HAL open science

A Framework for intuitive collaboration with a mobile manipulator

Benjamin Navarro, Andrea Cherubini, Aïcha Fonte, Gérard Poisson, Philippe Fraisse

To cite this version:

Benjamin Navarro, Andrea Cherubini, Aïcha Fonte, Gérard Poisson, Philippe Fraisse. A Framework for intuitive collaboration with a mobile manipulator. 2017. hal-01489029v1

HAL Id: hal-01489029

https://hal.science/hal-01489029v1

Preprint submitted on 14 Mar 2017 (v1), last revised 22 Feb 2019 (v2)

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

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

A Framework for intuitive collaboration with a mobile manipulator

Benjamin Navarro[1,2], Andrea Cherubini[1], Aïcha Fonte[2], Gérard Poisson[2], and Philippe Fraisse[1]

***Abstract*— In this paper, we present a control strategy that enables intuitive physical human-robot collaboration with mobile manipulators equipped with an omnidirectional base. When interacting with a human operator, intuitiveness of operation is a major concern. To this end, we propose a redundancy solution that allows the mobile base to be fixed when working locally and moves it only when the robot approaches a set of constraints. These constraints include distance to singular poses, minimum of manipulability and distance to objects and angular deviation. Experimental results with a Kuka LWR4 arm mounted on a Neobotix MPO700 mobile base validate the proposed approach.**

I. INTRODUCTION

Mobile manipulators benefit from the dexterity of a standard manipulator with the extended workspace of a mobile platform. Locomotion can be realized by wheeled, legged or flying bases. However, mobile manipulators are over-actuated robots that need specific control algorithms to deal with their redundancy. Several approaches have been proposed, depending on the type of mobile base that is used.

For wheeled bases, differential drive actuation introduces non-holonomic constraints due to the rolling without slipping of the wheels on the ground. These constraints limit the set of velocities that can be realized by the mobile base, and that need to be integrated in the controller. This has been addressed by several approaches, e.g., using a path planning strategy [1] or producing a complete kinematics model together with a redundancy scheme [2]. Instead, for mobile bases with steerable wheels, a global kinematics model cannot be used directly, since velocities on the steering axes do not induce velocities on the robot. This has been investigated in [3], where the Jacobian null space projection and a global input-output linearization with dynamic feedback have been tested and compared.

Legged robots equipped with an arm generally fall into two categories, biped and quadruped. Biped robots, usually adopt a humanoid structure. For these systems, locomotion and manipulation are tightly coupled, since the robot balance needs to be guaranteed. Generally, for dealing with the system's high redundancy, researchers use optimization with a set of tasks (e.g., base velocity and hand/s pose/s) and constraints (e.g., stability and self collision avoidance). This strategy has been applied in [4] and in [5], to achieve human-humanoid interaction. Quadruped robots equipped with a manipulator have been investigated since [6], where the base is modeled as a parallel robot to define the pose of the arm's base frame and inverse kinematics are used to control the whole robot.

Recently, researchers have embedded manipulators on aerial robots [7], [8]. In such scenarios, the dynamic effects of the arm motion must be taken into account, along with redundancy, to keep the robot stable.

Research have been conducted on physical interaction with mobile robots, such as in [**?**] were a fully omnidirectional wheeled robot has been made compliant using force control. Physical interaction with mobile manipulators has also been studied in [9], where force thresholds have been used to decide if the base, the arm or both of them have to move, or in [**?**] where the use of a tactile skin permits full body compliance.

Our work focuses on human-robot collaboration, where *intuitiveness of operation* is crucial, as it enables untrained operators to use the system. To the best of our knowledge, this case has not yet been investigated for mobile manipulators. To this end, we propose a damping controller, that allows the end effector to be moved manually, together with a redundancy solution that keeps the mobile platform fixed during local manipulations and moves it only when needed (i.e., when the arm approaches singular poses, low manipulability zones or workspace limitations). This controller has been validated on a robotic arm mounted on a mobile base with steerable wheels.

The paper is organized as follows. In Section II, we define the main variables and describe the proposed controller as well as the redundancy solution. In Section III, the arm motion constraints, which trigger motion of the base, are detailed. An experimental validation in both simulated and real environments is presented in Section IV. We summarize and conclude in Section V.

