

Document downloaded from:

<http://hdl.handle.net/10251/80118>

This paper must be cited as:

Andres De La Esperanza, FJ.; Gracia Calandin, LI.; Tornero Montserrat, J. (2012).
Implementation and testing of a CAM postprocessor for an industrial redundant workcell with
evaluation of several fuzzified Redundancy Resolution Schemes. *Robotics and Computer-
Integrated Manufacturing*. 28(2):265-274. doi:10.1016/j.rcim.2011.09.008.



The final publication is available at

<http://dx.doi.org/10.1016/j.rcim.2011.09.008>

Copyright Elsevier

Additional Information

Implementation and testing of a CAM postprocessor for an industrial redundant workcell with evaluation of several fuzzified Redundancy Resolution Schemes

Javier Andres*

Design and Manufacturing Institute
Universitat Politècnica de Valencia
Camino de Vera s/n
46022, Valencia, Spain
fraande@doctor.upv.es

*Corresponding author

Luis Gracia

Dept. of Systems Engineering and Control
Universitat Politècnica de Valencia
Camino de Vera s/n
46022, Valencia, Spain
luigraca@isa.upv.es

Josep Tornero

Dept. of Systems Engineering and Control
Universitat Politècnica de Valencia
Camino de Vera s/n
46022, Valencia, Spain
jtornero@isa.upv.es

Tel.: +34 963879942; fax: +34 963877060

ABSTRACT

This paper describes the implementation of a postprocessor to adapt the toolpath generated by a CAM system (NX™) to a complex workcell of eight joints (namely, a KUKA KR15/2 manipulator mounted on a linear track and synchronized with a rotary table), devoted to the rapid prototyping of 3D CAD-defined products. Previously, it evaluates several Redundancy Resolution Schemes at the joint-rate level for the configuration of the postprocessor, dealing not only with the additional joints but also with the redundancy due to the symmetry on the milling tool. The use of these redundancies is optimized by adjusting two performance criterion vectors related to both singularity avoidance and maintenance of a preferred reference posture, as secondary tasks to be done during the path tracking. In addition, two proper fuzzy inference engines actively adjust the weight of each joint in these tasks. The postprocessor is validated in a real prototyping of a *Valencian Falla*.

Keywords: Postprocessor, redundant robot, fuzzy, milling, CAM.

1. Introduction

Rapid prototyping with soft materials is of increasing importance in order to support the product development process in *industrial design engineering*, specifically to get physical replicas of CAD (*Computer Aided Design*) defined models. In this context, relatively large prototypes require redundant robotic workcells due to their high flexibility and large working areas. In conjunction, the implantation of more sophisticated CAD/CAM/ROB integrated manufacturing systems reduces the time invested in successive verifications, adjustments and translations in the machining process.

Leading commercial CAM (*Computer Aided Manufacturing*) systems plan off-line the cutting toolpaths as a discrete set of close-enough tool poses. As the milling tool has a symmetry axis that allows rotating the tool without affecting the task (see Fig. 1), these systems only specify five parameters to carry out the milling task: three location and two orientation coordinates of the cutter referred to a Cartesian coordinate system (*base*, {B}). Thus, at milling works the dimension of the *Task space* (T) is $t=5$. It is outstanding that the cutter's tracking data are directly related with the desired finish conditions of the workpiece. Thus, these data are mandatory and independent from the machine tool that will manufacture the workpiece (putting particular calibration efforts on one side [1]). However, this information has to be *postprocessed* (i.e., adapted) from the CAM system to the production system that is going to be used.

At the *Design and Manufacturing Institute* (IDF), in the *Universitat Politècnica de Valencia* (UPV), a sculpturing redundant workcell has been configured to test milling methods for rapid prototyping. As shown in the Fig. 1, an industrial KUKA KR15/2 arm with six revolute (R) joints is mounted on a linear track (d_L , being it an additional prismatic joint, P), and it works over a synchronized rotary table (θ_M , being it an additional R-joint) on which the initial blank of material is set. Due to the fact that P and R joints only allow one *degree of freedom* (DOF), both additional joints plus the six of the robotic arm complete a workcell with a *Joint space* (\mathfrak{S}) dimension of $n=8$.

To deal with a kinematic analysis, the table can be arbitrary regarded as fixed, while the other movable *end-effector* (EE) bears the cutter tool in the Cartesian *Operational workspace* (Ω). As the pose of a rigid body in Ω is specified with three location and three orientation coordinates, $\dim(\Omega)=m=6$.

The introduced workcell is *redundant* as $n > t$ (with $T \subseteq \Omega$) [2], with a degree of *kinematic redundancy* (r_k) of $r_k = n - t = 3$. The main difficulty of postprocessing a toolpath from the CAM system to this workcell focuses on managing the redundancy in order to avoid *manipulator postures* near singularities or limits of range, while reaching the successive *cutter poses* of the toolpath.

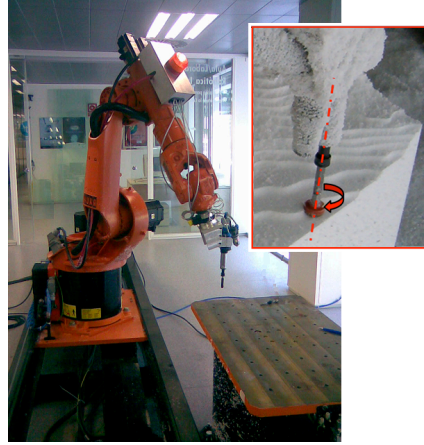


Fig. 1. View of the KUKA workcell at the IDF-UPV. Detail of the irrelevant axis of symmetry of the milling tool.

This paper revises diverse *Redundancy Resolution Schemes* (RRS) at joint rate level to deal with the postprocessing stage from the CAM system to the redundant workcell. After describing the workcell at Section 2, the implementation of different criteria for singularity avoidance and maintenance of a preferred posture is discussed at Section 3. These criteria also incorporate the Fuzzy Logics as a possible manner of tuning the RRS attending to an expert criterion. Subsequent to the explanation the algorithm used for continuous path tracking (Section 4), Section 5 presents an effective implementation of a CAM-Robotics integrated postprocessor for an automatic off-line generation of the robot instructions to carry out milling tasks. Section 6 makes a test on the designed postprocessor, which is finally used in a practical application, Section 7. Finally, Section 8 exposes several conclusions to this study.

2. Kinematic modelling of the workcell

A *kinematic model* is the mathematical description required to control adequately the *posture* of the chain and the associated *pose* of the EE while performing a task. The *Direct Kinematic Problem* (DKP) is the mapping from the *Joint* (\mathfrak{S}) to the *Operational* (Ω) *space*, i.e., determining the pose of the EE for a given manipulator posture. On the contrary, the *Inverse Kinematic Problem* (IKP) consists of determining the posture of the manipulator for a given pose of its EE.

At the displacement level, the DKP is straightforward. Thus, a point in \mathfrak{S} represents a unique pose of the EE referred to the base $\{B\}$ at Ω . The *standard* Denavit-Hartenberg (DH) model [3][4] is used in this study due to its simplicity and reputation in the robotics community. It represents the EE pose as a 4×4 homogeneous transformation matrix that results from the operation of the descriptive parameters of the workcell ($a_i, \alpha_i, d_i, \theta_i$) resumed in Fig. 2 and Table 1. For a revolute axis θ_i is the joint variable and d_i is constant, while for a prismatic joint d_i is variable and θ_i is constant.

