XXXVIII Jornadas de Automática
https://doi.org/10.17979/spudc.9788497497749.0694

USING AN UAV TO GUIDE THE TELEOPERATION OF A MOBILE MANIPULATOR

Josep-Arnau Claret
Institute of Industrial and Control Engineering (IOC), UPC, Barcelona, Spain; josep.arnau.claret@upc.edu

Luis Basañez
Institute of Industrial and Control Engineering (IOC), UPC, Barcelona, Spain; luis.basanez@upc.edu

Abstract

In this paper, a new teleoperation system consisting on the integration of a mobile manipulator, an UAV, and a haptic is presented. The camera of the UAV is used to give visual feedback to the operator. An algorithm is presented to allow the operator to command both the UAV and the mobile manipulator while keeping the point of view pointing towards the robot by only using a single haptic device. The presented algorithm combines a position-position and a position-velocity workspace mapping from the haptic to the mobile manipulator and the UAV, both in position and orientation. Further, a solution is presented that takes advantage of the null space of the mobile manipulator to keep the body and the arm of the mobile manipulator from occluding its own TCP and the object it is carrying, thus easing the teleoperation task. Experimentation has been carried on the system, both in a virtual and a real scenario, showing its potential in teleoperation scenarios. Overall, a novel teleoperation system and the ongoing progress towards its implementation in real situations is presented in this work.

keywords: Teleoperation, mobile manipulator, UAVs, workspace mapping, redundancy, null space

1 INTRODUCTION

The appearance of robots has had a deep impact in modern industry, allowing for a unprecedented level of productivity and opening new possibilities. Since its beginnings, robots have been confined in closed industrial cells, mainly due to security constraints. Since then, there has been a strong interest in the robotic community to allow the robots to operate in unbounded environments. This interest has naturally led to the study of the properties of mobile manipulators.

Low-cost Unmanned Air Vehicles (UAV) have become very popular in recent years. The coordination of multiple UAVs [1] and UAV planning [2] are active areas of research. Remarkably, the combination of an UAV with a mounted camera opens a wide range of applications like exploration of disaster scenarios [3].

Unmanned Air Vehicles and mobile manipulator (MM) have been and are currently used in teleoperation tasks [4, 5]. The coordination with or without teleoperation of a ground mobile robots and UAVs can be found in the literature for multiple tasks [6, 7, 8]. Remarkably, to the author's knowledge, no work allows to command the camera of an UAV to simultaneously teleoperate a mobile manipulator.

Workspace mapping algorithms are a key component in teleoperation systems. They map the input commands of the operator, usually from a haptic device or a joystick, to the robot workspace. This mapping is usually done in the position dimension, using a mapping from the haptic position to the robot position (*position-position* mapping), a *position-linear velocity* mapping, or an hybrid mapping, and it has to account for the scaling and the unbound workspace for mobile robots [9, 10]. Dynamic point of view, which deals with the orientation, can also be found for robot teleoperation [11] and in the exploration of virtual environments [12].

The Jacobian null space has been widely used in the robotic community to execute multiple tasks in a robot with different levels of priority [13, 14]. Its use ranges from the most common tasks to avoid singularities and object collisions, to convey emotions to users [15].

The first contribution of this work is to present a novel position and orientation workspace mapping algorithm which allows an operator to command a mobile manipulator while simultaneously and continuously changing the point-of-view of an UAV with a mounted camera from which the operator obtains visual feed-back from the scene. Second, the Jacobian null space of the mobile manipulator is studied to avoid the robot to occlude its own TCP, thus easing the teleoperation task.

The rest of this paper is organized as follows. First, Section 2 presents the mobile manipulator used in this work. Section 3 introduces the workspace mapping algorithm. The algorithm to exploit the null space of the robot is presented in Section 4. Section 5 presents the advances in the implementation. Finally, the conclusions can be found in Section 6.

