



HAL
open science

Constraint Compliant Control for a Redundant Manipulator in a Cluttered Environment

Sébastien Rubrecht, Vincent Padois, Philippe Bidaud, Michel de Broissia

► **To cite this version:**

Sébastien Rubrecht, Vincent Padois, Philippe Bidaud, Michel de Broissia. Constraint Compliant Control for a Redundant Manipulator in a Cluttered Environment. Jadran Lenarcic and Michael M. Stanisic. *Advances in Robot Kinematics: Motion in Man and Machine Proceedings of the 12th International Symposium on Advances in Robot Kinematics*, Springer, pp.367–376, 2010, 10.1007/978-90-481-9262-5_39 . hal-00586455

HAL Id: hal-00586455

<https://hal.science/hal-00586455>

Submitted on 15 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Constraint Compliant Control for a Redundant Manipulator in a Cluttered Environment

S. Rubrecht, V. Padois, P. Bidaud and M. de Broissia

^{1 2 3} *Institut des Systèmes Intelligents et de Robotique, Paris, France,*
e-mail: rubrecht@isir.upmc.fr

⁴ *Bouygues Travaux Publics, 78280 Guyancourt, France*

Abstract. In this paper, we present an iterative kinematics control law for redundant manipulators overcoming some usual problems associated to cluttered environments (constraints violations, oscillations, environment dilatation). The Constraints Compliant Control approach relies on a passive avoidance scheme (no motion generation for constraints avoidance) on a limited number of constraints selected from a vicinity analysis. A scaling solution based on the feasible motions with respect to the constraints enables to reach the frontiers of the workspace. Two missions described as sequences of key frames are simulated to compare the performances of the Constraint Compliant Control with state of the art control laws. The obtained computation times remain acceptable to consider a use in real time.

Key words: Redundancy resolution, Inverse velocity kinematics, Control.

1 Introduction

Industrial robotics manipulator missions have evolved from well defined tasks in structured environments to missions where the tasks are more complex (involving multiple objectives among which various constraints) and where the environments are not always known in advance and can be harsh and cluttered. As an example, the motivations of the presented work consider the use of a teleoperated manipulator aimed at working in a tunnel boring machine cutter head (see Figure 1 and [1]).

In such environments, the use of redundant manipulators enables to specify simultaneously various goals on different parts of the manipulator, explicitly or automatically. It gives the possibility to fulfill different tasks while strictly satisfying a certain amount of constraints (collision avoidance, joint position and velocity limits). However, to our knowledge, there is no multi-objective method that guarantees a safe behaviour without substantially reducing the motion capabilities, especially when the manipulator is expected to get close to the constraints surfaces (which may be operational surfaces). This is the problem tackled in this paper.

Consider a manipulator with n DOFs and a set of objectives (not necessarily imposed to the same part of the manipulator) involving m operational displacements

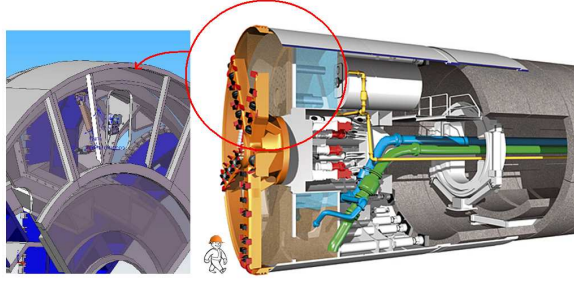


Fig. 1 Tunnel Boring Machine: manipulator in the excavation room.

δx . The relation between the operational and the joints displacements (considered here as the actuators input signal, extension to torque controlled actuators is trivial)

$$\delta x = J\delta q \quad (1)$$

where J is the jacobian associated to the objectives. In order to satisfy the constraints, usual avoidance strategies rely on an avoidance term δx_C derived from a potential based on the inverse of the distance to the constraint [2]. This strategy is called *active*, as a motion is generated to avoid the constraint. Thus, the expected behaviour will result from the combination of the operational tasks $((J_T, \delta x_T)$, concatenation of the tasks) and the constraints $((J_C, \delta x_C)$, concatenation of the constraints). In the particular cases of joints related constraints (joint position and velocity limits), the constraint jacobian matrix is $[0, \dots, 0, \pm 1, 0, \dots, 0]$, the ± 1 rank being at the joint number in the manipulator chain, the sign depending on the limit type (maximum or minimum).