The IKP is interesting since CAM systems specify the cutter tracking at Ω , while each particular robot controller works at \mathfrak{S} . The IKP at the *displacement* level is much more challenging since an infinite number of solutions may exist in the case of a redundant manipulator. The resolution of the IKP for the IDF's workcell is profusely described at [5] taking as entry arguments the current θ_M and d_L values.

Whitney [6] first introduced differential kinematic relationships to solve the motion control problem, namely, the *resolved-motion rate* control. The *forward kinematics* (DKP) between the EE velocity (t) and the joint velocity (\mathfrak{F}) is represented by the *linear algebraic* equation,

$$t = J \mathfrak{F} \quad (1)$$

where the coefficient matrix (Jacobian, J) is a non-linear function of joint angles. From the two different conceptions for the Jacobian matrix, namely the *geometric* and the *analytical* J [7], the first one suggested by Whitney [8] is used as it simplifies the computation [4].

This matrix maps the joint rates, grouped into the 8-dimensional vector $\dot{\phi} = [\dot{\phi}_M \ \dot{\phi}_L \ \dot{\phi}_1 \ \dots \ \dot{\phi}_6]^T$, into the *twist* vector ($t = [\omega \ v]^T$) [4], with $\omega = [\omega_x \ \omega_y \ \omega_z]^T$ and $v = [v_x \ v_y \ v_z]^T$ denoting the angular and linear velocities of the EE's reference frame relative to {B}, respectively (Fig. 2). Again, the IKP is most useful for path tracking, i.e. given the desired twist of the EE the aim is to obtain the joint velocities, with J known. In the case of non-redundant robots, J is a square matrix and the solution can be normally found with the inverse of J , namely $\dot{\phi} = J^{-1}t$. However, a difficulty arises in the case of *redundant* robots having a non-square J (i.e., more columns than rows). The solution $\dot{\phi}$ that better fits all the equations of the system with a minimum least-squares criterion is achieved with the use of the *right Moore-Penrose pseudo-inverse* ($J^\dagger \equiv J^T (JJ^T)^{-1}$), namely $\dot{\phi} = J^\dagger t$. Even so, a *homogeneous component* can be added in order to perform a *secondary task* at the cost of giving up the minimum-norm solution:

$$\dot{\phi} = J^\dagger t + (I_{n \times n} - J^\dagger J)h \quad (2)$$

By using Eq. (2), the manipulator is to be required to track desired target positions as *primary task*. The second part of Eq. (2) is an arbitrary vector from the *Null Space* of J , $\mathfrak{N}(J)$, being $(I - J^\dagger J)$ the *projection operator* on $\mathfrak{N}(J)$. Thus, one can try to achieve secondary goals by suitably choosing h to be projected into self-motions (namely, chain re-postures but maintaining EE's pose). Eq. (2) lead to different RRS.

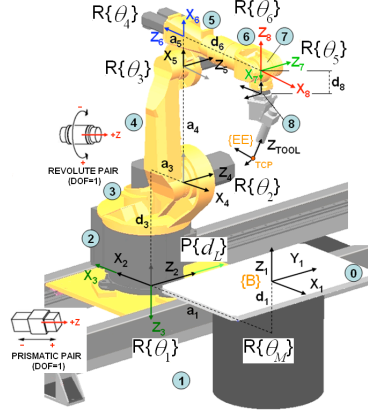


Fig. 2. Frame assignment for the Denavit-Hartenberg representation of the RP-6R workcell.

Some other variations of Eq. (2) could be considered in order to optimize the motion of the manipulator, namely those using a *weighted pseudo-inverse*, i.e. $J_W^\dagger \equiv W^{-1}J^T (JW^{-1}J^T)^{-1}$, and those using a *damped least-square (DLS-) inverse*, i.e. $J^{\dagger\lambda} = J^T (JJ^T + \lambda^2 I)^{-1}$ where λ is the damping factor. In the first case, W is a positive-definite weighting matrix to give greater weight to some joints over others, by converting Eq. (2) into

$$\dot{\phi} = J_W^\dagger t + (I_{n \times n} - J^\dagger J)h \quad (3)$$

The joint weights are tuned (e.g., in a similar way as it is described in Section 3.4) to improve the main milling task [10], which is different (but complementary) from the scope of this study. In this work, the weighting factors are tuned to improve the secondary tasks of *singularity avoidance* and *reference posture maintenance*, which are feasible in a redundant manipulator.

The *DLS-inverse* [9] could be considered as a way to overcome kinematic singularities by means of

$$\dot{\phi} = J^{\dagger\lambda} t + (I_{n \times n} - J^\dagger J)h \quad (4)$$

where $J^{\dagger\lambda}$ is a modified Jacobian that is non-singular in the whole workspace. Under this control, one obtains only an approximate inverse kinematic solution and the problem is to select a suitable value for the damping factor λ in order to reach a compromise between good behaviour and accuracy in the neighbourhood of singular points. While this method could apply for tasks with $t=5$ such as painting, it

does not apply for milling tasks where pose accuracy is a priority. For this reason, this method is not considered this work.

3. Redundancy Resolution Schemes

Prior to any control scheme, it may be interesting to visit the considerations that some authors [11][12] made on the concept of redundancy in welding 6R robots. They distinguish two types of redundancy: *intrinsic* ($r_I = n - m$ when $m > n$) and *functional* ($r_F = m - t$ when $m > t$). For the scope of this study, devoted to milling tasks, the workcell of Fig. 2 has a degree of intrinsic redundancy of two ($r_I = 8 - 6 = 2$) and a degree of functional redundancy of one ($r_F = 6 - 5 = 1$), associated to the symmetry axis of the cutter. Reasonably, the kinematic redundancy of the pair workcell-task is $r_K = r_I + r_F = n - t = 3$, as previously stated.

Putting the additional joints (θ_M, d_L) aside, J is a full rank square matrix ($\mathfrak{R}(J) = \emptyset$) for 6R manipulators, so the second term of Eq. (2) could not be used. Furthermore, the *functional* redundancy should be captured in the model. Nevertheless, this kind of redundancy is not present in Eq. (2) even with the columns added to J by the rotary table and the linear track. To capture the functional redundancy, two methods can be used: one by augmenting the dimension of \mathfrak{S} (*Virtual Joint Method, VJM*) and another one by reducing the dimension of the twist (namely, *Twist Decomposition Method, TDM*). The VJM merely adds a virtual joint on the symmetry axis of the tool (Fig. 3), in order to obtain an extra DOF of redundancy. Therefore, Eq. (2) can be rewritten as

$$\mathfrak{S}_v = \mathcal{J}_v^0 t + (I_{(n+1) \times (n+1)} - J_v^\dagger J_v) h \quad (5)$$

where J_v is an *augmented* Jacobian matrix by a virtual joint-rate \mathfrak{S}_v , namely

$$\mathfrak{S}_v = \begin{bmatrix} \mathfrak{S}_M & \mathfrak{S}_L & \mathfrak{S}_1 & \dots & \mathfrak{S}_6 \\ & & \mathfrak{S}_7 & & \end{bmatrix}^T \quad (6)$$

The TDM, developed by Baron and Huo [11][13], decomposes the angular velocity of the twist vector ($t, 2 \times R^3$) at the minimum norm solution of Eq. (2) into two orthogonal components: one in the task subspace (ω_t) and another lying into the orthogonal task subspace (ω_{t^\perp}). The later projection is irrelevant for the task (Fig. 3) and it is replaced with an arbitrary vector h properly projected and allowing a secondary task to be satisfied, namely

$$\mathfrak{S} = (J^\dagger T) t + J^\dagger (I_{6 \times 6} - T) J h; \quad T \equiv \begin{bmatrix} (I_{3 \times 3} - ee^T) & 0 \\ 0 & I_{3 \times 3} \end{bmatrix} \quad (7)$$

where T is the *twist projector* matrix (see [11][13]). The unit vector e denotes the orientation of the axis of symmetry along the milling tool.

