Document downloaded from:

http://hdl.handle.net/10251/78896

This paper must be cited as:

Andres De La Esperanza, FJ.; Gracia Calandin, LI.; Tornero Montserrat, J. (2011). Calibration and Control of a Redundant Robotic Workcell for Milling Tasks. International Journal of Computer Integrated Manufacturing. 24(6):561-573. doi:10.1080/0951192X.2011.566284.

The final publication is available at

http://dx.doi.org/10.1080/0951192X.2011.566284

Copyright Taylor & Francis

Additional Information

Calibration and control of a redundant robotic workcell for milling tasks

Javier Andres
Instituto IDF, Universidad Politécnica de Valencia (UPV), Camino de Vera s/n, 46022 Valencia, Spain
fraande@doctor.upv.es

Luis Gracia
Instituto IDF, Universidad Politécnica de Valencia (UPV), Camino de Vera s/n, 46022 Valencia, Spain
luigraca@isa.upv.es

Josep Tornero
Department of Systems Engineering and Control (DISA), Universidad Politécnica de Valencia (UPV), Camino de Vera s/n, 46022 Valencia, Spain
jtornero@isa.upv.es

For Peer Review Only

Calibration and control of a redundant robotic workcell for milling tasks

This paper deals with the tuning of a complex robotic workcell of eight joints devoted to milling tasks. It consists of a KUKA^{TM} manipulator mounted on a linear track and synchronised with a rotary table. Prior to any machining, the additional joints require an *in situ* calibration in an industrial environment. For this purpose, a novel planar calibration method is developed to estimate the external joint configuration parameters by means of a laser displacement sensor and avoiding direct contact with the pattern. Moreover, a redundancy resolution scheme on the joint rate level is integrated within a CAM system for the complete control of the workcell during the path tracking of a milling task. Finally, the whole system is tested in the prototyping of an *orographic model*.

Keywords: redundant workcell; robot calibration; postprocessing; CNC machining; CAD/CAM

For Peer Review Only

1. Introduction

Robotic arms are crucial in many manufacturing processes involving large volumes because of their high flexibility and large working areas (Patrick 2005). These properties are usually enhanced with the use of additional joints carrying the arm or the workpiece and making up what is known as an *industrial workcell*.

At the *Design and Manufacturing Institute* of the *Universidad Politécnica de Valencia* (IDF-UPV), a sculpturing robotic system has been tuned to test milling methods for rapid prototyping on soft materials such as expanded polystyrene (EPS). As shown in Figure 1, an industrial KUKA^{TM} KR15/2 arm with six revolute (6R) joints is mounted on a linear track, and works over a synchronized rotary table platform on which the initial blank of material is set. This provides a wide effective workspace for handling large objects with complex shapes. Because the revolute joints and the linear track joints only allow one *degree of freedom* (DOF) each, the dimension of the *joint space* ($\Im$) is n=8 for this workcell. Other examples of robotic systems for similar applications (such as polishing, arc-welding, spray-glazing, etc.) can be found in Feng-Yun and Tian-Sheng (2005), Huang and Lin (2003), Bidandaa *et al.* (1993), and Tsai *et al.* (1992).

Conventional CNC machining techniques can be adapted from high precision metal cutting to fast milling in EPS with robots. Sophisticated CAD/CAM/ROB integrated manufacturing systems mean that during the machining process less time is invested in successive verifications, adjustments, and data translation.

Leading commercial CAM (*computer aided manufacturing*) software is prepared for the off-line planning of 5-axis CNC operations (López De Lacalle *et al.* 2005). For precision and universality, the five parameters captured in a cutting toolpath are the three pose coordinates of the *tool centre point* (TCP) and the two orientation coordinates of the milling tool (considering the tool symmetrical along its axis of revolution, see Figure 1). Accordingly, at milling works the dimension of the *task space* (T) is t=*5*. This assumes no-uncertainty in tool orientation and positioning for a conventional CNC machine but, in every case, the toolpath must be adapted (*postprocessed*) to the milling system (Lee and She 1997, Nemec and Zlajpah 2008).

Accuracy is crucial in a robot program generated by a CAM system, but accuracy is lessened by a poor adjustment of the workcell, particularly in the *kinematic* factors related to the additional joints. Offsets and deviations in the assembly of the additional external joints of the workcell reduce accuracy (Khalil and Dombre 2002).

This paper exposes the fine-tuning of the redundant workcell shown in Figure 1 for milling tasks. After describing the kinematics of the workcell in Section 2, the calibration of the external joints is tackled in Section 3. Section 4 explains the redundancy resolution scheme (RRS) used to implement the algorithm for continuous path tracking in order to integrate a CAM-robotics postprocessor (Section 5). This postprocessor generates the robot instructions off-line for milling tasks with the correct use of the redundancies. The designed postprocessor is tested in Section 6 with a practical application, namely, the prototyping of an *orographic model*. Finally, the conclusions to this work are discussed in Section 7.

2. Kinematic modelling of the redundant workcell

A *kinematic model* is the mathematical description required to control the *posture* of the robot chain and the associated *pose* of the milling tool. When performing a task on the workcell, the table can be regarded as fixed while the other movable *end-effector* (EE) holds the cutter tool in the Cartesian *operational workspace* (Ω) whose base frame is {B}, see Figure 1. Note that the pose of a rigid body in Ω is defined by six coordinates (three locations and three orientation values) and so dim(Ω)=*m*=6.

The *direct kinematic problem* (DKP) at the displacement level is straightforward, mapping from $\Im$ to Ω. It consists of determining the pose of the EE for a given manipulator posture. The *standard* Denavit-Hartenberg (DH) model is widely utilized in this context due to its simplicity. It represents the pose of the EE as a 4×4 homogeneous transformation matrix *T* deduced from four parameters (α_i, a_i, θ_i, d_i) describing the relative position between consecutive links (Hartenberg and Denavit 1955).

Moreover, the robotic workcell described in Section 1 is *redundant* as $n > t$ $(with\ \mathrm{T} \subseteq \Omega)$ (Patel and Shadpey 2005), with a degree of *kinematic redundancy* (r_K) of $r_K = n - t = 3$. This redundancy appears because of the additional linear track (d_L) and rotary table (θ_M), as well as the cutter tool symmetry while spinning along its revolution axis, see Figure 1. In practical use, the redundancy along the milling tool axis means that an additional *virtual joint* can be considered for the kinematic model, increasing the dimension of $\Im$ (*n*=9). With these premises, the full DH model is summarised in Figure 2 and Table 1.

3. External joint calibration

Robot kinematic calibration consists in identifying the differences between the default geometrical parameters given by the manufacturer and those of the real workcell to achieve the required *accuracy* see ISO 9283:1998. It is notable that the off-line generation of programs for robotic workcells, as opposed to the traditional on-line teaching methods based on *repeatability,* has resulted in the emergence of different calibration techniques.

Existing calibration techniques can be broadly classified into *closed-loop* (or *pose-matching*) and *open-loop* (or *pose-measuring)* approaches (Fassi *et al.* 2006). The *closed-loop* methods impose some constraints on the EE and only use the joint position measurements to calibrate the robot. However, it is difficult to move a physically closed

For Peer Review Only

kinematic chain from one position to another while maintaining the physical constraints, such as touching a point or a plane with a probe. Hence, it is difficult to gather accurate joint readings (Fassi *et al.* 2006, Khalil *et al.* 1995). The *open-loop* methods traditionally require expensive and demanding devices to measure the EE pose, such as theodolites (Whitney *et al.* 1986), inclinometers, coordinate measuring machines (Driels *et al*. 1993, Zhong *et al*. 1994), sonic and visual sensors (Stone *et al*. 1986, Driels and Swayze 1994), and laser tracking systems (Prenninger *et al*. 1993, Vincze *et al*. 1994). An alternative open-loop method was developed by Sultan and Wager (2001) on the basis of independent-axis calibration and using two theodolites as data-acquisition media. However, the closed-architecture control system of commercial KR15/2 robots has encouraged the authors to develop a different procedure.

The KR15/2 manipulator has proven to be accurate, but the workcell assembly made *in situ* left some misalignments in the additional joints, namely the linear track and the rotary table. Consequently, a simple, fast, and accurate *calibration* is needed to update the robot controller with the correct robot-specific DH parameter values instead of the default values.