II. MOBILE COMANIPULATION FRAMEWORK

We consider a manipulator (open kinematic chain), with $\mathbf{q} \in \mathbb{R}^j$ its j joint values, mounted on an omnidirectional mobile platform with $\dot{\mathbf{x}}_{base} \in \mathbb{SE}(3)$ its cartesian control input. The manipulator tool control point (TCP) has pose $\mathbf{x} = \left[\mathbf{p}^\top \ \boldsymbol{\theta}^\top\right]^\top \in \mathbb{SE}(3)$ and velocity $\dot{\mathbf{x}} = \left[\mathbf{v}^\top \ \boldsymbol{\omega}^\top\right]^\top$. Both $\mathbf{x}$ and $\dot{\mathbf{x}}$ are expressed in the robot base frame, attached to the center of the mobile base. We also assume that it is possible to estimate (either directly or through joint torque measurement) the external wrench (forces and torques) applied to the TCP, and expressed in the tool center frame: $\mathbf{h}_{ext} = \left[\mathbf{f}_{ext}^\top \ \tau_{ext}^\top\right]^\top \in \mathbb{R}^6$. This information will be used to move the tool according to the wrench that the operator applies onto it.

[1]Laboratory for Computer Science, Microelectronics and Robotics LIRMM - Université de Montpellier CNRS, 161 Rue Ada, 34090 Montpellier, France. E-mail: firstname.lastname@lirmm.fr

[2]PRISME Laboratory, University of Orléans - INSA CVL, 63 avenue de Lattre de Tassigny, F-18020 Bourges Cedex, France. E-mail: firstname.lastname@univ-orleans.fr

A. Damping control

To let the operator move the tool we apply damping control, which is a particular case of admittance control [10]. It can be formulated in the tool frame T as:

$$ {}^{T}\dot{\mathbf{x}} = \mathbf{B}^{-1}\mathbf{h}_{ext} + {}^{T}\dot{\mathbf{x}}^{*}. \tag{1} $$

In (1), $\mathbf{B}$ is a diagonal matrix of strictly positive user-defined damping values and ${}^{T}\dot{\mathbf{x}}^{*}$ a reference velocity in the tool frame. For the robot to track velocity ${}^{T}\dot{\mathbf{x}}$, this is mapped to the robot base frame through:

$$ \dot{\mathbf{x}} = {}^{B}\mathbf{V}_{T}\ {}^{T}\dot{\mathbf{x}} = \begin{bmatrix} {}^{B}\mathbf{R}_{T} & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^{B}\mathbf{R}_{T} \end{bmatrix} {}^{T}\dot{\mathbf{x}}, \tag{2} $$

${}^{B}\mathbf{R}_{T}$ being the rotation matrix from tool to base frame.

B. Whole body control strategy

Because of redundancy, mobile manipulators inherently share some mobility of the cartesian space between the manipulator and the mobile base. In order to solve for such redundancy, we adopt the following strategy:

$$ \dot{\mathbf{x}}_{arm} = \mathbf{A}\dot{\mathbf{x}}, \tag{3} $$

$$ \dot{\mathbf{x}}_{base} = (\mathbf{I} - \mathbf{A})\dot{\mathbf{x}}, \tag{4} $$

$$ \mathbf{A} = \mathrm{diag}\{a_{v_x}, a_{v_y}, a_{v_z}, a_{\omega_x}, a_{\omega_y}, a_{\omega_z}\} \in \mathbb{R}^{6\times 6}. \tag{5} $$

In these equations, $\dot{\mathbf{x}}_{arm}$ is the velocity command for the arm and all six $a \in [0, 1]$. If the velocity in a direction (degree of freedom or dof) is only realized by the manipulator, the corresponding a is 1. If it is 0, the motion in that direction is obtained thanks to the mobile platform. For values in between, the motion is shared among the two. The derivation of the a values will be explained in section III.

In this work, the mobile base is limited to the 3 components of ground plane motion $[v_x\ v_y\ \omega_z]^{\top}$, whereas, outside singularities, the arm can generate velocities in any direction in space. Hence, we set:

$$ \mathbf{A} = diag(a_{v_x}, a_{v_y}, 1, 1, 1, a_{\omega_z}). \tag{6} $$

Inverse kinematics is used on the manipulator to map task space velocities to joint space:

$$ \dot{\mathbf{q}} = \mathbf{T}(\mathbf{q})\dot{\mathbf{x}}_{arm} \tag{7} $$

with $\mathbf{T}(\mathbf{q})$ a $j \times 6$ matrix. Since different kinematic structures exist to allow omnidirectional motion (Swedish or spherical wheels, legs, flying base, etc.) the actuation of the mobile base dof is not considered in this work so we assume that the mapping from $\dot{\mathbf{x}}_{base}$ to its joint values exists and is known.

