INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Automated Milling Path Tracking and CAM-ROB Integration for Industrial Redundant Manipulators

Regular Paper

Luis Gracia[1,*], Javier Andres[2] and Carlos Gracia[3]

1 Instituto IDF, Universitat Politècnica de València, Camino de Vera s/n, Valencia, Spain
2 Department of Mechanical Engineering and Construction, Universitat Jaume I, Avda Vicent Sos Baynat s/n, Castellón, Spain
3 Departamento de Organización de Empresas, Universitat Politècnica de València, Camino de Vera s/n, Valencia, Spain
* Corresponding author E-mail: luigraca@isa.upv.es

Received 19 May 2012; Accepted 27 Jun 2012

DOI: 10.5772/51101

Abstract The present paper explores the industrial capabilities of a CAM-ROB system implementation based on a commercial CAD/CAM system (NX™) for an industrial robotic workcell of eight joints, committed to the rapid prototyping of 3D CAD-defined models. The workcell consists of a KUKA™ KR15/2 manipulator assembled on a linear track and synchronized with a rotary table. A redundancy resolution scheme is developed to deal with the redundancies due to the additional joints of the robot, plus the one from the symmetry axis of the milling tool. During the path tracking, the use of these redundancies is optimized by adjusting two performance criterion vectors related to singularity avoidance and maintenance of a preferred reference posture, as secondary tasks to be done. In addition, two suitable fuzzy inference engines adjust the weight of each joint in these tasks. The developed system is validated in a real prototyping of a carving.

Keywords Postprocessor, redundancy, robot manipulator

1. Introduction

Redundant manipulators have extra degrees of freedom, resulting in multiple solutions of the joint angles for a given end-effector (EE) task. They are skilful and versatile, with a large working volume that is of interest for applications such as painting or milling. *Rapid prototyping* done by milling soft materials is of increasing interest to obtain physical replicas of CAD (*Computer Aided Design*) defined models, thus supporting the *industrial design* process.

In this context, a CAD/CAM/ROB integrated manufacturing system reduces the time invested in successive verifications, adjustments and translations in the machining process. Commonly, CAD/CAM (*Computer Aided Manufacturing*) systems only integrate and fully associate the labours of design (CAD) and manufacturing (CAM). As a final step in these systems, the CAM module generates a milling path 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, these systems only specify five parameters to carry out the milling task: three location and two orientation coordinates of the *tool centre point* (TCP) of the milling tool referred to a Cartesian coordinate system (*base*, {B}), see Figure 1. As the desired finish conditions of the workpiece are set in the CAM system, the generated data are compulsory and independent from the machine tool that will manufacture the workpiece.

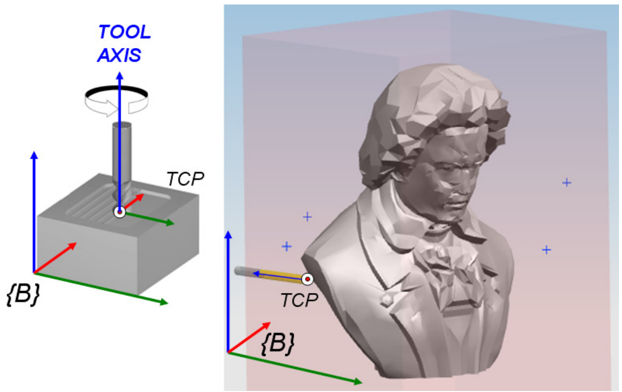

Figure 1. TCP on the tool tip. The milling toolpath is defined with three location and two orientation coordinates of the TCP

Subsequently, that data has to be *postprocessed* (i.e., adapted) from the CAM system to the production system that is going to be used. In this case, the *milling tool* attached at the EE of the robot is expected to track this path and the choice of the optimal posture of the manipulator, amongst all possible choices, has become the main interest in studying such postprocessing. Keeping the joints within their limits, singularity avoidance and torque optimization are some utilizations of redundancy as a second subtask, while the proper EE motion is the first.

The *Design and Manufacturing Institute* (IDF) of the *Technical University of Valencia* has configured a sculpturing redundant workcell for rapid prototyping from the NX™ (Siemens) system, see Figure 2. In it, an industrial KUKA™ KR15/2 arm with six revolute (R) joints is mounted on a linear track (d_L , being an additional prismatic joint, P) and it works over a synchronized rotary table (θ_M , being an additional R-joint) on which the initial blank of material is set. Because the 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 an 8-dimension *Joint Space*. Therefore, those two additional joints plus the irrelevant rotation around the symmetry tool axis make this workcell *redundant*, with a degree of *kinematic redundancy* of three [1].

