2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Semi-autonomous surface-tracking tasks using omnidirectional mobile manipulators

Carlos Suarez Zapico[1,2], Yvan Petillot[1,2] and Mustafa Suphi Erden[1,2]

Abstract— **Despite the potential of mobile manipulators and applications where robots require a force-controlled physical interaction with the environment, the majority of robot automation nowadays is still based on fixed manipulators for free-motion tasks (e.g. welding, pick and place, or painting). In this work, we propose a control solution for omnidirectional mobile manipulators in force-tracking tasks, interacting with unknown surface geometries and with a human teleoperator in the control loop. Keeping a teleoperator in the loop makes the system widely applicable to unstructured environments. With little effort, a human can take care of the mobile base navigation, self-collisions, and collisions with the environment, as well as selecting the area of the asset surface to process. The teleoperator interfaces with the robot platform by commanding motion in the mobile base to increase the arm's workspace and manoeuvrability. The operator can also command the movement of the end-effector, sliding on the surface geometry to process a specific area. Alternatively, he can let the controller execute a parametric trajectory (spiral or raster) for an autonomous area coverage and meanwhile telecommand the base in order to keep the arm in configurations with good dexterity. The autonomous controller, on the other hand, takes responsibility for following the unknown contour on the manipulated surface by only taking observations from a force/torque sensor attached to the arm's wrist, exerting a prescribed force, and handling the motion control in the base and the arm so that both can follow their respective task requests. Overall, we have developed a user-friendly control scheme, where an operator with little training and using a joystick, can guide the robot system to perform a physically interactive task on the surface of an asset.**

I. INTRODUCTION

Mobile manipulators have the capability to provide flexibility in the manufacturing process and task automation, yet they are not in widespread use, with most applications focusing on simplistic pick-and-place tasks [1]. Controlling the interaction forces between the robot arm and the environment opens new potential applications for mobile robot automation such as polishing, grinding, sanding, or Non-Destructive Evaluation (NDE). In applications such as sanding or polishing, by partially automating these tasks, the human operator would benefit from preventing health issues associated with holding vibrational tools regularly, inhaling dust particles generated in the process, or non-ergonomic postures.

Keeping a human operator in the loop with a semi-autonomous robot, on the other hand, provides supervision that can relax the need for complex navigation involved in deploying the mobile robot in unstructured, cluttered, and dynamic environments. For instance, a human teleoperator can easily determine which area needs to be processed and how to move the base to maximize the arm operational envelope whilst avoiding obstacles in the scene. The human-robot symbiosis as presented here can also increase productivity, as teleoperating the robot is less tiring and possibly more precise and faster than holding and using tools, such as a rotatory sander.

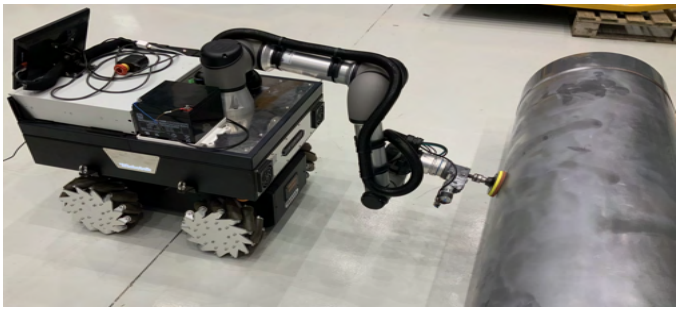

Fig. 1. Testbed used to validate the proposed control architecture. rbKairos omnidirectional mobile manipulator with an UR5e arm, polishing a PVC pipe with a spinning tool. The UR5e has a force/torque sensor embedded in the end-effector used in the force control and contour following.

In order to get the best process quality, the controller has to meet the following criteria. Once the tool establishes contact, it has to keep the tool perpendicular to the surface while exerting a specific force. To obtain homogeneous results across the processed surface, it is crucial to exert a steady and constant force, maintain a consistent angle of attack between the tool and the surface, and keep a constant tangential velocity to the surface. Therefore having a good contour following is paramount for the quality obtained after the operation is finished. Another important feature to obtain good results is to have a task-space path generator that ensures good surface coverage and that the tool has worked uniformly (removing the same amount of material).

If the mobile robot operates on flat and smooth ground, as typical industrial plantation floors, choosing an omnidirectional base greatly simplifies motion planning. This is because it eliminates the need to deal with nonholonomic constraints, which would otherwise require additional trajectory optimization planners to deal with these constraints.