2 THE BARCELONA MOBILE MANIPULATOR

The robot used in this work is the Barcelona Mobile Manipulator (BMM). The BMM consists of an omnidirectional mobile platform with an arm manipulator (Fig. 1). It is composed by a main body of 138 kg and $1\times0.78\times0.708$ m^3. The platform has three degrees of freedom: two independent translations and a rotation around the vertical axis. The arm manipulator used is a Kuka LWR 4+. It has seven degrees of freedom. Overall, the BMM has ten degrees of freedom. EtherCat is used to control the wheels while Orocos is used for the high level control in a Linux Xenomai environment over a PC with four Intel Core i5 CPUs at 3.1GHz. For further details on the platform see [16].

As ultimately it is intended to implement the proposed approach to teleoperate the BMM while the camera of a Parrot AR.Drone is sending the video of the scene to the operator, the kinematic model of the BMM has been in this work. The first steps towards this goal can be found in [17].

Figura 1: The Barcelona Mobile Manipulator.

Inverse kinematics

The algorithm to command the TCP is presented following [18]:

$$\dot{q} = J^{+}\dot{x} = J^{+}\begin{bmatrix} \dot{p}^{s}_{si} + K_P\, e_p \\ L^{-1}\left(L^T \omega^{s}_{si} + K_o\, e_o\right) \end{bmatrix} \qquad (1)$$

where:

- $e_p = p^s_{si} - p^s_r$.
- $e_o = \frac{1}{2}\left[n_R(q)\times n_D + s_R(q)\times s_D + a_R(q)\times a_D\right]$; with n, s, and a the columns of $R = R^s_r$ and $D = R^s_{si}$.
- $L = -\frac{1}{2}\left[S(n_D)\,S(n_R) + S(s_D)\,S(s_R) + S(a_D)\,S(a_R)\right]$.

where $S(\cdot)$ is the cross product matrix.

Once q is obtained by integrating $\dot{q}$, $T^s_r(q)$ can be computed using the direct kinematics. This algorithm allows a real time tracking of the robot TCP, both in position and orientation.

The computation of T^s_{si} to obtain p^s_{si} and R^s_{si} is presented in the next Section.

3 THE TELEOPERATION SYSTEM

3.1 OVERVIEW

Multiples frames need to be defined in this work prior to the introduction of the workspace mapping algorithm (Fig. 2): *mo*, the master inertial frame, a.k.a., the frame of the origin of the workspace of the haptic; *mi*, the frame of the tip of the haptic device; *s*, the inertial frame at the remote side, a.k.a., the mobile manipulator workspace frame; *so*, the frame of the origin of the haptic workspace in the remote inertial frame; *si*, the frame of the haptic tip in the remote inertial frame; *c*, the frame attached to the free-flying camera; and *r*, the robot end-effector frame.

Given an homogeneous transformation matrix $T^j_i \in SE(3)$, composed of a rotation matrix and a translation, $R^j_i \in SO(3)$ and $p^j_i \in \Re^3$, respectively, which expresses the frame i w.r.t. the frame j, $T^j_i \in SE(3)$ can be used to represent a position vector expressed in a frame i into a new frame j.

During a teleoperation task, T^{mo}_{mi} contains the information of the operator command to the system through the haptic. p^{mo}_{mi} is scaled by $K_S = \mathrm{diag}(k_{S_X}, k_{S_Y}, k_{S_Z})$ as $p^{so}_{si} = K_S\, p^{mo}_{mi}$. The scales need to be adjusted separately depending on the X, Y or Z dimension. This is convenient because it allows the adjustment between the workspace of the haptic and the manipulator. T^{so}_{si} corresponds to the input of the user from the haptic in the MM workspace and is composed of a translation and a rotation, $K_S\, p^{mo}_{mi}$ and R^{mo}_{mi}, respectively.