Using a classical (pseudo-)inverse of the manipulator task Jacobian $\mathbf{J}$ leads to large velocities near singular poses that could potentially harm the operator. To avoid such problem, we use, as in [11], adaptive damped least squares to compute $\mathbf{T}(\mathbf{q})$. This leads to:

$$ \mathbf{T}(\mathbf{q}) = \mathbf{J}^{\top}(\mathbf{J}\mathbf{J}^{\top} + \lambda^2\mathbf{I})^{-1}, \tag{8} $$

with λ^2 being calculated as:

$$ \lambda^2 = \begin{cases} 0 & \text{if } \sigma_m \geq \varepsilon, \\ (1 - (\frac{\sigma_m}{\varepsilon})^2)\lambda_{max}^2 & \text{otherwise.} \end{cases} \tag{9} $$

In (9), σ_m is the smallest singular value of $\mathbf{J}$, which can be obtained from its singular values decomposition, ε is a threshold that activates the damping effect, and λ_{max} is the maximum value for λ. The advantage of using (9) over a constant value for λ is that the arm performance is not degraded when it is away from a singularity, hence keeping a small velocity tracking error.

III. CONSTRAINTS

In this section, we explain how the a values are calculated, in order to satisfy a set of constraints: distance to singularities, minimum of manipulability, distance to objects and angular deviation. For each constraint, the goal is to move only the arm while the system is far from that constraint. This is a major difference with classic whole body control where all joints are actuated to perform a task. Our choice arises from the fact that for an operator manipulating the robot it is more intuitive that the base is fixed when working locally and moves only when a distant target needs to be reached.

In order to get a smooth evolution of the a values, we use an interpolation function described by:

$$ f(c,x^-,x^+,y^-,y^+) = \begin{cases} y^- & \text{if } c \leq x^-, \\ y^+ & \text{if } c \geq x^+, \\ f_p(c,x^-,x^+,y^-,y^+) & \text{otherwise.} \end{cases} \tag{10} $$

Here, $f_p(c, x^-, x^+, y^-, y^+)$ is a fifth-order polynomial with null first and second derivatives at x^- and x^+. Figure (1) gives an example of f.

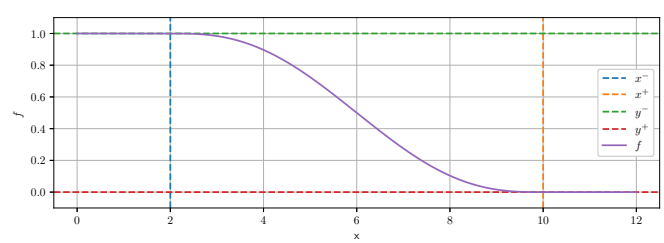

Fig. 1: Interpolation function f for $x^- = 2$, $x^+ = 10$, $y^- = 1$ and $y^+ = 0$.

A. Distance to singularities

Clearly, when the arm reaches a singular configuration, the mobile base has to take over. To this aim, we use σ_m as a measure of the distance to singular poses to derive the a values:

$$ a_{s,i} = f(s, 0, 1, 0, 1), \forall i \in \{v_x, v_y, ..., \omega_z\} \tag{11} $$

with $s = 1 - \lambda^2/\lambda_{max}^2$. Using (9) leads to s varying from 1 in non-sigular poses ($\lambda = 0$) to 1 in singularity ($\lambda = \lambda_{max}$).

B. Manipulability

The manipulability index defined by Yohikawa [12] is a largely used metric in mobile manipulation since it is related to the ability, for the arm, to produce velocities in the task space. It can be computed as:

$$\mu = \sqrt{det(\mathbf{J}\mathbf{J}^T)} = \prod_{i=1}^{M} \sigma_i \tag{12}$$

where σ_i is the i-th singular value of $\mathbf{J}$. This measure can be weighted with a factor decreasing near joint limits, as in [13]. We propose the following penalization cost:

$$\beta(\mathbf{q}) = \prod_{i=1}^{j} \left[1 - \left[\frac{2q_i - (q_i^+ + q_i^-)}{q_i^+ - q_i^-} \right]^2 \right] \in [0, 1], \tag{13}$$

where q_i^+ and q_i^- are the upper and lower limits of the i-th joint. We can then merge the two measures using the following:

$$m = [\beta(\mathbf{q})\alpha + (1 - \alpha)]\, \mu \tag{14}$$

with α a scalar value that can be adjusted from 0 (no penalization) to 1 (full penalization) depending on the desired effect of β on the manipulability.

