

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/305778862>

# Industrial implementation of a multi-task redundancy resolution at velocity level for highly redundant mobile manipulators

Conference Paper · June 2016

CITATIONS

2

READS

436

4 authors:



Christian Scheurer  
KUKA Roboter GmbH  
8 PUBLICATIONS 65 CITATIONS

[SEE PROFILE](#)



Mario Daniele Fiore  
KUKA Roboter GmbH  
2 PUBLICATIONS 2 CITATIONS

[SEE PROFILE](#)



Shashank Sharma  
KUKA Roboter GmbH  
5 PUBLICATIONS 16 CITATIONS

[SEE PROFILE](#)



Ciro Natale  
Università degli Studi della Campania "Luigi Vanvitelli"  
142 PUBLICATIONS 1,910 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



SAPHARI [View project](#)



LOCOMACHS [View project](#)

# Industrial implementation of a multi-task redundancy resolution at velocity level for highly redundant mobile manipulators

Christian Scheurer, Corporate Research, KUKA Roboter GmbH, Augsburg, Germany

Mario Daniele Fiore, Corporate Research, KUKA Roboter GmbH, Augsburg, Germany

Shashank Sharma, Corporate Research, KUKA Roboter GmbH, Augsburg, Germany

Ciro Natale, Industrial and Information Engineering, Second University of Naples, Aversa, Italy

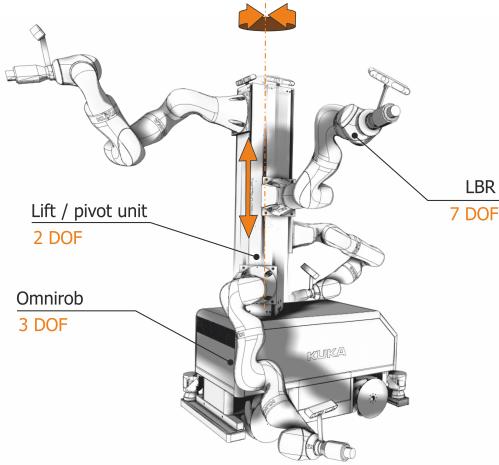
## Abstract

Mobile manipulators with many degrees of freedom (DOFs) have increasingly become of interest to industrial and service robotics due to their mobility and remarkable versatility. The redundancy of these robots can be conveniently exploited to generate internal joint motions that reconfigure the structure according to given task specifications. The availability of many DOFs allows the robot to execute various tasks at a time. Therefore, an efficient redundancy resolution becomes crucial to achieve multiple tasks simultaneously and use the capabilities of the complete robot system. Tasks can be of different nature and defined in different spaces; relative priorities among tasks are strictly enforced which is appropriate for situations that cannot tolerate compromises as in industrial environments. Null-space based behavioral approaches have been deeply investigated at velocity level as it is the core of the proposed control strategy. We implemented an effective priority strategy approach using a closed-loop version of the inverse kinematics of the robot together with Damped Least-Squares operators on Jacobian matrices. Different tasks with different priorities have been specified for our robot mimicking a typical industrial manufacturing setting. Simulations and real experiments on our 12 DOF mobile manipulator have been conducted to validate the implemented technique.

## 1 Motivation

Industrial robots have been developed for decades thinking of them as machines in a cell, where they carry out some fixed, planned tasks, moving and rotating a tool to precisely follow a predetermined Cartesian path. According to safety standards, no human being is allowed to stay in a cell while a robot is moving. With the pressing demand of robots in applications like service robotics, medical robotics, remote tele-operation, nuclear waste treatment and outdoor navigation, the trend is shifting from fixed base manipulators to flexible, multi-purpose mobile manipulators. Mobile manipulators have become of high interest also to the industrial sector because of their increased flexibility and effectiveness they offer due to the added mobility leading to unlimited horizontal workspace range. The goal of the European project named VALERI (Validation of Advanced, Collaborative Robotics for Industrial Applications) [1], for example, is to introduce a highly redundant mobile manipulator, the KUKA VALERI omniRob (Fig. 1) to the aerospace industry to help with the assembly of aerospace components and work hand-in-hand with humans on the production floor (Fig. 2). Therefore, industrial robots are required to overstep carefully arranged working cells and enter into semi-structured or even unstructured environments where they are supposed to safely interact with humans. Planning

robot motions completely offline in those dynamically changing environments with a high degree of uncertainty, may likely lead to losses in process quality or even to complete failures in the assigned tasks. Therefore, an efficient online redundancy resolution becomes crucial in industrial mobile manipulators in order to fully use their capabilities and to avoid dangerous situations. Its redundancy can conveniently be exploited to control the posture according to several simultaneous tasks like obstacle avoidance, self-collision avoidance, mechanical joint limit avoidance, compliant motion behavior as well as dexterity and versatility in the interaction with the environment are only some examples of constraint tasks which one may wish to fulfil along with the original and most important Cartesian end effector task. Optimization criteria for the joint motion have to be considered while tasks with different levels of priority can be specified. Additionally, the concurrent resolution of multiple tasks should be in strict hierarchy and errors on lower-prioritized tasks are only tolerated in the case of nearly conflicting tasks and only when not affecting the quality of the industrial process as such.