In this work, efficient implementation of a CAM-Robotics postprocessor is developed to assist the automatic off-line generation of the robot milling instructions.

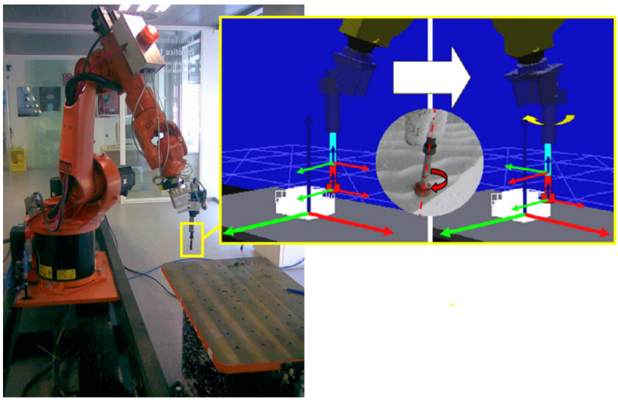

Figure 2. KUKA™ industrial workcell, with detail of the symmetry axis of the milling tool. leading to a virtual joint supposition

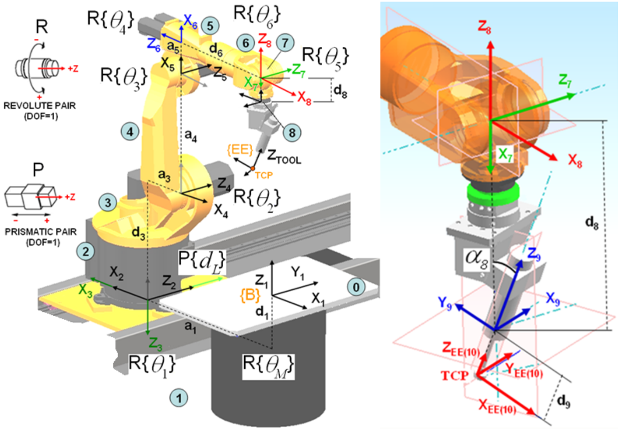

Figure 3. Frame assignment for the Denavit-Hartenberg representation of the RP-6R workcell

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

Table 1. Denavit-Hartenberg parameters of the workcell

2. Characterization of the industrial workcell

Before dealing with any physical analysis of the workcell, it is convenient to understand that the industrial KUKA™ KRC2 controller processes an own programming language: the KRL (KUKA™ Robot Language) [2]. This language incorporates a data structure to control the robot positioning for continuous path tracking (for internally controlled linear and circular movements of the TCP). In the case of having external axis, the structure becomes

$$\{\text{E6POS: X, Y, Z, A, B, C, E1, E2}\} \quad (1)$$

in which the TCP positioning and orientation coordinates (X, Y, Z, A, B, C) are referred to a work Cartesian coordinate system defined in {B} (also named as *Operational Space,* on the table, see Figure 3). Moreover, the external additional axis, E1 and E2, set the linear track and rotary table joint values respectively. Thus, the information postprocessed from the NX^{TM}–CAM system must bring the data to the robot with the following templates for linear and circular paths, respectively:

$$\text{LIN \{destination\_point \}} \quad (2)$$

$$\text{CIRC \{ auxiliary\_point \}\{ destination\_point \} CA} \quad (3)$$

where each point has a *E6POS* structure (1) and CA means the Circle Angle.

A *kinematic model* of a manipulator is the mathematical description required to satisfactorily control 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* to the *Operational Space*, i.e., determining the pose of the EE for a given manipulator posture. On the contrary, the *Inverse Kinematic Problem* (IKP) maps the data from the *Operational* to the *Joint Space.* Both problems can be considered at both *displacement* and *joint rate* levels.

As in many industrial implementations, with the provided templates it is not possible to control any of the joint speeds and to synchronize the full manipulator in order to get a desired TCP speed. In short, bringing the toolpath to the *Joint Space* (from the *Operational Space* in which it is generated by the CAM system) is not efficient for direct application as it may include errors on the finished surface. Instead, it is interesting for the analysis of the posture, leaving the calibration aside [3]. The conclusions of any kinematic analysis are done in parallel until its practical application: the setting of the convenient external joint values. Then, with these values the robot controller solves the posture of the rest of the previously calibrated 6R manipulator.

