Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

On the use of functional redundancy in industrial robotic manipulators for optimal spray painting ⋆

Andrea Maria Zanchettin * **Paolo Rocco** *

* *Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza L. Da Vinci 32, 20133, Milano, Italy*
{zanchettin, rocco}@elet.polimi.it

Abstract: Robotic spray painting is a well-established industrial application usually performed with a 6-axes industrial robot. This task leaves an extra degree of freedom which can be exploited in order to achieve any additional goal. Unfortunately, typical proprietary industrial robotic controllers do not allow the programmer to modify the inverse kinematics algorithm, and thus to solve task redundancy following any specified criterion. In this paper a method to enforce an arbitrary redundancy resolution criterion on top of an industrial robot controller is discussed and applied to the execution of a spray painting task. The extra degree of freedom is used to maximize the manipulability index. Simulations and experimental results achieved on the ABB IRB 140 industrial robot are presented.

Keywords: Inverse kinematic problem, Redundant manipulators, Industrial robots, Robot programming

1. INTRODUCTION

Automated robotic spray painting is a well-established application, especially in automotive and furniture manufacturing. Many commercial software packages are available to optimize off-line this task. In fact, the tool trajectory planning is a complex problem, involving several engineering techniques, ranging from non-linear programming to 3D surface modeling.
The quality of the paint coating highly depends on the geometric properties of the surface and on the cost function adopted to compute the gun trajectory. Commercial optimization tools aim at planning the tool trajectory to accomplish the task with the minimum cycle-time or with the minimum number of trajectory turns, subject to the desired coating thickness, but rarely take into account the robot properties. The decision variables are the route taken by the gun on the surface, the standoff distance and the path velocity.
Several research studies have been already proposed to further optimize the painting task. In From and Gravdahl (2007) it was shown that by allowing a small orientation error (< 20 %), the speed and the quality of the task can be improved. In Kim and Sarma (2003) the actuator speed limits are taken into account, while the work proposed in Suh et al. (1991) takes into account the robot dynamics to find an optimized trajectory.
Many numerical issues arise, however, in computing the solution of the optimization problem. In Xia et al. (2009) a surface segmentation is proposed to reduce the computational burden, while in Fa-zhong et al. (2009) the optimization problem is solved using Genetic Algorithms (GA).
A simple model of the painting task that is usually taken into account for optimization is depicted in Fig. 1, see Antonio et al. (1997), Arikan and Balkann (2000), Wei and Dean (2009). The typical profile of the paint deposition rate $f(r,h)$ is also shown, where R is spray cone radius and h is the TCP standoff distance.

⋆ The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No 230902 - ROSETTA.

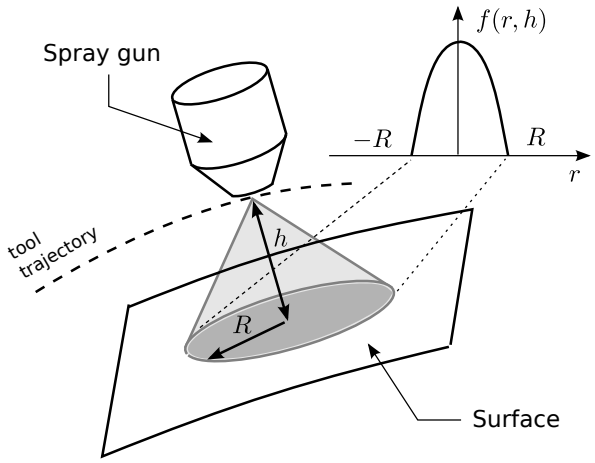

Fig. 1. Model of the painting task

From this simple model, using the tools of differential geometry, see Arikan and Balkann (2000), it is possible to compute the deposition of paint on any point of the surface as a function of the gun trajectory.
Assuming that the shape of the painting distribution is a cone, the axial symmetry of the tool allows a degree of freedom (i.e. the rotation around the tool axis of symmetry) that does not affect the task execution and can be exploited to further optimize the painting task. As an output of a non model-based optimization technique, the description of the trajectory of the spray gun requires 5 degrees of freedom, one less than those of a 6-axes industrial robot. Three degrees of freedom are used to place the tool in the Cartesian space, two degrees of freedom are necessary to make the spray gun perpendicular to the surface. Assume that the tool pose can be described in terms of Cartesian position of the TCP and a minimal description of the orientation of the tool, then the angle around the tool axis can be freely specified without affecting the task. Notice that the tool axis does not coincide in general with the axis of the last joint of the wrist, which makes the redundancy resolution