The proposed calibration method combines *pose-matching* and *pose-measuring* methods using a set of positions of the terminal point of the robot but avoiding the inconveniences of physical contact with the EE. These points are assumed to be on the same plane. A non-linear least squares (*NLSQ*) identification model has been derived from the consistency conditions of the planes and is detailed below. The DH parameters associated with the additional external joints will be held in vector β, i.e. $\beta = [\alpha_1\ a_1\ \theta_M\ d_1\ \theta_2\ d_L]^T$, in order to be accurately estimated using the calibration procedure. These parameters relate the base frame {B} with the additional frames linked to the additional external joints P{d_L} and R{θ_M}, see Figure 2. Nevertheless, the calibration method proposed below can be applied to the complete chain by magnifying the observed parameters. This method is suitable for in-situ calibration in an *industrial environment* and can be run autonomously.

3.1. Sensor used in the calibration

A laser displacement sensor (SICK^{TM} OD100-35P840) is rigidly attached to the robot flange. Due to its reduced size, the laser can be integrated within the same milling-tool holder and, thus, the calibration can be performed for the nominal operating conditions of the robot. The laser has a measuring range of 100 mm with 35 µm in resolution and is assumed to be aligned with the Z-axis of the EE. A coordinate system {LR} is chosen with the origin (TCP) in the laser line and with the X,Y-axis set at a convenient orientation, see Figure 3.

It is cumbersome to use *planar methods* if the calibration process includes the calibration of the laser itself, since the observability of certain parameters is not possible if the EE is moved in parallel over a plane. Thus, the availability of a fully calibrated laser sensor is assumed prior to the assembly of the new additional joints, i.e. ${}^{8}T_{LR}$ is assumed to be known. For example, ${}^{8}T_{LR}$ can be computed with the approach proposed by Zhu *et al.* (2004).

In practical terms of *industrial calibration*, the workspace base frame {B} is set at the centre of the rotary table using a squared pattern and several external measurements made in situ (theoretically, only three points per plane are required), see Figure 4.

For Peer Review Only

3.2. Non-contact planar constraint calibration procedure

It has been reported that partial EE pose information is sufficient for complete parameter identification (Zhong and Lewis 1995). However, Zhuang *et al.* (1999) showed that a single-plane constraint does not necessarily guarantee the observability of the unknown kinematic parameters of the robot, as the measurement data is biased towards a particular direction. These authors demonstrated that no less than three planes placed at a number of different orientations could replace a 3D-position measurement system.

We use the laser distance sensor to sweep the three constraint planes (Π_X, Π_Y and Π_Z) at an intended constant distance *d*, see Figure 5. To achieve this, a set of points $p_M{}^i$ $(i=1,...,np)$ is commanded to the controller forming three orthogonal meshes in their respective orthogonal planes. This is performed by setting the required position and orientation of the laser displacement sensor at $\{B\}$ for each point, with the pattern specified by the robot manufacturer (KUKA Corp., 2005):

$$p_M{}^i = (p_{M\,x}{}^i, p_{M\,y}{}^i, p_{M\,z}{}^i, p_{M\,A}{}^i, p_{M\,B}{}^i, p_{M\,C}{}^i) \qquad (1)$$

where the three first values represent the position, and the three last values represent the roll-pitch-yaw orientation coordinates, respectively.

The calibration procedure capitalises on the restriction imposed by the laser line, being almost perpendicular to each plane, and which is intended to be at the same distance *d* for all the p_M points commanded to the controller. Moreover, we can consider the laser TCP at a distance *d* from the laser sensor so that each commanded point p_M lies on its associated plane. The general equation of a plane Π containing the origin of $\{B\}$ is $ax+by+cz=0$, where *a, b, c* correspond to the plane coefficients. Since the TCP is lying in its corresponding plane, each commanded point $p_M{}^i$ should accomplish that $ap_{M\,x}{}^i + bp_{M\,y}{}^i + cp_{M\,z}{}^i = 0$. Moreover, the equations observed for the three pattern planes are, respectively, $p_{M\,x}{}^i = 0$, $p_{M\,y}{}^i = 0$ and $p_{M\,z}{}^i = 0$. Therefore, the commanded points p_M have the form:

$$p_{M\Pi_X}{}^i = (0, p_{M\,y}{}^i, p_{M\,z}{}^i, 0, -\pi/2, 0) \qquad (2)$$

$$p_{M\Pi_Y}{}^i = (p_{M\,x}{}^i, 0, p_{M\,z}{}^i, 0, 0, -\pi/2) \qquad (3)$$

$$p_{M\Pi_Z}{}^i = (p_{M\,x}{}^i, p_{M\,y}{}^i, 0, 0, 0, 0) \qquad (4)$$

A *non-linear least squares* (*NLSQ*) identification model has been derived from just the laser readings, without additional external measurements or joint recordings. The objective is to adjust the parameters of a *kinematic model function* that best fits all the *np* commanded points, see Figure 6. For each point, this *model function* has the form:

$$p_M{}^i = f(q^i, \beta); \quad i = 1,...,np \qquad (5)$$

The *mp* adjustable model parameters are held in the vector β. In the case studied, $\beta = [\alpha_1\ a_1\ \theta_M\ d_1\ \theta_2\ d_L]^T$ (*mp*=6).

For Peer Review Only

The fact that each set is constrained to lie on its corresponding plane leads to the construction of the identification model. Let ${p_S}^i$ represent the actual coordinates of the laser TCP in {B} that correspond to the configuration q^i acquired for the commanded point ${p_M}^i$, see Figure 5. Due to the perpendicular orientation commanded, the distance D_Π^i measured by the laser device approximates to the respective actual coordinate ${p_S}^i{}_x$, ${p_S}^i{}_y$ or ${p_S}^i{}_z$ at {B} (Figure 5). For each plane, the other five coordinates of the reached point ${p_S}^i$ are neither known nor approximated by any external measurement. Nevertheless, the error committed in all coordinates is reduced as the *mp* model parameters are tuned.

A residual, r^i, can be defined as the difference between the actual observed values ${p_S}^i$ and the model predicted values, namely $r^i = {p_S}^i - f(q^i, \beta)$. For each plane Π and each commanded ${p_M}^i$, we approximate each residual to be minimised with the reading , i.e. $r^i \approx D_\Pi^i$. Thus, the identification model relating the deviation of the EE location from the plane Π with the differential error in the geometric parameters ($\Delta\beta$) can be expressed as:

$$D_\Pi = J_{f\Pi}\,\Delta\beta \qquad (6)$$

with $J_{f\Pi} = \partial r_\Pi / \partial\beta$ being the *Jacobian* matrix of the residual regarding the identifiable parameters. Only the expression corresponding to the residual error r_Π in the perpendicular direction of each plane Π is taken into account.

In this non-linear system, the derivatives $\partial r_\Pi / \partial\beta$ are functions of both the commanded postures and the identifiable parameters, so these gradient equations do not have a closed solution. Instead, the default assembly values are chosen as initial guesses for the *mp* parameters, $\beta^{(0)} = [\pi/2, 803, \theta_M, -305, 0, d_L]^T$, and the readings are repeated for two different configurations of the additional joints, i.e. $\{\theta_M, d_L\} = \{0, 700\}$ and $\{\theta_M, d_L\} = \{\pi/4, 300\}$, see Figure 7. Thus, while minimising the residuals, the final value of β is refined iteratively by consecutive approximations:

$$\beta^{(s+1)} = \beta^{(s)} + \Delta\beta^{(s)} \qquad (7)$$

where s is the iteration number. As Equation (6) is applied for a sufficient number *np>>mp* of commanded points arranged on the three orthogonal planes, the resulting system to identify $\Delta\beta^{(s)}$ is:

$$\Delta p^{(s)} = W\left(\beta^{(s)}\right)\Delta\beta^{(s)} \qquad (8)$$

where W is the *observation matrix* (Khalil *et al.* 1995) of dimension *np*×*mp*. The *observation matrix* is an ordered composition of the Jacobians associated with the corresponding observation at each plane, $W = [J_{f\Pi_X}(\beta^{(s)})\ J_{f\Pi_Y}(\beta^{(s)})\ J_{f\Pi_Z}(\beta^{(s)})]^T$. Consequently, care must be taken in the configuration of Δp due to the fact that the measurements taken in each plane must correspond with the significance of each row of W. The *npx1* vector of the observed residuals in the three planes is

For Peer Review Only

$\Delta p^{(s)} = [D^{(s)}_{\Pi_X}\ D^{(s)}_{\Pi_Y}\ D^{(s)}_{\Pi_Z}]^T$. The increment $\Delta\beta$ can then be solved to obtain the least squares error solution:

$$\Delta\beta^{(s)} = W^{\dagger}\Delta p^{(s)} \tag{9}$$

where $W^{\dagger}$ is the *left pseudo-inverse* of W, namely $W^{\dagger} \equiv (W^T W)^{-1} W^T$.