2.1 Kinematic model of the industrial workcell at the displacement level

The DKP at the displacement level is straightforward: a *posture* (*Joint Space*) represents a unique *pose* of the EE referred to {B}. The *standard* Denavit-Hartenberg (DH) model [4] represents the EE pose as a 4×4 homogeneous transformation matrix that results from the operation of some descriptive parameters of the workcell (a_i, α_i, d_i, θ_i) resumed in Figure 3 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 at the *displacement* level is more challenging given that an infinite number of solutions may exist in the case of a redundant manipulator. For a 6R manipulator, algebraic methods can be applied [5], but in this case it is possible to apply a closed geometric method [6]. With the current θ_M and d_L joint values known, it is possible to establish ${}^{R}p_W$ = *[*W_X, W_Y, W_Z*]* between the manipulator's base *{R}* and the wrist, W (see Figure 4). The *gross positioning* (joints θ_1, θ_2 and θ_3) can be programmed with the following trigonometric relations:

$$\theta_1 = -\operatorname{atan2}(W_Y, W_X) \quad (rad) \quad (4)$$

$$\theta_2 = -(\alpha + \varepsilon) \quad (rad) \quad (5)$$

$$\theta_3 = \pi - \gamma + \varphi \quad (rad) \quad (6)$$

with φ=*atan*(155/600).

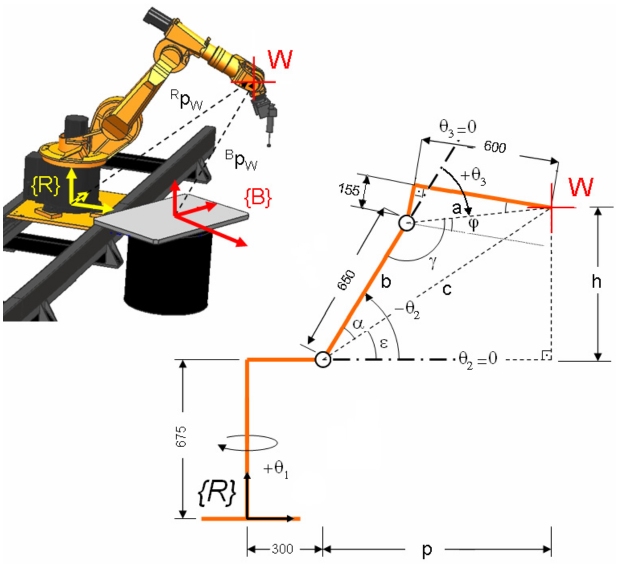

Figure 4. Geometric solution of the R joints in charge of the *gross positioning*

It is easy to demonstrate that the fine positioning (joints θ_4, θ_5 and θ_6, in rads.) can be obtained by a sequence of ZYZ Euler's rotations:

$$\theta_4 = \text{-atan2}({}^{W}A_6(2,3), {}^{W}A_6(1,3)) \quad (7)$$

$$\theta_5 = \text{atan2}((({}^{W}A_6(3,1))^2 + ({}^{W}A_6(3,2))^2)^{1/2}, {}^{W}A_6(3,3)) \quad (8)$$

$$\theta_6 = \text{-atan2}({}^{W}A_6(3,2), -{}^{W}A_6(3,1)) \quad (9)$$

where ${}^{W}A_6$ is the homogenous transformation matrix from *{R'''W}* to the robot flange. *{R'''W}*, placed at *{W}*, is achieved from *{R}* by means of two rotations $\rho_1 = -\theta_1$ and $\rho_2 = \pi/2 + \theta_2 + \theta_3$ (see Figure 5).