Given that $T^s_{si} = T^s_{so}\, T^{so}_{si}$ it follows that:

$$\begin{aligned} p^s_{si} &= p^s_{so} + R^s_{so}\, K_S\, p^{mo}_{mi} \\ R^s_{si} &= R^s_{so}\, R^{mo}_{mi} \end{aligned} \qquad (2)$$

From Fig. 2 it can also be noted that:

$$T^s_c = T^s_{so}\, T^{so}_c \qquad (3)$$

where T^{so}_c is a constant transformation that couples the frames c and so.

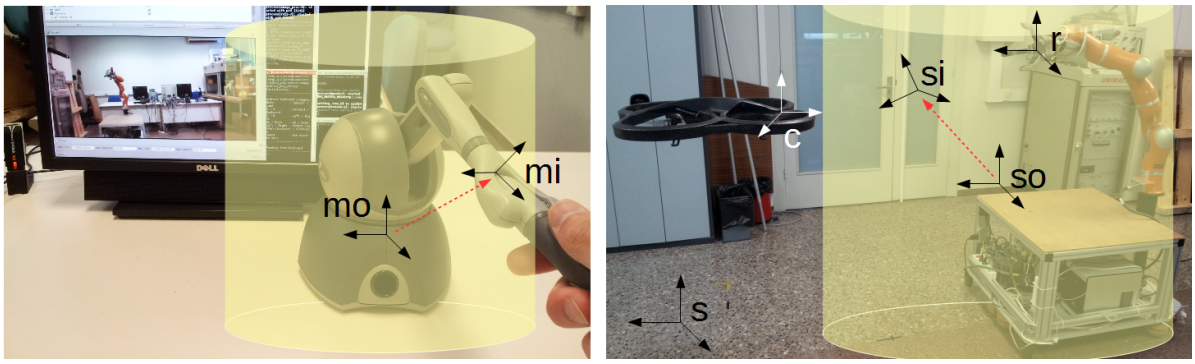

Figura 2: The teleoperation system. Left: The haptic and the screen with visual feedback from the UAV. Right: The haptic workspace (in yellow) is mapped into the workspace of the robot. Red arrows depict the user command.

Finally, T_r^s is obtained from the computation of the MM inverse kinematics (Sect. 2).

3.2 ADOPTED APPROACH

The proposed approach extends the solution presented in [19] to a mobile manipulator. But in this work the technique is also applied to the orientation by using a free-flying camera.

3.2.1 The position mapping

W.r.t. the position, when the user commands the tip of the haptic device *inside* the the mobile manipulator TCP is commanded using a *position-position* mapping between the haptic tip position and the robot TCP. When the boundary of the haptic workspace, roughly defined as a thick frontier containing the external workspace boundary, is reached, a *position-linear velocity* component is added to the previous *position-position*. This combination of inputs allows the user to command the TCP of the mobile manipulator in an unbounded workspace through the bounded workspace of the haptic device.

A "cylindric bubble" is used as haptic workspace to adapt the solution in [19] to the workspace of mobile manipulator (Figs. 2-3): the Z position is commanded by a *position-position* map (Eq. 2) and the *position-linear velocity* is activated in the XY plane when the distance from the haptic tip to the Z axis of the frame *mo*, D_V, is higher than a predefined radius R_V. Formally this can be expressed as:

$$\dot{p}_{so}^s = \left(1 - \frac{R_V}{D_V}\right) k_V \, u_{si_{XY}}^s = \\ = \left(1 - \frac{R_V}{D_V}\right) k_V \, R_{so}^s \, K_S \, u_{mi_{XY}}^{mo} \quad (4)$$

with $k_V > 0$, the maximum allowed velocity; $u_{mi_{XY}}^{mo}$, the unit vector of $p_{mi_{XY}}^{mo} = [p_{mi_X}^{mo}, p_{mi_Y}^{mo}, 0]^T$; and $D_V = \|p_{mi_{XY}}^{mo}\|$. The term $(1 - R_V/D_V)$ allows for increasing velocities from zero ($D_V = R_V$) to the maximum defined in k_V ($D_V \to \infty$). Thus, the farther away the haptic tip is from the vertical axis at its workspace, the fastest the MM will move in the same direction.

