

# TELEOPERATING A MOBILE MANIPULATOR USING A UAV CAMERA WITHOUT ROBOT SELF-OCCLUSIONS

Josep-Arnau Claret

Institute of Industrial and Control Engineering, BarcelonaTECH, josep.arnau.claret@upc.edu

Luis Basañez

Institute of Industrial and Control Engineering, BarcelonaTECH, luis.basanez@upc.edu

## Abstract

The paper presents a novel teleoperation system that allows the simultaneous command of a mobile manipulator and a free flying camera, implemented using a UAV, from which the operator can monitor the task execution in real-time. A novel use of the kinematic redundancy is presented to prevent the robot parts from occluding the end-effector to the operator view. Following an obstacle avoidance approach, an input is fed to the null space that keeps the robot links away from the end-effector in the image plane as a secondary task. Simulations and the implementation in a real setup show the goodness of the proposed approach.

**Keywords:** Mobile manipulation, UAV, teleoperation, redundancy, obstacle avoidance.

## 1 INTRODUCTION

Teleoperation has been one of the main robotics fields since its inception [6, 11]. One of the key aspects during a teleoperation task is the quality of the view the operator has of the scene [12, 2]. In particular, one aspect often ignored, but which can greatly affect the teleoperation of a robot, is the occlusion of the object at the TCP of the robot that is being commanded by the operator.

Occlusions of the mobile manipulator (MM) TCP can be due to obstacles in the scenario, but also to the robot itself, that is, its links and the platform: the different robot parts can be in a configuration such that they are located between the camera and the TCP, making the visibility of the commanded object more difficult, thus increasing the workload on the user, and degrading the overall performance of the teleoperation.

Redundancy allows robots to accomplish the main task and fulfill secondary tasks. The null space is a mean to achieve this. It has been used for several secondary tasks: multiple end-effector control [1], singularity [10] and object collision avoidance [8], joint constraints [9], manipulability optimization [13], emotion conveyance to users [3], etc.

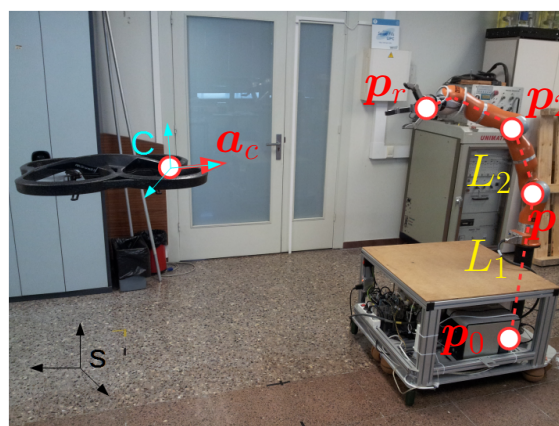


Figure 1: The BMM-I and the free-flying camera.

### 1.1 CONTRIBUTION

With the idea to free the user from the explicit teleoperation of the robot so that he/she can focus on the object being teleoperated, a need arises to improve as much as possible the view of the object. In order to do so, this paper introduces, to the authors knowledge, the first use of the robot null space to prevent the object attached to the TCP from being occluded by its own body and links.

The presented method, the *Object Best View Function* (OBVF), has been validated in simulations and implemented in a real setup. The solution is presented in a teleoperation system previously presented by the authors [4] consisting of the BMM-I (a MM with an omnidirectional platform and a 7 Degrees of Freedom -DOFs- manipulator), an Unmanned Aerial Vehicle (UAV) and a haptic device (Fig. 1). The operator commands the robot TCP using the haptic while getting visual feedback from the scenario through a camera mounted on the UAV, which allows the camera to track the MM without an explicit operator command, preventing the robot from leaving the camera field of view, and easing the teleoperation. The results show the goodness of the proposed solution.

Section 2 presents the proposed solution; Sect. 3 and 4, the simulations and experimentation. Section 5 contains the conclusions and future work.

## 2 PROPOSED SOLUTION

One of the leading aspects to address the aforementioned problem is its similarity with the obstacle avoidance problem, which goal is to move the robot from an initial to a final configuration in a scenario filled with objects with which the robot can collide. The challenge is to find a feasible trajectory or direction of movement that guarantees that the path or the next step is free of collisions.

The force field formulation [7] assigns to each object a potential field that generates a repulsive force between the object and the robot. A resultant force is the result of the influence of all the objects in the scene, which can be input to the robot through its null space and execute the avoidance task with low priority. The main difference with the classical obstacle avoidance problem is that in the OBVF the resultant force generated by the obstacles (the parts of the robot) has to be applied to the robot link itself, since the robot point that is intended to stay away from the obstacles is the robot TCP, which already has a trajectory to track as the higher priority task.