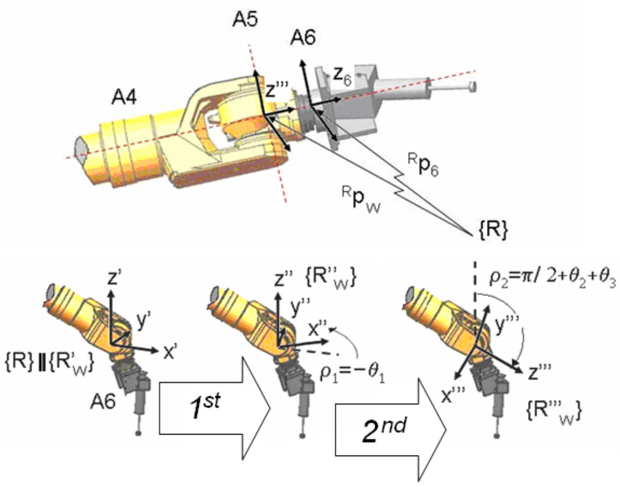

Figure 5. Achievement of {R'''W} from {R} by means of two rotations

2.2 Kinematic model of the industrial workcell at the joint rate level

The *forward kinematics* (DKP) between the joint velocity ($\dot{q}$) and the EE velocity (*t*) is represented by the *linear* algebraic equation

$$t = J \cdot \dot{q} \tag{10}$$

where the coefficient matrix (*geometric* Jacobian, J) is a non-linear function of joint angles [7]. This matrix maps the 8-dimensional joint rate vector $\dot{q} = [\dot{\theta}_M, \dot{d}_L, \dot{\theta}_1, ..., \dot{\theta}_6]^T$ into the *twist* vector $t = [\omega \ \ v]^T$, with $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ and $v = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ being the angular and linear velocities of the milling tool TCP's frame relative to {B}, see Figure 1.

The IKP is most useful for path tracking, i.e., given the desired twist of the milling tool the aim is to obtain the joint velocities, with *J* known. A difficulty arises in the case of *redundant* robots having a non-square *J* (i.e., more columns than rows). The solution $\dot{q}$ 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* $J^{\dagger} \equiv J^T \left(JJ^T\right)^{-1}$. Nevertheless, an additional *homogeneous addend* may perform *a secondary task* at the cost of giving up the minimum-norm solution:

$$\dot{q} = J^{\dagger} \cdot t \quad + \quad (I_{nxn} - J^{\dagger} J)h \tag{11}$$

To capture the additional redundancy due to the symmetry axis of the tool, in addition to those due to the additional joints (θ_M, d_L), the dimension of the *Joint Space* can be augmented by adding a virtual joint on the symmetry axis of the tool (see Figure 2), in order to obtain an extra DOF of redundancy (*virtual joint method* [8]). Therefore, (11) can be rewritten as

$$\dot{q}_v = \tilde{J}_v^{\dagger} \cdot t + (I_{(n+1)x(n+1)} - J_v^{\dagger} J_v) \cdot h \tag{12}$$

where J_v is an *augmented* Jacobian matrix by a virtual joint-rate $\dot{\theta}_7$, namely

$$\dot{q}_v = \begin{bmatrix} [\dot{\theta}_M, \dot{d}_L, \dot{\theta}_1, ..., \dot{\theta}_6]^T \\ \dot{\theta}_7 \end{bmatrix} \tag{13}$$

By solving (12), the manipulator tracks the desired target positions as *primary task*. The additional *homogeneous addend* is an arbitrary *performance criterion vector* (*h*) projected into the *Null Space* of J, $\aleph(J)$. To practical effects it can be considered as a *virtual force* re-posturing the manipulator away from a critical area in the *Joint Space* for the same EE's motion $\dot{q}$, as *secondary task*. The most widespread method to select *h* is the *gradient projection method* [9]. It takes the minimization of a posture-dependent scalar (*performance criterion index*, *p*) by means of its gradient vector, namely

$$h = -k \cdot \nabla p \ ; \ \text{with } \nabla p = \left[\frac{\partial p(q)}{\partial q_1}, \frac{\partial p(q)}{\partial q_2}, ..., \frac{\partial p(q)}{\partial q_n} \right]^T \tag{14}$$

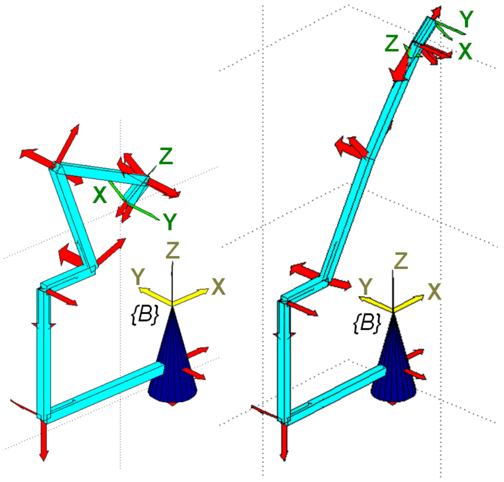

Figure 6. Best conditioned posture (left) and mechanical mid-joint posture (right) for the 6R KUKA™ KR 15/2 manipulator