Differentiating, translation in Eq. 2 and Eq. 4 can be unified:

$$\dot{p}_{si}^s = \left(1 - \frac{R_V}{D_V}\right) K_V \, R_{so}^s \, K_S \, u_{mi_{XY}}^{mo} \, \chi_{[R_V,\infty)}(D_V) + \\ + R_{so}^s \, K_S \, \dot{p}_{mi}^{mo} \quad (5)$$

where $\chi_C(x)$ is the indicator function.

Additionally, in order for the operator to feel that he has entered in the *position-linear velocity* area, a force F can be applied to the user, with the force pointing towards the haptic workspace Z axis, and proportional to the distance from the Z axis:

$$F = -k_f^V \, R_{so}^s \, \left(K_S \, u_{mi_{XY}}^{mo} - K_d \, \dot{u}_{mi_{XY}}^{mo}\right) \chi_{[R_V,\infty)}(D_V)$$

with:

$$k_f^V = \left(1 - \frac{R_V}{D_V}\right) K_f$$

Note that a damping has been added in the previous expression. The purpose of this it to prevent the platform from moving if the user releases the haptic tip.

3.2.2 The orientation mapping

The free-flying-camera allows the operator to use the *position-linear velocity* mode on the orientation, enabling an *orientation-angular velocity* map.

The *orientation-angular velocity* control is activated when the haptic orientation surpasses a predefined boundary. This boundary can be defined using the ZYX intrinsic Euler angles as:

$$R_{mi}^{mo} = \begin{pmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ c_\beta s_\gamma & s_\alpha s_\beta s_\gamma + c_\alpha d_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{pmatrix} \quad (6)$$

with $s_a = \sin a$ and $c_a = \cos a$. The Euler angles can be retrieved as $\alpha = \text{atan2}(r_{32}, r_{33})$, $\beta = \text{atan2}\left(r_{31}, \sqrt{r_{11}{}^2 + r_{21}{}^2}\right)$ and $\gamma = \text{atan2}(r_{21}, r_{11})$.

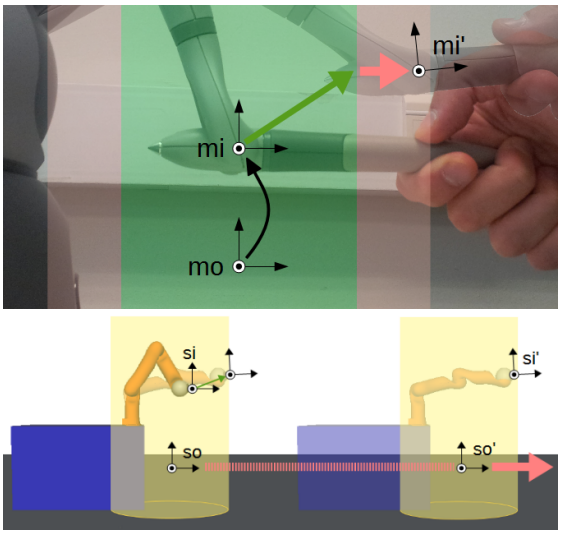

Figura 3: Position mapping. Top figure: in the green area the haptic position is mapped to a translation in the haptic workspace in the scenario (yellow volume in the bottom figure). Bottom figure: in the pink area (boundaries of the haptic workspace) the position is also mapped to the linear velocity and a gross motion on the platform is generated.

The *orientation-angular velocity* gets active when $|\beta| > \beta_B$ and $|\gamma| > \gamma_B$. Next, the frame R^s_{so} is rotated around $\hat{Z}$, the vertical axis, according to the rotation direction specified by the user. The angular velocity is proportional to the amount of penetration of the user into the *orientation-angular velocity* zone (Fig. 4, pink area) inside the workspace of the haptic.

