Transactions on Engineering Sciences vol 44, © 2003 WIT Press, www.witpress.com, ISSN 1743-3533

Methodology and simulation of the calibration of a five-axis machine tool link geometry and motion errors using polynomial modelling and a telescoping magnetic ball-bar

Y. Abbaszadeh-Mir, J.R.R. Mayer & C. Fortin
École Polytechnique, Canada

Abstract

Many parameters influence the positioning errors of machine tools. Amongst them are the link geometric errors that describe the relative location between the machine's successive rotary and prismatic joints and, the motion errors, describing the non-ideal motion of a joint. A method is proposed that uses a telescoping magnetic ball-bar to indirectly calibrate these errors on a five-axis machine tool. The motion error sources are individually modelled as polynomial functions of each joint coordinates. This modelling method conveniently incorporates the link geometric errors. Once the polynomial coefficients are identified, it is then possible to predict the tool versus workpiece position and orientation errors resulting from these error sources.

1 Introduction

Many parameters influence the positioning errors of machined tools. Amongst them are two groups of geometric error parameters causing tool positioning errors. The first group includes the link geometric errors, which describe the relative location between the machine's successive rotary and prismatic joints. Examples of these are squareness errors, parallelism errors, rotary joint offsets, and rotary axes separation errors. The second group includes the motion errors, describing the non-ideal motion of a joint (either of the X, Y, Z, A, B or C axis). Examples of these are scale errors, straightness errors and angular errors such as

yaw, pitch, and roll for prismatic axes, and, the angular, axial, radial and tilt errors for rotary axes.

Much research activity has been conducted to define, describe and identify the geometric inaccuracies of a machine tool and enhance its performance through compensation in order to produce accurate parts. Standards are available but most methods are quite demanding in terms of cost, time and skill in order to realise the measurements. In recent years, the telescoping magnetic ball-bar (TMBB) instrument has been used to measure inaccuracies of machine tools by analysing the tool-tip position relative to the part table for specific paths. The TMBB measures the compounded volumetric effect projected in the radial direction of a circular path caused by a number of motion and link geometric errors as well as other error sources [1], [2], [3], [4], [5], [9] and [6]. In 1994, Hai et al [7] applied a TMBB to measure three-axis machine tool motion errors using polynomials. This model succeeds in identifying all coefficients by repeating the circular test on several parallel planes for each pair of axes. In 1997, Pahk [8] used a TMBB and developed a polynomial concept to identify a three-axis machine tool errors. Pahk used polynomials to represent the geometric errors, its work concentrated on three-axis machine tools and modelled the error parameters as dependent polynomial functions. For example, Pahk represented the positioning error as polynomials without the zero terms, and angular errors as derivatives of straightness.

The use of a fixed length ball bar for the calibration of the misalignment, axis separation and joint offsets of rotary axes was demonstrated by Driels [10] for a robot made of six rotary axes modeled using a modified Denavit-Hartenberg (D–H) formulation.

In a previous paper we proposed and numerically simulated a method to calibrate the link geometric errors of a 5 axis machine tool using a TMBB [13] and [14]. We now present an extension of this approach to model and identify five axis machine tool geometric errors using independent polynomial functions and ball-bar readings not restricted to circular paths. The considered machine tool errors include both the link geometric and the joint motion errors.

2 Five-axis machine tool forward model

The kinematic model of a machine tool is used to calculate the relative location between a tool frame *t* rigidly attached to the tool attachment point and a part frame *w* rigidly attached to the table . From the base or foundation frame F we define a tool branch and a workpiece feature branch using their homogenous transformation matrix (HTM) ${}^{F}\boldsymbol{T}_{t}$ and ${}^{F}\boldsymbol{T}_{w}$ respectively. Thus, the pose of t relative to w is given as

$$ {}^{w}\boldsymbol{T}_{t} = [{}^{F}\boldsymbol{T}_{w}]^{-1} \cdot {}^{F}\boldsymbol{T}_{t} \tag{1} $$

For a machine tool with a serial architecture, each branch is modelled by a series of HTMs. For example the tool branch of a machine with a ZFYXAC topology is modelled as

$$^{F}\boldsymbol{T}_{t} = \boldsymbol{D}_{z}\boldsymbol{T}_{t} \tag{2}$$

and the workpiece feature branch as

$$^{F}\boldsymbol{T}_{w} = \boldsymbol{D}_{y}\boldsymbol{D}_{x}\boldsymbol{D}_{a}\boldsymbol{D}_{c}\boldsymbol{T}_{w} \tag{3}$$

where the $\boldsymbol{D}$ matrices are the individual link-joint combinations.

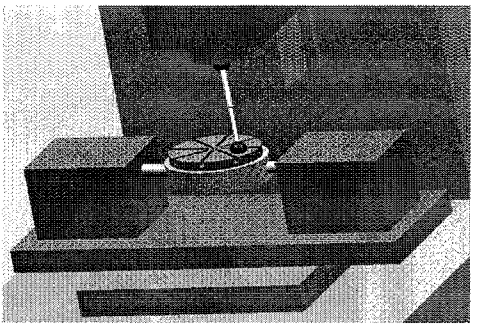

Figure 1 : Matsuura machine tool with a ball-bar set-up.

Figure 1 shows the corresponding machine solid model whereas Figure 2 illustrates the kinematic model. A particular tool versus workpiece feature location corresponds to a particular set of joint coordinate values as defined by $\theta = [z\ y\ x\ a\ c]^{T}$. At $\theta = [0\ 0\ 0\ 0\ 0]^{T}$ the reference frames of all joints coincide (see Figure 2).

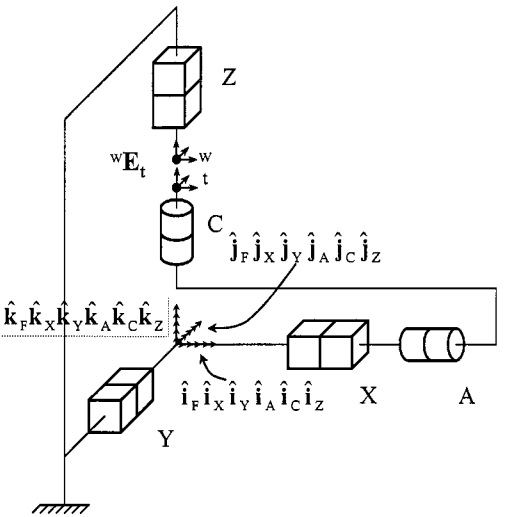

Figure 2 : Joint frames at zero coordinate $\theta=[0\ 0\ 0\ 0\ 0]^{T}$.