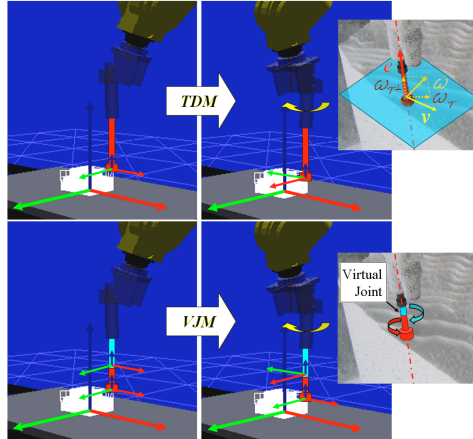


Fig. 3. Kinematic interpretation of the VJM and the TDM when considering the symmetry axis of the milling tool.

A final consideration can be done for the workcell studied, where both intrinsic and functional redundancies exist. It may be profitable to study the combination of the TDM with a projection on $\mathfrak{R}(J)$, in the following form

$$\mathfrak{R} = (J^\dagger T)t + J^\dagger (I_{6 \times 6} - T)J h_1 + (I_{nn} - J^\dagger J)h_2 \quad (8)$$

where h_1 and h_2 are the two feasible *performance criterion vectors* projected as functional and intrinsic redundant tasks, respectively. To the authors' knowledge, a solution like Eq. (8) has not been tested yet. Clearly, Eq. (7) is a particular case of Eq. (8) with $h_2 = 0$, and Eq. (2) corresponds to tasks with all translation and rotation coordinates of the EE required (with $T = I_{6 \times 6}$).

The *performance criterion vector* (h) can be considered as a *virtual force* that attempts to push the configuration of the manipulator away from a critical area in \mathfrak{S} [14], as secondary task. Nevertheless, h has a different signification in both VJM and TDM. In case of the VJM, fully based on Eq. (2), h is a motion projected on $\mathfrak{R}(J)$, i.e., the *tool centre point* or TCP, see Fig. 2, is not displaced by the action of h . In the TDM, this secondary motion can be appreciated through the symmetry axis of the tool tip from the base frame (Fig. 3).

The most widespread method used to select h is the *Gradient projection method* (GPM), introduced by Liégeois [15]. It takes the minimization of a position-dependent scalar (*performance criterion index*, p) by means of its gradient vector, namely

$$h = -k \cdot \nabla p ; \text{ with } \nabla p = \left[\frac{\partial p(q)}{\partial q_1} \quad \frac{\partial p(q)}{\partial q_2} \quad \dots \quad \frac{\partial p(q)}{\partial q_n} \right]^T \quad (9)$$

3.1. Performance criterion for singularity avoidance (h_{cond})

Several authors have used Yoshikawa's manipulability [16][17] as the distance criterion to avoid postures with poor *kinematic performance* inside the reachable workspace (those near to an *internal singularity* [18][19]). However, in some circumstances the determinant of a matrix is useless in assessing the invertibility of J . Instead, Angeles [4][20] preferred the *condition number of J* , $k(J)$, which can be considered as the rate at which \mathfrak{R} will change with respect to a small change in t in Eq. (1). The condition number also gives an upper bound for the round-off-error amplification in the solution [4]. Therefore, it is desirable for the robot to work in any posture minimizing $k(J)$. This scalar quantity measures how well the system behaves with regard to force and motion transmission through links and joints along the manipulator [4], so it also promotes the better efficiency of the cutter over the workpiece. Thus, it is encouraged the use of a complementary force-feedback control, such as that presented in [21], in order to tune the cutter speed to limit the forces transmitted through the robotic chain.

In this paper, k_F is taken into account to achieve a definitive h . It adopts a weighted Frobenius norm due to its lower computational cost compared with other norms. An expression valid for any $6 \times n$ Jacobian matrix, with $n \geq 6$, is [4]:

$$k_F(H) = \frac{1}{6} \sqrt{\text{tr}(HH^T) \text{tr}[(HH^T)^{-1}]} ; \text{ with } 1 \leq k_F < \infty \quad (10)$$

Compared to k_F , the $1/k_F$ has the advantage of being comprised between 0 and 1. In Eq. (10), H is a *homogeneous Jacobian*. In it, the lack of homogeneity of J is counteracted with the *characteristic length* (L) of the robot [22][23], by which we divide the Jacobian entries that have units of length. This parameter, defined as "the normalizing length that renders the condition number of the Jacobian matrix a minimum" [4], can be determined by using the Nelder-Mead simplex method (Matlab's function *fminsearch*) (see [22] for the procedure). The *straightforward problem* of determining L for the KR 15/2, leaving apart the additional joints (θ_M, d_L) which amount to a rigid-body motion of the overall 6R chain, gave a result of $L=350.6$ mm. The best conditioning achieved was $k_F=1.247$, which corresponds with the posture depicted in Fig. 4.

The performance criterion *index* is calculated as

$$p_{cond} = \frac{k_F}{2} (q - q_{Ts})^T W_{cond} (q - q_{Ts}) \quad (11)$$

This index is to be activated when the k_F value passes over a preset threshold value, ζ . At that instant, the corresponding configuration (q_{Ts}) is recorded to evaluate the distance to the actual posture at \mathfrak{S} :

$$h_{cond} = -\nabla p_{cond} = -W_{cond} k_F (q - q_{Ts}) \quad (12)$$

3.2. Performance criterion for reference posture maintenance (h_{jnt})

A certain constant reference arm posture (q^{ref}) may be desirable for avoiding collision with an obstacle [16] or reach the mechanical joint-limits by maintaining the manipulator as close as possible to the *mid-joint reference posture* [12]. A performance criterion *index* could be written as

$$p_{jnt} = \frac{1}{2} (q - q^{ref})^T W_{jnt} (q - q^{ref}) \quad (13)$$

thus,

$$h_{jnt} = -\nabla p_{jnt} = -W_{jnt} (q - q^{ref}) \quad (14)$$

The *mid-joint posture* of the KR 15/2 is not appropriate as a reference for milling tasks, neither the *best-conditioned posture* which is quite near of some mechanical joint limits (Fig. 4). For the scope of this article, the well-known HOME posture ($q_0 = [\pi \ 0 \ \pi \ -\pi/2 \ 0 \ 0 \ \pi/2 \ 0]^T$ rad, Fig. 11) is taken as the reference posture for Eq. (14) as a compromise between both objectives, i.e. $q^{ref} = q_0$.

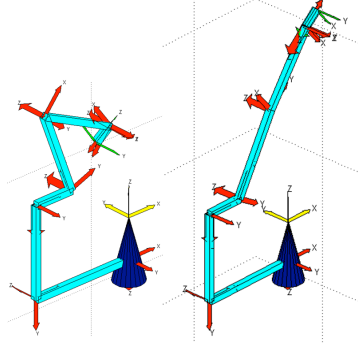


Fig. 4. Left, best conditioned posture for the 6R KR 15/2 manipulator. Right, mechanical mid-joint posture.