**Figure 1** KUKA VALERI mobile manipulator with 12 DOFs

## 2 State of the art online redundancy resolution

An investigation of online redundancy resolution strategies has been carried out. Methods are classified into velocity-based [2, 3, 4], acceleration-based [18] and force/torque-based [19] approaches. Velocity-based redundancy resolution principles have been one of the first local methods mentioned in literature and are widely used due to their good performance in simulations, in real experiments and the reduced computational burden [20]. They use first-order differential kinematic equations to solve redundancy in a least-square sense to minimize joint norm velocity [2, 3]. Specifically, the use of the pseudoinverse of the task Jacobian matrix guarantees optimal reconstruction of the desired task velocity while a homogeneous term can be added to exploit the redundancy of the robot to maximize a specified objective function. One drawback of these methods is that solutions suffer from high norms during transition into and out of singular configurations (*kinematic singularities*). Several approaches have been suggested to deal with kinematic singularities, i.e. using different kinds of Damped-Least-Square (DLS) inverse of the Jacobian matrices [12, 13]. Considerations made in [13] highlight that DLS solutions with variable damping factor and numerical filtering represent a good compromise between accurate tracking performance of the pseudoinverse solution and the capability of providing feasible joint velocities of the DLS inverse operators. However, the maximum damping factor is often required to be tuned manually and finding an optimal condition for the damping activation is not trivial.

Task-priority redundancy resolution techniques have been proposed to allow the specification of a primary task that is fulfilled with higher priority with respect to a secondary task. In [4, 5, 6] desired trajectories for both primary and secondary tasks are specified leading to a more

efficient solution for the two-tasks problem. The main drawback of this technique is that new singularities, named *algorithmic singularities*, may arise in configurations in which the two tasks are not compatible; this situation can even happen if the Jacobians of both tasks have still full rank. Similarly to the case of kinematic singularities, ill-conditioned and discontinuous joint velocity solutions may be encountered in the proximity of an algorithmic singularity, but can, on the other hand, be treated in a similar manner using DLS operators. Extension of the redundancy resolution scheme at velocity level to  $N$  generic tasks has been first proposed by Siciliano and Slotine [7] in an attractive recursive way, which is also affected from algorithmic singularities as it is an extension of [4].

## 3 Description of the implemented multi-task, velocity-based redundancy resolution

A general, hierarchical velocity-based redundancy resolution framework has been implemented on basis of [4, 7] that gives the user the possibility to specify tasks with priorities and deals with the execution of these. The prioritized set of tasks is depicted in Fig. 3. Tasks of high priority are usually those related to safety features, e.g., obstacle and joint limit avoidance, whereas the task of following a fully specified 6D end effector trajectory is basically one of the lower-prioritized tasks. Nevertheless, it is very important to follow a specified end effector trajectory as accurately as possible, especially in industrial settings where a process has to fulfil certain quality measures.

For the sake of simplicity, consider initially three tasks denoted with the subscript 1, 2 and 3 respectively, fully specified by the task functions  $\sigma_1(t), \sigma_2(t)$  and  $\sigma_3(t)$  as well as the first-order differential relations between task functions and robot joint vector  $q$ :

$$\begin{aligned} \sigma_1(t) &= f_1(q(t)) \in \mathbb{R}^{m_1}, & \dot{\sigma}_1(t) &= J_1(q(t))\dot{q}(t), \\ \sigma_2(t) &= f_2(q(t)) \in \mathbb{R}^{m_2}, & \dot{\sigma}_2(t) &= J_2(q(t))\dot{q}(t), \\ \sigma_3(t) &= f_3(q(t)) \in \mathbb{R}^{m_3}, & \dot{\sigma}_3(t) &= J_3(q(t))\dot{q}(t). \end{aligned} \quad (1)$$

The vector  $q(t) \in \mathbb{R}^n$  describes the robot joint configurations over time, whereas the matrices  $J_1 \in \mathbb{R}^{m_1 \times n}$ ,  $J_2 \in \mathbb{R}^{m_2 \times n}$  and  $J_3 \in \mathbb{R}^{m_3 \times n}$  represent the task Jacobians of the three tasks. Furthermore, the corresponding null space projectors for the first two tasks are given by:

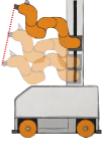
$$P_1 = (I_n - J_1^\dagger J_1), \quad P_2 = (I_n - J_2^\dagger J_2), \quad (2)$$

where  $\dagger$  is the right Moore-Penrose pseudoinverse operator and  $I_n$  is the  $(n \times n)$  identity matrix. Given desired task trajectories  $\dot{\sigma}_{1des}, \dot{\sigma}_{2des}, \dot{\sigma}_{3des}$ , the task-priority solution for the three tasks can thus be derived as:

$$\dot{q}_{des} = \dot{q}_1 + (J_2 P_1)^\# (\dot{\sigma}_{2des} - J_2 \dot{q}_1) + (J_3 P_2^\#)^\# (\dot{\sigma}_{3des} - J_3 \dot{q}_2), \quad (3)$$



**Figure 2** KUKA VALERI omniRob showing its extended reach (left) and in an aerospace industry mock-up (right).

Priority	Task	Associated aim
1	Obstacle Avoidance	 Do not injure people Avoid damages to both the robot structure and the environment
2	Joint Limits Avoidance	 Generate feasible joint space solution
3	End Effector Trajectory	 Execute the industrial assignment