The geometric parameters β are iteratively updated from the current pose error $\Delta p^{(s)}$ using Equations (9) and (7) until the elements of $\Delta\beta^{(s)}$ become smaller than a prescribed limit. This best-fit solution is saved in the controller. The common sense criterion for convergence is somewhat arbitrary, as for example

$$\left|\frac{\Delta\beta_j^{(s)}}{\beta_j^{(s)} - \beta_j^{(0)}}\right| < 0.001 \quad \forall\ j = 1,\ldots,mp \tag{10}$$

which is equivalent to specifying that each parameter should be refined to 0.1% precision, i.e. $\left|\beta_j^{(s)} - \beta_j^{(\infty)}\right| < 0.001 \cdot \left|\beta_j^{(s)} - \beta_j^{(0)}\right| \approx 0.001 \cdot \left|\beta_j^{(\infty)} - \beta_j^{(0)}\right|$, where $\beta_j^{(\infty)}$ is the asymptotic value of the parameter.

3.3. *Calibration results*

The calibration procedure described in this work has been run on an Intel® Core Duo PC with Matlab® 2007c. A total of 108 points have been measured, i.e., a regular mesh of 36 grid points (6x6) was commanded for each plane in Figure 7. Figure 8 shows the behaviour of the proposed calibration algorithm for the studied workcell. The calibration process shows a good convergence. The stop criterion given by Equation (10) is met at the 18th iteration and the value achieved for the parameters is $\beta^{(18)} = \beta^{(0)} + [0.05\ 0.01\ 0.06\ 0.01\ 0.07\ 0.08]^T$ (mm, rad).

To validate the improvement obtained with the calibration procedure, two comparisons are considered. For the first, a total of *nv*=96 points on the table plane were measured using the laser sensor and varying external joints randomly from 0 to 1000 mm for d_L and from 0 to π rad for θ_M. Figure 9 shows the performance for this test with and without calibration. After calibration the accuracy of the data clearly iºmproves: the average value is reduced from -3.342 to -0.600 mm; the root mean square (RMS) error is reduced from 5.429 mm to 0.998 mm; and the standard error deviation is reduced from 4.279 to 0.797 mm.

The second comparison consists of the milling of a small workpiece, see Figure 10. Although the motion of the additional joints is not required for this milling task, the operator manually displaces (by means of the control panel of the robot) the linear track 500 mm and rotates the table $\pi/4$ rad at a certain time while the lateral walls of the workpiece are being milled. Note that, after calibration (Figure 10, bottom right) the workpiece is better finished.

4. Inverse kinematic control of the redundant workcell

The *inverse kinematic problem* (IKP) is relevant because CAM systems specify the toolpath at the Cartesian workspace Ω, whereas the robot controller works in the joint space $\Im$. The IKP can be solved for each pose of the path, although at the *displacement level* the IKP is cumbersome because an infinite number of solutions exist for redundant

For Peer Review Only

manipulators. Instead, an iterative approach at the *joint rate level* is usually used (Whitney 1969).

The *forward kinematics* between the EE velocity *t* and the joint velocity $\dot{q}$ is represented by the *linear* equation $t = J \cdot \dot{q}$, where J is the *mxn Jacobian* matrix, which is a non-linear function of the joint angles that can be computed with the geometric procedure outlined by Angeles (2003), among others. For the redundant workcell considered in this work, the Jacobian matrix maps the 9-dimensional vector $\dot{q} = [\dot{\theta}_M, \dot{d}_L, \dot{\theta}_1, ..., \dot{\theta}_7]^T$ of joint rates into the EE velocity vector $t = \begin{bmatrix} \omega & v \end{bmatrix}^T$, with $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ and $v = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ denoting the linear and angular velocities of the EE reference frame relative to the base frame {B}, respectively. The objective of the IKP at the joint rate level is to obtain the required joint velocities $\dot{q}$ to achieve the desired EE velocity *t* for a given Jacobian matrix *J*. In the case of *redundant* robots, for which the Jacobian matrix J is non-square (*n*>*m*), the *right pseudo-inverse* $J^{\dagger} \equiv J^T \left(JJ^T\right)^{-1}$ gives the minimum least-squares solution $\dot{q}$ that fulfils the desired EE velocity *t*, namely $\dot{q} = J^{\dagger} \cdot t$. Moreover, an arbitrary vector from the *Null Space* of J, namely $\aleph(J)$, can be added to achieve *a secondary goal*:

$$\dot{q} = J^{\dagger} \cdot t + (I - J^{\dagger} J) h \qquad (11)$$

where I is the identity matrix of dimension *n*, $(I - J^{\dagger} J)$ is the projection operator on $\aleph(J)$, and *h* is the *performance vector* for the secondary task. At this point it is worth mentioning that, as the Jacobian matrix depends on the DH parameters, the workcell calibration is crucial for a proper robot control.

In the redundancy resolution scheme (RRS) given by Equation (11), the *performance vector h* can be considered as a *virtual force* that attempts to push the configuration of the manipulator away from a critical area in $\Im$ (Nemec and Zlajpah 2008) as a secondary task. The most widespread method used to select *h* is the *gradient projection method* (GPM) (Liégeois, 1977), which minimises a position-dependent scalar, the *performance index p*, by means of its gradient vector:

$$h = -k \cdot \nabla p \;\; ; \text{ with } \nabla p = \left[\frac{\partial p(q)}{\partial q_1}, \frac{\partial p(q)}{\partial q_2}, ..., \frac{\partial p(q)}{\partial q_n} \right]^T \qquad (12)$$

In this work, two performance criteria are combined into vector *h*:

$$h = h_{jnt} + h_{cond} = -\nabla\left(p_{jnt} + p_{cond}\right) = -\left(H_{jnt}\left(q - q^{\text{ref}}\right) + H_{cond}\, k \,\left(q - q_{Ts}\right)\right) \qquad (13)$$

where *H* is the weighting matrix for each criterion, e.g. constant diagonal matrices are considered in this work $H_{jnt} = H_{cond} \equiv diag(0.01)$. Note that the performance vector h_{jnt} given by performance index p_{jnt} tends to maintain the workcell as close as possible to a reference posture q^{ref} (e.g. the HOME posture q_0 shown in Figure 2) to avoid joint limits (Huo and Baron 2008). The performance vector h_{cond} given by performance index p_{cond} tends to maintain the workcell as far as possible from ill-conditioned postures (Angeles and López-Cajún 1992), i.e. postures with poor kinematic performances. The

For Peer Review Only

second performance vector h_{cond} is activated when the value of the *condition number* k of the Jacobian matrix (e.g. obtained using the Euclidean norm) passes over a preset threshold value, e.g. 0.5. At that instant, the corresponding configuration q_{Ts} is recorded to evaluate at $\Im$ the distance to the actual posture.

5. Postprocessor implementation for a redundant workcell

5.1. CAM system for toolpath generation

The software NX® is a digital development system from Siemens™ that integrates the tasks of design (CAD), simulation (CAE), and planning of milling tasks (CAM). NX interacts with the program codes in TCL and C++ that manipulate the path data (*event handler*) and gives a convenient format to the generated output (*definition file*), see Siemens Corp. (2009) and Figure 11. The treatment of the data has been programmed in Matlab® code with the aid of the Hemero toolbox (Maza and Ollero 2001), whereas the Robomove® software (Qdesign S.r.l. 2007) has been used to visualise the robot postures resulting from the implemented RSS.

5.2. Continuous path tracking

Once the CAM system has generated the toolpath as a discrete set of close-enough poses at Ω, the EE of the robot must track this path. A tangent, normal, and binormal unit *Frenet-Serret* vectors, i.e. *{t, n, b}*, is associated with every sample point of the trajectory and this indicates the required pose, see Figure 12. This trajectory data is stored as T_{CAM}.