The proposed solution thus follows the next ideas:

- To use the MM null space to avoid robot self-occlusions as a secondary task.
- To pose the problem as a collision avoidance task between the robot TCP and particular parts of the robot.
- To evaluate the proximity between the TCP and the robot links *projected in the image plane*. By measuring the distance in the image plane between the projected TCP and the links of the robot, the operator point of view is naturally taken into account.

To address the OBVF, an *escape* velocity (the repulsive force in the force field formulation) is generated to further be applied to the point of the robot where the axis of joints 3 and 4 intersect. The desired velocity input is designed to contain the information of the distance of all the links that may occlude the robot TCP and give more importance to the closest links because those are the links more likely to occlude the TCP.

Consider the points  $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$  and  $\mathbf{p}_r$  of the mobile manipulator, expressed in the world reference,  $s$ , such that each point  $\mathbf{p}_i = [p_{i_x} \ p_{i_y} \ p_{i_z}]^T \in \mathbb{R}^3$ ; and the homogeneous transformation matrix  $\mathbf{T}_s^c \in SE(3)$  from the world frame to the camera frame,  $c$ , with the origin  $\mathbf{p}_s^c \in \mathbb{R}^3$  and the orientation described by the rotation matrix  $\mathbf{R}_s^c \in SO(3)$ . Let  $\mathbf{a}_c$  denote the unit vector over the optical axis of the camera pointing towards the robot (Fig. 1).

The OBVF is activated when any  $\mathbf{p}_1$  or  $\mathbf{p}_2$ , without its Z coordinate, is in front of  $\mathbf{p}_r$  seen from the camera, that is, the coordinate system  $c$  origin. Let  $\mathbf{p}_{i_{XY}} = [p_{i_x} \ p_{i_y}]^T$  and  $\mathbf{a}_{c_{XY}} = [a_{c_x} \ a_{c_y}]^T \in \mathbb{R}^2$ , then  $\mathbf{p}_i$  will be in front of  $\mathbf{p}_r$  if

$$s_i = (\mathbf{p}_{i_{XY}} - \mathbf{p}_{r_{XY}})^T \mathbf{a}_{c_{XY}} > 0. \tag{1}$$

Note that the task activation and deactivation can lead to discontinuities; this is addressed below.

When the OBVF is activated, the next step is to change the points reference frame from the world to the camera:  $\mathbf{r}_i = (\mathbf{T}_s^c)^{-1} \mathbf{p}_i$  and  $\mathbf{r}_r = (\mathbf{T}_s^c)^{-1} \mathbf{p}_r$ . Let  $c2ph(\mathbf{r}, f_c)$  be a function that projects a point in the camera frame  $\mathbf{r} = [r_x \ r_y \ r_z]^T \in \mathbb{R}^3$  to a point  $\mathbf{c} \in \mathbb{R}^2$  in the image plane, considering a *pinhole camera model* as

$$\mathbf{c} = c2ph(\mathbf{r}, f_c) = \frac{f_c}{r_z} [r_x \ r_y]^T$$

where  $f_c$  is the focal length of the camera.

Let's also define the function  $ph2c(\mathbf{c}, z, f_c)$  that converts a projected point  $\mathbf{c} = [c_x \ c_y]^T \in \mathbb{R}^2$  in the image plane at the distance  $f_c$  into its 3D *unprojected* point  $\mathbf{r} \in \mathbb{R}^3$  at a distance  $z$  with

$$\mathbf{r} = ph2c(\mathbf{c}, z, f_c) = \frac{z}{f_c} [c_x \ c_y \ 1]^T.$$

The next step is to compute the force exerted on each link of the mobile manipulator. The considered links can be seen in Fig. 1 as  $L_1$  and  $L_2$ .  $L_1$  is the link defined by the points  $\mathbf{p}_0$  and  $\mathbf{p}_1$ , while  $L_2$  is the link defined by the points  $\mathbf{p}_1$  and  $\mathbf{p}_2$ .

Our interest is to compute the point  $\mathbf{r}_i^* \in \mathbb{R}^3$ , in every link  $i$ , such that its projection in the image plane  $\mathbf{c}_i^*$  is closest to the projection of the robot TCP also in the image plane,  $\mathbf{c}_r$ . Given  $\lambda_i \in [0, 1]$  any point  $\mathbf{r}_{L_i}$  in the link  $L_i$  can be expressed as

$$\mathbf{r}_{L_i} = \mathbf{r}_{i-1} + \lambda_i (\mathbf{r}_i - \mathbf{r}_{i-1}) = \mathbf{r}_{i-1} + \lambda_i \mathbf{n}_i \tag{2}$$

with  $i \in \{1, 2\}$ .