**Figure 3** Set of prioritized task.

where

$$\begin{aligned} \dot{q}_1 &= J_1^\dagger \dot{\sigma}_{1_{des}} \\ \dot{q}_2 &= \dot{q}_1 + (J_2 P_1)^\dagger (\dot{\sigma}_{2_{des}} - J_2 \dot{q}_1) \\ P_2^A &= I_n - (J_2^A)^\dagger J_2^A \end{aligned} \quad (4)$$

and  $J_2^A = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}$  being the augmented task Jacobian of the combined tasks 1 and 2. The null space projector  $P_2^A$  defined as in (4) ensures that the solution for task 3 is projected onto the null space of both tasks 1 and 2 simultaneously, so that it cannot affect any task of higher priority. This way of computing the null space projector is known as *augmented* projection [21]. A generalization to  $N$  generic tasks of the formulation can be written in a

recursive fashion as:

$$\begin{cases} \dot{q}_0 = 0 \\ \dot{q}_i = \dot{q}_{i-1} + (J_i P_{i-1}^A)^\dagger (\dot{\sigma}_{i_{des}} - J_i \dot{q}_{i-1}) & i = 1, \dots, N \\ \dot{q}_{des} = \dot{q}_N. \end{cases} \quad (5)$$

Also the computation of the augmented null space projectors can be carried out in a recursive way as follows:

$$\begin{cases} P_0^A = I_n \\ P_i^A = P_{i-1}^A - \bar{J}_i^\dagger \bar{J}_i & i = 1, \dots, N, \end{cases} \quad (6)$$

$$\text{where } \bar{J}_i = J_i P_{i-1}^A.$$

In order to avoid the well-known problem of numerical drift of the solution in real, discrete-time based implementations on the robot controller, a CLIK action has been integrated into the task-priority formulation. This integration can simply be carried out by replacing the generic desired task velocity  $\dot{\sigma}_{i_{des}}$  with the reference one defined as:

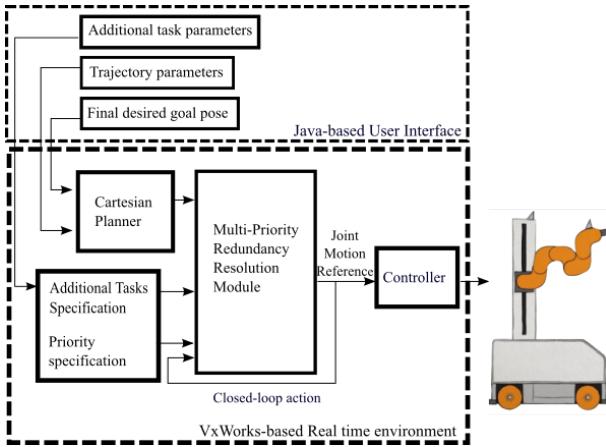
$$\dot{\sigma}_{i_{ref}} = \dot{\sigma}_{i_{des}} + \Gamma_i \tilde{\sigma}_i, \quad (7)$$

where  $\Gamma_i$  is a positive-definite gain matrix and  $\tilde{\sigma}_i$  is the  $i$ -th task error, generally defined as  $\tilde{\sigma}_i = \sigma_{i_{des}} - \sigma_i$ . The gain matrix  $\Gamma_i$  is usually defined as a scalar matrix with  $\Gamma_i = \gamma_i I_{m_i}$  and  $m_i$  being the dimension of  $\sigma_i$ .

To deal with the two kind of singularities, namely kinematic and algorithmic singularities that can occur in this formulation, Damped-Least-Squares (DLS) inverse operators are used. More specifically, variable damping and numerical filtering techniques have been implemented on basis of [13], but extending the filtering action to all singular values of the matrices that have to be pseudoinverted. As already mentioned in the previous section, the main drawback of this approach is that no optimal way for selecting the maximum damping factors and the activation threshold for DLS action are known so far. Therefore, they have to be chosen empirically.

The implemented module represents a low-level planner for reactive control: based on the desired end effector

Cartesian trajectory and different kind of available information like distance to joint limits and laser scan information for detecting unknown obstacles, the redundancy resolution module is able to compute the joint space trajectory that the manipulator has to follow in order to achieve the prioritized list of tasks. The computation happens online, allowing the robot to dynamically react to new and unpredictable situations. The module also represents an unified way of motion planner for a mobile manipulator since the joint space planning automatically involves all joints and each part of the robotic system is treated as a part of the same kinematic chain.



**Figure 4** Architecture on KUKA Sunrise real-time controller

## 4 Experiments in simulation and on our mobile manipulator

Experiments have been carried out in a Matlab-based simulation environment prior to executing the same on our VALERI omniRob. The following sections describe the tasks that can be considered in our framework and how they are implemented. Ordered with decreasing priorities these tasks are obstacle avoidance using laser scanner data mounted on the mobile platform, joint limit avoidance and the task of following a Cartesian end effector trajectory as precisely as possible.

The redundancy resolution scheme has been successfully integrated within the KUKA hard real-time Sunrise framework. This means it has been compiled for the real-time kernel running directly on the robot controller in a fixed 1ms cycle time. The implemented architecture can be seen in Fig. 4: a Cartesian planner computes a trajectory connecting an initial pose of the end effector with the final desired Cartesian pose defined by the user. The user can also set parameters for the specification of additional tasks from a java-based programming interface before the application starts; a redundancy resolution module is then used as low-level reactive planner to compute new joint space input for the robot according to the specification

of the prioritized tasks in each millisecond. Finally, a model-based controller takes care of making the robot follow that calculated joint motion reference.