problem significant.
This paper contributes in applying a recently developed method, reviewed in this paper, to enforce an optimal redundancy resolution criterion on top of an existing industrial controller. In particular, thanks to a slight modification to the input of the kinematic inversion algorithm, it is possible to select the redundant degree of freedom (i.e. the angle of the tool around its axis of symmetry) so that the painting task can be performed according to some kinematic optimization criterion for the given manipulator.
The remainder of this work is organized as follows. In Section 2 some material regarding redundancy resolution is reviewed and it is explained how to enforce an arbitrary user-oriented redundancy resolution criterion. Section 3 explains how this method can be applied to a spray painting task. In Sections 4 and 5 simulations and experiments are discussed.

2. BACKGROUND AND PREVIOUS RESULTS

Let q_i $(i = 1,\dots,6)$ denote the variable characterizing the position of the i-th joint. The posture of the robot is uniquely defined once the vector $q = [q_1\ q_2\ \dots\ q_6]^T$ is given. In order to describe the task, let

$$s = [x\ y\ z\ \psi\ \theta]^T \tag{1}$$

be a minimal set of task variables (e.g. Cartesian coordinates of the TCP, x, y, z, and the first two ZYZ Euler angles, ψ, θ) and

$$s = f(q) \tag{2}$$

be the direct kinematics mapping. The kinematic inversion problem is to find q for a given s such that the previous equation holds. Usually, this problem is addressed at velocity level. In other words, the first time derivative of (2) is taken into account:

$$\dot{s} = \frac{\partial f}{\partial q}\dot{q} \equiv J_s(q)\,\dot{q} \tag{3}$$

where J_s is the task-Jacobian [1] .
Since the painting task is redundant, both (2) and (3) have infinite solutions. This fact can be used to enforce additional requirements to make, for instance, the painting operation more efficient.
The so called *extended Jacobian*, proposed in Baillieul (1985), is aimed at the design of a redundancy resolution technique that enforces the minimality/maximality of a certain cost-function. Consider a differentiable objective function to be optimized, say $U(q)$, and let N be a null-space basis of the task Jacobian matrix J_s. The following optimization problem can be formalized:

$$\max_q U(q) \text{ subject to } s - f(q) = 0 \tag{4}$$

It can be easily proven, see Seraji (1994), that the solutions of (4) are such that

$$N^T \frac{\partial U}{\partial q} = 0 \tag{5}$$

Therefore, in order to enforce the holonomic constraint (5), the following inverse kinematic algorithm can be adopted:

$$\dot{q} = \begin{bmatrix} J_s \\ \dfrac{\partial}{\partial q}\left(N^T \dfrac{\partial U}{\partial q}\right) \end{bmatrix}^{-1} \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} \tag{6}$$

Unfortunately this is not usually feasible with current industrial controller architectures: any existing industrial controller implements its own proprietary kinematic inversion algorithms.

[1] Capital letters are going to be used to denote matrices. Their dependence on q, if any, will be omitted.

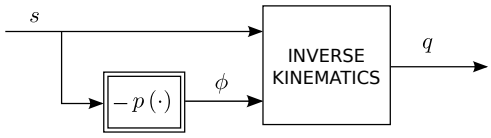

Fig. 2. General framework for redundancy resolution, Zanchettin and Rocco (2010)

Industrial controllers for redundant manipulators may implement redundancy resolution criteria which in general have no relations with criteria like (6). In particular, in the case of a 6-axis manipulator, the kinematic inversion algorithm is expected to be fed with a full description of the TCP pose, in terms of both Cartesian position and orientation. In other words, it usually happens that either the robotic programmer or the system integrator do not have access to the proprietary code in order to enforce his/her own inverse kinematics algorithm. As a consequence, when one or more task variables do not have to assume a prescribed value to accomplish the task (the third ZYZ Euler angle ϕ in the case of the painting task), the most commonly adopted solution is to specify for it/them a constant value, which does not help to exploit the redundancy in the desired way.
A previous work, see Zanchettin and Rocco (2010), addresses this problem. In particular, it shows that any user-oriented, as long as holonomic, redundancy resolution criterion, such as (5), can be enforced even when it is not possible to modify the embedded code in the industrial controller. The main result, as applied to the paintint task, can be stated as follows.