This is equivalent to solve:

$$\lambda_i^* = \arg \min_{\lambda_i} \|c2ph(\mathbf{r}_{i-1} + \lambda_i \mathbf{n}_i, f_c) - c2ph(\mathbf{r}_r, f_c)\| \tag{3}$$

Eq. 3 is also equivalent to obtain the value  $\lambda_i^*$  that minimizes the previous squared distance, which can be obtained analytically (see Appendix). The solutions  $\lambda_1^*$  and  $\lambda_2^*$  of the minimization problems must lie in  $[0, 1]$ . Otherwise, points in the same line but outside the link would be taken into account. Once  $\lambda_1^*$  and  $\lambda_2^*$  are obtained, the corresponding point in the link can be determined using Eq. 2:  $\mathbf{r}_i^* = \mathbf{r}_{i-1} + \lambda_i^* \mathbf{n}_i$ .

In order to determine the repulsive force  $f_i$  from the robot TCP to each link both its modulus and its direction need to be computed. The force direction can be expressed as the unit vector  $u_i = (p_i^* - p_r) / \|p_i^* - p_r\|$  with  $p_i^* = T_s^c r_i^*$ . Although the force direction points from the TCP to the point in the link, it will ultimately be the link which will be moved away from the TCP.

The modulus of the force,  $k_i$ , has to be inversely proportional to the distance between the TCP and the link in the image plane, so as to give the teleoperator a sense of coherency between the exerted force and the proximity of the object occlusion. The distance between the TCP and the link in the image plane is straightforward to compute:

$$d_i = \|c2ph(r_i, f_c) - c2ph(r_r, f_c)\|.$$

In order to compute the modulus of the force, a maximum value  $k_i^F$  of the force can be set when the distance is smaller than a certain value,  $d_i^m$ . For distances greater than  $d_i^M$  the modulus can be set to zero, *i.e.*,  $k_i = 0$ . The interval  $[d_i^m, d_i^M]$  is selected such that a smoothing transition for the force modulus between  $[0, k_i^F]$  can be implemented.  $k_i$  can be written similar as in [9]:

$$k_i = \begin{cases} 0 & \text{if } d_i > d_i^M \\ k_i^F f_s \left( \frac{d_i^M - d_i^m}{d_i^M - d_i}, \frac{d_i^M - d_i^m}{d_i - d_i^m} \right) & \text{if } d_i \in [d_i^m, d_i^M] \\ k_i^F & \text{else} \end{cases} \quad (4)$$

$$f_s(x, y) = \frac{1}{2} \left( 1 + \tanh \left( \frac{1}{2}(x - y) \right) \right).$$

Finally the force for each link  $L_i$  can be computed as  $f_i = k_i u_i$ , which can be interpreted as the velocity imposed to the corresponding point of the link which prevents it from getting close to the TCP in the image plane. Fig. 2 graphically depicts the elements needed to compute each  $f_i$ . The final force  $f$  is the sum of  $f_1$  and  $f_2$ . This ensures the continuity and smoothness of the force vector as both  $f_i$  are continuous and calculated using Eq. 4.

As stated above, the modulus of the force needs to change in a continuous way also w.r.t. the OBVF activation/deactivation. Otherwise, in a situation where the robot tip would be close to a link in the image plane, but still in front of it, if, due to the user command, the tip of the robot got behind a link (making a  $s_i > 0$  in Eq. 1) the force would suddenly trigger a discontinuity in the joints.

To address this, a threshold  $s_F$  has been defined for Eq. 1. This threshold sets up a transition zone where  $f$  continuously changes from zero to its appropriate value. This strategy can be implemented using a parameter  $k_F$  for determining

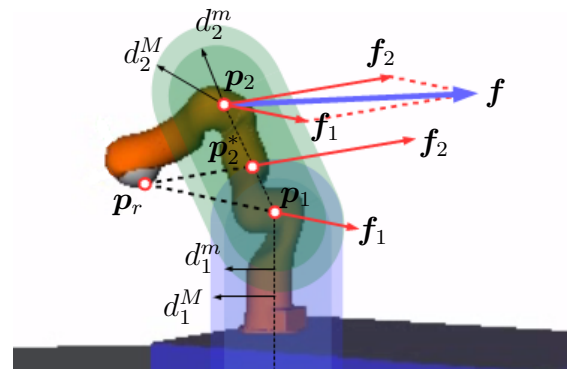


Figure 2: Exerted forces on the links by the proximity of the TCP. The TCP lies outside the area where the forces  $f_1$  or  $f_2$  are generated, which has been presented this way for ease of visualization.