3.3. Combined performance criterion

The two performance criteria described above can be combined into a unique vector, which simultaneously is to maintain the manipulator as close as possible to the q^{ref} posture and as far as possible of bad conditioned postures. This combined performance criterion, to be applied at VJM of Eq. (5), could be written as

$$h = -\nabla p = -\nabla (p_{jnt} + p_{cond}) = h_{jnt} + h_{cond} = -(W_{jnt} (q - q^{ref}) + W_{cond} k_F (q - q_{Ts})) \quad (15)$$

At this point, it is also possible to highlight the convenience of evaluating as h_1 and h_2 in Eq. (8) as the respective sub-tasks of h_{jnt} and h_{cond} .

3.4. Weighting factor

It is remarkable that the choice of the weighting diagonal matrices, W_{cond} and W_{jnt} , is a major difficulty when implementing the previous RRS due to subjectivity. By tuning both matrices, the relative importance amongst the joints in each sub-task is adjusted. It can be critical for the performance of the RRS and traditionally this task has been set based on trial and error [13]. Higher weights are to be assigned to those joints that are supposed to be more reactive when lowering the condition number or being far of the reference posture. Previous studies assigned constant weights [10][11][12][13]. Recently, Huo and Baron [24] have also reported the need of adjusting these weights. They present a self-adaptation algorithm to search an optimum-weighting vector in replacement of the low-effective manual tuning of weights. In practice, in case of milling tasks where the tool *pose* (and hence the robot *posture*) changes

constantly, it may be desirable to identify an appropriate value of W at each configuration in a reasonable time. Thus, alternatively the present paper uses *Fuzzy Logics* as a *cost-effective computational* strategy for the assignment of variable weights to the performance vector h . As it will be shown, a *fuzzy inference controller* that captures the practical knowledge of the technician devotes to this assignment.

4. Continuous path tracking

Once CAM system has generated a continuous trajectory as a discrete set of close-enough poses at Ω , the EE of the robot is expected track this path. A tangent, normal, and binormal unit vectors ($\{t, n, b\}$, respectively) can be associated with every sample point of the trajectory, namely the *Frenet-Serret* vectors, indicating the required pose (Fig. 5).

The joint angles of the robot have to be calculated along this continuous set of poses of the EE. In principle, an IKP could be solved at each sampled pose. However, with a redundant manipulator not giving rise to a simple solution, an alternative iteratively approach can be applied by using J :

$$J(q^i)\Delta q^i = \Delta t^i \quad (16)$$

with Δt^i defined as

$$\Delta t^i \equiv \begin{bmatrix} Q_k \text{vect}(Q_k^T Q_d) \\ \Delta p \end{bmatrix} \quad (17)$$

where Q_k represents the current rotation matrix from base frame to EE frame (namely, the first three rows and columns of the current homogeneous transformation matrix resulting from the DH model) and Q_d represents the desired rotation matrix. They have the following relationship

$$Q_d = Q_k \Delta Q \rightarrow \Delta Q = Q_k^T Q_d \quad (18)$$

Function $\text{vect}(\Delta Q)$ represents the axial vector of a 3x3 rotation matrix ΔQ , and is calculated as

$$\text{vect}(\Delta Q) \equiv \frac{1}{2} \begin{bmatrix} \Delta Q_{32} - \Delta Q_{23} \\ \Delta Q_{13} - \Delta Q_{31} \\ \Delta Q_{21} - \Delta Q_{12} \end{bmatrix} \quad (19)$$

Vector Δp is defined as the difference between the prescribed value p_d of the position vector of the operation point and its current value p_k . The relations amongst Q_d , p_d , Q_k , p_k , and ΔQ , Δp are shown in Fig. 5. The following generic algorithm is deduced in [4], from which some variations in the 8th step will be done in following sections depending on the RRS:

Algorithm 1

- 1) $q \leftarrow$ initial joint position
- 2) $\{p_d, Q_d\} \leftarrow$ desired EE pose
- 3) $\{p, Q\} \leftarrow$ DK(q)
- 4) $\Delta Q \leftarrow Q^T Q_d$
- 5) $\Delta p \leftarrow p_d - p$
- 6) $\Delta t \leftarrow \begin{bmatrix} Q \text{vect}(\Delta Q) \\ \Delta p \end{bmatrix}$
- 7) $J_g(q) \leftarrow$ DK(q)
- 8) $\Delta q \leftarrow$ Redundancy Resolution Scheme
- 9) if $\|\Delta q\| \leq \varepsilon \Rightarrow$ STOP, else
- 10) $q \leftarrow q + \Delta q$, and go to step 3

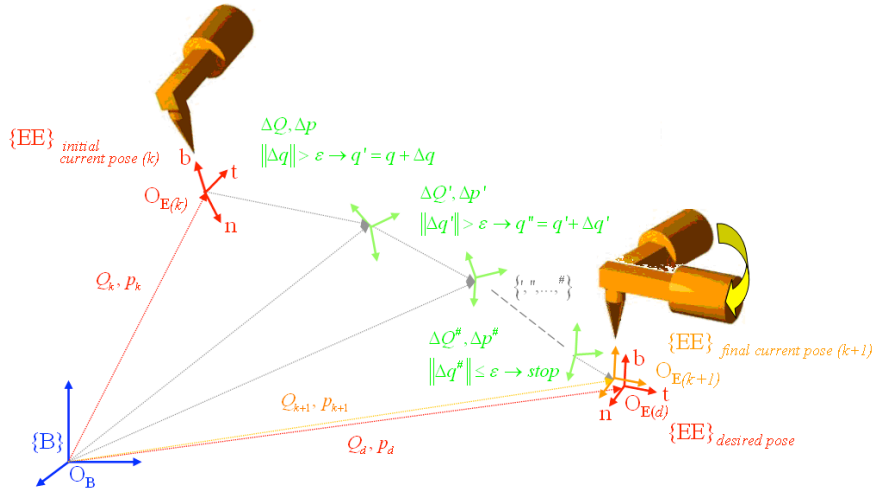


Fig. 5. Highlight of the loop leading from an initial current pose (k) to a desired final pose ($k+1$), both specified by the Frenet-Serret frames (tangent, normal and binormal).

5. Implementation

Besides the specific treatment of the CAD file imported to the CAM system (NX), some common guidelines about the used material and methods are exposed.

5.1. Material

NXTM (Siemens) is a digital product development system that integrates and fully associates the labours of design (CAD), simulation (CAE) and manufacturing (CAM). The CAM module makes possible the planning of milling tasks, but it also can interact with two program codes in TCL (*Tool Command Language*) (connected with C++ modules) that manipulate the path data (*Event Handler*) and give the convenient format to the generated output (*Definition File*) [25].

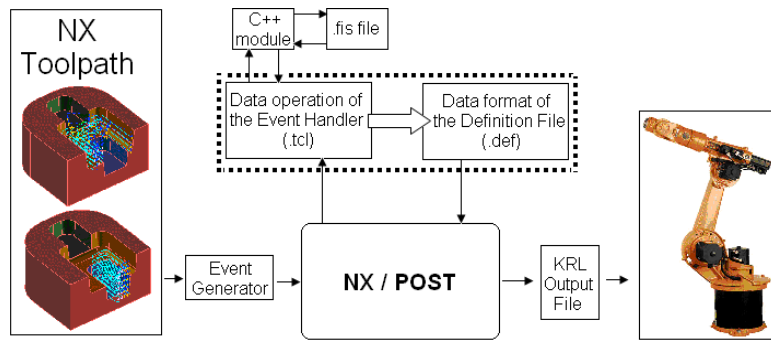


Fig. 6. The Definition File and the Event Handler programs postprocess the toolpath from NX to the KUKA Robot Language (KRL).