### 4.1 Obstacle avoidance using laser scanners of a mobile platform

An obstacle avoidance task has been implemented by extending the approach in [22] to our case of planar safety laser scanners. The VALERI omniRob has two proximity laser scanners enabling a 360° monitoring around the mobile platform with a resolution of 0.5°. The kinematic chain also including the mobile platform (as described in Tab. 1) is supposed to be surrounded by virtual springs, installed in different points of the structure in a distributed way. Each spring is attached to a proximity sensor and has a certain amount of pseudo-energy, according to the minimum distance to obstacles measured by the corresponding sensor. The pseudo-energy resulting from the  $k$ -th sensor acting on the kinematic chain is defined by the following continuous function:

$$\epsilon_k(q) = \begin{cases} \frac{1}{2}(d_k - r_k)^2 & \text{if } d_k < r_k, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where  $d_k$  is the distance information obtained by the  $k$ -th sensor mounted on the kinematic chain and  $r_k$  is a threshold to decide when measurements should have effect and be considered in a whole-body reaction (rest length of  $k$ -th virtual spring). It is worth noting that the total pseudo-energy will be zero only in case no obstacles are in volume around the manipulator monitored by the sensors. Therefore the desired value for the task function will always be zero over the time. The total energy of all springs used as task function for obstacle avoidance is then

$$\sigma_{OA} = f_{OA}(q) = \sum_{k=1}^{N_S} \epsilon_k(q). \quad (9)$$

The task Jacobian can be computed in a closed form with the gradient of the energy function regarding the kinematic model of the mobile robot resulting in

$$J_{OA} = \sum_{k=1}^{N_S} \frac{\partial \epsilon_k(q)}{\partial q} \quad (10)$$

with

$$\frac{\partial \epsilon_k(q)}{\partial q} = \begin{cases} -(d_k - r_k)v_{d_k}^T \frac{\partial p_{s_k}(q)}{\partial q} & \text{if } d_k < r_k, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

The quantity  $v_{d_k}(q) = (p_{o_k} - p_{s_k}(q)) / \| p_{o_k} - p_{s_k}(q) \|$  is the normalized vector connecting the  $k$ -th sensor to the closest obstacle point, where  $p_{o_k}$  is the position of the obstacle point measured by the  $k$ -th sensor and  $p_{s_k}$  is the position of the  $k$ -th sensor on the robot structure itself. In the case of obstacle avoidance with laser scanners mounted on

a mobile platform where the first two prismatic joints of the mobile base coincide with the  $x$  and  $y$  direction of the world frame and measurements are considered regarding the center point of the base, the task of obstacle avoidance automatically becomes to keeping the center point of the base away from the surrounding obstacles. Then, each laser beam can be treated as a proximity sensor mounted in the center of the base and  $\frac{\partial p_{sk}(q)}{\partial q}$  results in being a constant quantity equal to

$$\frac{\partial p_{sk}(q)}{\partial q} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \quad (12)$$

## 4.2 Joint limit avoidance

An individual task function has been defined for each joint as follows. For the  $i$ -th joint, the corresponding task function  $\sigma_{JL_i}$  is:

$$\sigma_{JL_i} = f_{JL_i}(q) = c_i(q_i) \quad (13)$$

$$c_i(q_i) = \begin{cases} \alpha(e^{\beta(q_i - q_{i_m})^2} - 1) & \text{if } q_i < q_{i_m} \\ 0 & \text{if } q_{i_m} \leq q_i \leq q_{i_M} \\ \alpha(e^{\beta(q_i - q_{i_M})^2} - 1) & \text{if } q_i > q_{i_M} \end{cases}$$

namely, it is zero when the  $i$ -th joint value is in a certain safety range  $[q_{i_m}, q_{i_M}]$  defined by soft joint limits - chosen far enough from the mechanical limits - and increase exponentially outside of this zone. The desired value for this function should therefore be always zero over the time. The specification in (13) also presents two positive scalars –  $\alpha$  and  $\beta$  – influencing the shape of the task function. Fig. 5 shows a principal shape of the task function for joint limit avoidance.

The first order relation defines then the task Jacobian for a single joint as

$$\dot{\sigma}_{JL_i} = J_{JL_i}(q)\dot{q} = \frac{dc_i(q)}{\partial q_i}\dot{q}. \quad (14)$$

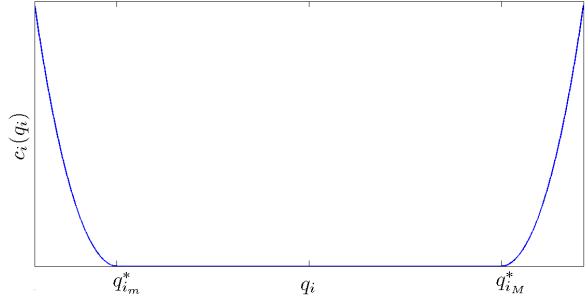
The task Jacobian will be zero when the  $i$ -th joint is inside its safety zone, whereas the  $i$ -th element of this row matrix increases exponentially as the  $i$ -th joint exceeds the range of  $[q_{i_m}, q_{i_M}]$ . Table 1 shows the set of safety limits that have been chosen for our VALERI omniRob.

The desired task velocities from all  $n$  single joints of the mobile manipulator are finally combined into one joint limit avoiding task with  $\dot{\sigma}_{JL} = [\dot{\sigma}_{JL_1}, \dots, \dot{\sigma}_{JL_n}]^T$ .