$f = k_F(f_1 + f_2)$ , with

$$k_{F_i} = \begin{cases} 0 & \text{if } s_i > 0 \\ f_s \left( \frac{-s_F}{s_F - s_i}, \frac{s_F}{s_i} \right) & \text{if } s_i \in [-s_F, 0] \\ 1 & \text{else} \end{cases} \quad (5)$$

and by finally setting  $k_F = \max(k_{F_1}, k_{F_2})$ .

The computed  $f$  can now to be fed to the null space. But the application point needs to be selected. In order to solve the problem of having to switch between several points, which would produce discontinuities and lead to instabilities during the teleoperation task, the application point has been set to  $p_2$ . Moreover, discontinuities on its derivative are also avoided with this choice.

Once the point to apply the  $f$  has been chosen, the robot Jacobian at this point  $J_{p_2}$  can be computed.

It would be tempting to obtain the joint velocities to feed the null space algorithm by trying to find the appropriate  $\dot{q}$  that solves the task

$$f = J_{p_2} \dot{q}.$$

But note that, since this algorithm attempts to increase the distance between the end-effector and the links in the image plane, it doesn't matter what happens to the distance in the Cartesian space, provided that, when projected in the image plane, it ultimately increases. This means that it is not needed that the closest link of the robot from the end-effector follows the direction pointed by the force  $f$ , which implies a three-DOFs task. It is only needed that it does so in the image plane, thus imposing a two-DOFs task.

For this reason, the Jacobian and the force need to be transformed into the camera reference frame, and only keep the XY directions. Projection ma-

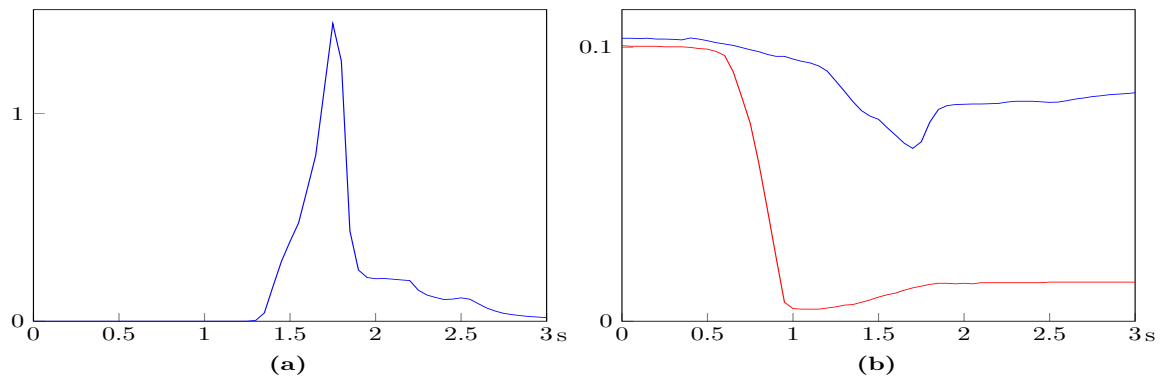


Figure 3: OBVF simulation 1: a) modulus of the force  $\mathbf{f}$  when the OBVF is active; b) minimum distances between  $d_1$  and  $d_2$ , *i.e.*,  $\min(d_1, d_2)$ , with the OBVF active (blue) and not active (red).

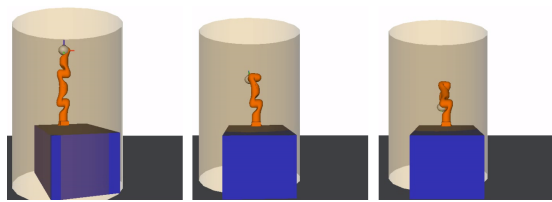


Figure 4: Simulation 1 without activation. Images follows from left to right.

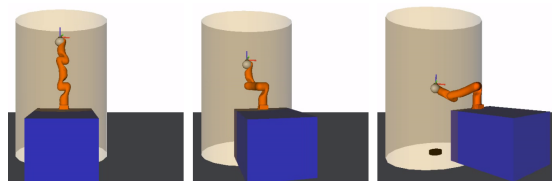


Figure 5: Simulation 1 with activation. Images follows from left to right.

trices can be defined for this purpose as

$$\mathbf{P}_{XY}^J = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{P}_{XY}^F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and the OBVF  $\dot{\mathbf{q}}$  can be obtained from

$$\mathbf{f}_c = \mathbf{P}_{XY}^F (\mathbf{R}_s^c)^T \mathbf{f} = \mathbf{P}_{XY}^J \mathbf{M} \mathbf{J}_{p_2} \dot{\mathbf{q}} = \mathbf{J}_c \dot{\mathbf{q}}$$

with  $\mathbf{f}_c$  being  $\mathbf{f}$  projected in the image plane;  $\mathbf{J}_c$ , the Jacobian of point  $p_2$  in the image plane; and

$$\mathbf{M} = \begin{bmatrix} (\mathbf{R}_s^c)^T & \mathbf{0} \\ \mathbf{0} & (\mathbf{R}_s^c)^T \end{bmatrix}$$

a matrix that rotates the vectors of the Jacobian so that they are referenced in the camera frame.