The joint angles of the robot have to be computed along this continuous set of poses of the EE. The IKP can be solved at each sampled pose, although the resolution is cumbersome because an infinite number of solutions exist for redundant manipulators. Instead, an iterative approach can be applied by using *J* (Angeles 2003):

$$J(q^i)\Delta q^i = \Delta t^i \text{ , with } \Delta t^i \equiv \begin{bmatrix} Q_k\, vect(Q_k^T Q_d) \\ \Delta p \end{bmatrix} \quad (14)$$

where Q_k (i.e., the first three rows and columns of the matrix *T* introduced in Section 2) is the current rotation matrix from {B} to {EE} (see Figure 1), and Q_d is the desired rotation matrix. Both matrices are related by:

$$Q_d = Q_k \cdot \Delta Q \rightarrow \Delta Q = Q_k^T \cdot Q_d \quad (15)$$

Function $vect(\Delta Q)$ represents the axial vector of the 3x3 rotation matrix ΔQ and is computed as:

$$vect(\Delta Q) \equiv \frac{1}{2}\begin{bmatrix} \Delta Q_{32} - \Delta Q_{23} \\ \Delta Q_{13} - \Delta Q_{31} \\ \Delta Q_{21} - \Delta Q_{12} \end{bmatrix} \quad (16)$$

Vector Δp is defined as the difference between the prescribed value p_d of the TCP position vector and its current value p_k. The relations amongst Q_d, p_d, Q_k, p_k, ΔQ and Δp are shown in Figure 12.

For Peer Review Only

Algorithm 1 below is programmed with these premises. It starts from the HOME posture (q_0, Figure 12) and iteratively uses the kinematic models of the single KR15/2 manipulator (sub-index *6R*) and the complete workcell (sub-index *Workcell*) to evaluate the condition number k and manage the motion of all the joints.

Algorithm 1

```
1) $q \leftarrow q_0$
for (each $i$-point of the trajectory, $\mathrm{T_{CAM}}(i)$)
    2) $\{p_d, Q_d\} \leftarrow \mathrm{T_{CAM}}$
    while $\|\Delta q\| > \varepsilon$
        3) $\{p, Q\} \Leftarrow \mathrm{DK}(q, \text{DH-Workcell})$
        4) $\Delta Q \Leftarrow Q^T \cdot Q_d$
        5) $\Delta p \Leftarrow p_d - p$
        6) $\Delta t \leftarrow \begin{bmatrix} Q \cdot vect(\Delta Q) \\ \Delta p \end{bmatrix}$
        7) $J_{\mathrm{Workcell}} \Leftarrow \mathrm{DK}(q, \text{DH-Workcell})$
        8) Determination of $k_F$
            8.1) $q_{6R} \leftarrow \{0, q(4), \ldots, q(8)\}$
            8.2) $J_{6R}(q) \Leftarrow \mathrm{DK}(q_{6R}, \text{DH-KR15/2})$
            8.3) $H_{6R} \underset{L}{\leftarrow} J_{g\ 6R}$
            8.4) $k_F \leftarrow H_{6R}$
        9) $\Delta q \Leftarrow \mathrm{RRS}$
        10) $q \Leftarrow q + \Delta q$
    endwhile
endfor
```

The RRS in the 9th step of Algorithm 1 is programmed using the Householder reflections. Direct calculation of the pseudo-inverse of the Jacobian $J^\dagger$ is avoided to keep the round-off errors as low as possible (Arenson 1998). Algorithm 2 below summarises this calculation using *Matlab*® code.

For Peer Review Only

Algorithm 2

```
1) $J^T \leftarrow J'$    (transpose of $J$ in Matlab)
2) $[Q\ R] = qr\left(J^T\right)$    (Matlab's orthogonal-triangular $QR$ - $decomposition$)
    2.1) $H \leftarrow Q'$
    2.2) $U \leftarrow R(1:6,:)$
3) $r \leftarrow \Delta t - J \cdot h$
4) $y_1 \leftarrow forward(U', r)$    ($U^T \cdot y_1 = r,$ solved by forward substitution)
5) $y \leftarrow [y_1;\ zeros(3,1)]$
6) $k \leftarrow H' \cdot y$
7) $\Delta q = k + h$
```

6. Application

The production system described above was implemented to mill an 8x13 meter orographic model of the tail end of a reservoir on the river *Mijares* (Spain). The design is imported from AUTOCAD® (widely used in topography) to the NX®-CAD module to restore the contour lines and interpolate a surface mesh. The quality of the resulting CAD file determines the efficiency of the results obtained in the subsequent steps of the milling process. As shown in Figure 13, the model is made by assembling 120 blocks of 1x1x0.5 meters of expanded polystyrene (EPS).

The toolpath generated in NX® has been simulated *with* and *without* the implemented postprocessor in the graphical Robomove® simulator and starting from the same HOME posture. In Figure 14 it can be seen that the end of the workpiece (red part) cannot be reached without moving the additional joints, i.e. d_L and θ_M. A better performance is achieved when the toolpath is postprocessed with the proposed RRS, which moves the additional joints to maintain a well-conditioned posture during the machining process.

Figure 15 (right) shows the final orographic model at the Hydraulic Engineering Department (DIHMA) of the Universidad Politécnica de Valencia, and which is used for simulating water courses under various climatic conditions.

For Peer Review Only

7. Conclusions

After introducing the capabilities of an industrial robotic workcell, two necessary requirements for setting up the workcell have been discussed in this paper: the *in situ* calibration of the robot's external joints; and the management of the redundancies (caused by the additional joints and the symmetry of the cutter tool).

For the calibration task, a novel *non-contact planar constraint* procedure has been developed on the basis of a planar pattern and a laser displacement sensor and has been successfully validated with two illustrative examples. The proposed calibration method is relatively cheap, can be implemented autonomously in most industrial robots, and is fast enough to be used in situ at regular intervals. Further work must be done to integrate the laser measuring device within the tool holder, in order to bridge the gap between the theoretical and practical applications on the shop-floor.

Moreover, a functional postprocessor has been programmed inside the CAM system for the control of redundancies during milling tasks. A real case-study has been considered to validate the effectiveness of these production systems for the milling of large prototypes using soft materials. The proposed postprocessor is expected to be easily applicable on any industrial robot, and useable for different applications such as welding or painting. The implementation details to adapt CAM modules from machining tasks to welding and painting tasks is a field for further research.

For Peer Review Only

REFERENCES

Angeles, J., 2003. *Fundamentals of robotic mechanical systems: theory, methods and algorithms*, New York: Springer.

Angeles, J. and López-Cajún, C. S., 1992. Kinematic Isotropy and the Conditioning Index of Serial Robotic Manipulators. *The International Journal of Robotics*, 11 (6), 560–570.

Arenson, N., 1998. *Real-time redundancy-resolution schemes for robotic manipulators.* Thesis (PhD). McGill University.

Bidandaa, B., Narayanana, V. and Rubinovitzb, J., 1993. Computer-aided-design-based interactive off-line programming of spray-glazing robots. *International Journal of Computer Integrated Manufacturing*, 6 (6), 357–365.

Driels, M.R., Swayze, L.W. and Potter, L.S., 1993. Full-pose calibration of a robot manipulator using a coordinate measuring machine. *International Journal of Advanced Manufacturing Technology*, 8 (l), 34–41.

Driels, M.R. and Swayze, W., 1994. Automated partial pose measurement system for manipulator calibration experiments. *IEEE Transactions on Robotics and Automation,* 10 (4), 430–440.

Fassi, I., Legnani, G., Tosi, D. and Omodei, A., 2006. Calibration of Serial Manipulators: Theory and Applications. *Industrial Robotics: Programming, Simulation and Applications*. Germany: Pro Literatur Verlag, 147-170.

Feng-Yun, L. and Tian-Sheng, L., 2005. Development of a robot system for complex surfaces polishing based on CL data. *International Journal of Advanced Manufacturing Technology*, 26 (9), 1132–1137

Hartenberg, R.S. and Denavit, J., 1955. A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22 (2), 215–221.

Huang, H.-k. and Lin, G.C.I., 2003. Rapid and flexible prototyping through a dual-robot workcell. *Robotics and Computer Integrated Manufacturing*, 19 (3), 263–272

Huo, L. and Baron, L., 2008. The joint-limits and singularity avoidance in robotic welding. *Industrial Robot: An International Journal*, 35 (5), 456–464.

International Standard ISO 9283:1998. *Manipulating industrial Robots: Performance criteria and related test methods*.

Khalil, W., Garcia, G. and Delagarde, J.F., 1995. Calibration of the geometric parameters of robots without external sensors, *IEEE International Conference on Robotics and Automotion*, 3039–3044.

Khalil, W. and Dombre, E., 2002. *Modeling identification and control of robots*, New York: Taylor & Francis Books Inc.

KUKA Corp., 2005. *KUKA System software (KSS): expert programming (KRC2 / KRC3)*, Release 5.2.

Lee, R.-S. and She, C.-H., 1997. Developing a postprocessor for three types of five-axis machine tools. *International Journal of Advanced Manufacturing Technology*, 13 (9), 658–665.

Liégeois, A., 1977. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, 7 (12), 868–871.