A. Related Work

A common approach when controlling mobile manipulators is to use redundancy resolution to manage the joint

[1]Heriot-Watt University, Institute of Sensors, Signals, and Systems, Edinburgh, United Kingdom
[2]Edinburgh Centre for Robotics, Edinburgh, United Kingdom
{cs377, y.r.petillot, m.s.erden}@hw.ac.uk
This research was funded by EPSRC, United Kingdom, through the ORCA Hub project with the Grant Reference EP/R026173/1

motion in the base and arm while executing various tasks with different priorities. While the primary task is normally associated with the task space movement in the end-effector, secondary tasks can be used to prevent self-collisions, joint limits, or to maintain good dexterity [2]. In [3], a motion planning control is proposed for mobile redundant manipulators in surface mechanization. By exploiting the redundancy to provide the maximum stiffness and centre of gravity stability, they improve the process quality. Similarly in [4], the null space is exploited to enhance the force exertion by maximizing the force manipulability for an admittance control that helps with mobility assistance for disabled people. In [5] and [6], the redundancy resolution is used as a dual trajectory controller for non-holonomic mobile manipulators. The framework is modified so that the base and arm can follow independent trajectories, prioritizing the arm when both trajectories can not be executed simultaneously. The redundancy resolution can also be extended for dynamics modelling, and this is what is implemented in [7] and [8] to obtain a whole-body cartesian impedance control.

Another approach in motion control for mobile manipulators is to use trajectory optimization instead of redundancy resolution to optimize secondary tasks. In [9], the mobile manipulation interaction is done in a Model Predictive Control framework. The optimization not only accounts for vehicle dynamics and collision avoidance but also includes the force controller as a dynamic constraint to handle the interaction at the end-effector. Trajectory optimization can also be used to handle nonholonomic constraints [10], [11]. In [10], it is also used to avoid collisions with surrounding static and dynamic obstacles.

One of the challenges when it comes to interaction with an unknown surface geometry is following the contour to keep the tool in contact and perpendicular to the surface. To estimate the surface normal, most solutions involve perception through point clouds, laser scans, position displacement, and force/torque [12] readings. In tasks such as polishing or brushing, the spinning tool provides feedback on how the tool is aligned with the surface normal, which helps reduce the perpendicularity error [13].

In our previous work [14], we put to test various interaction controllers with the goal of maintaining contact with a prescribed force at the end-effector of a manipulator. During the test, we introduce motion disturbances at its base by replaying recorded odometry from an underwater vehicle. In [15] we use a hybrid force-position control where a novel trajectory following algorithm is developed to follow a specific trajectory projected in an unknown surface geometry. Finally, in [16] and [17], we used particle swarm optimization to find optimal locations for a mobile base and provide the attached arm with the best working conditions.

B. Contribution

In this work, we offer a new semi-autonomous robotic application for force-tracking tasks with omnidirectional mobile manipulators. This human-robot symbiosis has the potential to increase productivity, execute these operations in scenarios inaccessible to human operators, and prevent health hazards derived from the long-term manual execution of these tasks.

We present a new control architecture composed of subsystems which, on their own, are quite popular in the field, but novel in the way we combine and configure them specifically for this application. This results in a system that is intuitive and easy to control by a human teleoperator to execute force-tracking tasks.

C. Control Architecture Overview

The control architecture is composed of various subsystems that work simultaneously to handle the whole body motion control while responding to the primary and secondary tasks, keeping the tool perpendicular to the surface, and exerting a desired force. Figure 2 schematizes the dimensions of the task space, $[x, y, z, \phi x, \phi y, \phi z]$ and the mobile base $[x_b, y_b, \phi_b]$. The primary task corresponds to the motion at the arm's end-effector whereas the secondary task contains the motion commands for the mobile base. We offer two separate operating modes for the motion at the task space, the teleoperated, and the automatic mode. The human operator can switch between these two modes. The difference is that in the teleoperated mode, the operator guides the motion of the end-effector along the surface, whereas in the automatic mode, a parametric trajectory (raster, spiral) is executed along the surface. The operator uses a joystick (Figure 2) to interface with the system. In the autonomous mode, he commands motion in $[x_b, y_b, \phi_b]$, while in the teleoperated mode he commands motion in $[x, y, x_b, y_b, \phi_b]$. So in both modes, the operator guides the mobile base. The rest of the dimensions are managed by other subsystems in the control architecture. $[\phi x, \phi y]$ which are controlled by the tool orientation module that maintains perpendicularity between tool and surface, and z which is defined by the force controller.

As opposed to mobile manipulators operating in underwater, aerial, or outer-space scenarios, the base on flat ground is not significantly disturbed by interactions with the environment, by the coupling dynamics between the base and the arm, or even by the weight of the tools that are used, as typically in this application. For that reason, the motion control of the entire robot is done through a whole-body kinematic model. For the force-tracking application, the control architecture is complemented with an admittance controller acting on the end-effector with a force/torque sensor closing the loop and located in the wrist of the arm.