Now the task can be fed as a lower priority task:

$$\dot{\mathbf{q}} = \mathbf{J}_1^+ \dot{\mathbf{x}}_1 + (\mathbf{J}_c \mathbf{N}_1)^+ (\mathbf{f}_c - \mathbf{J}_c \mathbf{J}_1^+ \dot{\mathbf{x}}_1).$$

with task 1 as the end-effector pose tracking.

$f_c$	$d_1^m$	$d_1^M$	$k_{F_1}$	$d_2^m$	$d_2^M$	$k_{F_2}$	$s_F$
1.0	0.10	0.15	1.5	0.05	0.10	1.0	0.1

Table 1: Simulation parameters.

### 3 SIMULATION RESULTS

A set of simulations has been run to assess the proposed algorithm. Table 1 shows the simulation values.  $d_1^m$ ,  $d_1^M$ ,  $d_2^m$  and  $d_2^M$  are distances in m: higher values would imply that the effect of the TCP on the links would be exerted from further distances.  $k_{F_1}$  and  $k_{F_2}$  have been chosen after experimentation so that the effect of the TCP on the links would be smooth, and  $s_F = 0.1$  corresponds to an angle of  $84^\circ$ , which means that the OBVF activation / deactivation zone corresponds to  $6^\circ$ .

The mobile manipulator used in this work, the Barcelona Mobile Manipulator 1 (BMM-I), is an omnidirectional mobile platform with a 7 DOFs industrial serial robot [5] (Fig. 1). The platform has 3 DOFs: 2 independent translations in the plane and 1 rotation around its vertical axis and so the system has a total number of 10 DOFs.

In the envisioned teleoperation system, a Parrot AR.Drone is used as the free flying camera (Fig. 1, left side). In order to evaluate the OBVF, an ideal scenario has been set: a UAV perfectly hovering, mimicked by a fixed camera, thus isolating the performance of the OBVF from the UAV motions and easing its evaluation.

The OBVF has been implemented as a secondary task through the null space of the first task. The first task is the command of the TCP by the operator using a haptic device as explained in [4].

#### 3.1 SIMULATION 1

This simulation consists in moving the BMM-I from the vertical configuration to a configuration

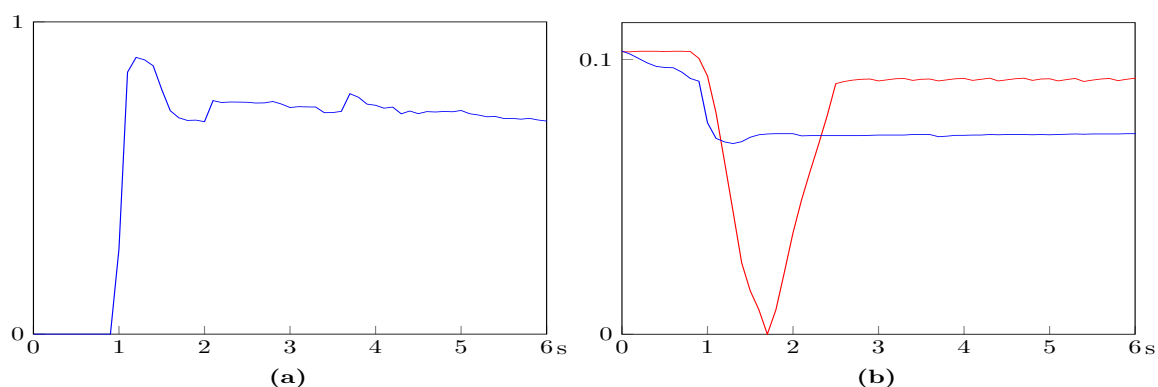


Figure 6: OBVF simulation 2: a) modulus of the force  $f$  when the OBVF is active; b) minimum of the distances  $d_1$  and  $d_2$  with the OBVF active (blue) and not active (red).