All the treatment of the data is programmed in MATLAB with the aid of the Hemero toolbox [26] and the Fuzzy Logic (FL) toolbox [27]. The FL-toolbox generates a .fis file (namely, *fuzzy inference system* file) which saves the characteristics of a designed fuzzy inference engine (input/output variables and rules). It also generates a stand-alone fuzzy inference engine in C++.

In addition, RobomoveTM (Qdesign) [28], was used to visualize the robot postures resulting from each RSS implemented (Fig. 11).

5.2. Method

5.2.1. RRS programming

The trajectory data, generated with NX, is kept as T_{CAM} . Starting from the HOME posture (q_0 , like shown in Fig. 2) and with the DH models of both the KR15/2 manipulator (*DH-KR15/2*) and the complete workcell (*DH-Workcell*) at hand, the programmed algorithm is summarized in Algorithm 2 on the basis of

Algorithm 1. In Algorithm 2 the sub-index *Workcell* refers to the complete kinematic chain of the workcell, i.e. including the linear track and the rotary table, while for the calculus of k_F in the 8th step, the sub-index *6R* refers to the isolated KR15/2 manipulator. This algorithm is customized in the 9th step for each of the two RRS studied. It is remarkable that the DH representation of the manipulators depends on the RRS selected, according to the significance explained in Section 4, by adding the additional joint or considering a fixed displacement up to the tool tip. For the VJM (Fig. 7, left), and additional line is added in the DH-model (Table 1). The TDM uses the actual DH-model, and therefore a final constant displacement matrix is required to know the position of the EE (Fig. 7, right).

Algorithm 2

```

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

```

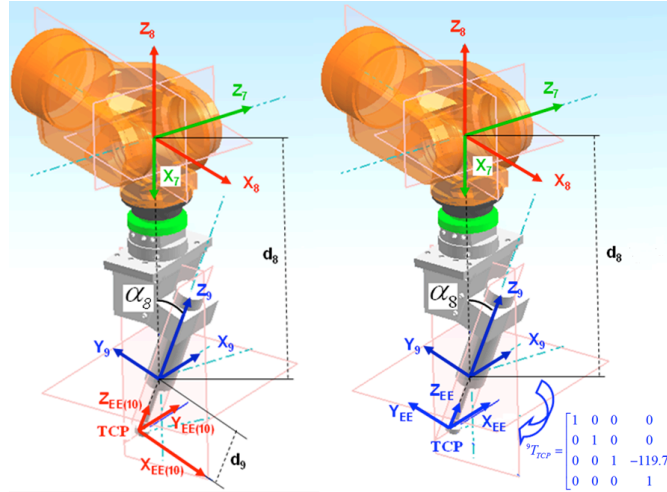


Fig. 7. Comparison of the DH frame assignment for the VJM (left) and the TDM (right).

Table 1. Table resumming the parameters for both standard DH-models.

Link	α_i (rad)	a_i (mm)	θ_i (rad)	d_i (mm)
1	$\pi/2$	803	θ_M	-305
2	$\pi/2$	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

5.2.2. Adapted Fuzzy weighting vector

Two *fuzzy inference engines* (FIE) can be conceived for the assignment of variable weights to each performance criterion vector, i.e. h_{jnt} and h_{cond} . In this labour, the consideration of the expert knowledge is of major importance. These engines were developed in MATLAB by means of its *FL-Toolbox* [27].

In the workspace over the table, the k_F is expected to decrease when the robot acquires a posture near the *extended-arm* or the *wrist* singularities [19][29]. In order to avoid these postures, joints θ_3 , θ_5 and the additional joints (θ_M, d_L) have a direct implication. However, in the case of the maintenance of a q^{ref} , it may be convenient to work near this reference posture of the joints doing the *gross positioning* ($\theta_1, \theta_2, \theta_3$) while a *fine orientation* ($\theta_4, \theta_5, \theta_6$) is being done [5], so it seems to be logical making different assignments for both groups of joints.

Based this reasoning, the output variables for each FIE are those weights associated to the joints related with the aspects highlighted, leaving a default value to those not taken into account (Table 2).

Table 2. Diagonal entrances of the *fuzzyfied* weighting matrixes, according to experience.

	θ_M	d_L	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	$\theta_{7(VJM)}$
W_{jnt}	w_{Mjnt}	w_{Ljnt}	0.01	0.01	w_{3jnt}	0.01	w_{5jnt}	0.01	0.01
W_{cond}	w_{Mcond}	w_{Lcond}	w_{gross}	w_{gross}	w_{gross}	w_{fine}	w_{fine}	w_{fine}	0.01

Based on those data and after studying which robot joints affect k_F and the maintenance of q^{ref} the most, the input variables are structured. Two inputs are defined in case of the conditioning (θ_3, θ_5) and three in case of the maintenance of the reference posture ($\theta_2, \theta_3, \theta_5$). Moreover, it is noteworthy that the absolute value of θ_5 , namely $|\theta_5|$, can be considered due to symmetry in the range of this joint.

In these input spaces, the number of clusters is related to the linguistic etiquettes assigned, according to the experience. In case of the input spaces (θ_2, θ_3 or θ_5), they are divided in three triangular clusters (Fig. 8). It can be noticed that the functions neither are equidistant nor have identical form, as they are tuned according to experience. Analogously, the output spaces are different for each fuzzy inference system, depending also on the experience.

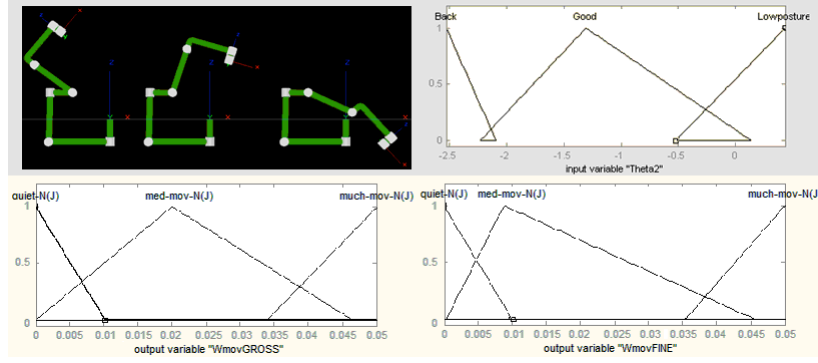


Fig. 8. Up, representation of the peak posture of the respective the three clusters in which the input space θ_2 is divided. Down, output space representation for the w_{gross} and w_{fine} assignment.

The last requirement to run the FIE consists in the compilation of a few “if-then” rules relating both input and output spaces, according to the experience. Giving too much rules can be cumbersome and moves away from the desired simplicity of a fuzzy inference system. For this test, four rules were created for the reference posture criterion, and two for the singularity avoidance criterion (Fig. 9).

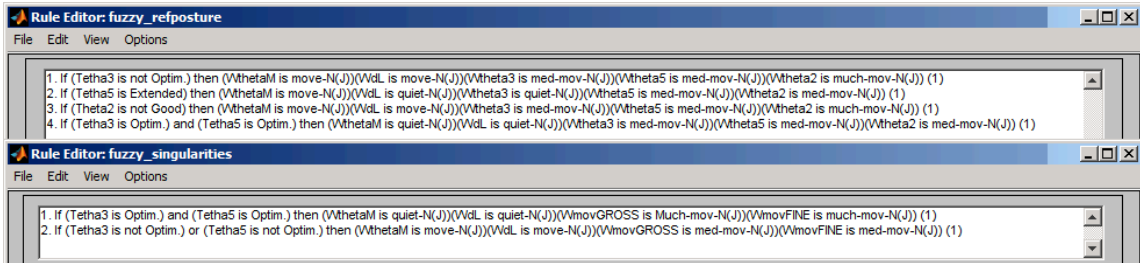


Fig. 9. A few rules (rule-base) were established for the reference posture criterion (up) and for the singularity avoidance criterion (down).