## 4.3 Cartesian end effector tasks

For the task of following a predefined Cartesian path, the task function  $\sigma_{EE}$  is representing an up to 6-dimensional end effector trajectory, whereas the task Jacobian  $J_{EE}$  is the well-known end effector Jacobian of the manipulator computed for a specific joint configuration by using the Denavit-Hartenberg description of the kinematic structure as proposed in table 1.

This section presents two experiments that have been conducted in simulation and on our physical robot to show the capability of the implemented redundancy resolution scheme.



**Figure 5** Principal shape of the task function for joint limit avoidance with  $q_{i_m}^*$  as lower and  $q_{i_M}^*$  as upper soft limit.

### 4.3.1 Experiment 1: Fixed end effector position with joint limit avoidance

In the first experiment the VALERI omniRob has to fix its end effector at the given Cartesian position (secondary task) while the joint limit avoidance task (primary task) is actually violated. In fact, the initial configuration of the manipulator is such that the 5-th joint exceeds its upper soft limit, which only for this simulation has been set to 0.7m in order to emphasize the effectiveness of the joint limit avoidance action:  $q(0) = q_0 = [0 \ 0 \ 0 \ -1.2 \ 0.92 \ 0.9 \ -1.2 \ -1.2 \ -1.1 \ 0.9 \ -1 \ 0.2]^T$ . Comparing the starting value of the 5-th joint with the soft limit of 0.7m, there is the necessity to move 0.22m down to be inside its safety zone.

The end effector task as a secondary task consists in keeping the end effector in the pose corresponding to the initial configuration  $q_0$  which is

$$\begin{aligned} \sigma_{EE}(0) &= [p_0 \ \phi_0]^T \\ p_0 &= [0.4821 \ -0.7405 \ 1.063]^T \\ \phi_0 &= [2.4824 \ -0.2883 \ -3.120]^T. \end{aligned}$$

The task-priority equation for the two tasks can therefore be written as

$$\dot{q}_{des} = J_{JL}^{\dagger\lambda} \dot{\sigma}_{ref JL} + (J_{EE} P_{JL})^{\dagger\lambda} (\dot{\sigma}_{ref EE} - J_{EE} J_{JL}^{\dagger\lambda} \dot{\sigma}_{ref JL}),$$

where the two reference task velocities  $\dot{\sigma}_{ref JL}$  and  $\dot{\sigma}_{ref EE}$  are implemented as closed-loop actions based on the actual task errors  $\tilde{\sigma}_{JL}$  and  $\tilde{\sigma}_{EE}$  generally described in eq. 7:

$$\begin{aligned} \dot{\sigma}_{ref JL} &= \dot{\sigma}_{JL_{des}} + \Gamma_{JL} \tilde{\sigma}_{JL} \\ &= \dot{\sigma}_{JL_{des}} + \gamma_{JL} I_{m_{JL}} \tilde{\sigma}_{JL} \end{aligned}$$

with the task error for joint limit avoidance being  $\tilde{\sigma}_{JL} = \sigma_{JL_{des}} - \sigma_{JL}$  and

$$\begin{aligned} \dot{\sigma}_{ref EE} &= \dot{\sigma}_{EE_{des}} + \Gamma_{EE} \tilde{\sigma}_{EE} \\ &= \dot{\sigma}_{EE_{des}} + \gamma_{EE} I_{m_{EE}} \tilde{\sigma}_{EE} \end{aligned}$$

with the task error for the end effector task computed using position and orientation errors. The dimensions of the two tasks are denoted with  $m_{JL}$  and  $m_{EE}$ , the gains of the

Joint	a[m]	$\alpha[\text{rad}]$	d[m]	$\theta[\text{rad}]$	Mech. Limit-	Mech. Limit+	Soft Limit- ( $q_{lm}^*$ )	Soft Limit+ ( $q_{lM}^*$ )
1	0	$-\pi/2$	$q_1$	0	No Limit	No Limit	-Inf [m]	Inf [m]
2	0	$\pi/2$	$q_2$	$-\pi/2$	No Limit	No Limit	-Inf [m]	Inf [m]
3	0.264	0	0.805	$q_3$	No Limit	No Limit	-Inf [rad]	Inf [rad]
4	0.151	0	0	$q_4$	$-5\pi/6$ [rad]	$2\pi/3$ [rad]	$-7\pi/9$ [rad]	$11\pi/18$ [rad]
5	0	$\pi/2$	$q_5$	$\pi/2$	0.05 [m]	0.95 [m]	0.15 [m]	0.85 [m]
6	0	$-\pi/2$	0.31	$q_6$	$-17\pi/18$ [rad]	$17\pi/18$ [rad]	$-8\pi/9$ [rad]	$8\pi/9$ [rad]
7	0	$\pi/2$	0	$q_7$	$-2\pi/3$ [rad]	$2\pi/3$ [rad]	$-11\pi/18$ [rad]	$11\pi/18$ [rad]
8	0	$\pi/2$	0.4	$q_8$	$-17\pi/18$ [rad]	$17\pi/18$ [rad]	$-8\pi/9$ [rad]	$8\pi/9$ [rad]
9	0	$-\pi/2$	0	$q_9$	$-2\pi/3$ [rad]	$2\pi/3$ [rad]	$-11\pi/18$ [rad]	$11\pi/18$ [rad]
10	0	$-\pi/2$	0.39	$q_{10}$	$-17\pi/18$ [rad]	$17\pi/18$ [rad]	$-8\pi/9$ [rad]	$8\pi/9$ [rad]
11	0	$\pi/2$	0	$q_{11}$	$-2\pi/3$ [rad]	$2\pi/3$ [rad]	$-11\pi/18$ [rad]	$11\pi/18$ [rad]
12	0	0	0	$q_{12}$	$-35\pi/36$ [rad]	$35\pi/36$ [rad]	$-11\pi/12$ [rad]	$11\pi/12$ [rad]
Flange	0	0	0.0785	0	-	-	-	-