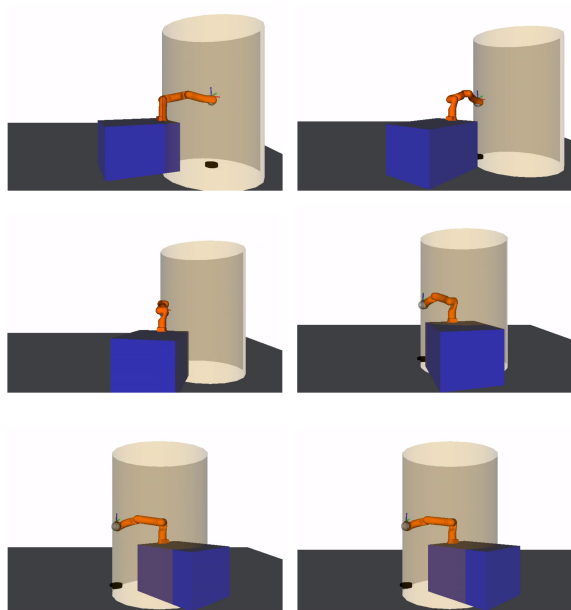


Figure 7: Simulation 2 without activation. Images follows from left to right and top to bottom.

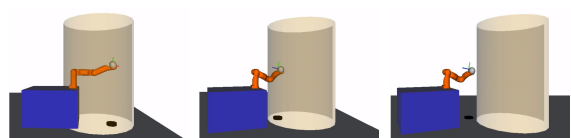


Figure 8: Simulation 2 with activation. Images follows from left to right.

with the arm extended horizontally and such that the TCP be hidden behind the robot from the operator perspective. Figs. 4 show the motion without the OBVF activated, and it can be clearly seen how the TCP is occluded by the robot. Figs. 5 show the same motion with the OBVF activated: the BMM-I does a self-motion such that it prevents the occlusion to occur.

Fig. 3a shows the modulus of  $f$  exerted during the motion, and Fig. 3b, the minimum distances  $d_1$  and  $d_2$  when the OBVF is active and not.  $\|f\|$  can be seen to increase when the distance between the TCP and the links decreases, peaking at the lowest distance (blue line), and stabilizing. It can also be seen how the distance remains far from zero when the OBVF is active, while it becomes almost zero otherwise.

Link with the simulation video: Simulation 1.

### 3.2 SIMULATION 2

In this simulation, the operator commands the TCP to move from right to left parallel to the image plane and such that, at the beginning of the motion, the TCP is at the back of the platform w.r.t. the camera. This required motion is solved by the inverse kinematics by first executing a general rotation motion around the Z axis of the platform to locate the TCP at the front of the motion, as if it was to pull the platform to the left side of the scenario. The particular motion of this simulation has been set such that the rotation moves the TCP behind the robot becoming occluded by the robot. To prevent the mentioned occlusion, the OBVF has been activated.

Figs. 7 and 8 show the motion of the simulation without and with the OBVF activated. When the OBVF is not active, an occlusion occurs during the teleoperation (Fig. 7, third image), which does not happen when the task is active.

Fig. 6b shows the minimum distances  $d_1$  and  $d_2$  when the OBVF is active. It is clear that the minimum distance becomes zero when the function is not active (Figs. 7, third image), which prevents the user from clearly seeing the object. When the function is active, the distance decreases until the force exerted by the OBVF is strong enough to prevent the TCP to pass behind the robot links.

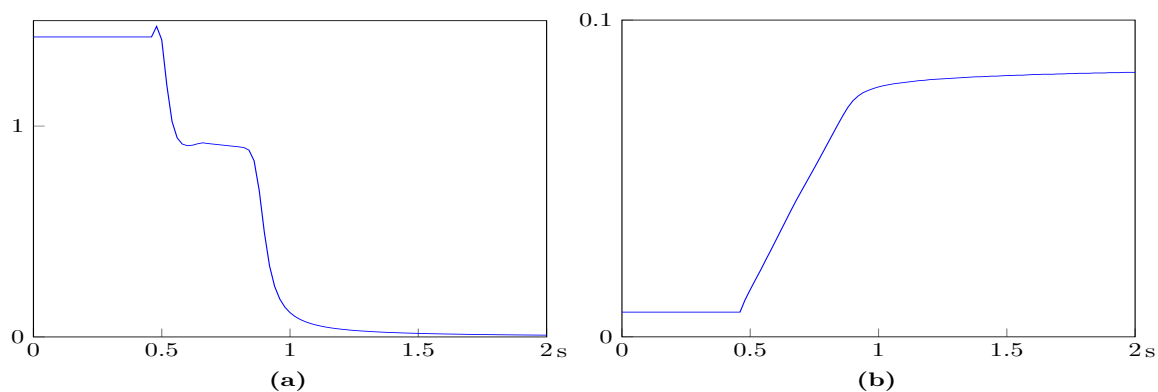


Figure 9: OBVF simulation 3: a) modulus of the force  $f$ ; b) minimum of the distances  $d_1$  and  $d_2$ .