These motions can be strictly prioritized in a multiobjective control law (see [4])

$$\delta q = J_1^\# \delta x_1 + (J_2 P_{J_1})^\# (\delta x_2 - J_2 J_1^\# \delta x_1) \quad (2)$$

where indexes 1 and 2 can be replaced by C and T and conversely. The $\#$ is a pseudoinversion operator (see [3]) and P_{J_1} is a projector on the kernel of J_1 (details about inversions and projectors are exposed in section 3.1).

Maciejewski [4] and Chaumette [5] proposed to put the constraints avoidance at the bottom of the task hierarchy ($1 \leftarrow T$, $2 \leftarrow C$; referenced as control law A). It enables to strictly fulfill the tasks, but in case of conflict between the tasks and the constraints, the manipulator may violate the constraints. To avoid this, Sentis [6] puts the constraints at the top of the hierarchy ($1 \leftarrow C$, $2 \leftarrow T$; referenced as control law B). However, this choice has a bad impact on the reachability of the manipulator as the avoidance motions need activation thresholds, which results in oscillations on the functional parts of the manipulator. Moreover, as the constraints are not homogeneous (obstacle avoidance involves body lengths that depends on the obstacle position w.r.t. the robot, while joint limits do not), it is not possible to use a bounded potential. It implies a consequent size of the threshold, as the potential

should grow from zero to infinity along it, and also the possible presence of infinite terms in the control law.

In order to avoid oscillations and infinite terms, Baerlocher uses in [7] an iterative approach based on an activation matrix that pre-multiplies the jacobian of the tasks. The algorithm deactivates the joints which violate their boundaries, so they are not included in the model inversion. However, this approach does not include other types of constraints and it often resorts to iterations in cases for which usual approaches find directly an *admissible* (i.e. constraints compliant) solution.

This paper presents a framework that extends the approach of Baerlocher to the collisions avoidance and satisfies the joint velocity limits; in particular, when the environment is not overconstrained (number of DOF sufficient to track the tasks while complying to the constraints), the control law is equivalent to the classical control law A . A particular attention has been paid to keep the computation times in the range of the one obtained with control law A and B .

In section 2, the Constraint Compliant Control (CCC), relying on the principle of passive avoidance, is introduced. Then, section 3 presents the simulations of two missions and the comparative results of the control laws presented previously (A , B and CCC). Finally, section 4 gathers the conclusions and the work perspectives.

2 Constraint Compliant Control

This section exposes our contribution. First, the passive avoidance principle is developed, according to which the robot should not move to avoid static constraints. Then, the control law expression is exposed; finally, the whole algorithm is presented.

2.1 Passive Avoidance

The approach of Baerlocher in [7] is equivalent to the insertion of a superior hierarchical level in which operational displacement would be null ($\delta x_C = 0$). It has the advantage to satisfy the joint boundaries in all cases. The extension to every static constraints (static obstacles, joint position and velocity limits) is done using the following control law

$$\delta q = J_C^\# 0 + (J_T P_{J_C})^\# (\delta x_T - J_T J_C^\# 0) = (J_T P_{J_C})^\# \delta x_T \quad (3)$$

which ensures a strict compliance with any constraints in J_C . This avoidance method is *passive* as no motion is generated by the proximity to a constraint. As what is described by the constraints expression is actually static in the physical world, no arbitrary motion is needed to satisfy them. Anyway, it is not often mandatory to forbid motions in all the constraints directions; it is not even desired, as it prevents

from getting closer to the constraints, but also from moving away from them. So, it is relevant to iterate on the constraints combinations to find the set of constraints that must be passively avoided to obtain an *appropriate* solution δq , *i.e.* that minimizes the operational error while being admissible. Let J_{CC} be a matrix containing a combination of lines of J_C . The control law then becomes

$$\delta q = (J_T P_{J_{CC}})^{\#} \delta x_T \quad (4)$$

Iterations are carried out to find the lines combination J_{CC} which yields an appropriate δq . In particular, the admissibility test is performed using Eq. (7).