Theorem 2.1. Consider a simply-connected open subset where the task Jacobian J_s is full rank and suppose that there exists a differentiable function $\phi(\cdot) : \mathbb{R}^6 \to \mathbb{R}$ such that the augmented Jacobian J_A, Egeland (1987):

$$J_A = \begin{bmatrix} J_s \\ \Phi \end{bmatrix} \tag{7}$$

is non-singular, where $\Phi = \partial\phi/\partial q$.
Let $\sigma(q) = 0$ be a holonomic constraint to be enforced during the motion. Then there exists a differentiable function $p(\cdot) : \mathbb{R}^5 \to \mathbb{R}$ such that the method

$$\dot{q} = J_A^{-1} \begin{bmatrix} \dot{s} \\ -\dot{p}(s) \end{bmatrix} \tag{8}$$

enforces the desired constraint $\sigma(q) = 0$.

Proof see Zanchettin and Rocco (2010).

Beside the technicality of such result, which can be also applied in the case of multiple degrees of redundancy, Theorem 2.1 simply states that every holonomic constraint can be enforced within a pre-coded inverse kinematics algorithm. In particular, the augmented inverse Jacobian is assumed here, but this result can be slightly extended to any algorithm such as the transpose Jacobian [2]

$$\dot{q} = J_A^T \left[\left(s^d - f(q)\right)^T \ \left(-p\left(s^d\right) - \phi(q)\right) \right]^T \tag{9}$$

or even a closed-form kinematic inversion technique

$$q = f_A^{-1}\left(s^d, -p\left(s^d\right)\right) \tag{10}$$

where $f_A(\cdot)^{-1}$ denotes the inverse kinematics function. This function exists away of singularities of the augmented Jacobian J_A in view of the inverse function theorem. Fig. 2 depicts the general framework, where it is clear that the input to the inverse

[2] The superscript d denotes the desired value.

kinematics algorithm, whatever it be, is composed making the desired value of the variable ϕ dependent on the desired value of the vector variable s through the user-defined function $p(\cdot)$. The existence of function $p(\cdot)$ is however not sufficient to enforce the desired holonomic constraint. Since usually such function cannot be computed in closed-form, simple guidelines are given in Zanchettin and Rocco (2010) to approximate the function in a least-squares sense. In particular, a *Monte Carlo* simulation, see Robert and Casella (2004), is adopted to extract joint configurations q satisfying the desired constraint $\sigma(q) = 0$, see the Appendix. Finally, using such joint configurations, function $p(\cdot)$ can be selected within a pre-defined functional space with a common least-squares approach. In fact, once a basis $p_1(\cdot), \cdots, p_w(\cdot)$ of the functional space has been defined, every function $p(\cdot) \in \mathscr{P}$ can be written as a linear combination of the basis functions, as follows:

$$\mathscr{P} = \{p(\cdot) = c_1 p_1(\cdot) + \cdots + c_w p_w(\cdot) = \hat{p}(\cdot)^T c\} \qquad (11)$$

where $c = [c_1 \ \dots \ c_w]^T$ and $\hat{p}(\cdot)^T = [p_1(\cdot) \ \dots \ p_w(\cdot)]$.
The function $p(\cdot)$ can be finally selected by solving the following optimization problem:

$$c^0 = \arg\min_{c \in \mathbb{R}^w} \sum_q \left(\hat{p}(f(q))^T c + \phi(q) \right)^2 \qquad (12)$$

3. APPLICATION TO A SPRAY PAINTING TASK

In this Section we first define the cost function adopted to solve the redundancy of the painting task. Then, applying the method described in Section 2, we elaborate on the way this method can be applied in the context of the spray painting a robot ABB IRB 140, see Fig. 3 and Tab. 1 for Denavit-Hartenberg parameters, equipped with an ABB RB1000-WSC painting tool shown in Fig. 4. Assuming that one would like to exploit the redundancy