A link–joint combination such as for the non existent U–axis $\boldsymbol{D}_U$ can be modelled by the product of 3 HTMs (Figure 3).

$$\boldsymbol{D}_U = \boldsymbol{D}_{1U}\,\boldsymbol{D}_{2U}\,\boldsymbol{D}_{3U}. \tag{4}$$

$\boldsymbol{D}_{1U}$ is the nominal link geometry HTM defined as:

$$\boldsymbol{D}_{1U} = \boldsymbol{trans}(\hat{\imath}, a_x(u))\ \boldsymbol{trans}(\hat{\jmath}, a_y(u))\ \boldsymbol{trans}(\hat{k}, a_z(u))\ \boldsymbol{rot}(\hat{k}, \alpha_z(u))\ \boldsymbol{rot}(\hat{\jmath}, \alpha_y(u))\ \boldsymbol{rot}(\hat{\imath}, \alpha_x(u)) \tag{5}$$

where for example ***trans***(**î**, $a_x(u)$) is a HTM for a linear transformation by a nominal value $a_x(u)$ along the **î** direction; ***rot***(**î**, $\alpha_x(u)$) is a HTM for a rotation by a nominal value $\alpha_x(u)$ around **î**. For the target Matsuura machine tool U represents either of Z, Y, X, A, or C and all nominal values are null.

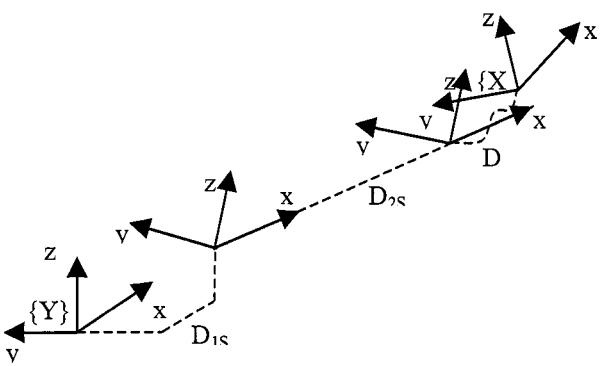

Figure 3 : Sub-HTM of each link-joint transformation.

The joint HTM $\boldsymbol{D}_{2U}$ describes the nominal motion of the joint frame,
$\boldsymbol{D}_{2U}$ = ***trans***(**dir**, *u*) for prismatic joint; $\boldsymbol{D}_{2S}$ = ***rot***(**dir**, *u*) for a rotary joint (6)
where for prismatic joints **dir** is either of **î**, $\hat{\mathbf{j}}$ or $\hat{\mathbf{k}}$ and *u* (the joint coordinate) is *x*, *y* or *z*, for X, Y or Z respectively. Whereas for rotary joints **dir** is **î** for A, and $\hat{\mathbf{k}}$ for C, and *u* is *a* or *c*, the displacement angle.

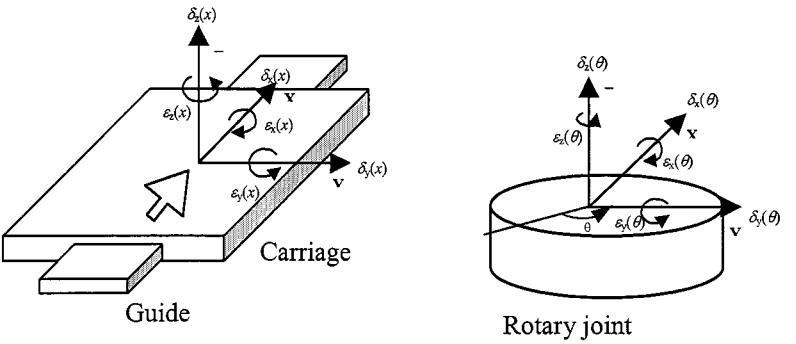

Figure 4 : Linear and angular errors for prismatic and rotary joints.

$\boldsymbol{D}_{3U}$ describes the joint motion errors (Figure 4),

$$\boldsymbol{D}_{3U} = \boldsymbol{trans}(\hat{i}, \delta_x(u))\ \boldsymbol{trans}(\hat{j}, \delta_y(u))\ \boldsymbol{trans}(\hat{k}, \delta_z(u))\ \boldsymbol{rot}(\hat{k}, \varepsilon_z(u))\ \boldsymbol{rot}(\hat{j}, \varepsilon_y(u))\ \boldsymbol{rot}(\hat{i}, \varepsilon_x(u)) \tag{7}$$

where $\delta_x(u)$, $\delta_y(u)$, and $\delta_z(u)$ are the linear errors as functions of the joint coordinate s along joint S, and similarly, $\varepsilon_x(u)$, $\varepsilon_y(u)$, and $\varepsilon_z(u)$ are the angular errors (see Figure 4). Considering the three prismatic joints of a five-axis machine tool there are 18 such motion errors and 3 link geometric errors i.e. one squareness error between the Y and X axes, and two squareness errors for the Z axis. Furthermore the geometric errors of the two rotary axis of the five axis machine tool are normally modelled using 12 motion errors and 5 link geometric errors. For a ZFYXAC machine topology, these normally include two alignment errors between the A and the X axes, one angular offset for the A axis, one squareness error between the C and A axes and 1 axis to axis distance between axes A and C. All 30 motion errors are functions of the joint coordinate and in this work are represented using 30 Chebyshev polynomial functions of joint coordinatesREF * MERGEFORMAT . This polynomial has orthogonal characteristics and can generate a better condition number for the sensitivity Jacobian matrix. The zero–order terms which are independent of the joint coordinate and some linear terms can represent the 8 link geometric errors [14].

The difference between the tool tip frame and the workpiece feature frame, ${}^{W}\boldsymbol{E}_t$ (see Figure 2) maybe represented as an HTM and calculated as follows:

$${}^{F}\mathbf{T}_t = {}^{F}\mathbf{T}_w\ {}^{W}\mathbf{E}_t \tag{8}$$

and so

$${}^{W}\boldsymbol{E}_t = [{}^{F}\boldsymbol{T}_w]^{-1}\ {}^{F}\boldsymbol{T}_t\,. \tag{9}$$

The HTMs of the machine kinematic model will be used to: a) simulate an imperfect machine and generate virtual measurement data for validation purposes, b) generate the sensitivity Jacobian matrix, which expresses the sensitivity of the ball-bar readings to the error parameters and c) predict the location errors of the tool-tip versus the part feature for a given set of joint coordinates, tool and workpiece geometry and machine geometric errors.