**Table 1** DH parameters, mechanical joint limits as well as soft joint limits used for the KUKA VALERI omniRob.

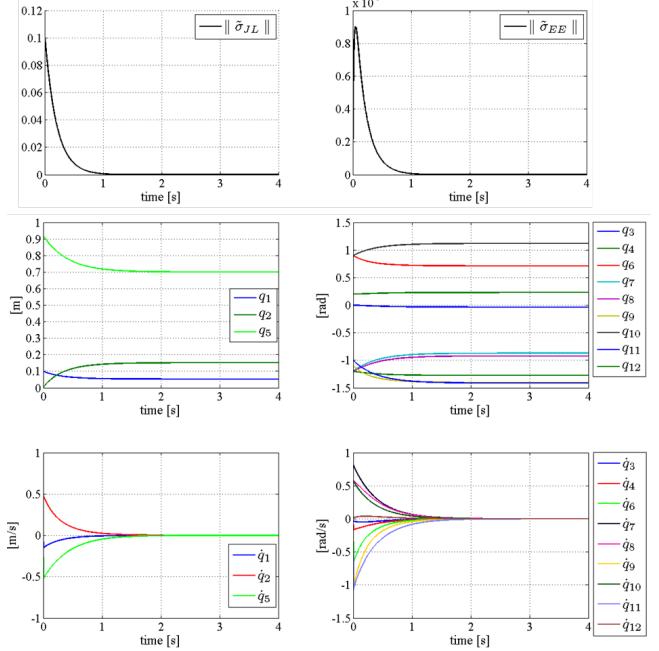
closed-loop actions has been set to  $\gamma_{JL} = 5$  and  $\gamma_{EE} = 50$ , respectively.

The joint motion reference (position and velocity) and a sequence of configurations assumed by the robot are reported in Fig. 6 and 7 respectively. Motion of the prismatic joints is illustrated on the left-side graphs, whereas the motion of the revolute joints can be observed on the right. From the evolution of the variable  $q_5$ , it can be noted how the 5-th joint – the additional linear axis – moves down until it reaches 0.70m while the redundancy of the unified mobile manipulator is exploited to keep the end effector in its initial pose. The evolution of the task errors  $\tilde{\sigma}_{JL}$  and  $\tilde{\sigma}_{EE}$  over the time is plotted in Fig. 6: the task error of the primary task is reported on the left, while the error of the secondary task is plotted on the right. It can be seen that both task errors exponentially tend to zero and that the end effector task error from a algorithmic point of view is in the magnitude of  $10^{-5}$  which is very important for quality demands of industrial processes.

#### 4.3.2 Experiment 2: Moving end effector along a straight line while avoiding unknown obstacles and joint limits

In this second experiment, an obstacle avoidance task has been inserted as primary task with highest priority. As in the previous experiment there is a joint limit avoiding task, but now as a lower, secondary task and a straight line following task as Cartesian end effector trajectory task as a third one with lowest priority. The task setup of the second experiment is actually represented in Fig. 3. So instead of just holding a defined Cartesian position with the end effector, the end effector has now a more complex task of following our desired Cartesian straight line. The resulting redundancy resolution scheme on joint velocity level with the three tasks of unknown obstacle avoidance  $OA$ , joint limit avoidance  $JL$  and end effector trajectory following  $EE$  corresponds to

$$\dot{q}_{des} = \dot{q}_{OA} + \dot{q}_{JL} + \dot{q}_{EE},$$

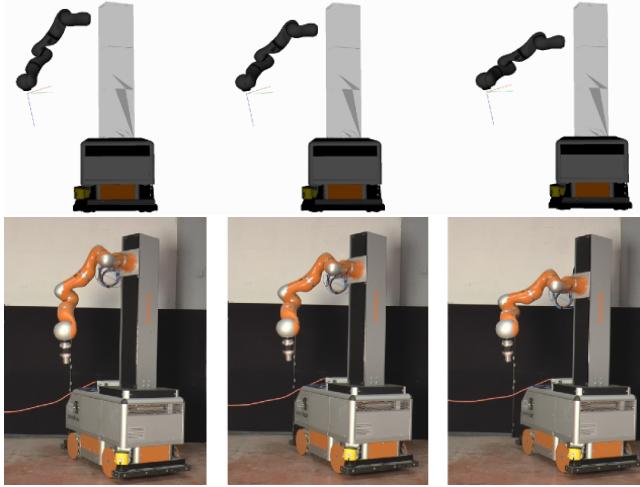


**Figure 6** Task errors (top) and joint motion reference  $q$  (middle) and  $\dot{q}$  (bottom) for experiment 1.

where the three terms of task velocities  $\dot{q}_{OA}$ ,  $\dot{q}_{JL}$  and  $\dot{q}_{EE}$  with the their appropriate null space projectors can be further detailed in form of the general  $N$  task formulation from eq. 5 and eq. 6 into

$$\begin{aligned}\dot{q}_{OA} &= J_{OA}^{\dagger\lambda} \dot{\sigma}_{ref_{OA}} \\ \dot{q}_{JL} &= (J_{JL} P_{OA})^{\dagger\lambda} (\dot{\sigma}_{ref_{JL}} - J_{JL} \dot{q}_{OA}) \\ \dot{q}_{EE} &= (J_{EE} P_{JL}^A)^{\dagger\lambda} [\dot{\sigma}_{ref_{EE}} - J_{EE}(\dot{q}_{OA} + \dot{q}_{JL})].\end{aligned}$$