2.3 Performance criterion for singularity avoidance and reference posture preservation

Two performance criterion vectors are combined simultaneously in order to maintain the manipulator as far as possible of *bad conditioned* postures (h_{cond}) and as close as possible to a *reference* posture (h_{jnt}):

$$h = -\nabla p = -\nabla\left(p_{cond} + p_{jnt}\right) = h_{cond} + h_{jnt} \tag{15}$$

For the first case, the *condition number of J*, $k_F(J)$, is taken into account to achieve a definitive *performance criterion vector for singularity avoidance* (h_{cond}). In [7] k_F is used as the rate at which $\dot{q}$ will change with respect to a small change in t at Eq. (10). This scalar also quantifies how well the system behaves with regard to force and motion transmission along the manipulator chain, so it also promotes the better efficiency of the milling tool at work. It also gives an upper bound for the roundoff-error amplification in the solution. An expression valid for any *6xn* Jacobian matrix, with $n \geq 6$, is:

$$k_F(H) = \frac{1}{6}\sqrt{tr(HH^T)\cdot tr[(HH^T)^{-1}]}\ ;\ 1 \leq k_F < \infty \tag{16}$$

H is the *Homogeneous Jacobian*. In it, the entries of *J* with units of length have been divided by *characteristic length* (*L*) of the manipulator [7]. Therefore, it is desirable to work at any manipulator posture minimizing *kF* (note that it can be more practical to evaluate the inverse, as $1 \geq {}^{1}\!/_{k_F} > 0$).

In the particular case of the KUKA™ KR 15/2, without the additional joints (θ_M, d_L) which amount to a rigid-body motion of the overall 6R chain, *L*=350.6 mm and the best-conditioned posture is shown in Figure 6 (*kF*=1.247).

Therefore, a performance criterion *index* can be calculated as

$$p_{cond} = \frac{k_F}{2}(q - q_{Ts})^T W_{cond}(q - q_{Ts}) \tag{17}$$

This index is to be activated when the k_F value passes over a preset *threshold value* (0.5 in the current study). At that instant, the corresponding configuration (q_{Ts}) is recorded to evaluate the distance to the actual posture in the *Joint Space*:

$$h_{cond} = -\nabla p_{cond} = -W_{cond}\cdot k_F \cdot (q - q_{Ts}) \tag{18}$$

For the second criterion (h_{jnt}), a certain constant reference arm posture (q^{ref}) may be desirable for avoiding collision with obstacles or reach the mechanical joint-limits [8]. A performance criterion *index* could be written as

$$p_{jnt} = \frac{1}{2}(q - q^{\mathrm{ref}})^T W_{jnt}(q - q^{\mathrm{ref}}) \tag{19}$$

thus, a *performance criterion vector for reference posture maintenance* (h_{jnt}) could be written as

$$h_{jnt} = -\nabla p_{jnt} = -W_{jnt}\left(q - q^{\mathrm{ref}}\right) \tag{20}$$

The *mid-joint posture* of the industrial KR 15/2 is not appropriate as a reference posture for milling tasks, and neither is the *best-conditioned posture*, which is near some mechanical joint limits (Figure 6). For this work, the HOME posture $q_0 = [+\pi,\ 0,\ +\pi,\ -\pi/2,\ 0,\ 0,\ +\pi/2,\ 0]^T$ rad (see Figure 3) is taken as the $\mathrm{q}^{\mathrm{ref}}$.

At (18) and (20), W_{cond} and W_{jnt} are two weighting diagonal matrices to be implemented. By subjectively tuning both matrices, one can adjust the relative importance amongst the joints in each sub-task by assigning higher weights to the most reactive joints (traditionally based on trial and error). For milling operations where the tool *pose* (and hence the robot *posture*) changes continually, it may be advantageous to assign appropriate weights at each configuration in a reasonable time. In the next section, this paper presents a *fuzzy inference controller* that captures the expert knowledge of a technician for a *computational cost-effective* assignment of variable weights at the performance vectors (*h*).

3. Implementation of the path tracking postprocessing

The NX™-CAM module makes possible the planning of milling tasks, but it can also interact with two program codes in TCL (*Tool Command Language*) that manipulate the path data (*Event Handler*) and give a convenient format to the output (*Definition File*) [10].