MATLAB’s FL-Toolbox, as a stand-alone application, allows defining a rule-base and testing it independently of the RRS, i.e., checking it prior to its integration with a RRS. For this purpose, several postures of the robot are supposed within the workspace and, afterwards the inference process, the expected major reactivity of each joint is checked with the weights assigned by the FIE. After this initial effort, a trained system is integrated with the RRS and the major advance is that the weights are dynamically adjusted along the path tracking.

5.2.3. Periodic revision by recursive IKP analysis.

Common milling tasks are made of a sequence of shorter paths. The breaks are used to revise and improve the posture for the following path. Thus, an additional trial is done considering a periodic posture revision to perform a better manage on the final k_F .

To practical effects, the non-linear analysis of the KR15/2 posture is done at a set of points by following [5], with the actual position on the additional joints known. Then, the additional joints θ_M and d_L are moved recursively to improve the k_F while maintaining the cutter pose (Fig. 10). Attending to precision in the manipulator, the major improvement is aimed with the table motion. After that, an additional smaller track motion is required. After the re-adjustment of the posture, the RRS keeps performing the postprocessing for the following period.

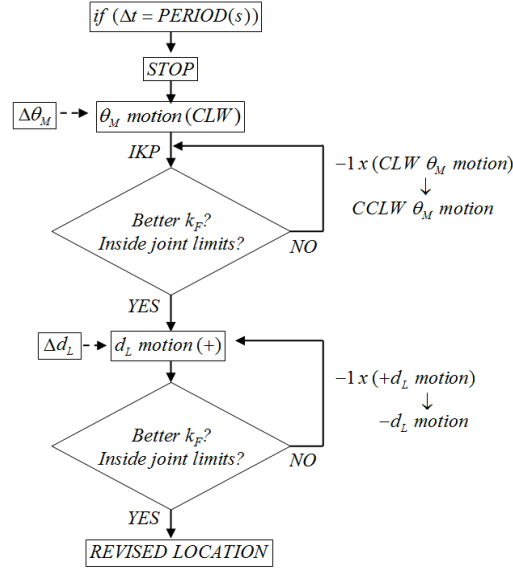


Fig. 10. Recursive posture revision for the studied workcell.

6. Testing and results

A challenging 5-axis milling was performed to test the postprocessor. It consists of a spherical shape to be milled through a continuous spiral path, aiming for the relocation with the additional joints meanwhile. As shown in Fig. 11, the orientation of the tool is required to be constantly the center of the sphere. This shape is supposed to be located into the manipulator's workspace on the rotary table, namely the base {B}. For the test, the center has been located at the coordinates $C = \{100, 200, 250\}$ (mm) and the radius of the sphere is $R = 150$ mm.

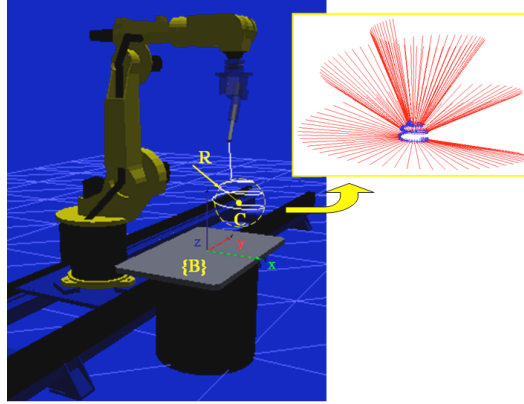


Fig. 11. Workcell at HOME posture (q_0) and representation of the test toolpath to be followed.

In order to perform the first trial, two constant diagonal weighting matrixes are assumed, namely $W_{jnt} = W_{cond} \equiv \text{diag}(0.01)$. A faster reaction can be expected in a second trial with $W_{jnt} = W_{cond} \equiv \text{diag}(0.1)$. Finally, a trial with the fuzzyfied weighting matrixes is performed. In all the trials, a threshold value of $\zeta = 0.5$ was considered to activate Eq. (12).

The resulting values of the inverse of k_F for each possible RRS deduced from Eq. (8) (VJM, TDM, and TDM with a projection on $\mathfrak{N}(J)$) were recorded for the two weighting matrixes during the path tracking. For the current implementation, the VJM (Fig. 12, left) maintains a more optimum and stable k_F during the path tracking than any feasible TDM implementation (Fig. 12, right). Thus, the VJM has a better performance for the current FIE, as the TDM seems to be more sensitive to weight adaptation, which provides more possibility on optimization. (A self-adaptation challenge of weighting matrix W is tackled in [24] in a different context, namely a 6-R decoupled manipulator without intrinsic redundancy.)

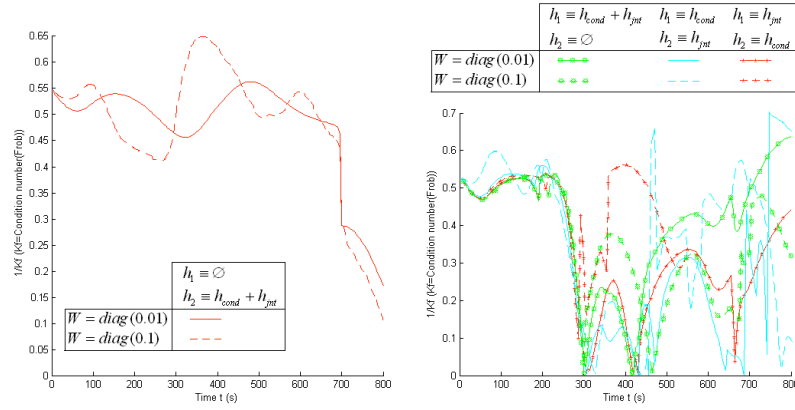


Fig. 12. Evaluation of the inverse of k_F for the VJM (left) and TDM (right).

The VJM procedure could apply better for the fuzzified tuning of the weighting vector in the present path tracking, taking into account the criterion of the proximity to the reference posture and the best value of k_F . Nevertheless, θ_3 reaches a mechanical limit when $W_{jnt} = W_{cond} \equiv \text{diag}(0.1)$ in the VJM (Fig. 13), so $W_{jnt} = W_{cond} \equiv \text{diag}(0.01)$ can be regarded as most convenient.

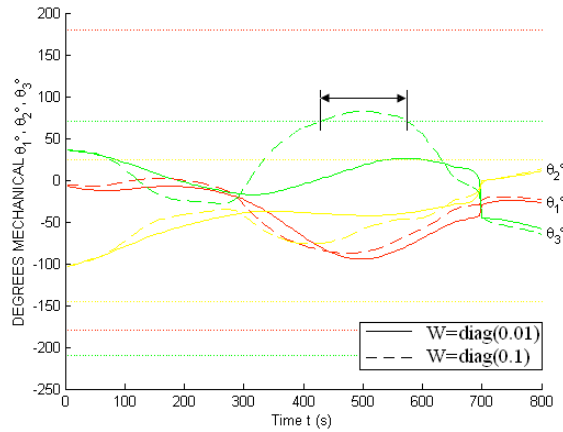


Fig. 13. Behaviour of the θ_1 , θ_2 and θ_3 joints during the spherical path tracking with the VJM.