3 Sensitivity Jacobian matrix generation

From the nominal link geometry and joint coordinates at each pose, the sensitivity Jacobian matrix, $\boldsymbol{J}$, is obtained using Newton-Euler's equation. This matrix determines the changes in the tool versus workpiece feature location $\boldsymbol{\delta\tau}$ resulting from changes in the machine error parameters $\boldsymbol{\delta p}$.

$$\boldsymbol{\delta\tau} = \boldsymbol{J}\,\boldsymbol{\delta p} \tag{10}$$

The sensitivity Jacobian matrix is built using transport matrices. A transport matrix is generated using a HTM and propagates the effect of an error twist $[\delta_x\ \delta_y\ \delta_z\ \varepsilon_x\ \varepsilon_y\ \varepsilon_z]^T$ occurring in one reference frame onto another frame rigidly connected to the former

$$^{\{B\}}\boldsymbol{\delta\tau}_B = {}^B_A\boldsymbol{C}\,{}^{\{A\}}\boldsymbol{\delta\tau}_A \tag{11}$$

where $^{\{B\}}\boldsymbol{\delta\tau}_B$ is the propagated twist (small displacement) at B, expressed in frame B, $^{\{A\}}\boldsymbol{\delta\tau}_A$ is the causal twist in A, expressed in frame A, and ${}^B_A\boldsymbol{C}$ is a 6-by-6 transport matrix [11], [12]. The transport matrix can be formed using sub-matrices of an HTM as follows :

$$if\ {}^A_B\boldsymbol{T} = \begin{bmatrix} {}^A_B\boldsymbol{R} & {}^{\{A\},A}\boldsymbol{P}_B \\ 000 & 1 \end{bmatrix} \Rightarrow {}^B_A\boldsymbol{C} = \begin{bmatrix} {}^A_B\boldsymbol{R}^T & {}^A_B\boldsymbol{R}^T\left[{}^{\{A\},A}\boldsymbol{P}_B\times\right]^T \\ 0\ 0\ 0 & \\ 0\ 0\ 0 & {}^A_B\boldsymbol{R}^T \\ 0\ 0\ 0 & \end{bmatrix}. \tag{12}$$

The transport matrix is a matrix form of the based on the Newton–Euler equations. It produces the well known rigid body kinematics effects and is built as follows. (a) A rotation at one point of a rigid body produces an equal orientation change at any other point of the same body. (b) A translation at one point of a rigid body produces an equal translation at any other point of the same body; (c) a translation at one point of a rigid body produces no rotation at any other point of the same body and finally (d) a small rotation at a point of a rigid body produces a small translation at any other point of that body depending on

the cross product of that rotation vector and the position vector originating at the point where the rotation occurs to the point where the resulting translational effect is analysed. The rotation matrices within the transport matrix are used to express the resulting twist in the reference frame where it occurs.

Since the tool path information (cutter location file) is presented in the tool frame, the error should be transformed to the tool-frame. For the machine considered, this can be accomplished by forming a sensitivity Jacobian matrix as follows :

$$ {}^{\{t\},f}\boldsymbol{\delta\tau}_t = \boldsymbol{J}\cdot\boldsymbol{\delta p} \tag{13}$$

where ${}^{\{t\},f}\boldsymbol{\delta\tau}_t$ is a twist with size 6*m*-by-1, $\boldsymbol{J}$ is the sensitivity Jacobian matrix with size (6*m*)-by-*n*, and $\boldsymbol{\delta p}$ is the error parameter column matrix with *n* elements. The number of poses and error parameters are *m* and *n* respectively.

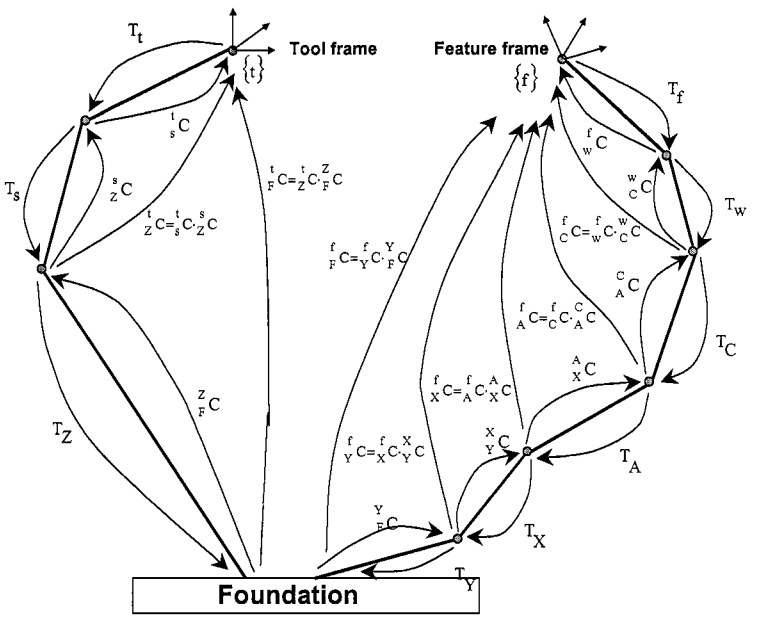

Figure 5 : Twist's origin and effect frames.

For example for a ZFYXAC machine the sensitivity Jacobian matrix is

$$ {}^{\{t\},f}_{t}\boldsymbol{J} = \left[-{}^{t}_{F}\boldsymbol{C}\cdot{}^{f}_{F}\boldsymbol{C}^{-1}\cdot{}^{\{f\},F}_{f}\boldsymbol{J} \quad {}^{\{t\},F}_{t}\boldsymbol{J} \right] \tag{14}$$

where

$$ {}^{\{f\},F}_{f}\boldsymbol{J} = \begin{bmatrix} {}^{f}_{F}\boldsymbol{C} & {}^{f}_{Y}\boldsymbol{C} & {}^{f}_{X}\boldsymbol{C} & {}^{f}_{A}\boldsymbol{C} & {}^{f}_{C}\boldsymbol{C} & {}^{f}_{w}\boldsymbol{C} & {}^{f}_{f}\boldsymbol{C} \end{bmatrix} \tag{15} $$

$$ {}^{\{t\},F}_{t}\boldsymbol{J} = \begin{bmatrix} {}^{t}_{F}\boldsymbol{C} & {}^{t}_{Z}\boldsymbol{C} & {}^{t}_{s}\boldsymbol{C} & {}^{t}_{t}\boldsymbol{C} \end{bmatrix} \tag{16} $$

and

$$ {}^{f}_{F}\boldsymbol{C}^{-1} = {}^{F}_{f}\boldsymbol{C} \tag{17} $$

In eqn (14), ${}^{\{f\},F}_{f}\boldsymbol{J}$ expresses the sensitivity of the feature frame location f observed from the foundation frame F and expressed in the feature frame f to causal error parameters within the feature branch. ${}^{\{t\},F}_{t}\boldsymbol{J}$ is defined in a similar manner. The two $\boldsymbol{C}$ matrices are used to combine the two branches into one continuous chain from the tool to the feature. In eqns (15) and (16) ${}^{f}_{f}\boldsymbol{C}$ and ${}^{t}_{t}\boldsymbol{C}$ are I_{6x6} and other elements are calculated as follows:

$$ \begin{aligned} &{}^{f}_{w}\boldsymbol{C} = {}^{f}_{f}\boldsymbol{C}\,{}^{f}_{w}\boldsymbol{C};\ {}^{f}_{C}\boldsymbol{C} = {}^{f}_{w}\boldsymbol{C}\cdot{}^{w}_{C}\boldsymbol{C};\ {}^{f}_{A}\boldsymbol{C} = {}^{f}_{C}\boldsymbol{C}\,{}^{C}_{A}\boldsymbol{C};\\ &{}^{f}_{X}\boldsymbol{C} = {}^{f}_{A}\boldsymbol{C}\,{}^{A}_{X}\boldsymbol{C};\ {}^{f}_{Y}\boldsymbol{C} = {}^{f}_{X}\boldsymbol{C}\,{}^{X}_{Y}\boldsymbol{C};\ {}^{f}_{F}\boldsymbol{C} = {}^{f}_{Y}\boldsymbol{C}\,{}^{Y}_{F}\boldsymbol{C}; \end{aligned} \tag{18} $$

Figure 5 shows the action of transport matrices for each axis and for both branches. For the C arrows, the origin of the arrow indicates the frame where the causal error parameters occur. The arrow tip indicates the frame where the resulting effect is propagated and calculated.

3.1 Generation of the extended sensitivity Jacobian Matrix for Coefficients

The extended sensitivity Jacobian matrix, $\boldsymbol{J}_e$ represents the sensitivity of the tool versus feature pose error to changes in the Chebyshev polynomial coefficients and it now has 30×4+12=132 columns.

From eqn (13) and the definition of a polynomial the first partial derivatives of a particular tool versus pose error twist component τ_i with respect to the Chebychev coefficients T_k are:

$$ \frac{\partial \tau_i}{\partial c_k} = \frac{\partial \tau_i}{\partial p_j(u)} \cdot \frac{\partial p_j(u)}{\partial c_k} = \frac{\partial \tau_i}{\partial p_j(u)} \cdot T_k(u); \quad i = 1,6m; \quad j = 1,30; \quad k = 0,3. \tag{19} $$

For example the sensitivity term of twist τ_i to the motion error $\delta_x(y)$ is replaced as follows

$$\frac{\partial \tau_i}{\partial \delta_x(y)} \Rightarrow \frac{\partial \tau_i}{\partial \delta_x(y)} \cdot \left[\delta_x(y)_0 \quad \delta_x(y)_1 \quad \delta_x(y)_2 \quad \delta_x(y)_3\right]. \tag{20}$$

4 Minimal and complete set of model variables

This initial model is referred to as the maximal-complete model because it can fully represent the geometric errors sources (of a cubic nature) but may contain an excessive number of variables. In Table 1, column **a** shows these error variables. There are 30x4 coefficients representing the machine tool error parameters and 12 location errors of the part on the table and the tool on the spindle (6 for each). The $\boldsymbol{J}_e$ matrix is ill-conditioned. Hereafter, we explain the variable reduction process of this model towards a minimal-complete model which can still completely explain the tool versus part feature pose errors. The ill-conditioning of $\boldsymbol{J}_e$ indicates that some columns are linear combinations of others. The variables corresponding to those columns form a confounded variable group and removing at least one variable of such a group is necessary. The $\boldsymbol{J}_e$ matrix columns corresponding to the removed variables are also removed to generate the reduced sensitivity Jacobian matrix $\boldsymbol{J}_r$.

4.1 Mathematical analysis of the extended sensitivity Jacobian matrix

A number of matrix properties are now proposed to reduce the number of variables. Using the notions of rank, condition number, size of a matrix, and the singular value decomposition (SVD), we can analyse whether or not a matrix is singular, how many independent columns exist, what columns (and so which variables) are dependent on each other, and to what degree of dependence.

To select the parameter or coefficients to be removed from a confounded group, the following strategies are used in the order listed.

I Part and tool related parameters

The error parameters pertaining to the tool and part locations are systematically kept, since they are often determined during a set-up procedure.

II Link geometric error parameters

Within a group of confounded variables, those coefficients corresponding to link geometric error definitions are systematically kept during the model reduction because they correspond to machine error parameters well recognised in the field of machine tools. For example the chosen coefficient corresponding to the squareness error X-Y is the zero order coefficient of the Y axis, $\varepsilon_z(y)_0$, see Figure 6. An alternative choice could have been $\delta_y(x)_1$.

Table 1: Summary of the variables in relation to the maximal-complete and minimal-complete models.