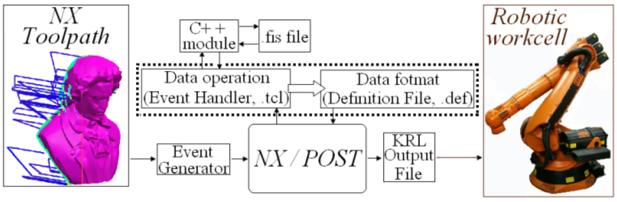

Figure 7. The Definition File and the Event Handler subroutines postprocess the toolpath from NX™ to KRL

The trajectory data, generated with NX™, is kept as T_{CAM} as a discrete set of close-enough poses. 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 tool pose (see Figure 1). All the treatment of the data is programmed in MATLAB™ with the assistance of the Fuzzy Logic toolbox [11].

Starting from the HOME posture and with the DH models of both the KR15/2 manipulator (*DH-KR15/2*) and the complete workcell (*DH-Workcell*) at hand, the loop leading from an initial current tool pose to the following one is programmed as follows [12]:

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) $J_{\text{Workcell}} \Leftarrow \mathrm{DK}(q, \text{DH-Workcell})$
    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) $\Delta q \Leftarrow$ Virtual Joint Method (VJM)
    10) $q \Leftarrow q + \Delta q$
  endwhile
endfor
```

where 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 lonely KR15/2 manipulator. Function $vect(\Delta Q)$ represents the axial vector of a 3x3 incremental rotation matrix ΔQ [7] and Δp is an incremental displacement between poses.

From this Algorithm 1, we can get the profitable values for the additional joints (linear track and rotary table) which were initially required in the template (1). This Algorithm 1, as complementary analysis of the data, is programmed in C++ and interacts with a *fuzzy inference file* (*.fis*, see Figure 7).

3.1 Adapted Fuzzy weighting vector

MATLAB™'s Fuzzy Logic toolbox allows defining and testing an *"if-then"* rule-base for weight assignment towards Algorithm 1, by means of a stand-alone *.fis* file (see Figure 7). For this, several postures of the robot are supposed within the workspace and the expected major reactivity of each joint is checked with the weights assigned by the *fuzzy inference engine* (FIE), which should be in accordance with *expert decisions*. In this case, two FIEs work in the active adjustment of variable weights at both performance criterion vectors, h_{jnt} and h_{cond} along the path tracking.

In this labour, expert knowledge is of major importance: 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 [13] [14]. Joints θ_3, θ_5 and θ_M, d_L have a direct implication in order to avoid these postures. Nevertheless, it seems to be convenient to work near the *gross positioning* ($\theta_1, \theta_2, \theta_3$) of the reference posture (q^{ref}) while the *fine orientation* ($\theta_4, \theta_5, \theta_6$) is being done with other relevance or weight. Thus, each FIE gives the weights associated to the stated joints, leaving a default value to those not taken into account, see Table 2.

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

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

3.2 Periodic revision by recursive IKP analysis

Real milling tasks are made of a sequence of concatenated paths in different regions of the workpiece. The breaks are helpful to travel between two regions and it gives a real opportunity to radically improve the posture for the following operation. Thus, an additional periodic posture revision is done to better manage the final k_F.