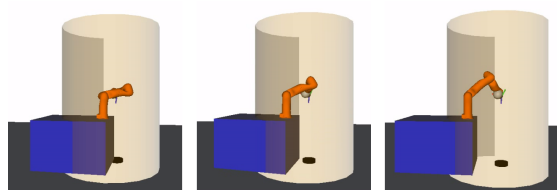


Figure 10: Simulation 3. Images follows from left to right.

Fig. 6a shows the modulus of  $f$ , which increases at the beginning as soon as the distance starts to diminish, peaks and stabilizes.

Differently from simulation 1, the force exerted does not become zero after it peaks (Fig. 6a) because the commanded trajectory is such that its TCP is always pushed against its links, so that the force  $f$  is never allowed to diminish. On the contrary, in simulation 1 the desired TCP trajectory is such that the OBVF exerts a force that moves the robot links fully away from the TCP, making then force  $f$  equal to zero.

Link with the simulation video: Simulation 2.

### 3.3 SIMULATION 3

This simulation consists in the *escape* of the BMM-I from a configuration in which, from the operator point of view, the TCP is very close to the links (Figs. 10, left image). Note that the links 2 and 3 are much closer to the TCP in the image plane than the first link. Suddenly, the OBVF gets activated at time 0.5 s. Being the scenario such that the links 2 and 3 are occluding the robot TCP, it is expected that the influence of the force  $f_2$  will be stronger than  $f_1$ , and so the robot joints will rotate to move the links 2 and 3 away from the TCP. Figs. 10 show the evolution of the robot configuration once the OBVF is activated, showing the expected behavior and freeing the TCP

from the occlusion.

Figs. 9 show the modulus of the exerted force  $f$  and the minimum distance between the TCP and the robot links, which increases until 0.08 m.

Note the shape of the force in the left image in Figs. 9: at the beginning, the  $f$  is 1.5 N, but since the OBVF is not active until the 0.5 s, it does not generate a motion on the robot nor an increase of the distance between the TCP and the links. Once the OBVF is active, the distance starts to increase and the force, to decrease. At this point, a high negative slope of the force is followed by a constant force at around 0.9 N during 0.3 s, followed by a high negative slope until the force becomes almost null. This *chair* shape is explained by considering that at the beginning  $f$  contains both the influence of  $f_1$  and  $f_2$  and that, since the TCP is very close to the link 2, the contribution of  $f_2$  is maximum, *i.e.*,  $f_2 = k_2 u_2$ . Once the OBVF gets active the force starts to move the links 1 and 2 away from the TCP. During the first high negative slope,  $f_2$  is constant, since the distance between the TCP and the link 2 is very small ( $d_2 < d_2^m$  in Eq. 5), but  $f_1$  is decreasing ( $d_1 \in [d_1^m, d_1^M]$ ). Once  $f_1$  becomes null ( $d_1 > d_1^M$ ), the only contribution to  $f$  is  $f_2$  so that for 0.3 s the force modulus is constant and the distance between the TCP and the robot links increases (in particular with link 2). Finally,  $d_2$  becomes such that  $f_2$  starts to diminish ( $d_2 \in [d_2^m, d_2^M]$ ) and becomes null ( $d_2 > d_2^M$ ).

## 4 EXPERIMENTAL VALIDATION

Two experiments have been executed to test the OBVF in a real setup, corresponding to the Simulations 1 and 2. The videos can be found in OBVF simulation 1 and OBVF simulation 2. A snapshot of the first is shown in Fig. 11.

The videos are divided in four areas. The left

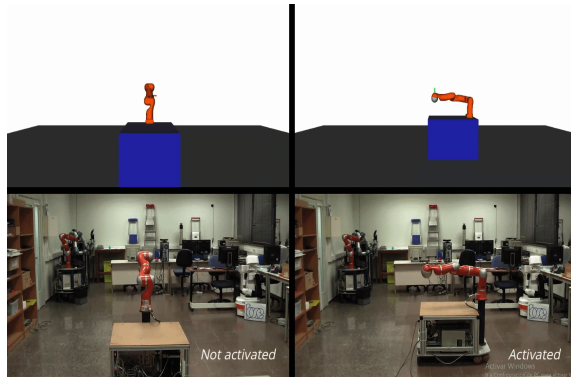


Figure 11: Snapshot of the simulation 1 video.

figures correspond to motions without the OBVF activation, and the right figures, with the activation. The top ones correspond to the simulated motions, and the bottom figures, in the real setup.