Fig. 3. ABB IRB 140 6-axes industrial manipulator

to optimize the manipulability measure, see Yoshikawa (1985), let J_ω be the robot Jacobian in the world frame and assume that, during the motion, the manipulability index $U(q)$ has to be maximized, where

$$U = \sqrt{\det(J_\omega J_\omega^T)} \qquad (13)$$

As explained by Seraji (1994) and reviewed in the previous Section, the maximization of a cost function can be turned into the following holonomic constraint:

Table 1. Denavit-Hartenberg Parameters

axis, i	$a_i[m]$	$d_i[m]$	α_i	θ_i
1	0.07	0	-90	q_1
2	0.36	0	0	q_2
3	0	0	-90	q_3
4	0	0.38	90	q_4
5	0	0	-90	q_5
6	0	0.065	0	q_6

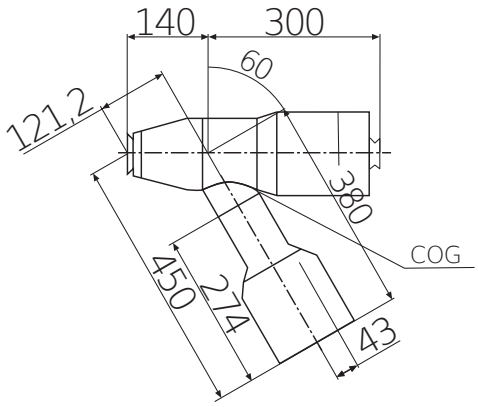

Fig. 4. ABB RB1000-WSC spray painting tool

$$\sigma(q) \equiv N^T \frac{\partial U}{\partial q} = 0 \qquad (14)$$

where N is any null-space basis of the task Jacobian J_s.
As already mentioned, any industrial controller is usually expected to be fed with a full description of the tool pose, namely three Cartesian coordinates and three ZYZ Euler angles have to be specified. As a matter of fact the third ZYZ Euler angle, ϕ, can be freely specified to perform the task.
The method described in the previous Section can be easily adopted to solve this problem. In particular, the constraint $\phi = -p(s)$ can be enforced on top of *any* existing implementation of the kinematic inversion algorithm, as sketched in Fig. 5, to achieve the optimal redundancy resolution requirement $\sigma(q) = 0$ in (14).
A number of $M = 5000$ Monte Carlo samples consistent with the optimality constraint (14) have been extracted according

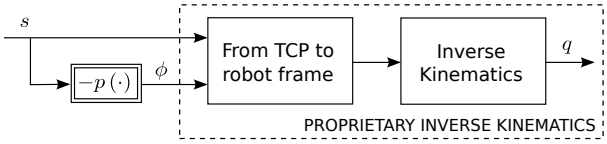

Fig. 5. Inverse kinematics algorithm

to the hit-and-run Monte Carlo simulation using the algorithm described in the Appendix. Then, the function $p(\cdot)$ in Fig. 5 has been selected within a functional space composed of polynomials in Cartesian variables of the task and a modified truncated Fourier series (which usually guarantees a faster convergence compared to the common Fourier series, see Huybrechs and Olver (2009)) in the orientation variables, yielding [3]

$$\begin{aligned} p(s) = {} & 0.99\cos(0.5\psi)\sin(\theta) + 0.50\cos(1.5\psi) \\ & + 1.42y - 0.63((x-0.4)/0.3 - 1) \\ & + 1.56\sin(\psi)\cos(0.5\theta) - 1.96\sin(\psi)\sin(\theta) \\ & - 0.40((z-0.2)/0.4 - 1) \\ & - 0.31((z-0.2)/0.4 - 1)^2 \end{aligned} \qquad (15)$$

[3] Only the most significant basis functions are taken into account.

with a root mean square error of less than 6 deg.

4. SIMULATIONS

In the following the output of two simulations is discussed. In the former, the algorithm in Fig. 5 is implemented with the function $p(\cdot)$ selected according to (15) (which has been selected to maximize the manipulability index during the motion). In the latter, $p(\cdot)$ was simply set to 0 yielding the constraint $\phi = 0$, which implies the tool to be aligned with the world's x-axis.