For practical effects, at a point between regions, the IKP is solved 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 (Figure 8). To avoid chain vibrations on the KUKA™ KR15/2 manipulator, the first and major improvement is aimed with the table motion. After that, an additional smaller track motion is required. After this revision of the posture, the RRS keeps performing the postprocessing for the current region.

4. Experimental results

With the aim of validating the designed postprocessor, the workcell is devoted to machine a carving of expanded polystyrene (EPS), see Figure 9. The figures are scanned and then treated with the NX™-CAD module. Two sample toolpaths of two perpendicular regions are shown in Figure 9.

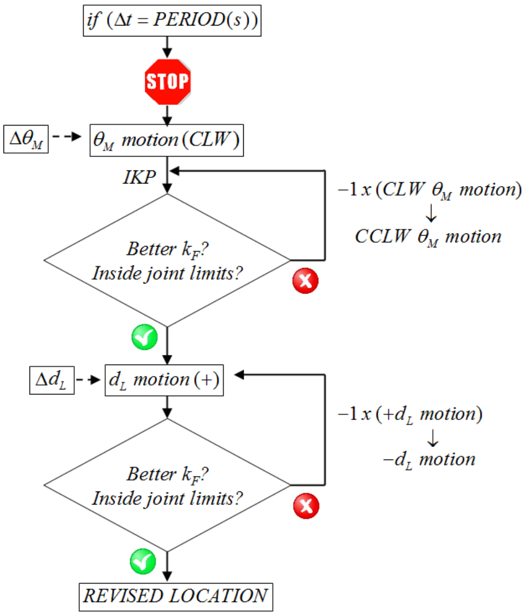

Figure 8. Recursive posture revision for the studied workcell

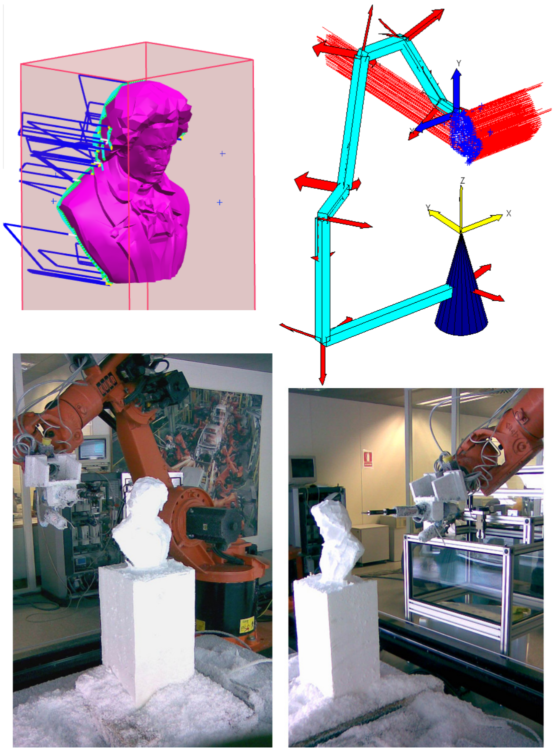

Figure 9. A toolpath is planned on two regions of the carving with a change on the orientation of the tool of 90 degrees. The orientation of the symmetry axis of the tool is depicted in red

After starting in the HOME position, as it can be appreciated $1/k_F$ is maintained at reasonable values while all joints are between the allowable limits. Moreover, in the void motion of the tool between regions, the posture is revised and the additional joints are moved to obtain an improved performance, see Figure 10 and Figure 11.

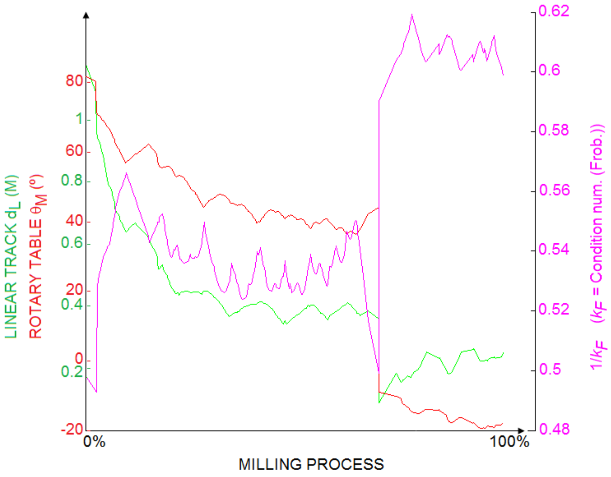

Figure 10. Conditioning of the manipulator while milling the carving

5. Conclusions

This paper has focused on the postprocessing of the information generated by the NX™-CAM system towards an industrial KUKA™ manipulator with additional joints and devoted to milling tasks. The main contribution, after studying the capabilities of the set, is the programming of a functional postprocessor. It deals with the virtual joint method to solve the redundancies, not only due to the additional linear track and rotary table, but also the one due to the symmetry axis of the milling tool.

From this analysis we can obtain the profitable values for the additional joints (linear track and rotary table) which are required by the industrial KUKA™ controller. These values have been traditionally assigned by experience or trial-and-error.

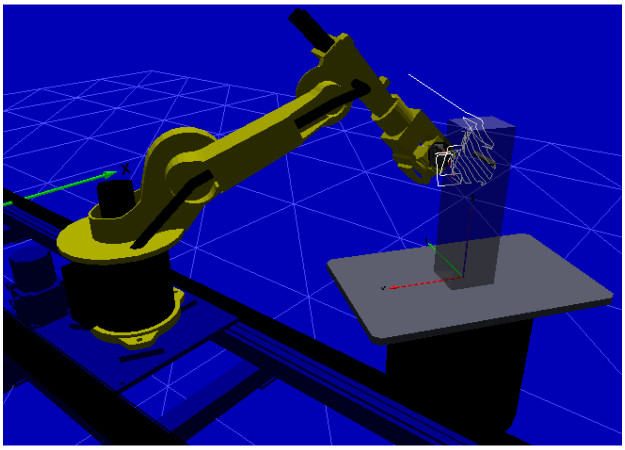

Figure 11. The white toolpath represents the reachable area during the milling process, which has been enhanced with the proposed postprocessor

For the generation of the self-motions to avoid bad conditioned postures on the reachable workspace, the *condition number* of the *Homogeneous Jacobian* has demonstrated its effectiveness. The better the conditioned posture, the better the manipulator behaves regarding motion and force transmission along the chain. It also means efficiency in the actuation of the tool over the workpiece and complementary to that promotes the future use of a force/torque sensor. This could tune the speed of the tool to limit the forces passed to the manipulator's chain.