Axis	All axis and set-up error terms for a 5-axes machine tool	**a** Maximal-complete model variables	**b** The coefficients removed	**c** Minimal-complete model variables	**d** Coefficient corresponding to link errors	**e** Coefficient corresponding to motion errors
Y	$\delta_{x,Y}$	$\delta_x(y)_0, \delta_x(y)_1, \delta_x(y)_2, \delta_x(y)_3$	$\delta_x(y)_0$, $\delta_x(y)_1$	$\delta_x(y)_2, \delta_x(y)_3$		$\delta_x(y)_2, \delta_x(y)_3$
	$\delta_{y,Y}$	$\delta_y(y)_0, \delta_y(y)_1, \delta_y(y)_2, \delta_y(y)_3$	$\delta_y(y)_0$	$\delta_y(y)_1, \delta_y(y)_2, \delta_y(y)_3$		$\delta_y(y)_1, \delta_y(y)_2, \delta_y(y)_3$
	$\delta_{z,Y}$	$\delta_z(y)_0, \delta_z(y)_1, \delta_z(y)_2, \delta_z(y)_3$	$\delta_z(y)_0$, $\delta_z(y)_1$	$\delta_z(y)_2, \delta_z(y)_3$		$\delta_z(y)_2, \delta_z(y)_3$
	$\varepsilon_{x,Y}$	$\varepsilon_x(y)_0, \varepsilon_x(y)_1, \varepsilon_x(y)_2, \varepsilon_x(y)_3$	$\varepsilon_x(y)_0$	$\varepsilon_x(y)_1, \varepsilon_x(y)_2, \varepsilon_x(y)_3$		$\varepsilon_x(y)_1, \varepsilon_x(y)_2, \varepsilon_x(y)_3$
	$\varepsilon_{y,Y}$	$\varepsilon_y(y)_0, \varepsilon_y(y)_1, \varepsilon_y(y)_2, \varepsilon_y(y)_3$	$\varepsilon_y(y)_0$	$\varepsilon_y(y)_1, \varepsilon_y(y)_2, \varepsilon_y(y)_3$		$\varepsilon_y(y)_1, \varepsilon_y(y)_2, \varepsilon_y(y)_3$
	$\varepsilon_{z,Y}$	$\varepsilon_z(y)_0, \varepsilon_z(y)_1, \varepsilon_z(y)_2, \varepsilon_z(y)_3$		$\varepsilon_z(y)_0, \varepsilon_z(y)_1, \varepsilon_z(y)_2, \varepsilon_z(y)_3$	$\varepsilon_z(y)_0$	$\varepsilon_z(y)_1, \varepsilon_z(y)_2, \varepsilon_z(y)_3$
X	$\delta_{x,X}$	$\delta_x(x)_0, \delta_x(x)_1, \delta_x(x)_2, \delta_x(x)_3$	$\delta_x(x)_0$	$\delta_x(x)_1, \delta_x(x)_2, \delta_x(x)_3$		$\delta_x(x)_1, \delta_x(x)_2, \delta_x(x)_3$
	$\delta_{y,X}$	$\delta_y(x)_0, \delta_y(x)_1, \delta_y(x)_2, \delta_y(x)_3$	$\delta_y(x)_0$, $\delta_y(x)_1$	$\delta_y(x)_2, \delta_y(x)_3$		$\delta_y(x)_2, \delta_y(x)_3$
	$\delta_{z,X}$	$\delta_z(x)_0, \delta_z(x)_1, \delta_z(x)_2, \delta_z(x)_3$	$\delta_z(x)_0$, $\delta_z(x)_1$	$\delta_z(x)_2, \delta_z(x)_3$		$\delta_z(x)_2, \delta_z(x)_3$
	$\varepsilon_{x,X}$	$\varepsilon_x(x)_0, \varepsilon_x(x)_1, \varepsilon_x(x)_2, \varepsilon_x(x)_3$		$\varepsilon_x(x)_0, \varepsilon_x(x)_1, \varepsilon_x(x)_2, \varepsilon_x(x)_3$	$\varepsilon_x(x)_0$	$\varepsilon_x(x)_1, \varepsilon_x(x)_2, \varepsilon_x(x)_3$
	$\varepsilon_{y,X}$	$\varepsilon_y(x)_0, \varepsilon_y(x)_1, \varepsilon_y(x)_2, \varepsilon_y(x)_3$		$\varepsilon_y(x)_0, \varepsilon_y(x)_1, \varepsilon_y(x)_2, \varepsilon_y(x)_3$	$\varepsilon_y(x)_0$	$\varepsilon_y(x)_1, \varepsilon_y(x)_2, \varepsilon_y(x)_3$
	$\varepsilon_{z,X}$	$\varepsilon_z(x)_0, \varepsilon_z(x)_1, \varepsilon_z(x)_2, \varepsilon_z(x)_3$		$\varepsilon_z(x)_0, \varepsilon_z(x)_1, \varepsilon_z(x)_2, \varepsilon_z(x)_3$	$\varepsilon_z(x)_0$	$\varepsilon_z(x)_1, \varepsilon_z(x)_2, \varepsilon_z(x)_3$
A	$\delta_{x,A}$	$\delta_x(a)_0, \delta_x(a)_1, \delta_x(a)_2, \delta_x(a)_3$	$\delta_x(a)_0$	$\delta_x(a)_1, \delta_x(a)_2, \delta_x(a)_3$		$\delta_x(a)_1, \delta_x(a)_2, \delta_x(a)_3$
	$\delta_{y,A}$	$\delta_y(a)_0, \delta_y(a)_1, \delta_y(a)_2, \delta_y(a)_3$		$\delta_y(a)_0, \delta_y(a)_1, \delta_y(a)_2, \delta_y(a)_3$	$\delta_y(a)_0$	$\delta_y(a)_1, \delta_y(a)_2, \delta_y(a)_3$
	$\delta_{z,A}$	$\delta_z(a)_0, \delta_z(a)_1, \delta_z(a)_2, \delta_z(a)_3$	$\delta_z(a)_0$	$\delta_z(a)_1, \delta_z(a)_2, \delta_z(a)_3$		$\delta_z(a)_1, \delta_z(a)_2, \delta_z(a)_3$
	$\varepsilon_{x,A}$	$\varepsilon_x(a)_0, \varepsilon_x(a)_1, \varepsilon_x(a)_2, \varepsilon_x(a)_3$	$\varepsilon_x(a)_0$	$\varepsilon_x(a)_1, \varepsilon_x(a)_2, \varepsilon_x(a)_3$		$\varepsilon_x(a)_1, \varepsilon_x(a)_2, \varepsilon_x(a)_3$
	$\varepsilon_{y,A}$	$\varepsilon_y(a)_0, \varepsilon_y(a)_1, \varepsilon_y(a)_2, \varepsilon_y(a)_3$		$\varepsilon_y(a)_0, \varepsilon_y(a)_1, \varepsilon_y(a)_2, \varepsilon_y(a)_3$	$\varepsilon_y(a)_0$	$\varepsilon_y(a)_1, \varepsilon_y(a)_2, \varepsilon_y(a)_3$
	$\varepsilon_{z,A}$	$\varepsilon_z(a)_0, \varepsilon_z(a)_1, \varepsilon_z(a)_2, \varepsilon_z(a)_3$	$\varepsilon_z(a)_0$	$\varepsilon_z(a)_1, \varepsilon_z(a)_2, \varepsilon_z(a)_3$		$\varepsilon_z(a)_1, \varepsilon_z(a)_2, \varepsilon_z(a)_3$
C	$\delta_{x,C}$	$\delta_x(c)_0, \delta_x(c)_1, \delta_x(c)_2, \delta_x(c)_3$	$\delta_x(c)_0$	$\delta_x(c)_1, \delta_x(c)_2, \delta_x(c)_3$		$\delta_x(c)_1, \delta_x(c)_2, \delta_x(c)_3$
	$\delta_{y,C}$	$\delta_y(c)_0, \delta_y(c)_1, \delta_y(c)_2, \delta_y(c)_3$	$\delta_y(c)_0$	$\delta_y(c)_1, \delta_y(c)_2, \delta_y(c)_3$		$\delta_y(c)_1, \delta_y(c)_2, \delta_y(c)_3$
	$\delta_{z,C}$	$\delta_z(c)_0, \delta_z(c)_1, \delta_z(c)_2, \delta_z(c)_3$	$\delta_z(c)_0$	$\delta_z(c)_1, \delta_z(c)_2, \delta_z(c)_3$		$\delta_z(c)_1, \delta_z(c)_2, \delta_z(c)_3$
	$\varepsilon_{x,C}$	$\varepsilon_x(c)_0, \varepsilon_x(c)_1, \varepsilon_x(c)_2, \varepsilon_x(c)_3$	$\varepsilon_x(c)_0$	$\varepsilon_x(c)_1, \varepsilon_x(c)_2, \varepsilon_x(c)_3$		$\varepsilon_x(c)_1, \varepsilon_x(c)_2, \varepsilon_x(c)_3$
	$\varepsilon_{y,C}$	$\varepsilon_y(c)_0, \varepsilon_y(c)_1, \varepsilon_y(c)_2, \varepsilon_y(c)_3$	$\varepsilon_y(c)_0$	$\varepsilon_y(c)_1, \varepsilon_y(c)_2, \varepsilon_y(c)_3$		$\varepsilon_y(c)_1, \varepsilon_y(c)_2, \varepsilon_y(c)_3$
	$\varepsilon_{z,C}$	$\varepsilon_z(c)_0, \varepsilon_z(c)_1, \varepsilon_z(c)_2, \varepsilon_z(c)_3$	$\varepsilon_z(c)_0$	$\varepsilon_z(c)_1, \varepsilon_z(c)_2, \varepsilon_z(c)_3$		$\varepsilon_z(c)_1, \varepsilon_z(c)_2, \varepsilon_z(c)_3$
w	$\delta_{x,w}$	$\delta_x(w)$		$\delta_x(w)$		
	$\delta_{y,w}$	$\delta_y(w)$		$\delta_y(w)$		
	$\delta_{z,w}$	$\delta_z(w)$		$\delta_z(w)$		
	$\varepsilon_{x,w}$	$\varepsilon_x(w)$		$\varepsilon_x(w)$		
	$\varepsilon_{y,w}$	$\varepsilon_y(w)$		$\varepsilon_y(w)$		
	$\varepsilon_{z,w}$	$\varepsilon_z(w)$		$\varepsilon_z(w)$		
Z	$\delta_{x,Z}$	$\delta_x(z)_0, \delta_x(z)_1, \delta_x(z)_2, \delta_x(z)_3$	$\delta_x(z)_0$	$\delta_x(z)_1, \delta_x(z)_2, \delta_x(z)_3$	$\delta_x(z)_1$	$\delta_x(z)_2, \delta_x(z)_3$
	$\delta_{y,Z}$	$\delta_y(z)_0, \delta_y(z)_1, \delta_y(z)_2, \delta_y(z)_3$	$\delta_y(z)_0$	$\delta_y(z)_1, \delta_y(z)_2, \delta_y(z)_3$	$\delta_y(z)_1$	$\delta_y(z)_2, \delta_y(z)_3$
	$\delta_{z,Z}$	$\delta_z(z)_0, \delta_z(z)_1, \delta_z(z)_2, \delta_z(z)_3$	$\delta_z(z)_0$	$\delta_z(z)_1, \delta_z(z)_2, \delta_z(z)_3$		$\delta_z(z)_1, \delta_z(z)_2, \delta_z(z)_3$
	$\varepsilon_{x,Z}$	$\varepsilon_x(z)_0, \varepsilon_x(z)_1, \varepsilon_x(z)_2, \varepsilon_x(z)_3$	$\varepsilon_x(z)_0$	$\varepsilon_x(z)_1, \varepsilon_x(z)_2, \varepsilon_x(z)_3$		$\varepsilon_x(z)_1, \varepsilon_x(z)_2, \varepsilon_x(z)_3$
	$\varepsilon_{y,Z}$	$\varepsilon_y(z)_0, \varepsilon_y(z)_1, \varepsilon_y(z)_2, \varepsilon_y(z)_3$	$\varepsilon_y(z)_0$	$\varepsilon_y(z)_1, \varepsilon_y(z)_2, \varepsilon_y(z)_3$		$\varepsilon_y(z)_1, \varepsilon_y(z)_2, \varepsilon_y(z)_3$
	$\varepsilon_{z,Z}$	$\varepsilon_z(z)_0, \varepsilon_z(z)_1, \varepsilon_z(z)_2, \varepsilon_z(z)_3$	$\varepsilon_z(z)_0$	$\varepsilon_z(z)_1, \varepsilon_z(z)_2, \varepsilon_z(z)_3$		$\varepsilon_z(z)_1, \varepsilon_z(z)_2, \varepsilon_z(z)_3$
t	$\varepsilon_{x,t}$	$\varepsilon_x(t)$		$\varepsilon_x(t)$		
	$\varepsilon_{y,t}$	$\varepsilon_y(t)$		$\varepsilon_y(t)$		
	$\varepsilon_{z,t}$	$\varepsilon_z(t)$		$\varepsilon_z(t)$		
	$\varepsilon_{x,t}$	$\varepsilon_x(t)$		$\varepsilon_x(t)$		
	$\varepsilon_{y,t}$	$\varepsilon_y(t)$		$\varepsilon_y(t)$		
	$\varepsilon_{z,t}$	$\varepsilon_z(t)$		$\varepsilon_z(t)$		