2.2 Active avoidance in additional objective

The solutions admissibility can be increased by adding an extra term of active avoidance on the constraints that are not avoided passively:

$$\delta q = \underbrace{(J_T P_{J_{CC}})^{\#} \delta x_T}_{\text{Passive avoidance}} + \underbrace{(J_{\overline{CC}} P_{[J_{CC}]}^{\perp})^{\#} (\delta x_{\overline{CC}} - J_{\overline{CC}} (J_T P_{J_C})^{\#} \delta x_T)}_{\text{Active avoidance}} \quad (5)$$

where $J_{\overline{CC}}$ is the complement of the lines of J_{CC} in J_C , $\delta x_{\overline{CC}}$ is the desired operational avoidance displacement associated to $J_{\overline{CC}}^{-1}$ and $P_{[J_{CC}]}$ is the projector on the kernel of the concatenation of J_{CC} and J . The extra term tends to move the manipulator away from the constraints as long as the tasks are not impacted. In particular, when J_C is empty, the control law behaves like control law A and a single iteration is needed when the situation is not overconstrained. The avoidance coefficients norm $|\delta x_{\overline{CC},i}|$ can be limited to avoid values tending to infinity. Finally, there are no oscillations on the functional part as the active avoidance term is under the task related term in the hierarchy.

2.3 Particular case of the joint velocity limit - Scaling

The joint velocity limits should not be addressed with passive avoidance, as it would stop a joint to prevent it from going too fast. If a joint displacement is too high, the solution is scaled to reach the maximal admissible displacement. To keep the motion coherency, the scaling is done in a way that preserves the operational direction

$$\delta q \leftarrow \delta q \min_{0 \leq i \leq n} \frac{\delta q_{i,max}}{\|\delta q_i\|} \quad (6)$$

¹ In a practical aspect, $J_{\overline{CC}}$ (resp. $\delta x_{\overline{CC}}$) can be replaced by J_C (resp. δx_C , desired operational avoidance displacement associated to J_C) in Eq. (5) without any consequence on the result.

where $\delta q_{i,max}$ is the maximum displacement of joint i on one iteration. This step is carried out only if a joint displacement δq_i is higher than $\delta q_{i,max}$.

The test of admissibility with respect to the constraints is performed through

$$J_C \delta q \leq l \quad (7)$$

where l is the concatenation of the maximum instantaneous displacements. In order to increase the validity of the solutions, the scaling is extended to all the solutions

$$\delta q \leftarrow \delta q \min_{\substack{0 \leq i \leq m \\ (J_C \delta q)_i > 0}} \left(\frac{l_i}{(J_C \delta q)_i} \right) \quad (8)$$

The scaling is done only if Eq. (7) is not verified. This step enables to obtain an admissible solution for each constraints combination in J_C . Actually, as all the constraints are in the form of Eq. (7), the solutions space is convex and contains the null solution (no motion). So, in every direction of the joint displacement space, there exists an admissible solution, which norm is null in the worst case. This method enables to get as close as possible to the constraints if there is a persistent demand in that way. The general algorithm of a control iteration is given by Algorithm 1.

Algorithm 1 : CCC

```

 $J_C \leftarrow$  constraints under the active avoidance threshold
 $\delta q_{comb} \leftarrow 0$  rad;  $\delta q \leftarrow 0$  rad
Errcomb  $\leftarrow 0$  m; Err  $\leftarrow 1$  m;  $\epsilon \leftarrow 10^{-2}$  m;
for all Lines Combinations  $J_{CC}$  in  $J_C$  do
     $\delta q_{comb}$  Computation - Eq. (5)
    Admissibility Test - Eq. (7)
    Scaling - Eq. (8)
    Errcomb  $\leftarrow \|J \delta q - \delta x_T\|$ 
    if Errcomb  $\leq$  Err then
        Err  $\leftarrow$  Errcomb;  $\delta q \leftarrow \delta q_{comb}$ 
        if Err  $\leq \epsilon$  then
            break
        end if
    end if
end for
send  $\delta q$ 
    
```

3 Implementation and Comparative results

This section presents the results of control laws A , B and CCC on two missions in the same environment. The proposed environment is composed of a column and a wall; the manipulator has 7 DOFs, all the links being 1 DOF rotational joints. The

environment and the manipulator are represented on Figure 2. The presented experi-

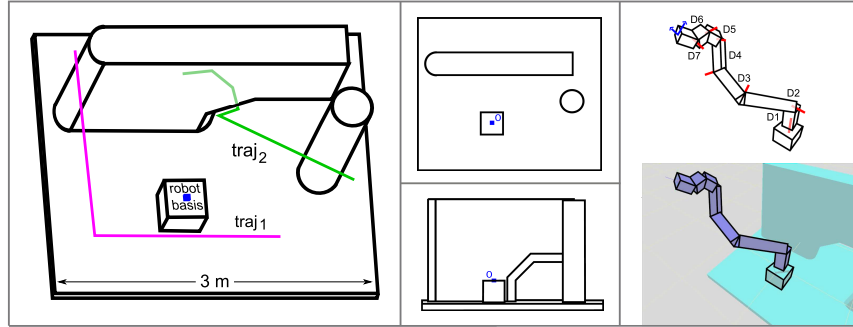


Fig. 2 Views of the environment and the two trajectories to track; manipulator schemes.