The implementation of *fuzzy inference engines* has also demonstrated to be an alternative to manage those parameters somehow related with experience. In this case, two engines adjusted the weights of the *performance vectors* for every robot situation in order to improve the robot performance.

The developed postprocessor has been effectively tested in a real prototyping of a carving by means of a (3+2)-axes milling operation, i.e., two regions machined with two different tool orientations.

With the same guidelines of this work, the proposed approach is expected to be useful not only to other industrial robots, but also for different applications such as painting or welding tasks.

6. Acknowledgments

This research is partially supported by research project DPI2009-14744-C03-01 of the Spanish Government, project PROMETEO 2009/063 of Generalitat Valenciana, and research projects PAID-05-11-2640 and PAID-00-12-SP20120159 of the Universitat Politècnica de València.

7. References

[1] R. V. Patel, F. Shadpey (2005) Control of Redundant Robot Manipulators: Theory and Experiments. Springer, New York.

[2] KUKA Roboter GMBH (2005) KUKA System Software (KSS): Expert Programming (*KRC2 / KRC3*), Release 5.2, KUKA Corp.

[3] J. Andres, L. Gracia, J. Tornero (2011) Calibration and control of a redundant robotic workcell for milling tasks. Int. J. of Computer Integrated Manufacturing, Vol. 24, No. 6, pp. 561–573.

[4] R. S. Hartenberg, J. Denavit (1955) A kinematic notation for lower pair mechanisms based on matrices. Journal of Applied Mechanics, Vol. 77, pp. 215–221.

[5] Y. Luo, W. Yi, Q. Liu (2012) Inverse Displacement Analysis of a General 6R Manipulator Based on the Hyper-chaotic Least Square Method. International Journal of Advanced Robotic Systems, Vol. 9, pp. 1-6.

[6] J. Andres, L. Gracia, J. Tornero (2009) Inverse kinematics of a redundant manipulator for CAM integration. An industrial perspective of implementation. Proceedings of the 2009 IEEE International Conference on Mechatronics, Malaga.

[7] J. Angeles (2003) Fundamentals of robotic mechanical systems: theory, methods and algorithms. Springer, New York.

[8] L. Huo, L. Baron (2008) The joint-limits and singularity avoidance in robotic welding. Industrial Robot: An International Journal, Vol. 35, No. 5 pp 456–464.

[9] A. Liégeois (1977) Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-7, pp. 245-250.

[10] Siemens Corp (2009) NX Documentation. In: {$UGII_base_dir}\UGDOC

[11] The MathWorks, Inc. (2008) Fuzzy Logic Toolbox: User's Guide; Revised for Version 2.2.7 (Release 2008),.

[12] J. Andres, L. Gracia, J. Tornero (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 Vol. 28, No. 2, pp. 265–274.

[13] L. Gracia, J. Andres, J. Tornero (2009) Trajectory tracking with a 6R serial industrial robot with ordinary and non-ordinary singularities. Int. Journal of Control, Automation and Systems, Vol. 7, No. 1, pp. 85–96.

[14] H. Zhou, Y. Cao, B. Li, M. Wu, J. Yu, H. Chen (2012) Position-singularity analysis of a class of the 3/6-Gough-Stewart manipulators based on singularity-equivalent-mechanism. International Journal of Advanced Robotic Systems, Vol. 9, pp. 1-9.