In the following sections, we explain the details of our proposed controller, starting with the whole-body differential inverse kinematics with redundancy resolution. The third and fourth sections describe the interaction control acting in the task space of the end-effector, and the base motion control, respectively. In Section V we show the results from an experiment that illustrates how the controller performs on a rbKairos mobile platform (Figure 1).

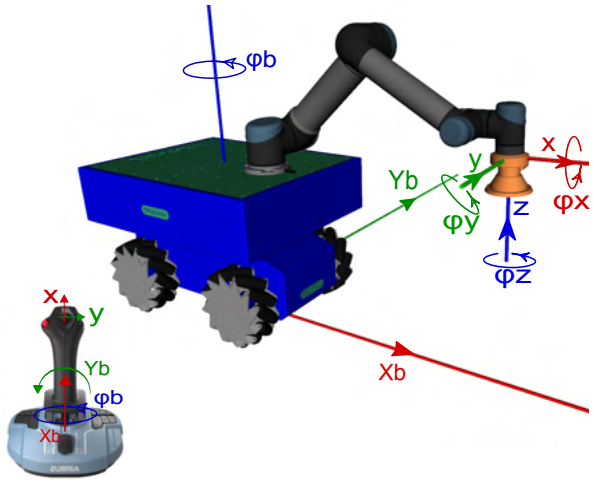

Fig. 2. Controlled dimensions in the task space $[x,y,z,\phi x,\phi y,\phi z]$ and the mobile base $[x_b,y_b,\phi_b]$. A joystick is used as an interface to send motion commands in velocity for $[x,y,x_b,y_b,\phi_b]$

II. REDUNDANCY RESOLUTION

Movement in the end-effector is fully determined with the 6 DOFs of the task space. The omnidirectional base provides control in the 3 DOFs of the floor and in addition to the joints of the arm (6 DOFs in the case of the rbKairos robot), it results in a system with redundancy that can be exploited with a redundancy resolution scheme.

The motion of the robot is achieved through differential inverse kinematics by using the robot Jacobian, J, which results from the concatenation of the arm and the base Jacobians (J_a and J_v, respectively). Given the current arm configuration q_a we can map the task space velocities $\dot{r}$ to the joint space $\dot{q}$ with the Jacobian:

$$\dot{r} = \begin{bmatrix} J_v & J_a(q_a) \end{bmatrix} \begin{bmatrix} \dot{q_v} \\ \dot{q_a} \end{bmatrix} \rightarrow \dot{r} = J(q)\dot{q} \tag{1}$$

In redundant systems, there is a null space of joint velocities that have no effect on the task space motion. We can exploit this null space by using the null space projector $(I_n - J^+J)$. We can include this and extend Expression (1) as:

$$\dot{q} = J^+\dot{r} + (I_n - J^+J)\dot{s} \tag{2}$$

where J^+ is the Moore-Penrose inverse Jacobian and $\dot{s}$ is an arbitrary joint speed normally used to optimize secondary tasks while following the primary one ($\dot{r}$).

The differential inverse kinematics breaks down when we reach a singularity or try to go beyond the arm's workspace, resulting in high speeds and unstable motions. We can handle these situations by embedding the Damped Least Square (DLS) [18] method into the previous expression:

$$\dot{q} = (J^TJ + \lambda^2 I)^{-1}J^T\dot{r} + (I_n - (J^TJ + \lambda^2 I)^{-1}J^TJ)\dot{s} \tag{3}$$

where I_n is an identity matrix with a size equal to the number of DOFs (9 for the rbKairos). I denotes the identity matrix with a size of 6, corresponding to the dimensionality of the task space. And λ is the damping term from the DLS method, which is defined by an adaptive law that acts when the robot approaches singularities.

$$\lambda^2 = \begin{cases} 0 & \text{when } \sigma_{lower} \geq \varepsilon \\ (1 - (\frac{\sigma_{lower}}{\varepsilon})^2)\lambda^2_{max} & \text{otherwise} \end{cases} \tag{4}$$

The adaptive law remains inactive if the lowest singular value of the Jacobian matrix σ_{lower} is greater than a predetermined threshold ε. In situations where σ_{lower} falls below ε, the damping factor is increased up to a maximum predetermined value λ_{max} as σ_{lower} decreases further.