The initial configuration of the robot for this experiment is  $q(0) = [0 \ 0 \ 0 \ -1.2 \ 0.5 \ 0.9 \ -1.2 \ -1.2 \ -1.1 \ 0.9 \ -$



**Figure 7** Sequence of joint configurations for experiment 1: simulation (top) and on the real robot (bottom)

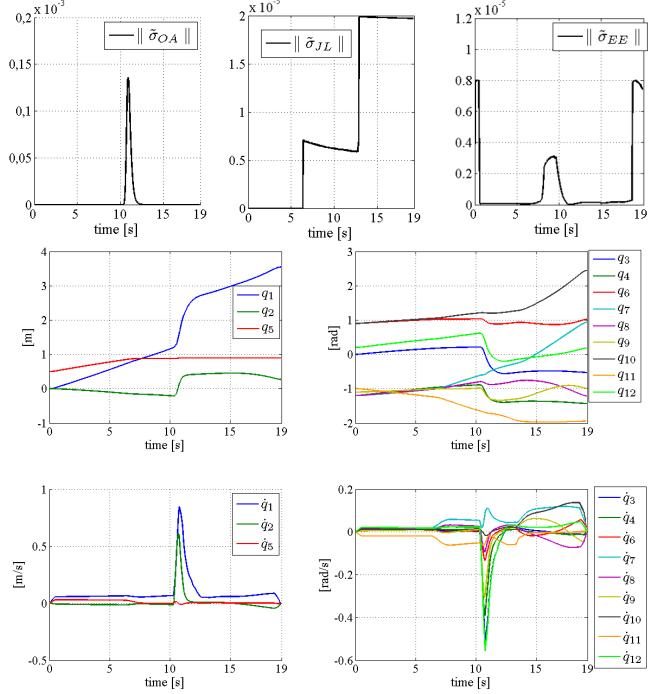
$1 \ 0.2]^T$ , corresponding to the initial pose

$$\begin{aligned}\sigma_{EE}(0) &= [p_0 \ \phi_0]^T \\ p_0 &= [0.4821 \ -0.7405 \ 0.6429]^T \\ \phi_0 &= [2.4824 \ -0.2883 \ -3.120]^T.\end{aligned}$$

For the additional obstacle avoidance task, all the activation thresholds  $r_k$  have been set to 0.8m and the gain for the closed-loop action of obstacle avoidance is set to  $\gamma_{OA} = 25$ . The soft limits in Tab.1 have been used for the joint limit avoidance task, with a gain  $\gamma_{JL} = 5$  as in experiment 1. Finally, the end-effector task consists in reaching the final pose

$$\begin{aligned}\sigma_{desEE}(t_{final}) &= [p_f \ \phi_f]^T \\ p_f &= [3.9821 \ -0.7405 \ 1.3429]^T \\ \phi_f &= [2.4824 \ -0.2883 \ -3.120]^T.\end{aligned}$$

along a planned straight line trajectory with trapezoidal velocity profile. The closed-loop action gain is  $\gamma_{EE} = 50$ . Fig. 8 shows the obtained task errors and joint motion reference. The pseudo-energy of the virtual springs associated to the obstacle avoidance task (top left) increases as soon as an obstacle enters in the circular field of radius  $r_k$  surrounding the mobile platform, and decreases progressively as the platform starts to react moving around the obstacle. The joint limit avoidance task generates first a contribution to prevent the 5-th joint and then the 11-th joint to exceed their upper and lower soft limit respectively. From the graph in the middle of the figure it can be noted that both joints reach their respective soft limits without violating them. Finally, the graph at top right shows an accuracy of  $10^{-5}$  in following the end effector trajectory, which is again really important for industrial assignments.