For both the simulations, the input of the inverse kinematic algorithm was an S-shaped trajectory laid on a curved smooth surface, resembling a car door. The path is covered from left to right in 6 s. We assume that the trajectory is given as the typical output of a non model-based optimization technique for spray painting, and thus cannot be further optimized, but still it allows one degree of freedom when covered with the tool mounted on the manipulator.

In Figs. 6 and 7 the manipulability ellipsoids are shown (together with a measure of their volume, i.e. the manipulability index U in (13)) along the followed path, using the two redundancy resolution techniques. As one can see, apart from the first turn, using the optimal redundancy resolution technique in (15) the manipulability can be significantly increased. In Fig. 8 the

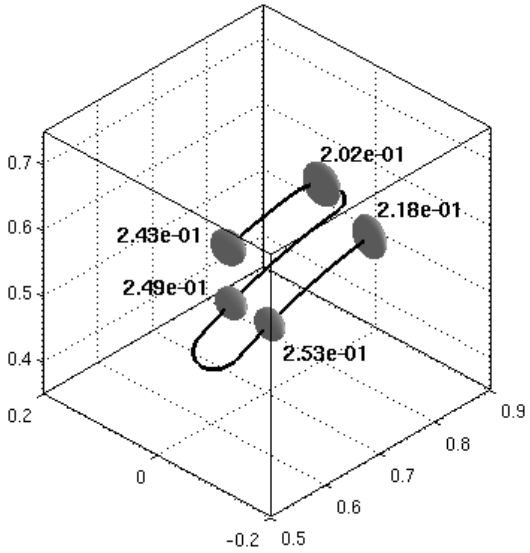

Fig. 6. Optimal redundancy resolution: manipulability ellipsoids

time history of the manipulability index is shown and compared with the maximum achievable. As one can see using the redundancy resolution technique with the function $p(\cdot)$ in (15) the optimality of the manipulability index is obtained, confirming the accuracy of the approximation that led to (15). On the other hand, the simulated trajectory corresponding to $p(\cdot) \equiv 0$ shows how smaller the manipulability index can be compared to its maximum value, if the redundant degree of freedom is not accurately planned. For instance, at time $t = 3.5$ s, corresponding to the second turn, the optimal manipulability index is 400% greater than the one obtained with a non-optimal redundancy resolution. Since usually the programmed task velocity around turn is as higher as possible, a small manipulability index might cause very high joint velocities, as actually happens at time $t = 3.5$ s, as shown in Fig. 9.

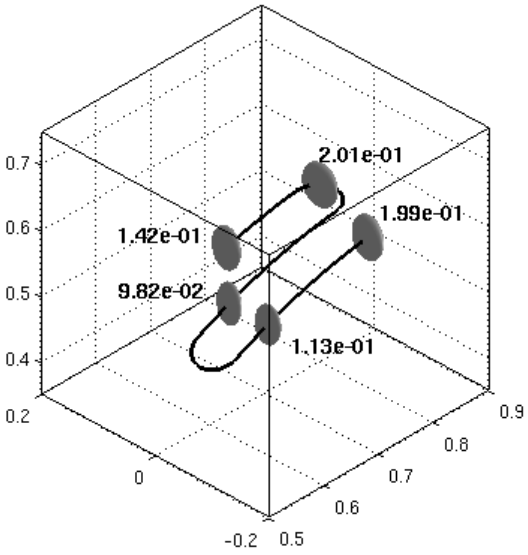

Fig. 7. Non-optimal redundancy resolution: manipulability ellipsoids

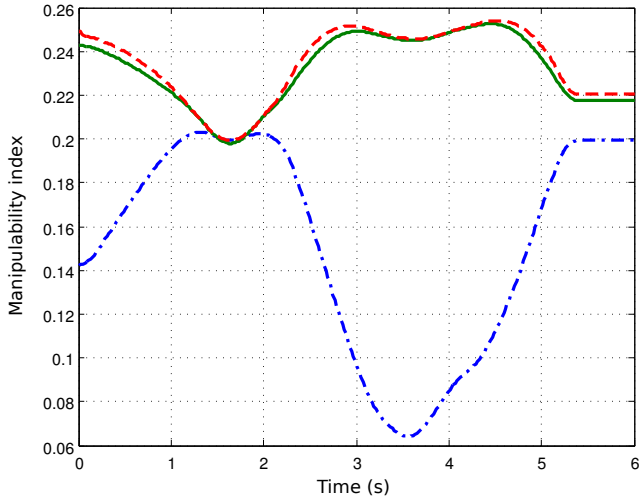

Fig. 8. Manipulability index: redundancy resolution using $\phi = -p(\cdot) \equiv 0$ (dash-dotted blue), optimal redundancy resolution using $p(\cdot)$ in (15) (solid green), maximum manipulability index (dashed red)