López De Lacalle, L.N., Lamikiz, A., Muñoa, J. and Sánchez, J.A., 2005. The CAM as the centre of gravity of the five-axis high speed milling of complex parts. *International Journal of Production Research*, 43 (10), 1983–1999.

Maza, J.I. and Ollero, A., 2001. Herramienta MATLAB-Simulink para la simulación y el control de robots manipuladores y móviles, *Actas de las XXI Jornadas de Automática*, Sevilla.

Meggiolaro, M.A., Scriffignano, G. and Dubowsky, S., 2000. Manipulator calibration using a single endpoint contact constraint. *Proceedings of the 26th Biennial Mechanisms and Robotics Conference of the 2000 ASME Design Engineering Technical Conferences*, Baltimore-Maryland.

Nemec, B. and Zlajpah, L., 2008. Robotic cell for custom finishing operations, *International Journal of Computer Integrated Manufacturing*, 21 (1), 33–42.

Patel, R.V. and Shadpey, F., 2005. *Control of redundant robot manipulators: theory and experiments*, Germany: Springer.

Patrick, W., 2005. Shop-floor productivity. *Manufacturing Engineering*, 135 (1). Available from: http://www.sme.org/cgi-bin/find-articles.pl?&ME05ART38&ME&20050710&&SME& [Accessed 12 August 2010].

Prenninger, J.R., Vincze, M. and Gander, H., 1993. Contactless position and orientation measurement of robot end-effector. *IEEE International Conference on Robotics and Automation,* Atlanta, 1, 180–185.

Qdesign S.r.l., 2007. *CAD-CAM off line programming for industrial robots*. ROBOmove on line help v. 2.0.

Siemens Corp., 2009. *NX Documentation*, *In*:{$UGII_base_dir}\UGDOC.

Stone, H.W., Sanderson, A.C. and Neuman, C.P., 1986. Arm signature identification. *International Conference on Robotics and Automation,* San Francisco, CA, 41–48.

For Peer Review Only

Sultan, I. A. and Wager, J. G., 2001. A technique for the independent-axis calibration of robot manipulators with experimental verification. *International Journal of Computer Integrated Manufacturing*, 14 (5), 501–512.

Tsai, M.J., Lin, S.-D. and Chen, M.-C., 1992. Mathematical model for robotic arc-welding off-line programming system. *International Journal of Computer Integrated Manufacturing*, 5 (4), 300–309

Vincze, M., Prenninger, J.R. and Gander, H., 1994. A laser tracking system to measure position and orientation of robot end effectors under motion. *International Journal of Robotics Research*, 13 (4), 305–314.

Whitney, D.E., 1969. Resolved motion rate control of manipulators and human prostheses, *IEEE Transactions on Man-Machine Systems*, 10 (2), 47–53.

Whitney, D.E., Lozinsky, C.A. and Rourke, J.M., 1986. Industrial robot forward calibration method and results. *ASME Journal of Dynamic Systems, Measurement and Control*, 108 (1), 1–8.

Zhong, X.L., Lewis, J.M. and Rea, H., 1994. Neuroaccuracy compensator for industrial robots. *IEEE International Conference on Neural Networks*, Orlando, Florida, 5, 2797–2802.

Zhong, X.L. and Lewis, J.M., 1995. A new method for autonomous robot calibration. *IEEE International Conference on Robotics and Automation*, 1790–1795.

Zhu, Z., Tang, Q., Li, J. and Gan, Z., 2004. Calibration of laser displacement sensor used by industrial robots. *Optical Engineering*, 43 (1), 12–13.

Zhuang, H., Motaghedi, S.H. and Roth, Z S., 1999. Robot calibration with planar constraints. *IEEE International Conference on Robotics and Automation*, Detroit, MI, 805–810.

For Peer Review Only

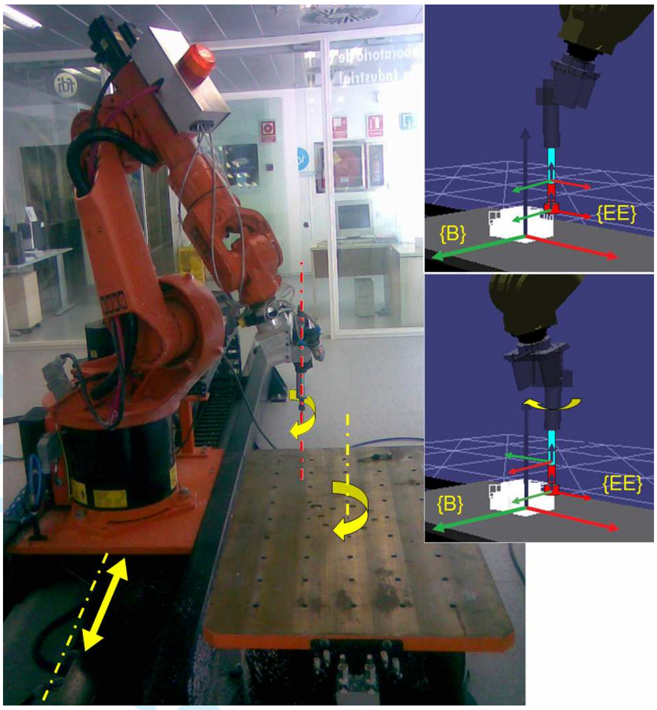

Figure 1. View of the KUKA^{TM} workcell at the IDF-UPV supplied with two additional joints. The insets show the irrelevant axis of symmetry of the milling tool.

For Peer Review Only

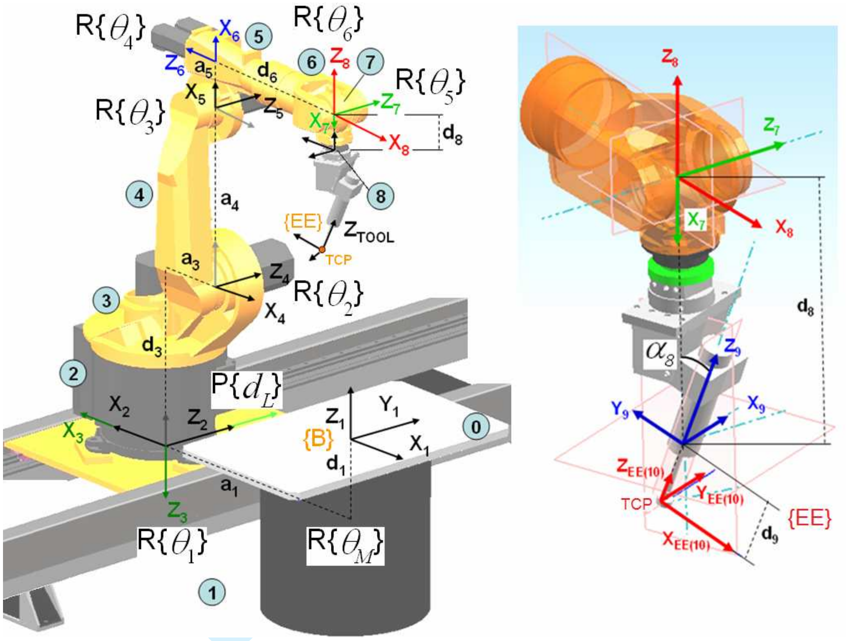

Figure 2. Frame assignment for the DH representation of the RP-6R workcell.

eer Review Only

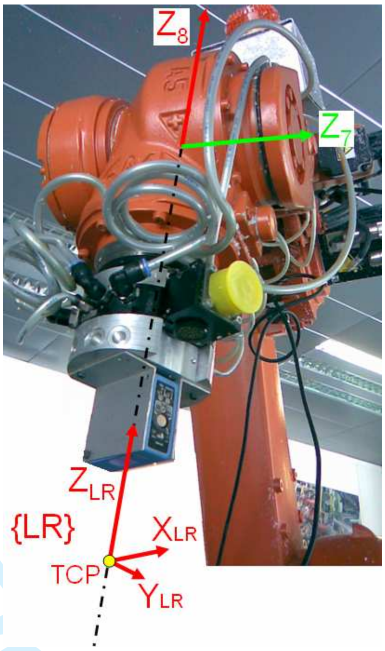

Figure 3. Laser displacement sensor attached to the robot flange.