ments simulate two inspection missions involving trajectories close to the obstacles. For each mission, the manipulator must track a 3 coordinates trajectory: it is assumed that the effector (camera) has the orientation DOFs needed to observe the points to be inspected. For the sake of simplicity, the results presented here do not integrate joint boundaries avoidance even though the proposed framework can deal with this type of constraint without any specific difficulty.

- **Mission 1: Go around the wall by the left side.** The environment is barely constrained in that area, the manipulator tracks a trajectory (traj_1 , in Figure 2) of 330 points on a single way of 3.50 m, so a displacement of 1.1 cm is expected for each iteration. The mission is *feasible*, i.e. the number of DOFs of the manipulator enables to fulfill the mission while avoiding the constraints.
- **Mission 2: Reach a point behind the wall.** The trajectory (traj_2 , in Figure 2) has 560 points, for a go and come back trajectory (getting out of a very constrained configuration can be a problem). The total distance is 5.20 m long, so the expected displacement is 0.9 cm at each iteration. The mission is not *feasible* as the manipulator is not long enough to reach the furthest point.

3.1 Implementation

The implementation is done in C++, and uses KDL (see [8]) and Boost::ublas² libraries. We detail the following implementation elements:

- **Desired displacement δx_T :** difference between the current position and the current trajectory point.

² http://www.boost.org/doc/libs/1_41_0/libs/numeric/ublas/doc/index.htm

- **Pseudoinversion:** in the control law Eq. (5), in order to avoid inversion problems in the neighborhood of singularities, the pseudoinversions with exponent $\#$ are done using the Damped Least Square (DLS) method (see [9]), where the damping factor λ is chosen as 0.50. A common way of computing P_J is given by $P_J = (I - J^\# J)$; in practice, the DLS method induces an error that distorts the projection, making the influence of lower priority objectives on upper priority objectives possible, which is not acceptable in our case. A safe way to compute P_J can be obtained directly from the SVD of J which provides an access to the projector on the kernel of J without requiring its inversion (as mentioned in [10]). For the same reasons, a product $(J_a P_{J_b})^\#$ is always pre-multiplied by P_{J_b} .
- **Active avoidance :** for control laws A and B , the active avoidance threshold is fixed to 15 cm, the gains are proportional (factor $2.5 \cdot 10^{-3}$) to the inverse of the distance to the constraint. For the CCC , the active avoidance is fixed to 4 cm, the gains are the same than for control laws A and B but the maximum value of the avoidance magnitude δx_C is fixed to 0.25 (distance of 1 cm between the manipulator closest point and the obstacle).
- **Distance computation :** the distance computation is provided by the collision detection package SWIFT++³. The information given by SWIFT++ is, for each segment, the point of the segment that is the closest to the obstacle. The approach that consists in constraining only the closest point to the obstacle (based on [4] for the whole manipulator and on [11] for convex segments) is valid in continuous time. In discrete time, progressive displacements can violate constraints because of segment rotations. To our knowledge, no work has been carried out on that topic; a contribution of Kanehiro and al. in [12] deals with not strictly convex segments, in which the problems are similar to those encountered when getting from continuous to discrete time. To avoid the effects of this phenomenon, an envelope of 2 cm has been added around the environment.
- **Joint instantaneous displacement limit :** δq_{max} has been fixed to 0.02 rad.

3.2 Results and Analysis

Figures 3 and 4 present the results obtained on the 2 missions with control laws A , B and CCC . The computation times are given comparatively as they depend on the implementation and computing power.

- **Mission 1: Go around the wall by the left side.**
 - **Control law A .** As the behavior is identical to the CCC on mission 1 (see 2.2), it is not represented on Figure 3.
 - **Control law B .** The operational position mean error is 3.5 cm, the tracking is not optimal especially at the end where the effector gets close to the wall: oscillations are generated due to the thresholds (observable on both graphs) and the operational error grows up to 13.1 cm.

