. A. (1982). LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Soft.* **8** (1), 43-71

Powell, M. J. D. (1993): The uniform convergence of thin plate spline interpolation in two dimensions. Report DAMTP 1993/NA16, University of Cambridge.

Soons, J. A., Theuws, F. C. and Schellekens, P. H. J. (1992): Modelling the errors of multi-axis machines: a general methodology, *Precision Eng.*, **14**, No. 1, 5-19.

Trapet, E. and Wäldele, F. (1989): Determination of the parametric errors of co-ordinate measuring machines and machine tools using a reference object, *VDI Berichte*, No 761, 163-175.

Zhang, G., Ouyang, R., Lu, B., Hocken, R., Veale, R. and Donmez, A. (1988): A displacement method for machine geometry calibration, *Annals of the CIRP*, **37**, 515-518.