5. IMPLEMENTATION AND EXPERIMENTS

The algorithm sketched in Fig. 5 has been implemented on top of the industrial controller without adding any additional hardware. The redundancy resolution criterion has been coded directly in RAPID [4] and the function $p(\cdot)$ has been implemented as a function call in the execution code.

While the proposed algorithm should be executed between the trajectory generator and the proprietary inverse kinematic algorithm, in the actual implementation the function $p(\cdot)$ is computed prior to the trajectory interpolator. Moreover, some additional software has been coded in order to transform unit quaternion in ZYZ Euler angle and vice-versa. In the RAPID interpreter, in fact, only two different representations are available: unit quaternion and XYZ Euler angles. The details of the actual implementation are further discussed in the Appendix.

Two experiments, corresponding to the simulations discussed in

[4] RAPID is the programming language for ABB robots.

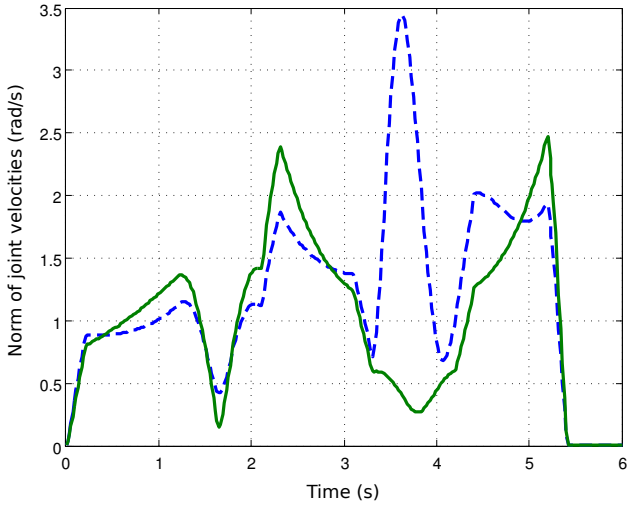

Fig. 9. Norm of joint velocities $\sqrt{\dot{q}^T\dot{q}}$: redundancy resolution using $\phi = -p(\cdot) \equiv 0$ (dashed blue), optimal redundancy resolution using $p(\cdot)$ in (15) (solid green)

the previous Section, have been performed. In Fig. 10 the time histories of the acquired manipulability indexes are compared to the simulated ones. As one can see, the behavior is quite similar. The only differences are mainly due to the trajectory interpolator. Since the acquisition functionality has been implemented within the RAPID code as a low-priority interrupt handler, the sampling frequency and its jitter are not suitable to compute joint velocities. For this reason, we will not compare joint velocities as previously done in simulation.

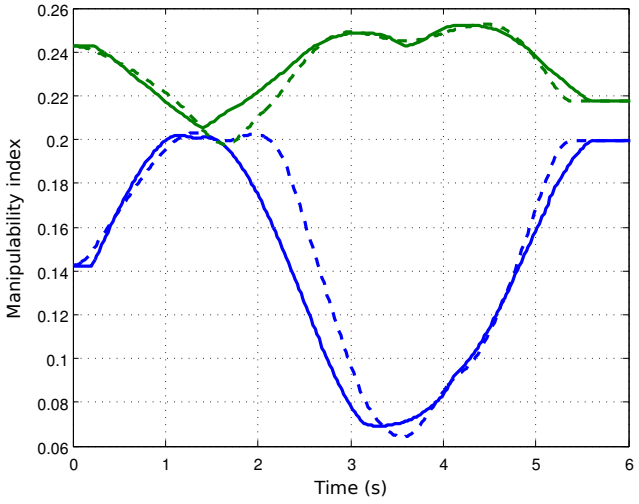

Fig. 10. Manipulability index: redundancy resolution using $\phi = -p(\cdot) \equiv 0$ (blue), optimal redundancy resolution using $p(\cdot)$ in (15) (green), either experimental (solid) or simulated (dashed)