To account for manipulability, we set the a values to:

$$a_{m,i} = f(m, m_{min}, m_{th}, 0, 1), \forall i \in \{v_x, v_y, ..., \omega_z\}, \tag{15}$$

m_{min} being the smallest manipulability allowed and m_{th} the manipulability threshold at which velocities start to be transfered to the mobile base.

C. Distance to objects

The manipulator workspace can be limited by real (e.g. the mobile base body, to avoid self-collisions) or virtual (e.g. a virtual wall limiting the arm motion) physical constraints. We consider them all as geometric objects (e.g., planes, spheres, etc). We attach to the end effector a virtual sphere, with a radius large enough to contain any tool the robot may be carrying. Then, we define the set of n physical constraints (objects) limiting the arm's workspace. Finally, we compute the distance between each object and the sphere, using the GJK algorithm [14]. This algorithm outputs the pair of closest points, p_{obj} and p_{sphere}, that respectively belong to the surface of the object and of the sphere. To compute the a values, we first evaluate for each object k a distance vector $d^k = [d_x^k\ d_y^k\ d_z^k]^\top$ using algorithm 1.

In this algorithm, d_{min} and d_{th} represent the minimum and threshold distances used by the interpolator. We also impose $d_{min} > 0$ so that $|\Delta p_i| = 0$ is true only when the i-th axis is unconstrained. Finally, we can compute the a values realizing the distance constraint with:

$$a_{d,v_i} = \prod_{k=1}^{n} d_i^k, \forall i \in \{x, y, z\} \tag{16}$$

```
for k ← 1 to n do
    Δp^k = p^k_obj − p_sphere
    for i ∈ {x, y, z} do
        if |Δp^k_i| > 0 then
            d^k_i = f(|Δp^k_i|, d_min, d_th, 0, 1)
        else
            d^k_i = 1
        end
    end
end
```

Algorithm 1: Workspace distance computation

D. Angular deviation

In order to let the operator rotate the mobile base when needed, we constrain the angular deviation to the reference orientation $\boldsymbol{\theta}^*$. To do so, we define:

$$a_{d,\omega_i} = f(\Delta\theta_i, \Delta\theta_{th}, \Delta\theta_{max}, 1, 0), \forall i \in \{x, y, z\} \tag{17}$$

where $\Delta\theta_i = |\theta_i^* - \theta_i|$, $\Delta\theta_{th}$ is the angular activation threshold and $\Delta\theta_{max}$ is the maximum angular error. In (17), the a values vary from 1 at the angular threshold to 0 at the maximum deviation.

E. Constraint deactivation

Since all the constraint values depend solely on the current robot state, the arm can be locked in or more task space directions if the corresponding a values approach zero (e.g., if the operator stretches it to the singular configuration, s/he cannot move it afterwards). To solve this problem, we propose a general deactivation strategy that allows the manipulator to move again if the generated velocity $\dot{\mathbf{x}}$ tends to move the robot away from the constraint. For this we need to define a virtual manipulator with joint values $\mathbf{q}^v \in \mathbb{R}^j$, end effector pose $\mathbf{x}^v \in \mathbb{SE}(3)$ and associated Jacobian $\mathbf{J}^v$. Each sample time, we update the virtual robot with:

$$\mathbf{q}^v = \mathbf{T}(\mathbf{q})\dot{\mathbf{x}} * T_s + \mathbf{q}, \tag{18}$$

$$\mathbf{x}_{arm}^v = f_x(\mathbf{q}^v), \tag{19}$$

$$\mathbf{J}^v = f_J(\mathbf{q}^v). \tag{20}$$

In these equations, T_s is the controller sample time and f_x, f_J are the forward kinematics algorithms for extracting the manipulator's pose and Jacobian. Then, constraints (11), (15), (16) and (17) are computed for both real and virtual arms. For each pair, if the virtual arm constraint value is greater than the real arm one, the deactivation mechanism is triggered. This results in:

$$a = \begin{cases} a^r & \text{if } a^v < a^r \text{ or } t > t_{end}, \\ f(t, t_{start}, t_{end}, a_{i,start}, 1) & \text{otherwise,} \end{cases} \tag{21}$$

with a^r and a^v the constraint values for the real and virtual arm respectively, t the current time, t_{start} the time at which the mechanism was triggered and its associated value $a_{i,start}$ and t_{end} (initialized to 0) the time at which the a value will reach 1. By doing this, we ensure a smooth transfer of the velocities from the mobile base to the arm.

F. Merging the constraints

In order to use different constraints at the same time, we propose to multiply them to derive the value to inject in (5). This translates to:

$$a = \prod_{i \in C} a_i, \tag{22}$$

with C being the set of constraints to include.