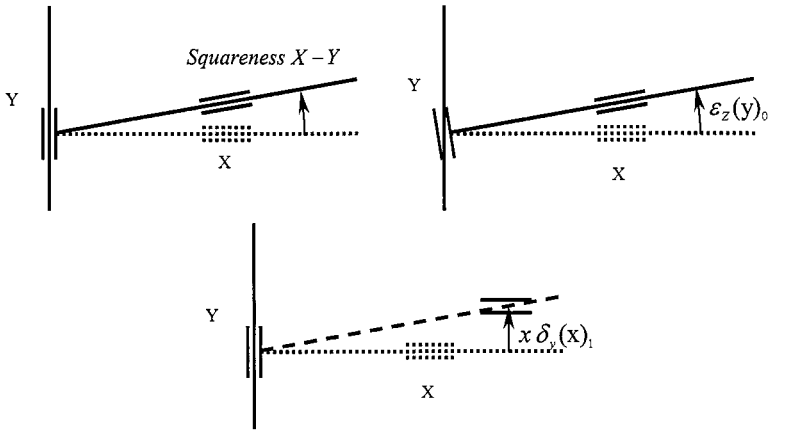

Figure 6 : Correspondence between the squareness X-Y and $\varepsilon_z(y)_0$ or $x\delta_y(x)_1$

III The condition number of the extended sensitivity Jacobian matrix

We remove the coefficient leading to the smallest conditioning number.

4.2 Selection procedure

A Matlab® simulation was used to generate a series of 600 machine configurations as well as the associated extended sensitivity Jacobian matrix which had the following characteristics : Dimensions=(600×6)-by-132; Rank=104; Conditioning number = ∞.