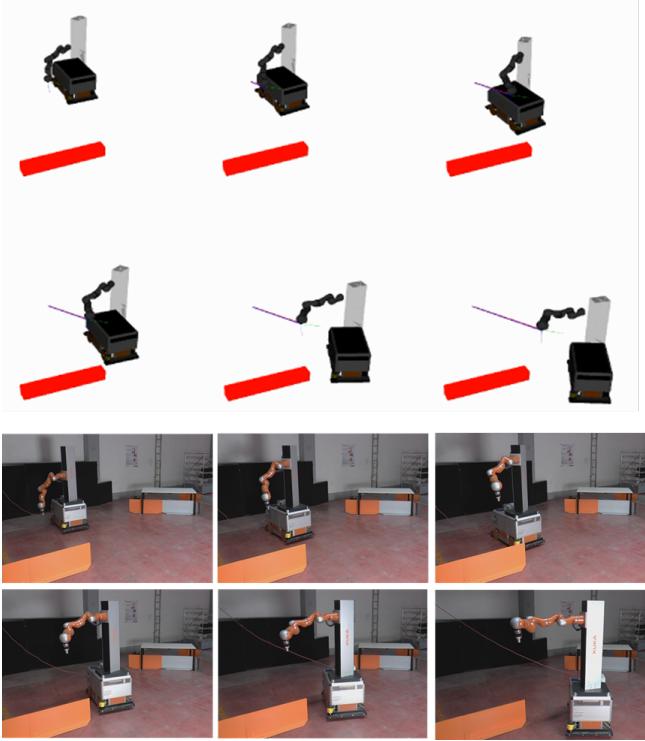


**Figure 8** Task errors (top) and joint motion reference  $q$  (middle) and  $\dot{q}$  (bottom) for experiment 2.

## 5 Conclusion

We have implemented an effective velocity-based redundancy resolution approach for mobile manipulators with many degrees of freedom such as our KUKA VALERI omniRob that preserves strict hierarchy among different prioritized tasks. This is especially important for industrial use-cases where robustness, safety issues as well as a high accuracy on the Cartesian task are of major concern. We have considered tasks like unknown obstacle avoidance with onboard safety laser scanners mounted on the mobile platform, joint limit avoidance and various Cartesian end effector tasks. In the future we would also like to integrate self-collision checking as another safety-relevant task. Both experiments have been successfully performed in our Matlab-based simulation environment as well as directly on our VALERI robot running with a Sunrise-based real-time controller in a hard one millisecond cycle rate. Task errors on all levels of priority are of very small magnitude even on the lowest priority which is usually the Cartesian end effector task. So our implemented approach is suitable for industrial processes where we have a high demand for process quality.

In the future we will consider general, spline-based representations to fully specify 6-dimensional Cartesian trajectories. Empirical accuracy measurements performed with our VALERI robot using a motion tracking system could also be undertaken. Finally, comparing the results of the implemented velocity-based formulation to higher order formulations such as acceleration-based and torque-based methods should also kept in mind as future



**Figure 9** Sequence of whole-body robot motion for experiment 2 of moving the end effector along a straight line while avoiding obstacles: simulation (top) and on the real robot (bottom).

work.

## 6 Literature

- [1] Validation of Advanced, Collaborative Robotics for Industrial Applications (VALERI project), Public Deliverables, <http://www.valeri-project.eu/results/public-deliverables/>.
- [2] D. Whitney, "Resolved motion rate control of manipulators and human prostheses", IEEE Trans. on Man-Machine Systems, vol. MMS-10, no. 2, pp. 47–52, 1969.
- [3] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms" IEEE Trans. on Sys., Man, and Cybern., vol. SMC-7, no. 12, pp. 868–871, 1977.
- [4] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot with redundancy," Preprints IFAC, 8th Triennial World Congress, vol. 14, pp. 78–83, 1981.
- [5] A. Maciejewski and C. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments", Internat. Journ. of Robot. Research vol. 4, no. 3, pp. 109-117, 1985.
- [6] Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators", Internat. Journ. of Robot. Research vol. 6, no. 2, pp. 3-15, 1987.
- [7] B. Siciliano and J. J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems", 5th Internat. Confer. on Advan. Robot., 1991.
- [8] G. Antonelli, F. Arrichiello and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems", Intel. Serv. Robot. vol. 1, no. 1, pp. 27-39, 2008.
- [9] G. Antonelli, "Stability Analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems", IEEE Trans. on Robot. vol. 25, no. 5, pp. 985-994, 2009.
- [10] L. Sciavicco and B. Siciliano, "Coordinate transformation: a solution algorithm for one class of robots," IEEE Trans. on Sys., Man, and Cybernet., vol. SMC-16, no. 4, pp. 550–559, 1986.
- [11] P. Falco and C. Natale, "On the stability of closed-loop inverse kinematics algorithms for redundant robots," IEEE Trans. on Robotics, vol. 27, no. 4, pp. 780–784, 2011.
- [12] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," ASME Journ. of Dyn. Sys., Measurem. and Contr., vol. 108, pp. 163–171, 1986.
- [13] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for realtime kinematic control of robot manipulators," IEEE Trans. on Rob. and Autom., vol. 13, no. 3, pp. 398–410, 1997.
- [14] Y. Nakamura and H. Hanafusa, „Optimal redundancy control of robot manipulators“, International Journal of Robotics Research vol. 6, no. 1, pp. 32-42, 1990.
- [15] Antonelli, Gianluca and Arrichiello, Filippo and Chiaverini, Stefano, "The null-space-based behav-ioral control for autonomous robotic systems", Intel-ligent Service Robotics vol. 1, no. 1, pp. 27-39, 2008.
- [16] G. Antonelli, „Stability Analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems“, IEEE Transactions on Robot-ics vol. 25, no. 5, pp. 985-994, 2009.
- [17] B. Siciliano and J. J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," 5th International Conference on Advanced Robotics, 1991.
- [18] A. D. Luca, G. Oriolo and B. Siciliano, "Robot redundancy resolution at the acceleration level", Laborat. Robot. and Automat., vol. 4, pp. 97–106, 1992.
- [19] O. Khatib, "A unified approach for motion and force control of robot manipulators: the operational space formulation," IEEE Journ. of Robot. and Automat., vol. RA3, no. 1, pp. 45–53, 1987.
- [20] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, Operational space control: a theoretical and empirical comparison, The Internat. Journ. of Robot. Research, vol. 27, no. 6, pp. 737757, 2008.

- [21] A. Dietrich, C. Ott, and A. Schäffer, “An overview of null space projections for redundant, torque-controlled robots”, *The Internat. Journ. of Robot.Research*, pp. 1–16, 2015.
- [22] P. Falco and C. Natale, “Low-level flexible planning for mobile manipulators: a distributed perception approach”, *Advanced Robotics*, vol. 28, no. 21, pp. 1431–1444, 2014.