By differentiating the equation in Eq. 2 the equation $\omega^s_{si} = \omega^s_{so} + R^s_{so}\ \omega^{so}_{si}$ is obtained, with each ω corresponding to an angular velocity. Then the *orientation-angular velocity* mode can be imposed with:

$$\omega^s_{so} = \text{sign}(\gamma)\ k_R \left(1 - \frac{\gamma_B}{|\gamma|}\right) \hat{Z}\ \chi_{[\gamma_B,\infty)}(\gamma)$$

From Eq. 3, T^s_c can be obtained by updating p^s_{so} and R^s_{so}.

3.3 SUMMARY

Summarizing the algorithm can be presented as:

1. $p^{mo}_{mi_{XY}} = [p^{mo}_{mi_X}, p^{mo}_{mi_Y}, 0]^T$
2. $D_V = \|p^{mo}_{mi_{XY}}\|$
3. if $D_V > R_V$:
   $p^s_{so} = p^s_{so} + \left(1 - \frac{R_V}{D_V}\right) k_V\ R^s_{so}\ K_S\ \frac{p^{mo}_{mi_{XY}}}{D_V}\ \Delta t$
4. $\beta = \text{atan2}\left(r_{31}, \sqrt{r_{11}{}^2 + r_{21}{}^2}\right)$
   $\gamma = \text{atan2}(r_{21}, r_{11})$

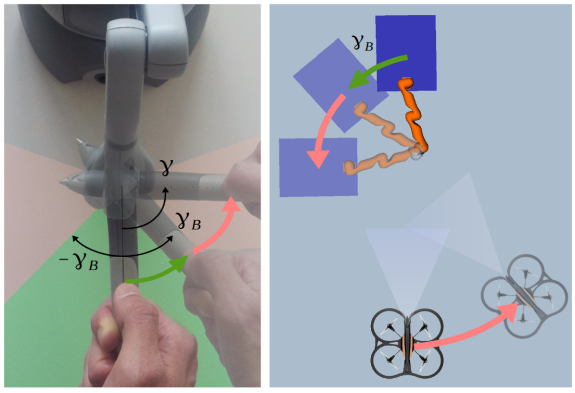

Figura 4: Rotation mapping. The haptic orientation around the Z axis in the green area is mapped to the camera orientation. In the pink area the orientation is also mapped to an angular velocity.

5. if $|\beta| > \beta_B$ and $|\gamma| > \gamma_B$:
   $R^s_{so} = \left[I + \text{sign}(\gamma)\ k_R \left(1 - \frac{\gamma_B}{|\gamma|}\right) S(\hat{Z})\ \Delta t\right] R^s_{so}$
6. $T^s_{si} = (p^s_{so} + R^s_{so}\ K_S\ p^{mo}_{mi}\ ,\ R^s_{so}\ R^{mo}_{mi})$
7. $T^s_c = T^s_{so}\ T^{so}_c$

4 EXPLOITING THE NULL SPACE

This particular configuration between a mobile manipulator, a haptic, and a UAV which gives visual feedback to the operator opens new possibilities.

For instance, the high degree of redundancy of the mobile manipulator can be exploited to improve the user experience and to ease the teleoperation task. The Jacobian null space is an elegant and well known framework to exploit the redundancy of a robotic system [13, 14].

In this work, the null space has been exploited to allow the operator to have a better view of the TCP of the robot and thus the object it is carrying. This is done by minimizing the chances that the robot body and arm intersect the line of sight of the operator.

With this approach, the first task of the robotic system is set to track the TCP and to rotate the camera as explained in the workspace mapping algorithm, while the robotic redundancy is exploited to keep the links and the platform of the robot away from the line that goes from the robot TCP to the focal point as a secondary task.

The two tasks stated above implemented in our robotic system can be formalised by extending the solution in Eq. 1 using the null space as:

$$\dot{q} = J^+ \dot{x} + N \dot{q}_0 \qquad (7)$$

with the null space projection operator $N = I - J^+ J$ and $\dot{q}_0$ containing the joint velocity that keeps the

robot away from occluding the TCP object from the user sight, as explained next.