As the motion accuracy of the base is much worse than the arm, we want to penalize the error in movement on the base DOFs. We use Weighted-Least Norm (WLN) [19] to integrate this feature into our formulation. Expression (3) is now transformed into:

$$\dot{q} = W^{-1}J^T(JW^{-1}J^T + \lambda^2 I)^{-1}\dot{r} - (I - W^{-1}J^T(JW^{-1}J^T + \lambda^2 I)^{-1}J)\dot{s} \tag{5}$$

where W is a diagonal square matrix that allows us to penalize each DOF independently by associating a null or very high value w_i.

$$W_q = \begin{bmatrix} w_1 & 0 & \dots & 0 & 0 \\ 0 & w_2 & \dots & 0 & 0 \\ : & : & \dots & : & : \\ 0 & 0 & \dots & w_n & 0 \\ 0 & 0 & \dots & 0 & w_{n+1} \end{bmatrix} \tag{6}$$

With this mathematical scheme, joint limits can also be easily handled by assigning a penalizing value when a joint is approaching its own limits [20]. In our case, the primary task velocity $\dot{r}$ is given in the mobile axis, so we have to provide the mobile axis Jacobian in 5.

With this whole-body kinematic controller, as long as all of the joints follow the joint velocity reference $\dot{q}$, we will not have motion errors. But because of motion inaccuracies, mostly occurring at the base DOFs, or in scenarios where the platform climbs up or down a slope, we need to close the motion control loop by feeding back the motion error in the task space. As an example, (2) can be modified to compensate for a tracking error e_p with a proportional controller as follows:

$$\dot{q} = J^+(\dot{r} + K_p e_p) + (I_n - J^+J)\dot{s} \tag{7}$$

Force-tracking only requires motion definition over five of the six dimensions in the task space. This provides us with one undefined DOF in the task space (ϕz), that we can exploit to our benefit. However, with this mathematical scheme, we cannot directly exploit this DOF. A higher-level planner working on top of this formulation could exploit this task space DOF.

III. TASK SPACE VELOCITY: HYBRID FORCE-POSITION CONTROL

When the robot end-effector establishes contact and starts the force-tracking operation, we divide the dimensions of the task space into position or force control. Figure 2 shows the

six dimensions of the task space [x, y, z, ϕx, ϕy, ϕz] and the notation that we use to refer to them. x and y are position-controlled, used for covering a specific area of the surface that we are interacting with. ϕz does not require motion definition, it can be set arbitrarily or exploited for secondary tasks. ϕx and ϕy are used to keep the tool perpendicularly to the surface and z is where we control the exerting force.

During the interaction, if the tool is misaligned with the surface normal, the torque increases in ϕx and ϕy. The tool orientation controller relies on this principle and works by minimizing these torques. This is achieved by using a simple proportional gain on the torque readings.

$$\dot{\phi}_x = K\tau_x \qquad \dot{\phi}_y = K\tau_y \tag{8}$$

where τ_x and τ_y are the torque readings at the interaction point frame, obtained after computing the wrench conversion from the Force/Torque sensor. K is a proportional gain and $\dot{\phi}_x$ and $\dot{\phi}_y$ are the output angular velocity to modify the orientation of the tool. With a higher gain, the robot can adapt faster when moving through high curvatures, performing a better contour following, but when the velocity is low, or there is little curvature, this can lead to an undesired wiggle behaviour in the orientation control of the tool. For better performance, an adaptive law for the gain K can be set so that it increases when the velocity is higher and high curvature is encountered and vice-versa. In the experiments presented in this paper, we apply a varying K gain, proportional to the tangential velocities of the end-effector, $\dot{x}$ and $\dot{y}$. This provides a simple solution to improve surface tracking and reduce the undesired wiggle effect compared to a fixed gain.

Finally, the direction z is where we run a mass-damping admittance controller to handle the impact and force control. The control in this axis is the most decisive for the success of the interaction in terms of safety, avoiding contact losses, and maintaining stable contact. As shown in [21] the mass-damping admittance system naturally evolves to the desired force F_d.

$$M_d\ddot{z} + B_d\dot{z} = \Delta F \tag{9}$$

where M_d and B_d are the virtual mass and damping respectively, and ΔF is the force error between the measured and desired force.

IV. SECONDARY VELOCITY: BASE MOTION

For the motion control of the mobile base, we feed the teleoperator velocity commands through the secondary velocity, $\dot{s}$ in the redundancy resolution (5). In the secondary task, an arbitrary whole-body joint velocity is provided. This joint velocity should be followed as best as possible after what has been left in the joint space to meet the priority task. Given the velocity commands for the base motion $V_{x_{base}}$, $V_{y_{base}}$, ω_{base}, we define $\dot{s}$ as:

$$\dot{s} = [V_{x_{base}}, V_{y_{base}}, \omega_{base}, 0, 0, 0, 0, 0, 0] \tag{10}$$