For Peer Review Only

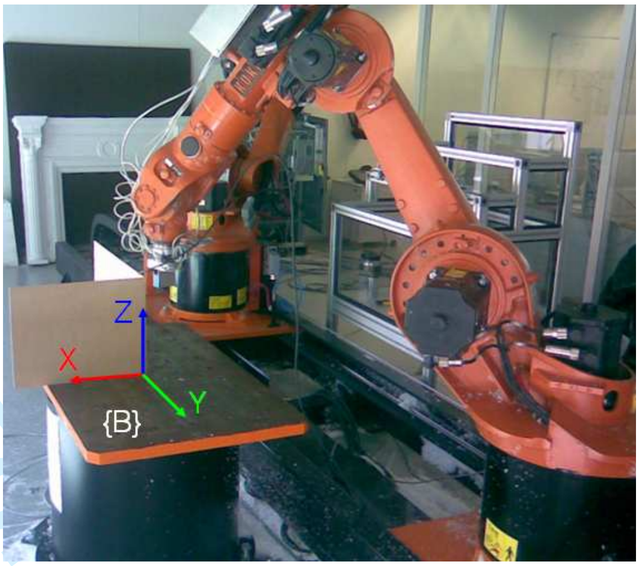

Figure 4. Squared pattern on the workspace base frame {B}.

For Peer Review Only

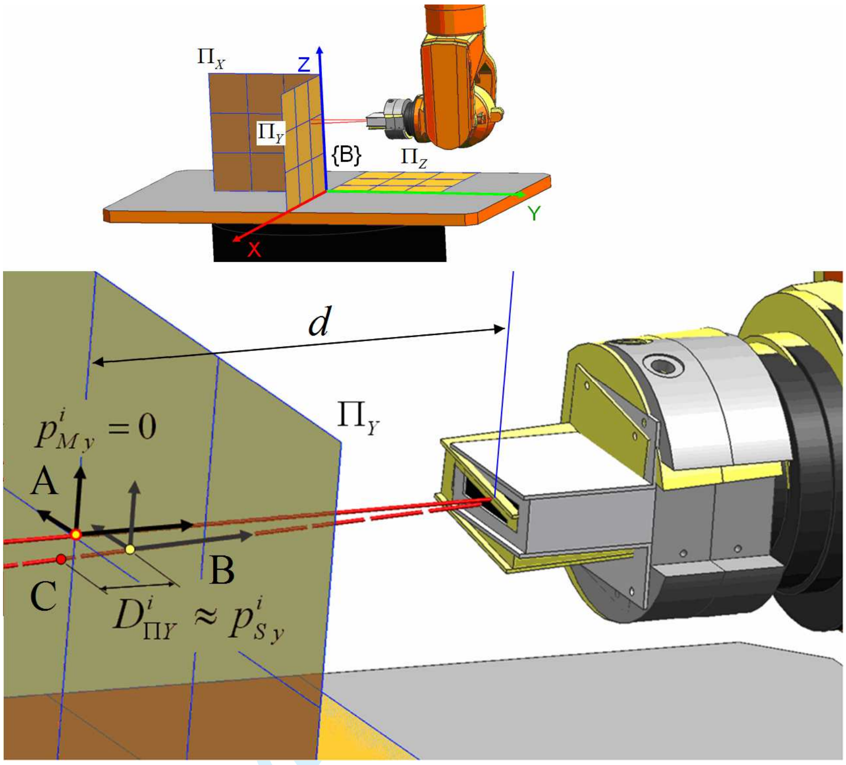

Figure 5. Three commanded meshes on the three respective planes are shown at the top; the coordinate $p_{S\,y}^{\,i}$ approximated by the distance $D^i_{\Pi_Y}$ to Π_Y is shown underneath.

er Review Only

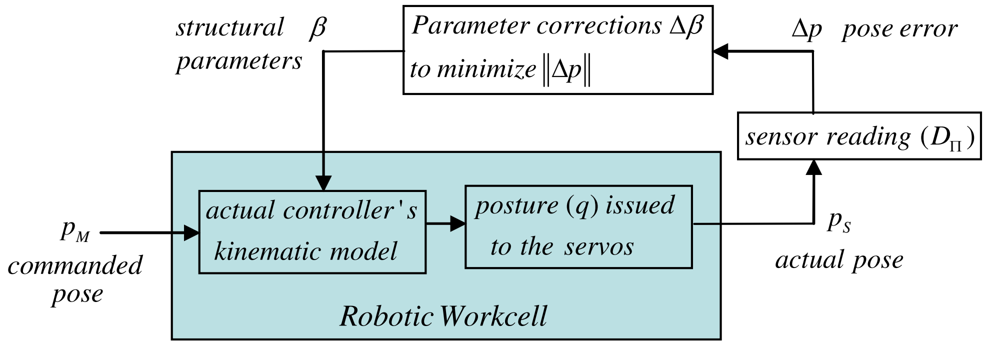

Figure 6. Open-loop procedure proposed for the workcell calibration.

For Peer Review Only

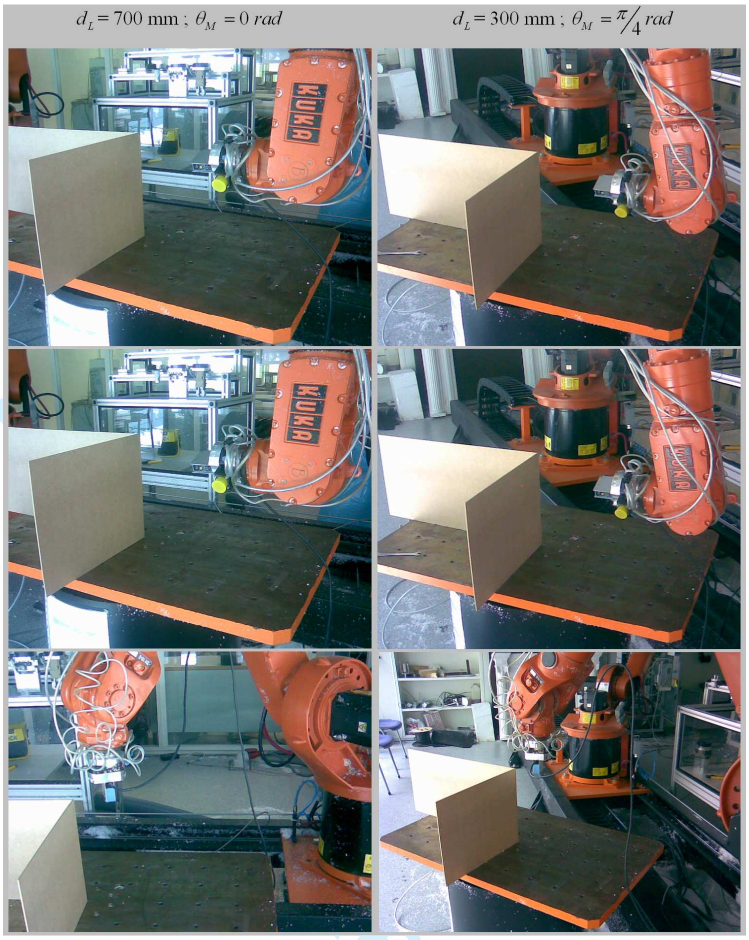
$d_L = 700$ mm ; $\theta_M = 0$ *rad*
$d_L = 300$ mm ; $\theta_M = {}^{\pi}\!/_{4}$ *rad*

Figure 7. Sweeping the reference planes with two configurations of d_L and θ_M .

For Peer Review Only

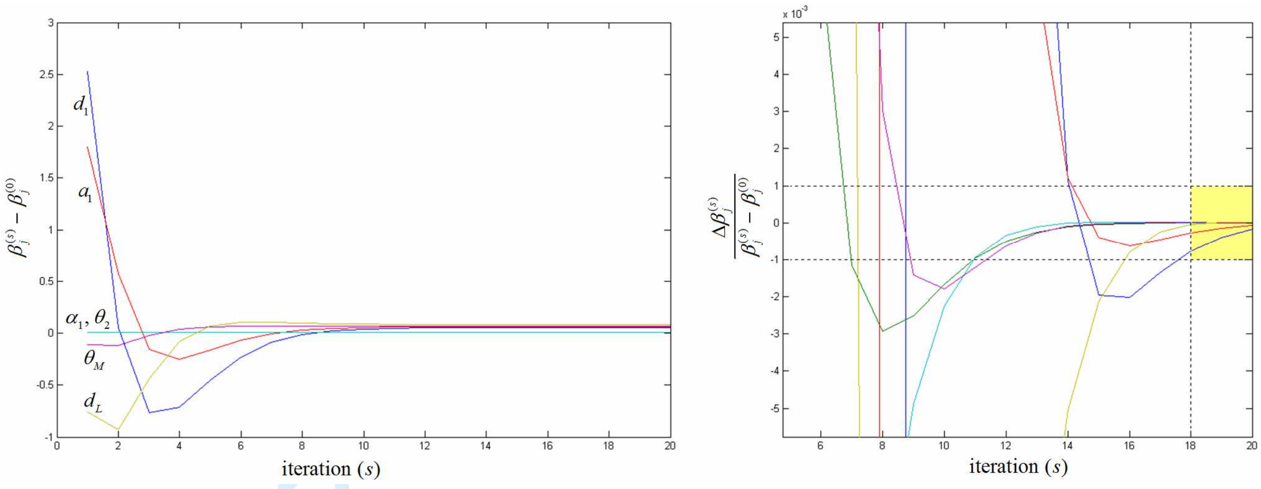

Figure 8. Left, the value $\beta_j^{(s)}$ of each parameter tends to an asymptotic value; right, the calibration algorithm meets the stop criterion at the18th iteration.