4.1 THE OBJECT BEST VIEW APPROACH

To compute $\dot{q}_0$ it is necessary to define a function that carries the information of the object and the camera, and that outputs the robot motion required to maximize some object view quality measure. To do so, the object best view task can be implemented using the force field formulation, in a similar way as a joint limit avoidance task.

With the force field formulation, an *escape* velocity in the Cartesian space is generated to be applied to the point of the robot which position is between the object and the operator and which position in the image is closest to the object position in the image plane.

The main idea is to compute the distance *in the image* between the projected TCP and each of the links of the robot and the platform. With this approach, some advantage can be obtained by previously discarding those links that are behind the TCP, as they won't pose any problem for the operator task.

From the set of computed distances, the smallest is selected and, if it is lower than a predefined threshold, it is considered that the teleoperated object is occluded by the corresponding robot link. Next, the escape velocity is computed parallel to the image plane and moving the link away from the TCP.

Following, the described algorithm is presented in a more formal way.

1. For each link $i = 1,2,3$, the points p_i^H are computed where:
   - p_1^H is at the centre of the platform surface.
   - p_2^H is the intersection of the links 1 and 2.
   - p_3^H is the intersection of the links 3 and 4.

   A set P_H is then constructed: $P_H = p_1^H, p_2^H, p_3^H$.

2. The p_i^H that are not between the TCP and the image plane are removed from P_H, that is, if

$$(p_{1_{xy}}^H - p_{TCP_{xy}})^T n_{CAM_{xy}} > 0$$

   where:
   - a_{XY} is the projection of vector a in the XY-plane.
   - p_{TCP} is the position of the mobile manipulator TCP.
   - n_{CAM} is the normal vector of the image plane, pointing towards the TCP.

3. p_{TCP} is transformed to the camera image plane reference frame, as the remaining points in P_H, obtaining c_{TCP}, and c_i^H, respectively.

4. For each point c_i^H:
   (a) For each link the point c_i^L is computed, which corresponds to the projection of the previous point in the robot structure, projected in the image plane.
   (b) The point, c_i^*, in each link i which is closest to the TCP in the image plane is computed as $c_i^* = c_i^L + \lambda_i\, n_i$, with $n_i = c_i^H - c_i^L$, easily obtained with the λ_i that minimizes the squared distance

$$\left(\frac{c_{i_x}^L + \lambda_i\, n_{i_x}}{c_{i_z}^L + \lambda_i\, n_{i_z}} - \frac{v_{TCP_x}}{v_{TCP_z}}\right)^2 + \left(\frac{c_{i_y}^L + \lambda_i\, n_{i_y}}{c_{i_z}^L + \lambda_i\, n_{i_z}} - \frac{v_{TCP_y}}{v_{TCP_z}}\right)^2$$

   and with v_{TCP} as the vector from the image focal point to the TCP of the mobile manipulator, p_{TCP}.
   (c) If the distance $d_i = \|c_i^* - c_{TCP}\| < \varepsilon_i$ the point c_i^* is retained, as it is occluding the TCP, and, thus, the teleoperated object.

5. The c_i^* with the smallest d_i is selected, that is, $c_i^m = \min_d(c_i^*)$.

6. The *escape* velocity, v_{esc}, is computed as: v_{esc} is applied to p_i^*, the point corresponding to c_i^* in the world frame at the third priority level. Thus the intermediate Jacobian $J_2 = {}^{\partial c_i^*}/_{\partial q}$ is computed in position as the linear interpolation (interpolating with λ_i) of the Jacobians of p_{i-1}^H and p_i^H, and with the orientation part of the Jacobian of p_{i-1}^H.

Finally, $\dot{q}_0$ can be computed as $\dot{q}_0 = J_2^+\, v_{esc}$, and fed into Eq. 7.