In order to completely define the motion control given by (5), we need to provide the weight matrix W that penalizes motion in the joints. When no motion penalization is associated with a joint, a value of 1 is assigned. The opposite can be obtained by assigning a very high value. By penalization, we refer to the impact of the motion of a joint on the primary task. With a high penalization, the joint will stay stationary or blocked, even if the primary task can not be resolved with the rest of the unblocked joints. Another effect that we obtain with this configuration, is that if we define a reference velocity in the secondary task ($\dot{s}$), the blocked joints will follow this reference without caring about the primary task, and the unblocked joints will execute the primary task and compensate for the displacement generated in the blocked joints. For the specifications of our application, the joints of the base are highly penalized to only follow the secondary velocity $\dot{s}$, and the joints of the arm are totally in charge of meeting the primary task requests. To enforce this behaviour, we assign a sufficiently high penalization term of 10^8 to the DOFs of the rbKairos base. Consequently, the weighting matrix is defined as follows:

$$W = diag\{10^8, 10^8, 10^8, 1, 1, 1, 1, 1, 1\} \tag{11}$$

The limitation of this setting is that if the teleoperator is not careful, he can command the mobile base to a position where the arm can no longer follow the primary task, and the arm ends up blocked at a near out-of-workspace configuration. However, the advantage is that the system remains very intuitive for the user as his commands are perfectly followed by the mobile base, without the redundancy resolution interfering in the DOFs of the base at any moment, not even when the primary task is violated. This configuration makes the system very easy to teleoperate and allows fast and agile motions or avoidance of collisions in cluttered spaces. We could be tented to set an intermediate penalizing value to the joints of the base, but not to the point of totally blocking them. For instance setting the weighting matrix like:

$$W = diag\{100, 100, 100, 1, 1, 1, 1, 1, 1\} \tag{12}$$

This configuration shows very similar behavior as before, with the only difference being that the redundancy resolution will act on the mobile base DOFs if the primary task is about to be violated, taking control over the base DOFs, and leaving none to the operator. A priori this might look like a better configuration. After all the controller is preventing the operator from moving the base away and making it unfeasible for the arm to meet the primary task request. However, this happens in a context where the arm is in a near out-of-workspace singular configuration and the controller will have to use the DOFs of the base to respond to the demands of the force controller that is acting in the task space, resulting in a very precarious scenario. With our proposed configuration (11), the arm will instead block near the singularity, but the teleoperator never loses control over the base DOFs, resulting in a safety feature of the system.

If we run this mathematical model in a simulated environment on an ideal robot that perfectly follows the motion reference in all its DOFs, we will observe that it executes the

requested motions both for the primary task and for the base perfectly accurately. We could check this in a simulation that we created as a previous step to the implementation in the real robot. However, with a physical system, the major source of error comes from the base DOFs following the velocity references. This error is directly transmitted to the end-effector and needs to be compensated by the force controller. If the error is too high, the force controller will not be able to compensate for it and this can generate high-force spikes that are dangerous for the robot's integrity, jeopardize the quality of the process, or in the best of cases run into a fail-safe and stopping the robot. To prevent this, we add velocity constraints to the mobile base commanded signal in order to reduce errors in the z task space dimension. Moreover, we smooth the reference signal to a middle point where the base can easily follow while keeping the control agile and intuitive for the operator, without too much time delay.

A. *Motion constraints on the base*

For safety reasons, a velocity saturation is set on the commanded reference to the actuated DOFs of the base. Another safety feature is to put extra care on the motion errors in the z dimension of the task space. This dimension is the most critical for the success of the interaction and also in terms of safety for the robot and asset, as it is perpendicular to the surface of the asset being treated, and motion errors can induce high peak forces.

For that reason, we add another motion constraint (13) built on top of the previous one, which is dependent on the current configuration of the robot as seen below:

$$\dot{r} = \begin{bmatrix} J_v & J_a(q_a) \end{bmatrix} \begin{bmatrix} V_{x_{base}} \\ V_{y_{base}} \\ \omega_{base} \\ 0 \\ \dots \\ 0 \end{bmatrix} \rightarrow \dot{z} \leq v_{zLim} \tag{13}$$

If the commanded velocity $V_{base} = [V_{x_{base}}, V_{y_{base}}, \omega_{base}]$ does not comply with the constraint (13), we scale it down by multiplying with the ratio between the velocity limit v_{zLim} and the induced task velocity in the z dimension. The saturation for the three actuated dimensions of the base is obtained with the following expression:

$$V_{saturation-base} = (v_{zLim}/\dot{z}) * V_{base} \tag{14}$$

where $\dot{z}$ is previously calculated in (13) given the current arm configuration q_a and the commanded velocity for the base V_{base}. This way, we not only can limit much better the motion error that the force control has to compensate for, but we also keep the motion direction intended by the operator and keep the machine interface intuitiveness.

The major source of error in the motion at the end-effector is the poor velocity-reference tracking in the base. In free motion tasks, we can compensate for this error by closing the loop with global odometry and feeding it back as in (7). But in this case, with fast interaction dynamics, it is best to execute the open loop motion as synchronized and accurately as possible. For that reason, it is paramount that the base follows the secondary velocity signal that is fed through $\dot{s}$ in (5). To meet this criterion we apply an acceleration limitation filter to the reference signal, both for the acceleration and deceleration phases. Figure 3 shows how the velocity tracking improves after applying the smoothing to a square signal and saturating in velocity for both directions of the x_b dimension in the base.