Next, each weighting matrix (W_{jnt} , W_{cond}) were tuned with the two *fuzzy inference controllers* designed. Again, the VJM again offers the best and more robust k_F . Fig. 14 shows the conditioning achieved during the test:

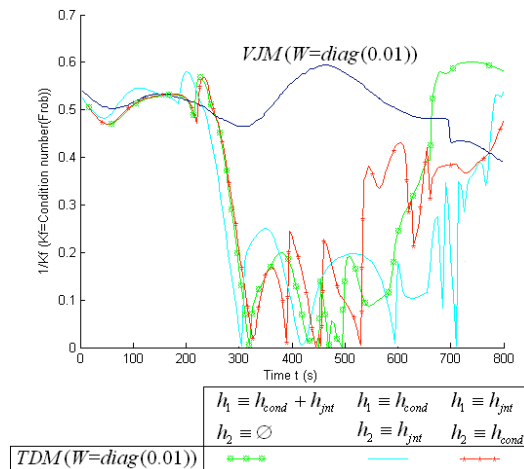


Fig. 14. Conditioning results for the different RRS studied with the adapted Fuzzy weighting vector of Table 2 for the combined performance criterion

Fig. 15 resumes the conditioning achieved with VJM in the previous trials. The use of the fuzzy adapted weighting vector (Fig. 15-b) improves the value of k_F , even more at the end of the test if compared with the one with constant weighting matrix $W = \text{diag}(0.01)$ (Fig. 15-a). To practical effects, a periodic revision can be implemented for real purposes (Fig. 15-c). In the test, with a period of 100s, the major k_F -average achieved.

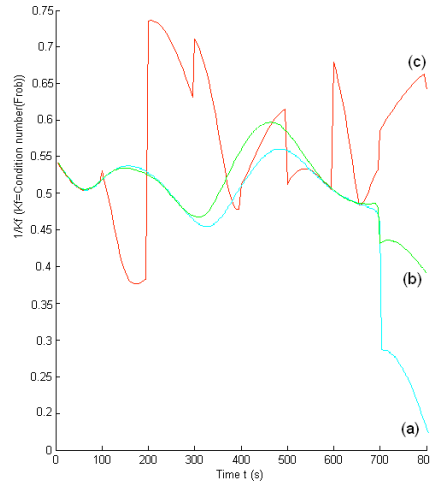


Fig. 15. Comparison of the conditioning achieved with the different VJM trials: (a) with constant weighting vector; (b) with fuzzy adapted weighting vector; (c) with fuzzy adapted weighting vector and a periodic IKP revision.

Moreover, while the worst conditioned posture has almost the same conditioning value ($k_F = 0.4$), the corresponding posture achieved with a continuous VJM (Fig. 16, left) has a most unpleasant presentation for the real robot than the one achieved with an IKP periodic revision (Fig. 16, right)

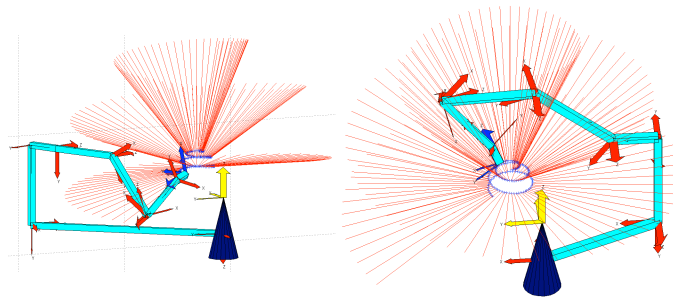


Fig. 16. Worst conditioned postures for the continuous VJM (left) and the VJM with an IKP periodic revision (right).

7. Applications

With the aim of validating the postprocessor designed, the workcell is devoted to machine a carving (namely, a Valencian *Falla*) with expanded polystyrene (EPS) (Fig. 17). The *Fallas* is a traditional celebration in praise of St. Joseph in Valencia, Spain (on March 19th). The term *Fallas* refers to both the celebration and the monuments created for the celebration. Prior to the celebration, much time is spent preparing the *ninots* (namely, *dolls*) that are assembled to compose a *Falla*. The figures are scanned and then treated by means of the NX-CAD module. The sample toolpath is shown at Fig. 17, in which the tool orientation is defined as normal to the surface along the tracking.

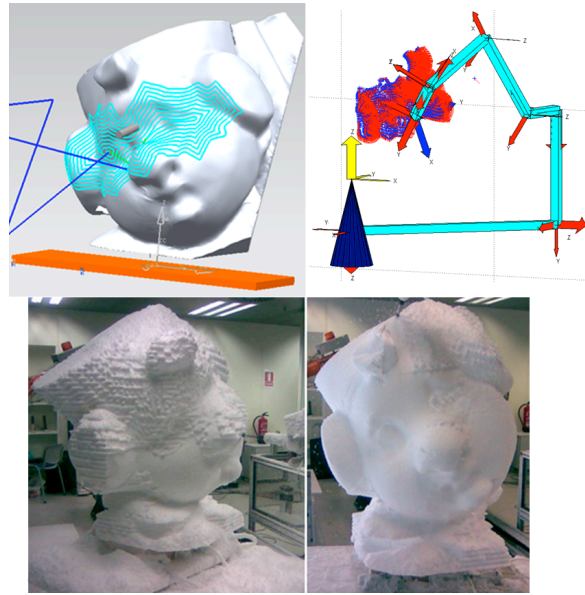


Fig. 17. A 5-axes toolpath is planned on the ninot to test the postprocessor.

The same task is attempted *without* and *with* the implemented postprocessor, starting from the same HOME posture (Fig. 18). Clearly, it can be appreciated much better performance within the second case, where the additional joints are moved to obtain a better performance while maintaining a well conditioned posture during the machining process.

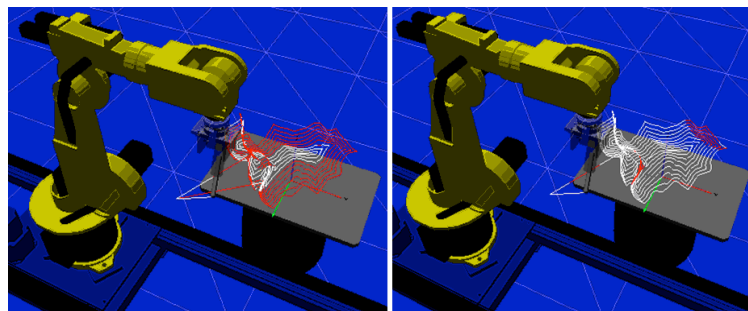


Fig. 18. The white toolpath, representing the area being reachable during the machining process, is enhanced with the programmed algorithm (left), if compared with a fixed table and track (right).

All joints are maintained between the allowable limits, while $1/k_F$ is kept between reasonable values, as shown in Fig. 19.

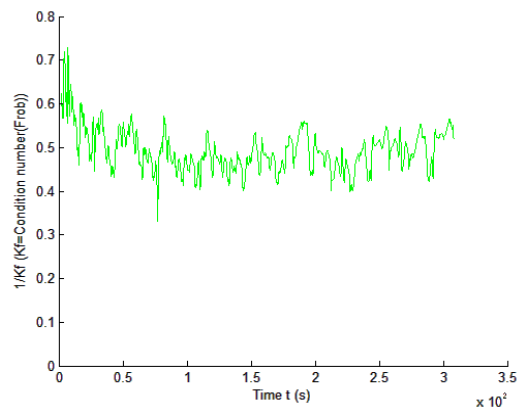


Fig. 19. Conditioning of the manipulator while the milling of the *Falla*.

8. Conclusions

The present paper has been focused on the postprocessing of the information generated by NX-CAM system towards the KUKA controller of a complex workcell devoted to milling tasks. As main contribution, after studying the capabilities of the set, a functional postprocessor is programmed inside the CAM system.