The minimal value could also be used instead of the product but this would not allow the operator to feel that a new constraint is approaching and react to it if the effect is not the desired one.

IV. EXPERIMENTS

In this section, we present the experiments assessing the correct behavior of the proposed framework. In IV-A, we present simulations for a given reference trajectory $\dot{\mathbf{x}}^*$ with each constraint taken separately; then, in IV-B, we introduce the setup for real robot validation, and we comment the results in IV-C.

A. Validation

1) Distance to singularity: Figure 2 presents simulation results when only the singularity constraint is activated. The velocity command $\dot{\mathbf{x}}$ extends the arm to reach a singular configuration, where the mobile base starts moving, then retracts it to a non-singular pose. The relevant parameters are the following: $\epsilon = 0.1$, $\lambda_{max} = 0.1$ and $C = \{s\}$. At $t = 6.65s$, the singularity constraint is activated and velocities are progressively transferred from the arm to the mobile base, until $t = 9.75s$, when the mechanism is deactivated. From this instant, the a_s values increase up to 1 where the mobile base is stopped and only the arm moves.

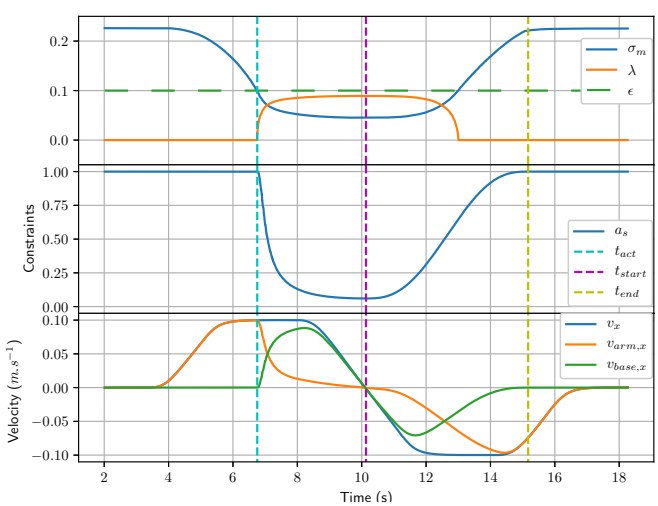

Fig. 2: Distance to singularity simulation. Top: smallest singular value σ_m and damping factor λ, middle: singularity constraints $a_s = a_{s,v_x} = a_{s,v_y} = a_{s,\omega_z}$, bottom: velocity commands along the x axis. t_{act} is the time at which the constraint gets activated.

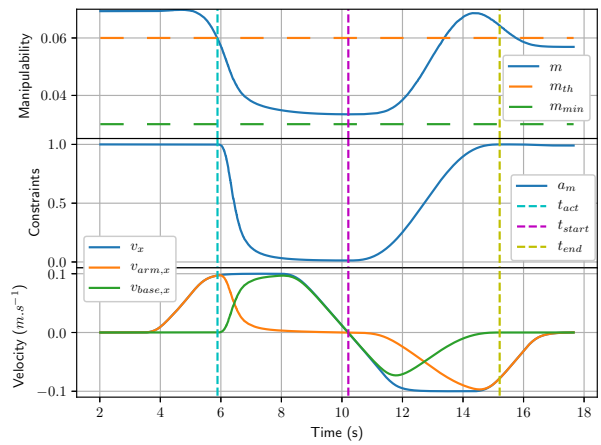

Fig. 3: Manipulability simulation. Top: manipulability measure m, middle: manipulability constraints $a_m = a_{m,v_x} = a_{m,v_y} = a_{m,\omega_z}$, bottom: Velocity commands along the x axis. t_{act} is the time at which the constraint gets activated.

2) Manipulability: Results for the manipulability constraint test are displayed in Fig. 3. The same velocity command $\dot{\mathbf{x}}$ as in IV-A.1 is used. The parameters for this test are $m_{th} = 0.06$, $m_{min} = 0.03$, $\alpha = 0.2$ and $C = \{m\}$. As expected, the arm follows the velocity trajectory until the manipulability measure drops below the threshold m_{th} and stops when m_{min} is reached. At this point, the mobile base fully tracks $\dot{\mathbf{x}}$ and the arm is at rest. At $t = 10s$ the arm starts moving again.