## 5 CONCLUSIONS

The OBVF has been presented, developed and tested in different scenarios. In all cases the proposed solution has prevented the TCP from being occluded by the robot parts. This work shows that it is possible to teleoperate a mobile manipulator while using the robot null space to prevent robot self-occlusions of the object, improving the operator view of the object, and thus the performance of the teleoperation.

Future work will include the generalization to mobile manipulators with any number of joints, and experiments with a free-flying camera.

## Appendix

Given a link  $L$  defined by two points  $\mathbf{r}^M, \mathbf{r}^m \in \mathbb{R}^3$  such that any point  $\mathbf{r}_L \in \mathbb{R}^3$  in the link is  $\mathbf{r}_L = \mathbf{r}^m + \lambda \mathbf{n}$  with  $\lambda \in [0, 1]$  and  $\mathbf{n} = \mathbf{r}^M - \mathbf{r}^m$ , and given a point  $\mathbf{r} \in \mathbb{R}^3$ , the aim is to find the  $\lambda^*$  that minimizes the distance  $D$  of the projection in the image plane between  $L$  and  $\mathbf{r}$  considering a pinhole camera model.

The solution can be analytically computed since  $\lambda^* = \arg \min_{\lambda} (D) = \arg \min_{\lambda} (D^2)$ . Differentiating  $D^2$  w.r.t.  $\lambda$ , and equating to zero,  $\lambda^* = A/B$ :

$$\begin{aligned}
 A &= (r_x^m)^2 r_z n_z - r_x^m r_z^m r_x n_z - r_x^m r_z^m r_z n_x \\
 &+ (r_y^m)^2 r_z n_z - r_y^m r_z^m r_y n_z - r_y^m r_z^m r_z n_y \\
 &+ (r_z^m)^2 r_x n_x + (r_z^m)^2 r_y n_y, \\
 B &= r_x^m r_x (n_z)^2 - r_x^m r_z n_x n_z + r_y^m r_y (n_z)^2 \\
 &- r_y^m r_z n_y n_z - r_z^m r_x n_x n_z - r_z^m r_y n_y n_z \\
 &+ r_z^m r_z (n_x)^2 + r_z^m r_z (n_y)^2.
 \end{aligned}$$

## Acknowledgement

\*This work was partially supported by the Spanish Government through the project DPI2016-80077-R.

## References

- [1] Baerlocher, P. and Boulic, R., (1998) Task-priority formulations for the kinematic control of highly redundant articulated structures, *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol 1, pp 323-329.
- [2] Chen, J. Y. C., Haas, E. C. and Barnes, M. J., (2007) Human Performance Issues and User Interface Design for Teleoperated Robots, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol 37, num 6, pp 1231-1245.
- [3] Claret, J. A., Venture, G., and Basañez, L., (2017) Exploiting the Robot Kinematic Redundancy for Emotion Conveyance to Humans as a Lower Priority Task, *International Journal of Social Robotics*, vol 9, num 2, pp 277-292.
- [4] Claret, J. A., Zaplana, I. and Basañez, L., (2016) Teleoperating a mobile manipulator and a free-flying camera from a single haptic device, *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp 291-296.
- [5] Clos, D. and Martínez, J., (2007) Plataforma Mòbil amb Rodes Esfèriques per al Robot "Lightweight Robot 4" de Kuka Roboter, Institute of Industrial and Control Engineering - Technical University of Catalonia.
- [6] Goertz, R. C., (1952) A force reflecting positional servo mechanism, *Nucleonics*, vol 10, part II, pp 43-45.
- [7] Khatib, O., (1986) Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *The International Journal of Robotics Research*, vol 5, num 1, pp 90-98.
- [8] Maciejewski, A. and Klein C., (1985) Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments, *The International Journal of Robotics Research*, vol 4, num 3, pp 109-117.
- [9] Mansard, N., Khatib, O. and Kheddar, A., (2009) A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks, *IEEE Transactions on Robotics*, vol 25, num 3, pp 670-685.

- [10] Nemeč, B. and Zlajpah, L., (2000) Null Space Velocity Control with Dynamically Consistent Pseudo-inverse, *Robotica*, vol 18, num 5, pp 513-518.
- [11] Nuño, E. and Basañez, L., (2009) Nonlinear Bilateral Teleoperation: Stability Analysis, *2009 IEEE International Conference on Robotics and Automation*, pp 3718-3723.
- [12] Shiroma, N., Sato, N., Chiu, Y. and Matsuno, F., (2004) Study on effective camera images for mobile robot teleoperation, *ROMAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication*, pp 107-112.
- [13] Yoshikawa, T., (1985) Dynamic manipulability of robot manipulators, *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol 2, pp 1033-1038.



© 2019 by the authors.  
Submitted for possible  
open access publication  
under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).