4.2 IMPLEMENTATION

Considering the implementation of the presented robotic system, a Sensable Phantom Omni has been used as the haptic device. The UAV used has been the Parrot AR.Drone. The main algorithm, that is, the workspace mapping and the computation of the null space inputs, has been computed in an Intel Core i7 at 2.80GHz with Debian OS. The software has been implemented in C++.

The ROS framework has been used to integrate the different components of the system, mainly the haptic, the UAV, the BMM, and the workspace mapping. The nodes containing the haptic drivers and the workspace mapping, and the node to command the UAV have been executed in the main computer.

The input obtained from the haptic node is fed to the workspace mapping algorithm. The components obtained from this node are then processed in a manager node that generates the input signals to the free-flying camera, the Parrot AR-Drone, and to the mobile manipulator, the BMM, divided in to the commands of the

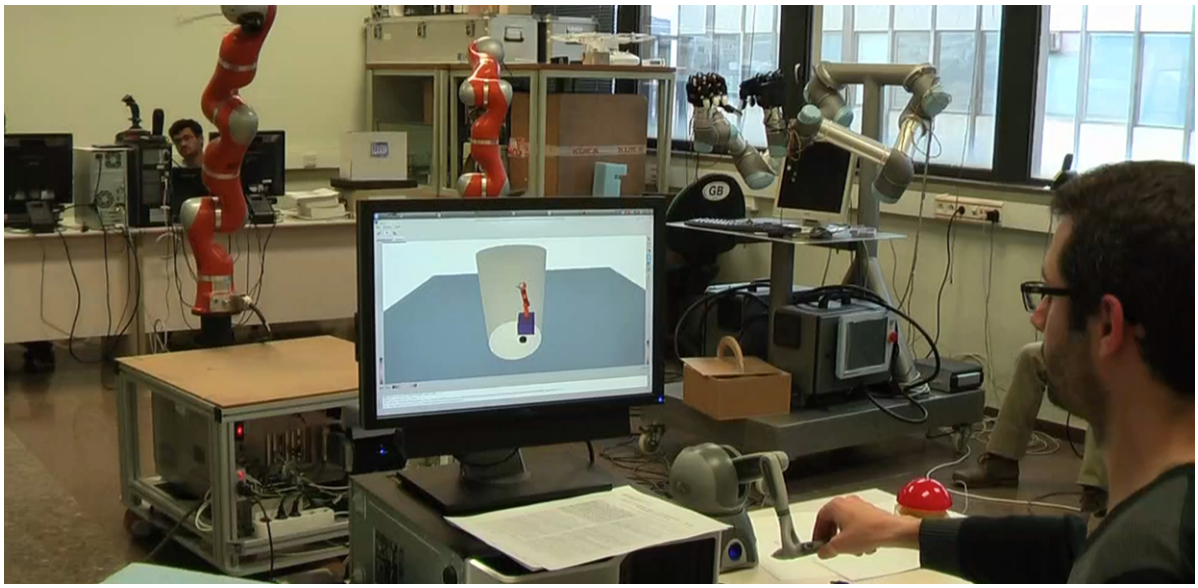

Figura 5: A scenario with a teleoperation task. The task is visualized in a virtual scenario.

platform and to the Kuka LWR. Optionally, the Kautham Project [20] has been configured to visualize the system in a virtual environment 5.

With this setup, several experiments have been done to test the system, initially solely in the virtual environment, and lately also in real situations.

5 CONCLUSIONS

A new robotic system that integrates a mobile manipulator, an UAV and a haptic device is presented. This system allows the teleoperation of the mobile manipulator using the visual feedback of the UAV camera.

An algorithm is presented that allows the operator to command the TCP of the mobile manipulator and the orientation of the UAV with respect to the robot using solely the haptic.

The Jacobian null space of the robot is exploited to ease the teleoperation task by avoiding the robot own occlusions as a secondary task.