3) Distance to objects: To assess the behavior of the workspace constraint we use only one object ($n = 1$), a $1 \times 1 \times 2$ m box, centered at the base frame origin. The same velocity profile as in IV-A.1 is used but with opposite sign to first send the TCP against the mobile base and then move away from it. The TCP virtual sphere radius is set to 15cm and we use $d_{th} = 0.05$, $d_{min} = 0.001$ and $C = \{d\}$. Results are presented in Fig. 4. As in the previous experiments, the arm tracks the velocity profile until the threshold distance is reached. Then, velocities are progressively transferred to the mobile base. It is only when $\dot{\mathbf{x}}$ becomes positive to send the TCP away from the mobile base that the arm starts moving again and the base starts to decelerate and finally stop.

4) Angular deviation: For the angular deviation test, we use a reference rotational velocity ω_z^* that rotates the TCP above the maximum allowed orientation error $\Delta\theta = 1rad$. We use $C = \{a\}$ It can be seen from Fig. 5 that the constraint gets activated when $\Delta\theta_z$ crosses $\Delta\theta_{th} = 0.5rad$ and that the velocities are correctly transfered from the arm to the mobile base. As in the previous examples, deactivation occurs when the velocity sign changes.

B. Real experiment setup

To validate the proposed approach, we set up an application where an operator needs to move the TCP, attached to the arm's end effector, to a distant area, outside the

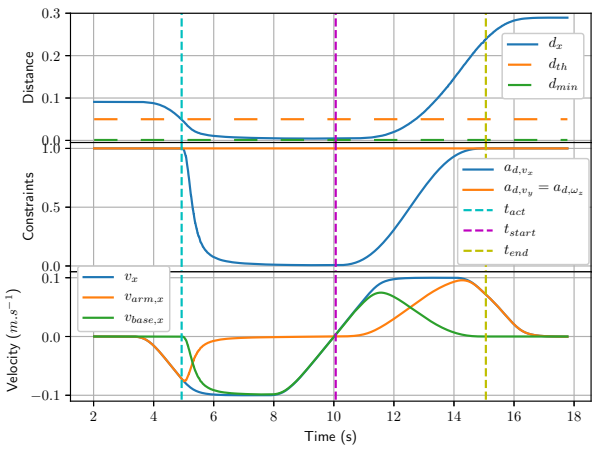

Fig. 4: Distance to objects simulation. Top: Minimal distance d_x, middle: workspace constraints, bottom: Velocity commands along the x axis. t_{act} is the time at which the constraint gets activated.

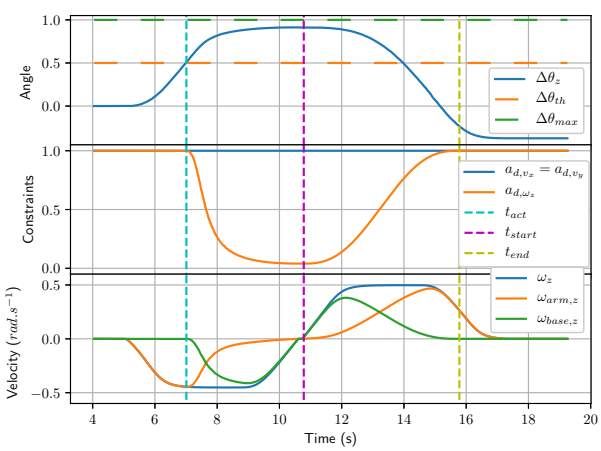

Fig. 5: Angular deviation simulation. Top: Angular error $\Delta\theta_z$, middle: workspace constraints, bottom: Velocity commands (z axis). t_{act} is the time at which the constraint gets activated.

manipulator's workspace. To this end, we used the LIRMM BAZAR[1] platform, shown in Fig. 6. BAZAR is composed of a Neobotix MPO700 mobile base, two Kuka LWR4 arms, two Shadow Hands and cameras. Only the right arm and the mobile base were used for this experiment. The external wrench $\mathbf{h}_{ext}$ is estimated through the FRI interface[2]. The implementation has been realized on a computer with an i7-6700HQ processor running Linux with the *Realtime Preemption patch*[3]. All the code was written in C++ using the Knowbotics Framework, currently under development at LIRMM. The FRI library was used to communicate with the Kuka arm while a UDP bridge was set up to send cartesian velocities to the computer embedded in the mobile base. The controller sample time was T=1 ms and the average computation time was 0.15 ms. Due to technical limitations in the mobile base low level controller, this was only updated every 25 ms.

Fig. 6: The BAZAR mobile manipulator.