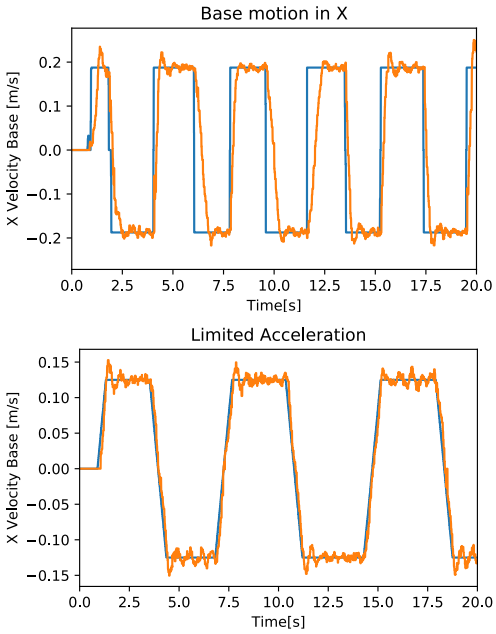

Fig. 3. Upper graph shows a commanded motion reference with no intermediate smoothing. The lower graph shows the same test with a smoothed reference by limiting it with a predefined maximum acceleration. Reference signal and odometry are represented in blue and orange respectively.

Equation (15) shows how the smoothing works. The smoothed reference signal s_t is obtained by bounding the maximum acceleration a_{lim}:

$$s_t = \max(\ \min(r_t,\ s_{t-1} + a_{lim}\Delta t),\ \ s_{t-1} - a_{lim}\Delta t)) \tag{15}$$

where r_t is the original user-commanded motion reference, and Δt is the sampling time.

V. EXPERIMENT FOR VALIDATION: PIPE BRUSHING

In this section, we show the performance obtained after implementing the controller on the rbKairos robot platform and we validate the subsystems of the controller working all together. To demonstrate its capabilities we brush a PVC pipe with $0.5m$ in diameter and $3m$ in length. The size and high curvature of the pipe make it a challenging scenario to put the whole controller to the test. During the experiment, we brush the front area of the pipe. Part of the coverage is made with the end-effector being teleoperated, then the operator switches to an autonomous mode which executes a raster trajectory to cover the rest of the front pipe area. In both modes, the user is still in the loop to teleoperate the motion of the base. The test shown in Figure 4 starts