6. CONCLUSIONS

When spray painting is executed with a symmetrical gun, it allows an extra degree that can be exploited to optimize the task execution. Unfortunately this is not usually feasible with current industrial controller architectures: any existing industrial controller implements its own proprietary kinematic inversion algorithms.
In this paper, we applied redundancy resolution techniques to a spray painting task. A method which can be implemented on top of the existing kinematic controller was adopted in order to exploit the task redundancy of this operation. In particular, the redundant degree of freedom was used to maximize the manipulability index during the motion.
Simulations have been run and showed that using the optimal redundancy resolution the generated joint trajectory is smoother and potential peaks in joint velocities, for instance during fast turns, can be avoided. Experiments confirmed the applicability of the proposed method and its potential impact in spray painting applications.

7. APPENDIX

7.1 Pseudo-code of the Monte Carlo sampling procedure

In this paper, an adapted version of the so-called *hit-and-run* method has been considered, see Chen and Schmeiser (1996). Under very simple assumptions, this method generates randomly selected samples from a quasi-uniform distribution in $\mathscr{D} = \{|\sigma(q)| < \varepsilon\}$. A pseudo-code of the algorithm is given in the following.

Hit-and-run sampling Suppose that $k-1 < N$ samples have been already generated in $\mathscr{D}$. Then, in order to generate the $k-th$ sample:

(1) select the number of steps, $T \gg 1$;
(2) pick a random sample $q^{(p)}$ from the ones generated previously $p = 1, \cdots, k-1$;
(3) set $s \leftarrow 0$ and $q^{(k),0} \leftarrow q^{(p)}$;
(4) generate a random direction v, $\|v\| = 1$ in the null-space of $\sigma(q) = 0$, i.e. such that $[\partial\sigma/\partial q]\, v = 0$
(5) move along the null space $q^{(k),s+1} = q^{(k),s} + \mu v$;
(6) set $s \leftarrow s+1$;
(7) if $s = T$ set $q^{(k)} \leftarrow q^{(k),T}$, else goto step 4);
(8) return $q^{(k)}$.

Notice that if $T \gg 1$ is selected large enough, then, since the point $q^{(k),s+1}$ is very close to $q^{(k),s}$, this algorithm spreads rapidly within a simply-connected region $\mathscr{D}$, where the constraint $\sigma(q) \approx 0$ is enforced. In the literature, these algorithms are also referred to as *Markov Chain Monte Carlo* methods. The graph consists in configurations q such that $|\sigma(q)| < \varepsilon$. The sequence of randomly explored nodes of the graph constitutes the Markov Chain. A detailed review on the use of these methods to generate uniform distributions can be found in Tempo et al. (2005).

7.2 RAPID code for redundancy resolution

In the following the implementation details of the block diagram in Fig. 5 are presented. In particular Fig. 11 describes the algorithm actually implemented on top of the kinematic inversion procedure of the industrial controller. The input is a full description of the robot pose, in terms of Cartesian position of the TCP and orientation of the tool (expressed using unit quaternions). The first two ZYZ Euler angle, i.e. ψ, θ, are extracted from the quaternion, while the third is discarded and computed using $p(\cdot)$ in (15). Then, the corresponding set of Euler angles is turned into a unit quaternion which is used to feed the trajectory interpolator and the subsequent kinematic inversion algorithm. An excerpt of the RAPID code implementation is given in the following.