C. Results

For this experiment, all three constraints were used $[C = \{s, m, d, a\}$, with the following parameters:

- distance to singularity constraint: $\epsilon = 0.1$, $\lambda_{max} = 0.1$,
- manipulability constraint: $m_{th} = 0.04$, $m_{min} = 0.02$, $\alpha = 0.2$,
- allowed workspace constraint: $d_{th} = 0.05m$, $d_{min} = 0.001m$, $\theta_{max} = 1rad$, $\theta_{th} = 0.5rad$,
- deactivation mechanism: $t_{end} - t_{start} = 2s$.

Results from this experiment are shown in Figures 7, 8 and in the video attached to this paper[4]. During the first 12 seconds the operator moves the arm freely and the mobile base stays fixed (Fig. 7a). Then, since the TCP approaches a singular configuration, the manipulability and singularity constraints are activated to transfer the velocities to the mobile platform. This allows the operator to move the robot to another location. During this phase, both translations and rotations of the mobile base are performed in order to reach the desired configuration (Fig. 7b). At $t = 30s$, the deactivation mechanism is triggered and the operator can again control the manipulator (Fig. 7c). At $t = 38s$, the end effector is pushed toward the mobile base, activating the workspace constraint. Until $t = 47s$, the mobile platform moves backward but the arm is still allowed to move in the unconstrained directions (Fig. 7d). Finally, the deactivation mechanism is enabled a second time to stop the mobile base and unconstrain the arm motion (Fig. 7e).

V. CONCLUSIONS

The key point of this paper is the proposal of a redundancy solution for mobile manipulators to enhance physical human-robot collaboration. The tool velocity is modified through

[1]Bimanual Agile Zany Anthorpomorphic Robot.
[2]http://cs.stanford.edu/people/tkr/fri/html
[3]https://rt.wiki.kernel.org/index.php/Main_Page

[4]also on http://bit.do/mobilecomanip

(a) Base is fixed, arm is moving. $t < 12s$

(b) Base is moving, arm is fixed. $12s < t < 30s$

(c) Base is fixed, arm is moving. $30s < t < 38s$

(d) Both base and arm are moving. $38s < t < 47s$

(e) Base is fixed, arm is moving. $t > 47s$

Fig. 7: Snapshots of the experiment.