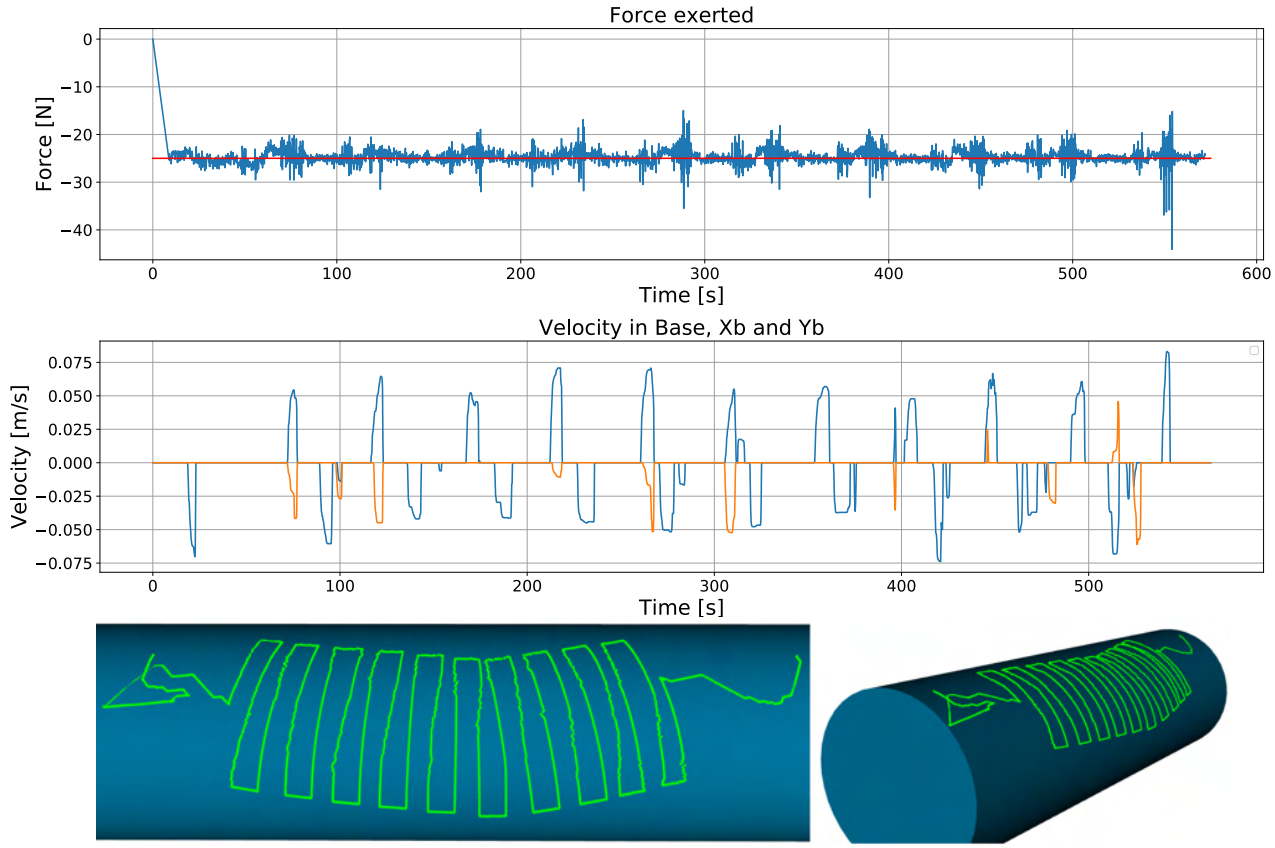

Fig. 4. Results obtained from brushing the front area of a PVC pipe. The upper graph shows the exerted force perpendicular to the surface where the desired force was set to 25 N. The second graph shows the linear velocities of the mobile base in x_b and y_b in blue and orange respectively. The bottom graph depicts the trace left by the tool during the brushing operation.

in teleoperation mode, later on, the operator switches to autonomous mode to execute a predefined raster trajectory, and finally, in the last phase, the user switches back to teleoperation mode. We can see when the autonomous mode is switched on and off by looking at the raster pattern left on the trace. The desired force is set to -25N. We can observe in the exerted force graph, a noisy signal, accuracy errors, and error spikes. Reasons for these errors can be explained by the rotatory brush, the tool-perpendicularity and force controllers working simultaneously and conflicting with each other, vibrations generated from the mecanum wheels and imperfections on the ground, or modeling errors in the whole-body kinematic model. Another reason is that there is not much passive compliance in the system, only the inherent compliance of the tool, arm, base, and pipe. Adding more compliance in the tool would reduce force spikes and slow down the contact dynamics which makes it easier for the controller to handle impacts.

VI. CONCLUSIONS

A semi-autonomous control architecture in mobile manipulators is proposed in this paper that takes advantage of the robot and human skills to create a symbiosis with the potential to further automate and improve force-tracking tasks. The architecture is decoupled into two separate controllers, one acting on the end-effector where the force-tracking is taking place, while the second handles the base motion based on teleoperator commands. A redundancy resolution scheme is used in a novel way to harmonize motions between the arm and base. Full motion control of the base is provided to the teleoperator by blocking the direct contribution of these DOFs to the motion in the end-effector. This presents a very intuitive interface for the operator to generate very agile motions in the platform, but in case of incompatible motions between the base and arm, the arm will stop following the task space requirements and end up being stuck in a near out-of-workspace singularity, so it is the task of the operator or a parallel supervisor process to prevent this from happening. This configuration provides safety and maintains full control in the hands of the operator. It avoids the redundancy resolution from taking over the mobile base in singular configurations to compensate for the primary task motion request.

In future work, we also plan to develop a fully automated version, by removing the human teleoperator out of the control loop. We can achieve this by optimizing the trajectory of the base to provide maximum maneuverability to the arm while the arm executes its assigned task in the end-effector.

REFERENCES

[1] Ghodsian, Nooshin, Khaled Benfriha, Adel Olabi, Varun Gopinath, and Aurélien Arnou. 2023. "Mobile Manipulators in Industry 4.0: A Review of Developments for Industrial Applications", Sensors 23, no. 19: 8026. doi:10.3390/s23198026

[2] C. Scheurer, M. D. Fiore, S. Sharma and C. Natale, "Industrial implementation of a multi-task redundancy resolution at velocity level for highly redundant mobile manipulators", Proceedings of ISR 2016: 47st International Symposium on Robotics, Munich, Germany, 2016, pp. 1-9.

[3] Bai, Q., Li, P., Tian, W. et al. "Coordinated motion planning of the mobile redundant manipulator for processing large complex components", Int J Adv Manuf Technol 121, 6703–6721 (2022). doi:10.1007/s00170-022-09785-x

[4] Hongjun Xing, Ali Torabi, Liang Ding, Haibo Gao, Zongquan Deng, Vivian K. Mushahwar, Mahdi Tavakoli, "An admittance-controlled wheeled mobile manipulator for mobility assistance: Human–robot interaction estimation and redundancy resolution for enhanced force exertion ability", Mechatronics, Volume 74, 2021, 102497, ISSN 0957-4158, doi:10.1016/j.mechatronics.2021.102497.