Different RRS on the joint-rate level have been tested and the best performance for the current CAM application have been achieved with the VJM. This RRS deals with the *functional redundancy* (the one due to the symmetry axis of the cutter) by considering an additional (virtual) joint in the referred symmetry axis.

In the design of secondary tasks, namely self-motions to avoid bad conditioned postures on the reachable workspace, the *condition number* of the *Homogeneous Jacobian* has given better results than the other performance indices. In fact, the better the posture is conditioned, the better the manipulator behaves regarding to motion and force transmission through the robotic chain. In this sense, it is advisable to implement a complementary force-feedback control to tune the cutter speed to limit the forces transmitted through the robotic chain. This issue remains as further work.

Also the expert *fuzzy systems* have demonstrated to be an alternative to actively manage some parameters related with experience, opposed to those previously fixed, even in those systems where it is difficult (or even impossible) to find a mathematical model of the process to control. In this study, a fuzzy inference engine improved the adjustment of the weights of the *performance vectors* for every robot situation, boosting the performance of the RRS and thus, of the postprocessor module. It is also worth mentioning the great influence on the result of the performance vector h .

The implemented postprocessor has been effectively tested with a graphical simulation of a demanding machining. Finally, it also has been successfully validated in a real prototyping of a *Valencian Falla*, done in EPS by means of 5-axes milling operations.

With the same guidelines, this postprocessor is expected to be easily applicable not only on any industrial robot, but also for different applications such as welding or painting labours.

Acknowledgements

This research is partially supported by the Technical University of Valencia (PAID-00-09), project PROMETEO 2009/063 of Generalitat Valenciana and research project DPI2009-14744-C03-01 of the Spanish Government.

REFERENCES

- [1] J. Andres, L. Gracia, J. Tornero; Calibration and control of a redundant robotic workcell for milling tasks, *Int. J. of Computer Integrated Manufacturing*, Vol. 24, No. 6, pp. 561–573, 2011.
- [2] R. V. Patel and F. Shadpey; *Control of Redundant Robot Manipulators: Theory and Experiments*; Springer 2005.
- [3] R. S. Hartenberg, J. Denavit; A kinematic notation for lower pair mechanisms based on matrices; *Journal of Applied Mechanics*, vol. 77, pp. 215–221, 1955.
- [4] J. Angeles; *Fundamentals of robotic mechanical systems: theory, methods and algorithms*; Springer, New York, 2003.
- [5] J. Andres, L. Gracia, J. Tornero; Inverse kinematics of a redundant manipulator for CAM integration. An industrial perspective of implementation; *Proceedings of the 2009 IEEE International Conference on Mechatronics*, Malaga, 2009.
- [6] D. E. Whitney; Resolved motion rate control of manipulators and human prostheses; *IEEE Trans. Man-Machine Syst.*, Vol. 10, No. 2, pp. 47–53, 1969.
- [7] L. Sciavicco, B. Siciliano; *Modelling and Control of Robot Manipulators*; Springer, London, 2000, pp. 79–87.
- [8] D. E. Whitney; The mathematics of coordinated control of prosthetic arms and manipulator; *ASME J. Dynamics Systems, Measurement and Control*, Vol. 94, No. 4, pp. 303–309, 1972.
- [9] S. Chiaverini, G. Oriolo, I. D. Walker, B. Siciliano, O. Khatib; *Kinematically redundant manipulators*; *Springer Handbook of Robotics*, Springer-Verlag, pp. 253, 2008.
- [10] M. Honegger, A. Codourey; Redundancy resolution of a cartesian space operated heavy industrial manipulator; *IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2094–2098, 1998.

- [11] L. Huo, L. Baron; Kinematic inversion of functionally-redundant serial manipulators: application to arc-welding; Transactions of the Canadian Society for Mechanical Engineering, 2005, Canada.
- [12] L. Baron; A joint-limits avoidance strategy for arc-welding robots; International Conference on Integrated Design and Manufacturing in Mechanical Engineering, Montreal, Canada, May 2000.
- [13] L. Huo, L. Baron; The joint-limits and singularity avoidance in robotic welding; Industrial Robot: An International Journal; Vol. 35, No. 5, pp 456–464, 2008.
- [14] Z. Kemeny; Mapping, detection and handling of singularities for kinematically redundant serial manipulators; Periodica Polytechnica Ser. El. Eng. Vol. 46, No. 1, pp. 29–45, 2002.
- [15] A. Liégeois; Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms; IEEE Trans. Syst., Man, Cybern., vol. SMC-7, pp. 245–250, 1977.
- [16] T. Yoshikawa; Analysis and control of robot manipulators with redundancy, Robotics Research: The First International Symposium, pp. 735–747, 1984.
- [17] T. Yoshikawa; Manipulability of Robotic Mechanisms, Int. J. of Robotics Research, Vol.4, No.2, pp. 3–9, 1985.
- [18] L. Gracia, J. Andres, and J. Tornero; Trajectory Tracking with a 6R Serial Industrial Robot with Ordinary and Non-ordinary Singularities; International Journal of Control, Automation, and Systems Vol. 7, No. 1, pp. 85–96, 2009
- [19] M. D. J. Hayes, M. L. Husty, P. J. Zsombor-Murray; Singular Configurations of Wrist-Partitioned 6R Serial Robots: a Geometric Perspective for Users; CSME Transactions, Vol. 26, No. 1, 41–55, 2002.
- [20] J. Angeles, C. S. López-Cajún; Kinematic Isotropy and the Conditioning Index of Serial Robotic Manipulators; The International Journal of Robotics Research; Vol. 11, No. 6, pp. 560–571, 1992.
- [21] J. G. Garcia, A. Robertsson, J. G. Ortega, R. Johansson; Self-calibrated robotic manipulator force observer; Robotics and Computer-Integrated Manufacturing, Vol 25, No. 2, pp. 366–378, 2009.
- [22] A. Khan Waseem, J. Angeles; The Kinetostatic Optimization of Robotic Manipulators: The Inverse and the Direct Problems; Journal of Mechanical Design; Vol. 128, No. 1, pp. 168–178, 2006.
- [23] E. Ranjbaran, J. Angeles, M. A. Gonzalez-Palacios, R. V. Patel; The mechanical design of a seven-axes manipulator with kinematic isotropy; J. of Intelligent and Robotic Systems, Vol. 14, No. 1, pp. 21–41, 1995.
- [24] L. Huo, L. Baron; The self-adaptation of weights for joint-limits and singularity avoidances of functionally redundant robotic-task, Robotics and Computer-Integrated Manufacturing, Vol. 27, No. 2, pp. 367–376, 2011.
- [25] Siemens Corp, 2009 ; NX Documentation ; In: {SUGII_base_dir}\UGDOC.
- [26] J. I. Maza, A. Ollero, Herramienta MATLAB-Simulink para la simulación y el control de robots manipuladores y móviles, Actas de las XXI Jornadas de Automática, Sevilla, 2000.
- [27] Roger Jang J. S., Gulley, N.; Fuzzy Logic Toolbox: User’s Guide; Revised for Version 2.2.7 (Release 2008), The MathWorks, Inc. 2008.
- [28] “CAD-CAM off line programming for industrial robots”, ROBOmove on line help v. 2.0, Qdesign S.r.l., 2007.
- [29] J. Andrés, L. Gracia, H. Marti, J. Tornero; Toolpath postprocessing for three axes milling in redundant robotic workcells by means of fuzzy integration in a CAM platform, Proceedings of the 2009 IEEE International Conference on Mechatronics, Malaga, 2009.