For Peer Review Only

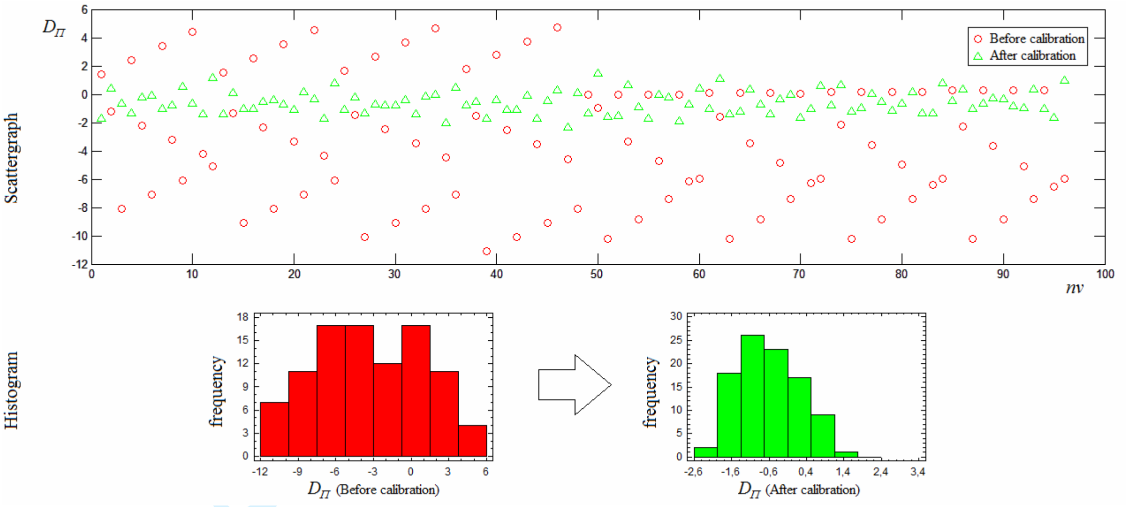

Figure 9. Validation of the calibration: scattergraph and histogram of the distance measurements D_Π to the table surface.

For Peer Review Only

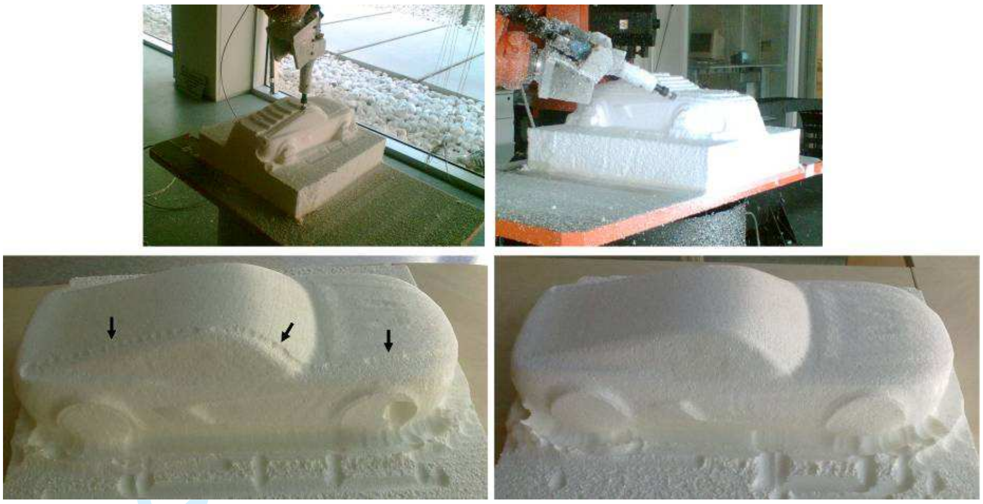

Figure 10. Milling of a small workpiece: bottom left, result without calibration; bottom right, result with calibration.

For Peer Review Only

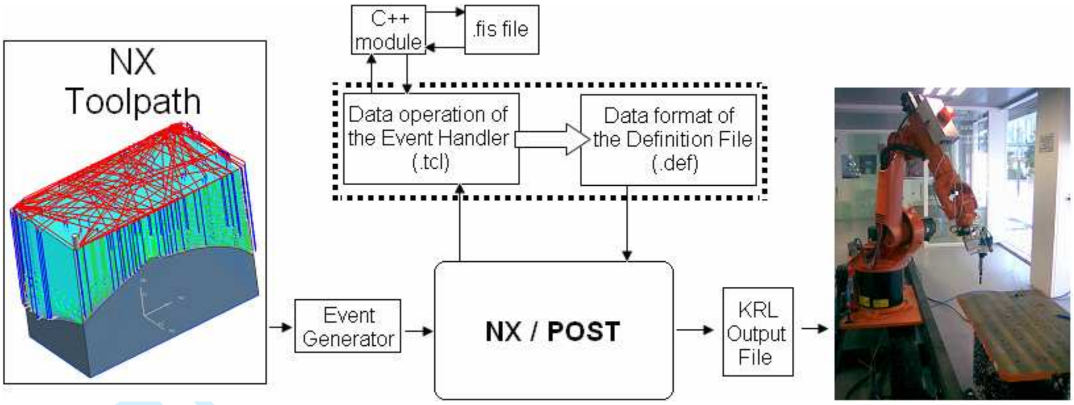

Figure 11. Integrated postprocessing in NX®: the ‘definition file’ and the ‘event handler’ adapt the toolpath to the controller’s Kuka Robot Language (KRL).

For Peer Review Only

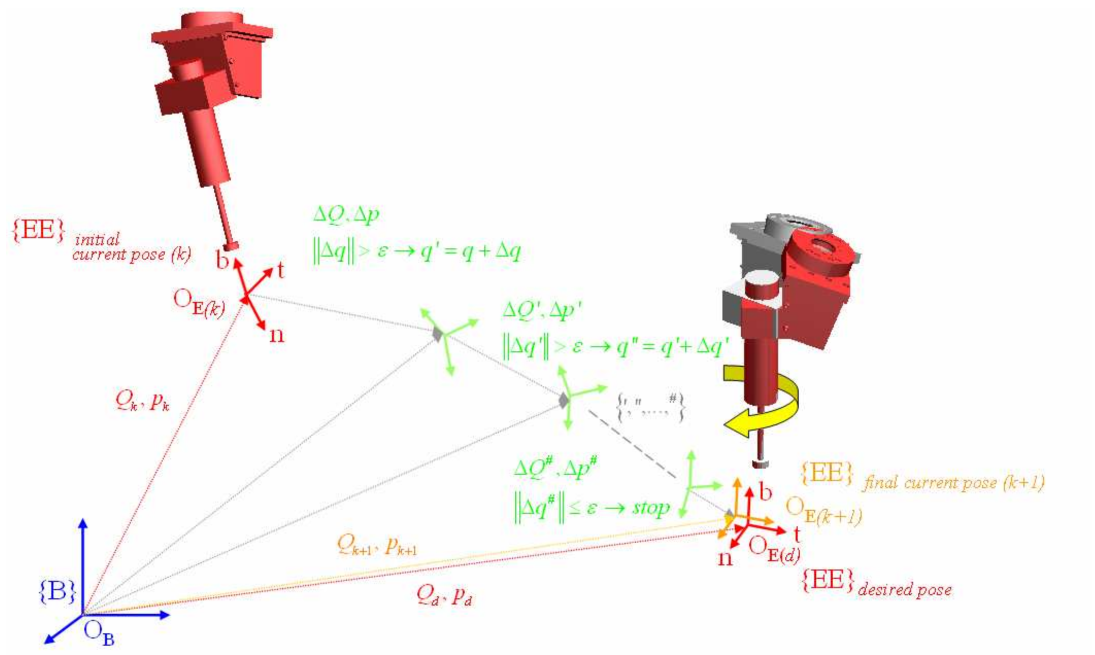
{EE} initial current pose (k)
b
t
n
$O_{E(k)}$
$\Delta Q, \Delta p$
$\|\Delta q\| > \varepsilon \rightarrow q' = q + \Delta q$
$\Delta Q', \Delta p'$
$\|\Delta q'\| > \varepsilon \rightarrow q'' = q' + \Delta q'$
$\{', '', \ldots, \#\}$
Q_k, p_k
$\Delta Q^{\#}, \Delta p^{\#}$
$\|\Delta q^{\#}\| \leq \varepsilon \rightarrow stop$
{EE} final current pose (k+1)
b
$O_{E(k+1)}$
t
n
$O_{E(d)}$
{EE} desired pose
Q_{k+1}, p_{k+1}
Q_d, p_d
{B}
O_B