In Table 1, column **a** shows all error parameters and coefficients, column **b** shows those parameters removed, column **c** shows all parameters and coefficients to be identified, column **d** shows the coefficients of column **c** representing link geometric errors, and column **e** shows the coefficients of column **c** representing motion errors. There are 104 parameters and coefficients to be identified (including 92 coefficients plus 2×6 errors for feature and tool mounting locations).

The numerical simulation shows that the minimal-complete model consists in 104 variables (parameters and coefficients) when using a cubic polynomial representation.

4.3 Remarks on the number of coefficients and parameters

To calibrate a multi-axis machine tool modelled as polynomial functions for its geometric errors parameters, the minimal number of parameters and coefficients N_{min} to be identified can be calculated from the following equations:

$$N_{min} = N_L + N_M \tag{21}$$

The number of link geometric error parameter N_L was established by Everett [15]:

$$N_L = 4R+2P+6 \tag{22}$$

Where R and P are the number of rotary and prismatic joints respectively. In our study the value of N_L for a machine of ZFYXAC topology is 20 [13], [14] since R=2 and P=3 as is the case with most five-axis machine tools. The number of coefficients N_M necessary to model the motion errors with polynomial functions of degree n=3 can then be established at 104-20=84. The value for N_M can be explained as follows. Initially all motion error parameter polynomials have n+1 coefficients. However all zero order terms are position independent, thus are not motion errors, and as such are accounted for in N_L. Also in principle for prismatic joints, the linear terms of the straightness errors (2 per joint) are not motion errors and so these first-order terms are not required for motion error modelling. With regards to rotary joints, all above zero-order terms are required. Therefore, the number of coefficients can be calculated as:

$$N_M = P(6(n+1-1)-2) + R(6(n+1-1)). \tag{23}$$

By substitution of N_L and N_M in eqn (21) we have:

$$N_{min}=4R+6n(R+P)+6 \tag{24}$$

For example, the ZFYXAC machine having 2 rotary joints and 3 prismatic joints, and using Chebyshev polynomials of degree 3, N_{min} =104.

5 Using a ball-bar to identify the parameters and coefficients

The TMBB only detects the projection of the pose error along its measurement axis (Figure 1) so in our model the tool position error must be projected along the TMBB axis using a dot product. As a result, the new component corresponding to the aligned axis directly provides the virtual distance measurement of the TMBB. This mathematical technique allows the generation of an identification Jacobian matrix $\boldsymbol{J}_I$ where each of the m lines (instead of the $6m$ lines of $\boldsymbol{J}_\tau$) represents the effect of a machine parameter coefficient onto the ball to ball distance thus maintaining the availability of a system of linear equations. However $\boldsymbol{J}_I$ is less rich, thus the need to measure more poses for the machine calibration.

In order to generate the TMBB measurement at each configuration, the position of the tool branch ball relative to the workpiece branch ball is calculated for two cases; 1) with the machine tool parameters at their estimated values from which ${}^w\boldsymbol{T}_t^{predicted}$ is calculated, and 2) for a true machine tool from which ${}^w\boldsymbol{T}_t^{true}$ is calculated. The difference between the ball to ball distances obtained from ${}^w\boldsymbol{T}_t^{true}$ and ${}^w\boldsymbol{T}_t^{predicted}$ respectively, represents the virtual measurement value for one configuration. These differences for all test configurations are concatenated

vertically to form the measured data column matrix $\Delta \boldsymbol{b}$. The ball to ball distance is calculated directly from the norm of the translation vector contained within the HTMs ${}^{w}\boldsymbol{T}_{t}^{true}$ and ${}^{w}\boldsymbol{T}_{t}^{predicted}$.

Each ball of the ball-bar only requires a position error and no orientation error. Considering this, the number of unknowns of the minimum-complete set to be identified will decrease to 98 parameters and coefficients (104-3-3=98).

A single socket strategy (a single attachment point at the tool and at the workpiece table) was found to produce a rank of 78, short of the required 98.

However, the measurement data set can be enriched using additional TMBB set-ups through the use of different ball positions at the tool attachment point (the spindle) and at the workpiece table.

5.1 Three-set-up method to acquire data

A three ball-bar set-up strategy was found successful. As seen in Figure 7 a three set-up strategy adds 6 position error parameters per set-up, 3 for the spindle ball and 3 for the workpiece table ball which gives a total of 110 unknowns to be identified.

It is important to point out here that such a procedure relies on the hypothesis that the machine errors are primarily of a systematic nature i.e. that the machine repeatability is relatively good.

6 Procedure to identify the parameters and coefficients of the minimal-complete model

To identify the parameters and coefficients, eqn (25) must be solved iteratively. In this equation, $\Delta \boldsymbol{b}$ are the ball-bar readings as a column matrix, and $\boldsymbol{J}_I$ is a rectangular matrix. A pseudo-inverse matrix has been applied to solve for the unknowns :

$$\cdot \delta \boldsymbol{P} = \boldsymbol{J}_I^{+} \, \Delta \boldsymbol{b} \tag{25}$$

where $\boldsymbol{J}_I^{+}$ is called pseudo-inverse of *Moore-Penrose* of $\boldsymbol{J}_I$ which can be calculated from its SVD:

$$\boldsymbol{J}_I^{+} = \boldsymbol{V} \; {}^{T}\boldsymbol{U}^{T} . \tag{26}$$

The Newton method is used to find the best solution for $\delta \boldsymbol{P}$ through an iterative procedure.