³ <http://gamma.cs.unc.edu/SWIFT++>

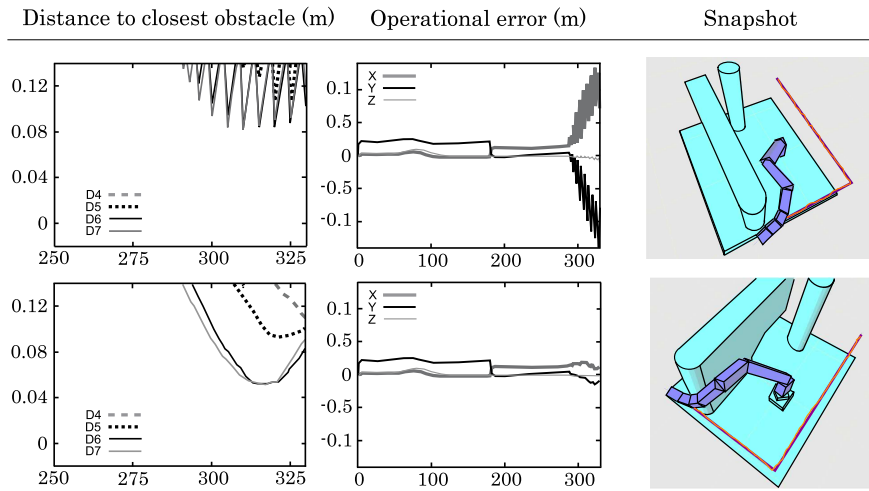


Fig. 3 Mission 1 results. Graphs of line 1 are obtained with control B , line 2 with control CCC

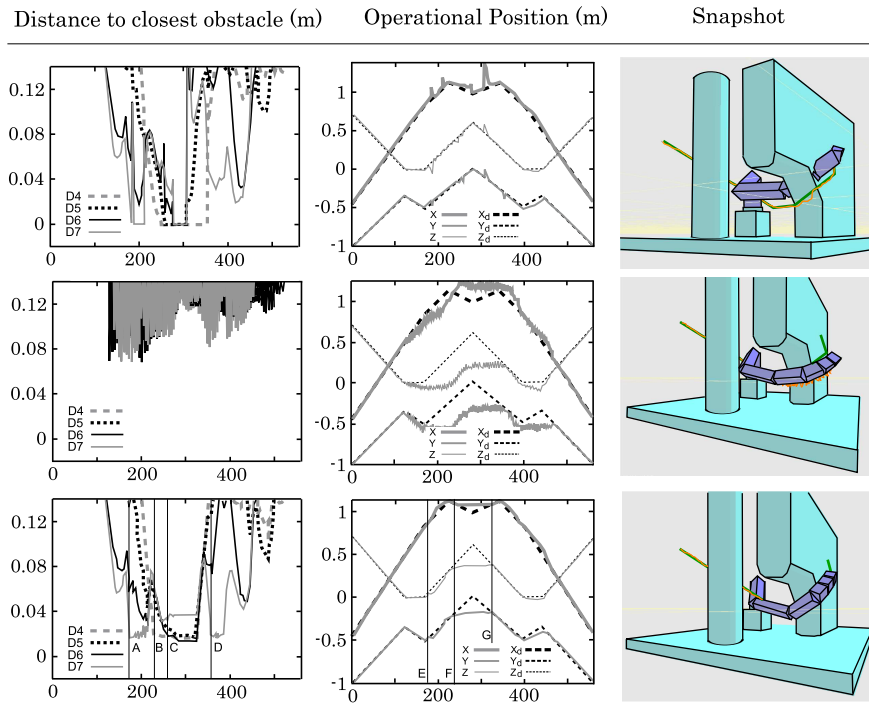


Fig. 4 Mission 2 results. Graphs of line 1 are obtained with control A , line 2 with control B , line 3 with control CCC

- **Control law CCC.** The behavior is identical to control law A: as there is no collisions when tracking the path with active avoidance at a lower level, the passive avoidance is not used and the manipulator has the same tracking error along the trajectory (operational position error inferior to 2.6 cm).
- **Mission 2: Join a point behind the wall.**
 - **Control law A.** The operational position mean error is 3.8 cm and reaches 35 cm on peaks. When the situation gets overconstrained, collisions occur (the distances to the obstacle reach 0 for DOFS 4, 5 and 7) as it is not possible to satisfy tasks and constraints; the peaks on the operational positions comes from avoidance terms tending to infinity.
 - **Control law B.** The oscillation phenomenon is higher than for mission 1 and the manipulator remains far from the path (operational position mean error 22.8 cm, max error 59.4 cm)
 - **Control law CCC.** When the manipulator comes close to the environment (enveloppe distance: 2 cm), the passive avoidance clamps the directions to the obstacles (point A, B, C and D on the first column graph) and the concerned segments moves along the orthogonal directions. When the manipulator seems completely stuck (point E on the second column graph), the scaling step enables little displacements to track as much as possible the desired displacements. Finally, when the direction is inversed (point F on the 3rd column graph), the manipulator is able to get away from the constraints directly. The computation time grows up 6.7 times w.r.t. the ones obtained with control law A on mission 1.