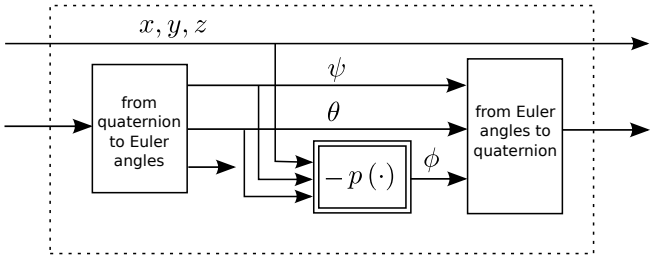

Fig. 11. Actual implementation

```
! painting task
PROC main()

 ! go to start position
 MoveL p(p0), v100, fine, sprayGun;

 ! start spraying
 MoveL p(p1), v100, z10, sprayGun;
 MoveC p(p2), p(p3), v100, z10, sprayGun;
 MoveL p(p4), v100, z10, sprayGun;
 MoveL p(p5), v100, z10, sprayGun;
 MoveC p(p6), p(p7), v100, z10, sprayGun;
 MoveL p(p8), v100, z10, sprayGun;
 MoveL p(p9), v100, fine, sprayGun;

ENDPROC


! function p for redundancy resolution
FUNC robtarget p(robtarget t)

 ! from quaternion to ZYZ Euler angles
 n     := t.rot.q1;
 ex    := t.rot.q2;
 ey    := t.rot.q3;
 ez    := t.rot.q4;
 r13   := 2*(ex*ez+n*ey);
 r23   := 2*(ey*ez-n*ex);
 f_deg := Atan2( r23 , r13 );
 f_rad := f_deg*pi/180;
 s_deg := Atan2( Sqrt(Pow(r13,2)+
           Pow(r23,2)), 2*(Pow(n,2)+
           Pow(ez,2))-1 );
 s_rad := s_deg*pi/180;
 x     := t.trans.x/1000;
 y     := t.trans.y/1000;
 z     := t.trans.z/1000;

 ! third Euler angle, function p
 t_deg := 180/pi*evalp( x,y,z,f_deg,
                        s_deg );

 ! modifying null-space
 t.rot := fromZYZtoQuat( f_deg,s_deg,
                        -t_deg );

 ! return optimal gun orientation
 RETURN t;

ENDFUNC
```

REFERENCES

Antonio, J., Ramabhadran, R., and Ling, T. (1997). A framework for optimal trajectory planning for automated spray coating. *International Journal on Robotics and Automation*, 12, 124–134.

Arikan, M.S. and Balkann, T. (2000). Process modeling, simulation and paint thickness measurement for robotic spray painting. *Journal of Robotic Systems*, 17, 479–494.

Baillieul, J. (1985). Kinematic programming alternatives for redundant manipulators. In *IEEE Conf. on Robotics and Autom., ICRA*, 722–728.

Chen, M. and Schmeiser, B. (1996). General hit-and-run Monte Carlo sampling for evaluating multidimensional integrals. *Operations Research Letters*, 161–169.

Egeland, O. (1987). Task-space tracking with redundant manipulators. *IEEE Journal on Robotics and Automation*, 1, 471–475.

Fa-zhong, L., Dean, Z., and Guihua, X. (2009). Trajectory optimization of spray painting robot based on adapted genetic algorithm. In *International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2009*, 907–910.

From, P.J. and Gravdahl, J.T. (2007). General solutions to functional and kinematic redundancy. In *IEEE International Conference on Decision and Control, CDC 2007*, 5779–5786.

Huybrechs, D. and Olver, S. (2009). Rapid function approximation by modified Fourier series. *Highly Oscillatory Problems*, 51–71.

Kim, T. and Sarma, S. (2003). Optimal sweeping paths on a 2-manifold: a new class of optimization problems defined by path structures. *IEEE Transactions on Robotics and Automation*, 19, 613–636.

Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer texts in Statistics.

Seraji, H. (1994). Kinematic functions for redundancy resolution using configuration control. In *US Patent*.

Suh, S., Woo, I., and Noh, S. (1991). Development of an automatic trajectory planning system (atps) for spray painting robots. In *IEEE International Conference on Robotics and Automation, ICRA 1991*, 209–216.

Tempo, R., Calafiore, G., and Dabbene, F. (2005). *Randomized algorithms for analysis and control of uncertain systems*. Springer.

Wei, C. and Dean, Z. (2009). Tool trajectory optimization of robotic spray painting. In *Second International Conference on Intelligent Computation Technology and Automation, ICICTA 2009*, 419–422.

Xia, W., Wei, C., and Liao, X. (2009). Surface segmentation based intelligent trajectory planning and control modeling for spray painting. In *International Conference on Mechatronics and Automation, ICMA 2009*, 4958–4963.

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *International Journal of Robotics Research*, 4(2), 3–9.

Zanchettin, A.M. and Rocco, P. (2010). A general user-oriented framework for holonomic redundancy resolution in robotic manipulators using task augmentation. Under review.