The implementation setup of the proposed solution in the real system is presented.

Future work will also address a gross control of the UAV, and the integration of the proposed solution in a stack of tasks with a higher number of prioritized tasks.

Acknowledgments

*This work was partially supported by the Spanish Government through the projects DPI2013-40882-P, DPI2014-57757-R and DPI2016-80077-R, and the predoctoral grants BES-2012-054899.

Bibliografia

[1] L. Merino, F. C. and. J. R. Martinez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, pp. 165–184, 2006.

[2] L. Lin and M. A. Goodrich, "UAV intelligent path planning for wilderness search and rescue," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 709–714, Oct 2009.

[3] S. M. Adams and C. J. Friedland, "A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management," in *Ninth International Workshop on Remote Sensing for Disaster Response*, 2011.

[4] D. Lee, C. Ha, and Z. Zuo, "Backstepping control of quadrotor-type UAVs and its application to teleoperation over the Internet.," in *IAS (2)* (S. Lee, H. Cho, K.-J. Yoon, and J. Lee, eds.), vol. 194 of *Advances in Intelligent Systems and Computing*, pp. 217–225, Springer, 2012.

[5] J. Park and O. Khatib, "A haptic teleoperation approach based on contact force control.," *I. J. Robotic Res.*, vol. 25, no. 5-6, pp. 575–591, 2006.

[6] H. G. Tanner, "Switched UAV-UGV cooperation scheme for target detection," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3457–3462, April 2007.

[7] R. Rao, V. Kumar, and C. Taylor, "Visual servoing of a UGV from a UAV using differential flatness," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ In-*

ternational Conference on, vol. 1, pp. 743–748 vol.1, Oct 2003.

[8] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle," *J. Intell. Robotics Syst.*, vol. 61, pp. 221–238, Jan. 2011.

[9] M. Frejek and S. B. Nokleby, "A methodology for tele-operating mobile manipulators with an emphasis on operator ease of use," *Robotica*, vol. 31, pp. 331–344, 5 2013.

[10] F. Conti and O. Khatib, "Spanning large workspaces using small haptic devices," in *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pp. 183–188, March 2005.

[11] A. Pérez and J. Rosell, "An assisted resynchronization method for robotic teleoperated tasks," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 886–891, May 2011.

[12] C. Ware and S. Osborne, "Exploration and virtual camera control in virtual three dimensional environments," *SIGGRAPH Comput. Graph.*, vol. 24, pp. 175–183, Feb. 1990.

[13] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans Syst, Man, Cybern, Syst*, vol. 7, pp. 868–871, Dec 1977.

[14] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, vol. 2, pp. 1211–1216, June 1991.

[15] J.-A. Claret, G. Venture, and L. Basañez, "Exploiting the robot kinematic redundancy for emotion conveyance to humans as a lower priority task," *International Journal of Social Robotics*, vol. 9, no. 2, pp. 277–292, 2017.

[16] D. Clos and J. Martínez, "Plataforma Mòbil amb Rodes Esfèriques per al Robot "Lightweight Robot 4" de Kuka Roboter," tech. rep., Institute of Industrial and Control Engineering - Technical University of Catalonia, June 2007.

[17] J. A. Claret, I. Zaplana, and L. Basañez, "Teleoperating a mobile manipulator and a free-flying camera from a single haptic device," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 291–296, Oct 2016.

[18] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st ed., 2008.

[19] L. Dominjon, A. Lecuyer, J. M. Burkhardt, G. Andrade-Barroso, and S. Richir, "The "bubble" technique: interacting with large virtual environments using haptic devices with limited workspace," in *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pp. 639–640, March 2005.

[20] J. Rosell, A. Pérez, A. Akbari, Muhayyuddin, L. Palomo, and N. García, "The kautham project: A teaching and research tool for robot motion planning," in *IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1–8, Institute of Electrical and Electronics Engineers (IEEE), Sep 2014.