4 Conclusion and Perspectives

The CCC is an iterative control method that solves the hierarchical multi-objective control problem while satisfying any number of fixed constraints: obstacles, joint boundaries, joints velocity limits. The passive avoidance principle and the solutions scaling enables to overcome the drawbacks of active avoidance at the top (optimality loss, oscillations) or at the bottom (constraints violation, infinite terms) of the hierarchy, while ensuring a computation time low enough to consider its use in real time on classical manipulators.

The CCC performances can be compared to the one obtained with convex optimization algorithms (even if the latter cannot ensure strict priorities between the hierarchy levels). As an example, the algorithm QuadProg++⁴ satisfies the constraints with a maximum computation time of 3 times the one of control law A. Nevertheless, the insufficiency of the avoidance method (see section 3.1) added to the algorithm trend to run along the constraints make the manipulator not able to get away of the most constrained configuration: it does not track the second part of the trajectory.

⁴ <http://sourceforge.net/projects/quadprog/>

The work perspective includes:

- The continuity can be handled by including the limits of joint acceleration constraints but they introduce incompatibilities with obstacles and joint position limits; the use of virtual constraints (dampers) is under evaluation.
- Locally, the constraints combination choice can be guided by the constraints criticity or by favouring combinations retained for previous iterations, etc.
- In a more global scope, this choice can be adapted to the missions and it can depend on the parts of the robot: active avoidance for better motion capabilities, passive avoidance to work close to the constraints.

Acknowledgements This work is a part of the TELEMACH project. It is supported by the French National Research Agency (ANR), Interactive Systems and Robotics Program 2007 (PSIROB07).

References

1. Rubrecht, S. and Padois, V. and Bidaud, P.: Evolutionary Design of a Robotic Manipulator for a Highly Constrained Environment. In IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Exploring new horizons in Evolutionary Design of Robots. pp. 59–64 (2009)
2. Khatib, O.: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. The International Journal of Robotics Research. **5**(1), pp. 90–98 (1986)
3. Ben Israel, A. and Greville, T.N.E.: Generalized Inverses: Theory and Applications. Springer (2nd edition) (2003)
4. Maciejewski, A. and Klein, C.: Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. The International Journal of Robotics Research. **4**(3), pp. 109–117 (1985)
5. Chaumette, F. and Marchand, E.: A New Redundancy-based Iterative Scheme for Avoiding Joint Limits Application to visual servoing. In: IEEE International Conference on Robotics and Automation, **2**, pp. 1720–1725 (2000)
6. Sentis, L. and Khatib, O.: Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. The International Journal of Humanoid Robotics. **2**(4), pp. 505–518 (2005)
7. Baerlocher, P. and Boulic, R.: An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels. The Visual Computer: International Journal of Computer Graphics. **20**(6), pp. 402–417 (2004)
8. Bruyninckx, H.: Open Robot Control Software: the OROCOS project. In: IEEE International Conference on Robotics and Automation, **3**, pp. 2523–2528 (2001)
9. Nakamura, Y. and Hanafusa, H.: Inverse kinematics solutions with singularity robustness for robot manipulator control. Journal of Dynamic Systems, Measurement and Control. **108**, pp.163–171 (1986)
10. Salaun, C. and Padois, V. and Sigaud, O.: Control of Redundant Robots Using Learned Models: An Operational Space Control Approach. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 878–885 (2009)
11. Faverjon, B. and Tournassoud, P.: A Local Based Approach for Path Planning of Manipulators With a High Number of Degrees of Freedom. In: IEEE International Conference on Robotics and Automation. **4**, pp. 1152-1159 (1987)
12. Kanehiro, F. and Lamiraux, F. and Kanoun, O. and Yoshida, E. and Laumond, J-P.: A Local Collision Avoidance Method for Non-strictly Convex Polyhedra. In: Proceedings of Robotics: Science and Systems IV (2008).