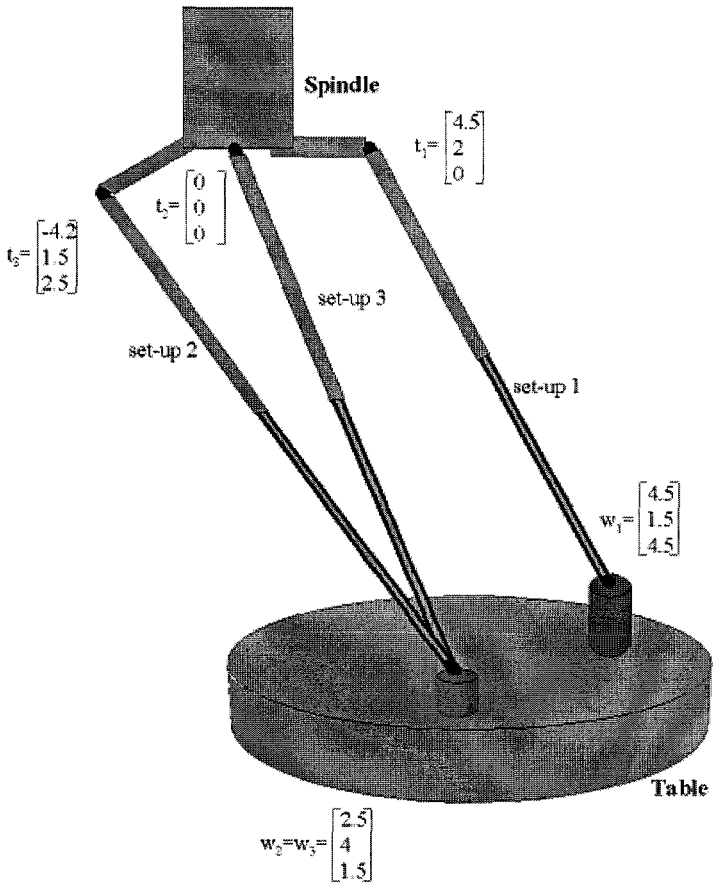

Figure 7 : Three set-up strategy. Each set-up is performed separately (not simultaneously as shown here for convenience).

7 Simulation procedure and results

Two calibration simulations were conducted using specially written Matlab® code. Simulation A involves imposing non-null small values for all 138 potential parameters and coefficients of the maximal-complete model and then attempting to identify through calibration the 110 parameters and coefficients of the minimal-complete model in order to obtain a model that perfectly mimics the true

machine erroneous behaviour. A series of 180 poses have been used. At the 14^{th} iteration the variable change is smaller than 10^{-14}. Because of confounded variables in the maximal complete model, and as expected, the identified variables of the minimal complete model differ from their true values. However the difference between the pose errors calculated for the calibration configurations, using the true variable values and the identified ones was smaller than 10^{-14}. Similar results are obtained when using configurations other than those used for calibration.

Simulation B involves imposing non-null small values only to the 110 parameters and coefficients of the minimal-complete model and then performing the calibration. In this second simulation, we would anticipate the identified model not only to mimic the true machine behaviour but also to have exactly the same parameter and coefficient values as the true machine. The unknowns parameters and coefficients are all identified to better than 10^{-13}.

In all the tests, the identification Jacobian matrix has a condition number below 3500 and a rank of 110.

8 Conclusion

A model has been developed to calibrate a five-axis machine tool for position dependent and position independent geometric errors using the type of data acquired from a telescoping magnetic ball-bar. These error sources were modelled by Chebyshev polynomials and integrated in a sensitivity Jacobian matrix which expresses the sensitivity of the tool-tip versus workpiece feature position to the machine geometric error sources. The Chebyshev polynomials are used because of their better numerical conditioning due to their orthogonal properties .

Because there are initially more polynomial coefficients than is strictly necessary to model the error sources, a strategy is proposed to reduce the number of coefficients to a minimal-complete model set. A formulae for the required number of parameters is also proposed.

Then, a three-socket ball-bar strategy was used to identify these parameters and simulations are conducted using specially written Matlab® programs and functions. The unknown coefficients and ball position parameters were calculated through a system of linear equation made of the identification Jacobian matrix and the ball-bar readings. Then, the identified polynomials are used to predict the tool-tip error for a number of machine configurations. The results confirm that the machine error polynomials can be successfully calibrated.

Acknowledgements

This work was made possible through the financial support of Rene Mayer's NSERC Grant N° 155677-98 and Clément Fortin's NSERC Grant 36396-98. The authors which to thank Guy Cloutier and Richard Gourdeau for the use of their Matlab® Polybox libraries coded in Matlab® developed under NSERC Grant N° OGP0138478 and OGP0121639.

References

[1] Bryan J. B., 1982, A simple method for testing easuring machine and machine tools Part 1: Principles and applications, *Precision Engineering*, 4(2).

[2] Bryan J. B., 1982, *A simple method for testing measuring machine and machine tools Part 2: Construction details*, *Precision Engineering*, **4**(2).

[3] Knapp, W., 1983, "Circular test for Three-Coordinate Measuring Machines and Machine Tools," *Precision Enginerring*, V17, pp.115-124.

[4] Knapp, W., 1983, "Test of the Three-Dimensional Uncertainty of Machine Tools and Measuring Machines and Its Relation to the Machine Errors," *Annals of CIRP*, V32(1), pp.459-464.

[5] Kakino Y., Ihara Y. and Shinohara A., 1993, *Accuracy Inspection of NC Machine Tools by Double Ball Bar Method*, Hanser Publishers.

[6] Kunzman, H., and Waldele, F., 1983, On Testing Coordinate Measuring Machines (CMM) with Kinematic Referenece Standards (KRS), *Annals of CIRP*, V32(1).

[7] Hai, N., Yuan, J., and Ni, J., 1994, "Revese Kinematic Analysis of Machine Tool Error Using Telescoping Ball Bar"., PED-Vol.68-1, *Manufacturing Science and Engineering*, Volume 1, ASME.

[8] Pahk H. J., Kim Y. S. and Moon J. H., 1997, A New Technique for Volumetric Error Assessment of CNC Machine Tools Incorporating Ball Bar Measurement and 3D Volumetric Error Model, *Int. J. Mach. Tools Manufac.*, **37**(11), pp. 1583-1596.

[9] Burdekin, M. and Park, I., A computer Aided System for Assessing the Contouring Accuracy of NC Machine Tools, *Proc. of the 28th MATADOR Conf.*, Manchester, 1988.

[10] Driels M. R., 1993, "Using Passive End-Point Motion Constraints to Calibrate Robot Manipulators, *Trans. of the ASME*, Vol. 115.

[11] Craig, J. J., *Introduction to Robotics, Mechanics and Controls*, Second Edition, Addison-Wesley Publishing Company.

[12] Hayati, S., Tso K. and Roston G., 1988, Robot Geometry Calibration, *Proc. of the IEEE international Conference on Robotics and Automation*, Vol. 2, April 24-29 1988, Philadelphia, Pennsylvania.

[13] Mayer, J.R.R., Mir, Y.A., Fortin, C., 2000, Calibration of a Five-Axis Machine Tool for Position Independent Geometric Error Parameters Using a Telescoping Magnetic Ball-Bar, *Proc. of the MATADOR International Conference*, 2000, UK.

[14] Abbaszadeh-Mir, Y, Mayer , J. R. R., Cloutier, G., and Fortin, C., 2002, Theory and simulation for the identification of the link geometric errors for a five-axis machine tool using a telescoping magnetic ball bar, *Int. J. of Production Research*, **40**(18), pp. 4781-4797.

[15] Everett L. J. and Suryohadiprojo A. H., 1988, A study of kinematic models for forward calibration of manipulators, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 798-800.