[5] M. Mashali, R. Alqasemi and R. Dubey, "Task priority based dual-trajectory control for redundant mobile manipulators", 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), Bali, Indonesia, 2014, pp. 1457-1462, doi:10.1109/ROBIO.2014.7090539.

[6] F. Farelo, R. Alqasemi and R. Dubey, "Optimized dual-trajectory tracking control of a 9-DoF WMRA system for ADL tasks", 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 2010, pp. 1786-1791, doi:10.1109/ROBOT.2010.5509554.

[7] Y. Wu, E. Lamon, F. Zhao, W. Kim and A. Ajoudani, "Unified Approach for Hybrid Motion Control of MOCA Based on Weighted Whole-Body Cartesian Impedance Formulation", in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3505-3512, April 2021, doi:10.1109/LRA.2021.3062316.

[8] J. Zhao, A. Giammarino, E. Lamon, J. M. Gandarias, E. D. Momi and A. Ajoudani, "A Hybrid Learning and Optimization Framework to Achieve Physically Interactive Tasks With Mobile Manipulators", in IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 8036-8043, July 2022, doi:10.1109/LRA.2022.3187258.

[9] J. Pankert and M. Hutter, "Perceptive Model Predictive Control for Continuous Mobile Manipulation", in IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 6177-6184, Oct. 2020, doi:10.1109/LRA.2020.3010721.

[10] M. Spahn, B. Brito and J. Alonso-Mora, "Coupled Mobile Manipulation via Trajectory Optimization with Free Space Decomposition", 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 12759-12765, doi:10.1109/ICRA48506.2021.9561821.

[11] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control", 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3411-3417, doi:10.1109/ICRA.2017.7989388.

[12] Wang Z, Liu W, Cui B, He J, Li Z, Zhao Y. "Research on Robot Surface Tracking Motion Based on Force Control of Six-Axis Wrist Force Sensor", Advances in Mechanical Engineering. 2015;7(1). doi:10.1155/2014/249696

[13] J. Li, Y. Guan, H. Chen, B. Wang and T. Zhang, "Robotic Polishing of Unknown-Model Workpieces With Constant Normal Contact Force Control", in IEEE/ASME Transactions on Mechatronics, 2022, doi:10.1109/TMECH.2022.3216314.

[14] Cetin, Kamil Zapico, Carlos Tugal, Harun Petillot, Yvan Dunnigan, Matthew Erden, Mustafa Suphi. (2021). "Application of Adaptive and Switching Control for Contact Maintenance of a Robotic Vehicle-Manipulator System for Underwater Asset Inspection", Frontiers in Robotics and AI. 8. doi:10.3389/frobt.2021.706558.

[15] J. Moura, W. Mccoll, G. Taykaldiranian, T. Tomiyama and M. S. Erden, "Automation of Train Cab Front Cleaning With a Robot Manipulator", in IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3058-3065, Oct. 2018, doi:10.1109/LRA.2018.2849591.

[16] Harun Tugal, Kamil Cetin, Yvan Petillot, Matthew Dunnigan, Mustafa Suphi Erden, "Contact-based object inspection with mobile manipulators at near-optimal base locations", Robotics and Autonomous Systems, Volume 161, 2023, 104345, ISSN 0921-8890, doi:10.1016/j.robot.2022.104345.

[17] Tugal, H., Cetin, K., Petillot, Y. et al. "Manipulation at optimum locations for maximum force transmission with mobile robots under environmental disturbances", Auton Robot 46, 769–782 (2022), doi:10.1007/s10514-022-10050-z

[18] S. Chiaverini, B. Siciliano and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator" in IEEE Transactions on Control Systems Technology, vol. 2, no. 2, pp. 123-134, June 1994, doi:10.1109/87.294335.

[19] Tan Fung Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," in IEEE Transactions on Robotics and Automation, vol. 11, no. 2, pp. 286-292, April 1995, doi:10.1109/70.370511.

[20] Zhang, Xinglei, Binghui Fan, Chuanjiang Wang, and Xiaolin Cheng. 2021. "An Improved Weighted Gradient Projection Method for Inverse Kinematics of Redundant Surgical Manipulators", Sensors 21, no. 21: 7362. doi:10.3390/s21217362

[21] Stanisic, R., Fernández, Á. (2012). "Adjusting the parameters of the mechanical impedance for velocity, impact and force control", Robotica, 30(4), 583-597. doi:10.1017/S0263574711000725