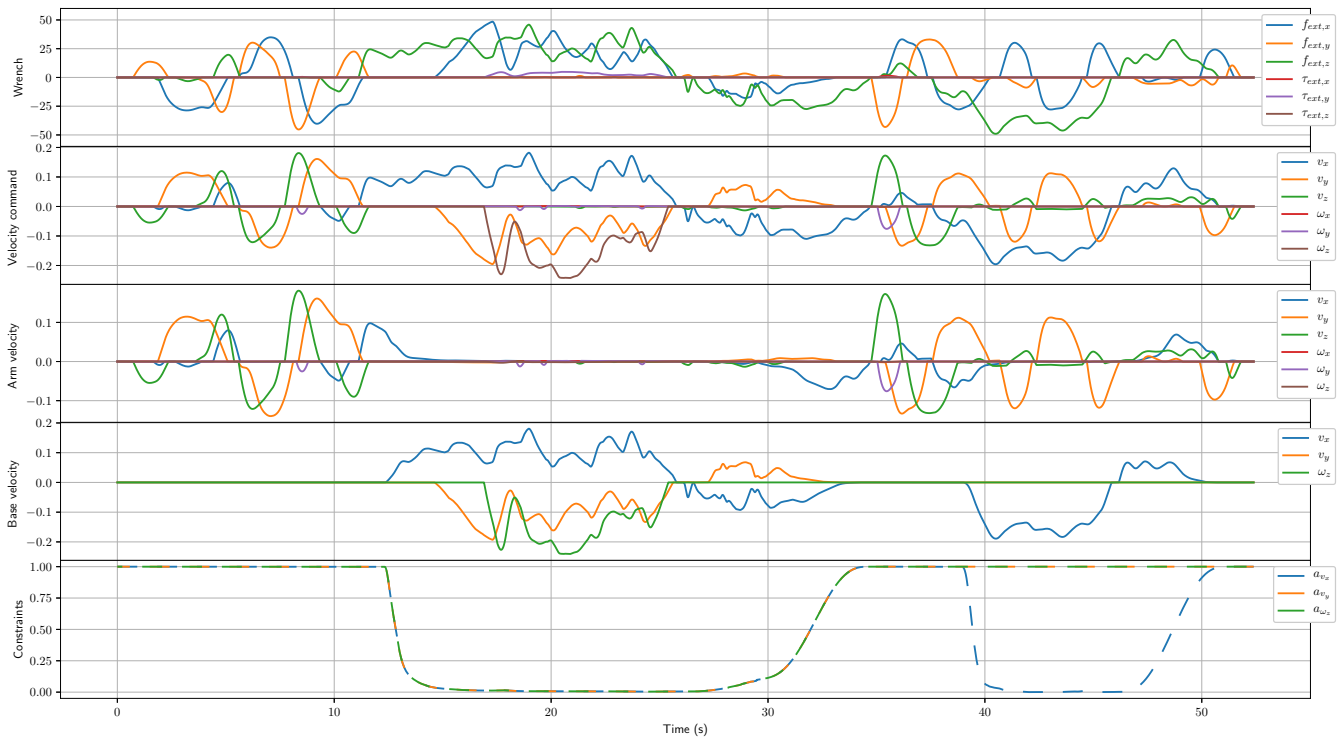

Fig. 8: Experimental results. From top to bottom: external wrench ${}^{T}\mathbf{h}_{ext}$ (N, Nm), velocity command $\dot{\mathbf{x}}$ (m/s, rad/s), arm velocity command $\dot{\mathbf{x}}_{arm}$ (m/s, rad/s), base velocity command $\dot{\mathbf{x}}_{base}$ (m/s, rad/s) and constraint values a_{v_x}, a_{v_y} and a_{ω_z}.

a reference trajectory or by interaction forces applied by a human operator. Four constraints have been proposed to exploit redundancy: distance to singularity, minimum of manipulability, distance to objects and angular deviations. The framework has been validated in both simulated and real environments. In future work, we will study different scenarios where new constraints may be needed. These include navigation in cluttered environments (e.g., for assistance to disabled people or to workers in factories).

ACKNOWLEDGMENTS

This work has been supported by the ANR (French National Research Agency) SISCob project ANR-14-CE27-0016.

REFERENCES

[1] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos. Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Transac. on Robotics and Automation*, 19(1):53–64, Feb 2003.

[2] A. De Luca, G. Oriolo, and P. R. Giordano. Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. In *2006 IEEE Int. Conf. on Robotics and Automation, 2006. ICRA 2006.*, pages 1867–1873, May 2006.

[3] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Kinematic control of nonholonomic mobile manipulators in the presence of steering wheels. In *2010 IEEE Int. Conf. on Robotics and Automation*, pages 1792–1798, May 2010.

[4] J. Vaillant, K. Bouyarmane, and A. Kheddar. Multi-character physical and behavioral interactions controller. *IEEE Transac. on Visualization and Computer Graphics*, PP(99):1–1, 2016.

[5] D. J. Agravante, A. Sherikov, P. B. Wieber, A. Cherubini, and A. Kheddar. Walking pattern generators designed for physical collaboration.

In *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1573–1578, May 2016.

[6] H. Adachi, N. Koyachi, T. Arai, and K. I. Nishimura. Control of a manipulator mounted on a quadruped. In *1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 883–888 vol.2, Nov 1996.

[7] A. E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano. Control of an aerial robot with multi-link arm for assembly tasks. In *2013 IEEE Int. Conf. on Robotics and Automation*, pages 4916–4921, May 2013.

[8] J. U. Álvarez-Muñoz, Nicolas Marchand, Fermi Guerrero-Castellanos, Sylvain Durand, and A. E. Lopez-Luna. Improving control of quadrotors carrying a manipulator arm. In *XVI Congreso Latinoamericano de Control Automático (CLCA 2014)*, page 6, -, Mexico, October 2014.

[9] Y. Jia, H. Wang, P. Strmer, and N. Xi. Human/robot interaction for human support system by using a mobile manipulator. In *2010 IEEE Int. Conf. on Robotics and Biomimetics*, pages 190–195, Dec 2010.

[10] N. Hogan. Impedance control: An approach to manipulation: Part II-Implementation. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):8–16, 1985.

[11] S. Chiaverini, O. Egeland, and R. K. Kanestrom. Achieving user-defined accuracy with damped least-squares inverse kinematics. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 672–677 vol.1, June 1991.

[12] T. Yoshikawa. Manipulability and redundancy control of robotic mechanisms. In *Proceedings. 1985 IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1004–1009, Mar 1985.

[13] Bong-Huan Jun, Pan-Mook Lee, and Jihong Lee. Manipulability analysis of underwater robotic arms on rov and application to task-oriented joint configuration. In *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, volume 3, pages 1548–1553 Vol.3, Nov 2004.

[14] Adrien Escande, Sylvain Miossec, Mehdi Benallegue, and Abderrahmane Kheddar. A strictly convex hull for computing proximity distances with continuous gradients. *IEEE Transac. on Robotics*, 30(3):666–678, June 2014.