Figure 12. Highlight of the loop leading from an initial current pose (*k*) to a desired final pose (*k+1*), both specified by the Frenet-Serret vectors *{t, n, b}*.

eer Review Only

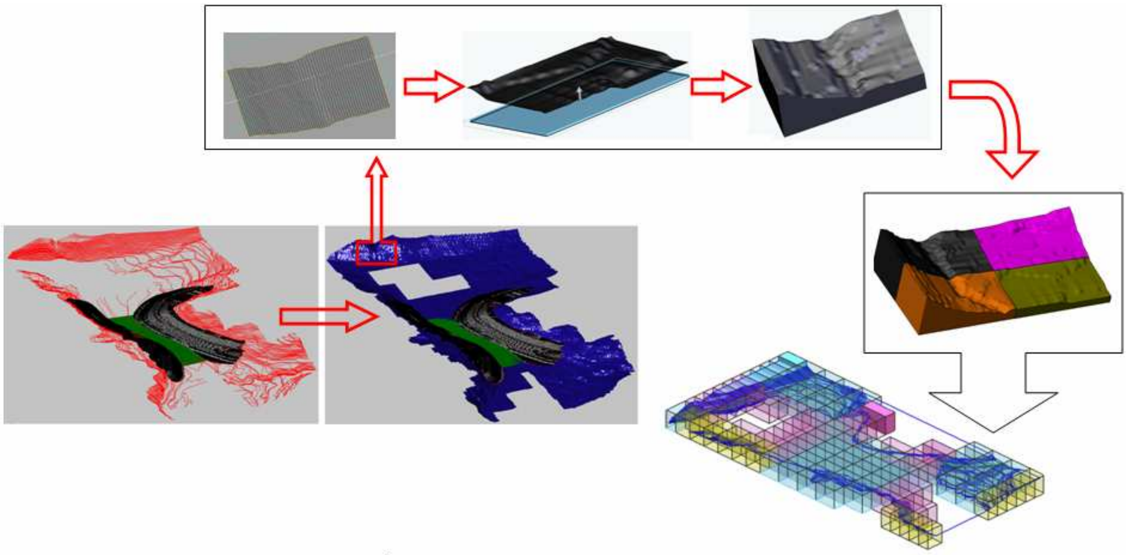

Figure 13. The orographic model is produced by assembling 120 blocks of 1x1x0.5 meters of EPS.

For Peer Review Only

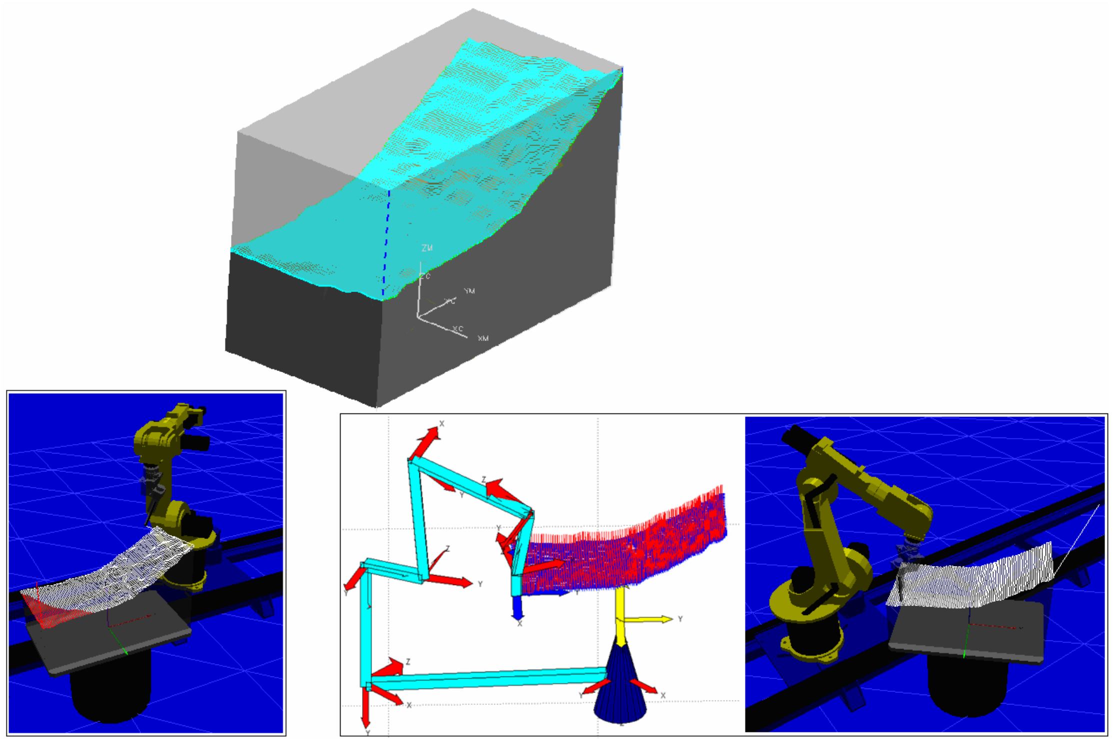

Figure 14. Toolpath postprocessed *without* (left) and *with* (right) the proposed RRS.

For Peer Review Only

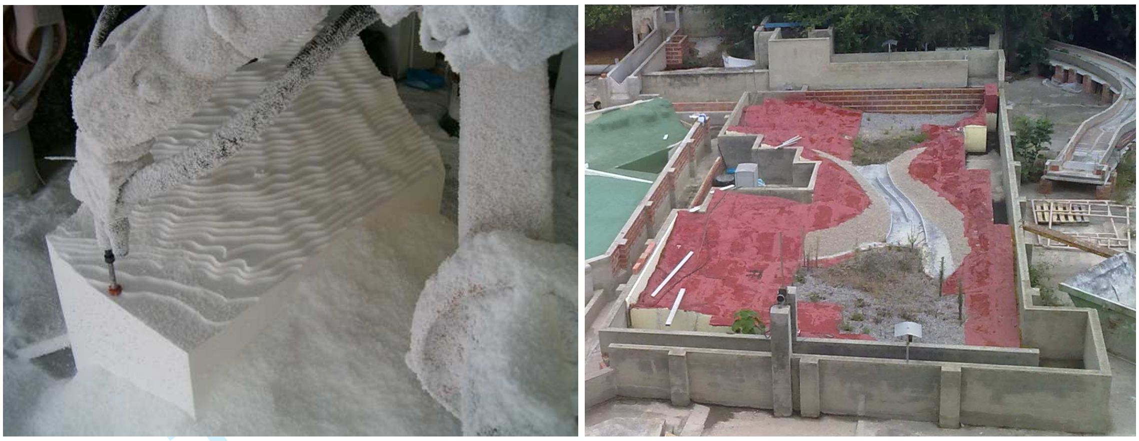

Figure 15. Left, milling process of an EPS block; right, final 1:75 model used for flowing simulation (real dimensions of 8x13 m).

For Peer Review Only

Table 1. Workcell parameters for the standard DH-model.

Link	α_i (rad)	a_i (mm)	θ_i (rad)	d_i (mm)
1	$\pi/2$	803	θ_M	-305
2	$\pi/2$	0	0	d_L
3	$\pi/2$	300	θ_1	-675
4	0	650	θ_2	0
5	$\pi/2$	155	θ_3	0
6	$\pi/2$	0	θ_4	-600
7	$\pi/2$	0	θ_5	0
8	0.3564	0	θ_6	-443.4
TCP	0	0	$\theta_{7(VJM)}$	-119.7

For Peer Review Only