Teleoperating a robot for removing asbestos tiles on roofs: insights from a pilot study

Serena Ivaldi, Edoardo Ghini

To cite this version:

Serena Ivaldi, Edoardo Ghini. Teleoperating a robot for removing asbestos tiles on roofs: insights from a pilot study. IEEE ARSO - 2023 IEEE International Conference on Advanced Robotics and Its Social Impacts, Jun 2023, Berlin (Germany), Germany. 10.1109/ARSO56563.2023.10187476 . hal-03996168v2

HAL Id: hal-03996168
https://hal.science/hal-03996168v2

Submitted on 26 Apr 2023

Teleoperating a robot for removing asbestos tiles on roofs: insights from a pilot study

Serena Ivaldi[1], Edoardo Ghini[1]

***Abstract*— Construction robots may one day replace workers in dangerous operations such as the removal of asbestos cement tiles on roofs, an operation that exposes workers to several risks to their health. We argue that such robots will be teleoperated, to enable expert workers to supervise or directly control the operations, leveraging their knowledge of the field and facilitating the adoption of robots. We developed a prototype of the teleoperation and control interface of a prototype robot for roof tiles removal operations, combining a graphical interface, joysticks and a digital twin. We report on the rationale behind our choices, as well as the lessons learned after a pilot study.**

I. INTRODUCTION

Asbestos is a naturally occurring mineral made of thin microscopic fibers with fire-resistant and insulating properties, which made it widely used in the last century in construction as an additive. Unfortunately, exposure to asbestos fibers can cause serious health problems such as lung cancer and mesothelioma. The problem with asbestos in existing buildings is that the material can deteriorate over time, releasing fibers into the air and putting the building's occupants and those working on the removal at risk of exposure. This is a worldwide problem with significant health and economical implications: in France, for example, asbestos is responsible for 2200 new cancers and 1700 deaths per year [1].

The cost of removing asbestos from existing buildings is high, as it requires specialized workers, equipment, and procedures to safely remove the hazardous material. The standard procedure consists in sealing off the site, installing a ventilation system that removes airborne fibers during the removal process, removing the contaminated material and placing it in sealed containers for disposal. These procedures are complex but feasible for indoor and closed spaces but become even more complex for outdoor and large buildings that are directly exposed to the air. Workers have to wear extremely uncomfortable insulation suits and respirators to not breathe the toxic powder released by the disgregation of the asbestos. Putting on and off these suits is time-consuming. Safety regulations to keep workers away from asbestos exposure generally require that each worker cannot work more than a couple of hours consecutively in a day, and with the long period of decontamination after each shift, the current operations result in high costs for companies.

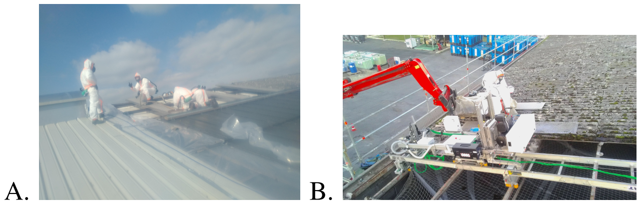

Fig. 1: Removing asbestos tiles on roofs. A: specialized workers wearing protective suits. B: prototype of RODDE, the mobile robot built by Isotop Etancheité, to replace workers in dangerous operations. [Credits: Isotop Etancheité]

This situation led to a growing demand for alternative methods to remove asbestos, and robotics technologies are among the promising leads. In particular, teleoperated robots could replace workers for some operations, thus reducing the exposure of workers to asbestos. A robotic system that is partially operated/supervised from a remote safe location by an expert operator of the field is a solution that facilitates the adoption of the robots, as well as exploiting the operators' knowledge of the field to deal with difficult situations. Teleoperating a remote robot has been widely explored for nuclear robots [2], space robots [3], and even humanoid robots for disaster response and telepresence [4].

While in recent years many construction robots have been proposed, for example for beam assembly [5] and ground excavation [6], robots for asbestos removal do not yet exist on the market, so it is very important to learn lessons from early prototypes and pilot studies. The Bots2Rec project [7] pioneered the construction of a mobile platform to assist in the rehabilitation of rooms and corridors, by providing disk grinding for the removal of plastering or paint from walls and ceilings, and surface scarification and grinding of ceramic tiles. The Telemovtop project [8] focused on the teleoperation of a mobile platform to replace workers on roofs in the process of removing asbestos cement roof tiles. This procedure is currently done manually by workers equipped with protections, unscrewing and carrying heavy asbestos cement sheets at the edge of the roof. In addition to the exposure to asbestos fibers, it is a risky process because it is carried out at height and is non-ergonomic because it involves manipulations bent on the knees and carrying heavy objects in constrained positions. A prototype robot, RODDE, was built by IsoTop Etancheité (see Fig. 1). This paper outlines the teleoperation and control of RODDE, and reports on the preliminary results of a pilot study.

In particular, our contribution is as follows: 1) We propose

---

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France `serena.ivaldi@inria.fr`

This work was partially supported by the Region Grand Est in France (project TELEMOVTOP), the European Union's Horizon 2020 and Horizon Europe research and innovation programmes under GA No. 731540 (An.Dy), No. 101070596 (EuRobin) and the CHIST-ERA project HEAP (CHISTERA-17-ORMR-003).

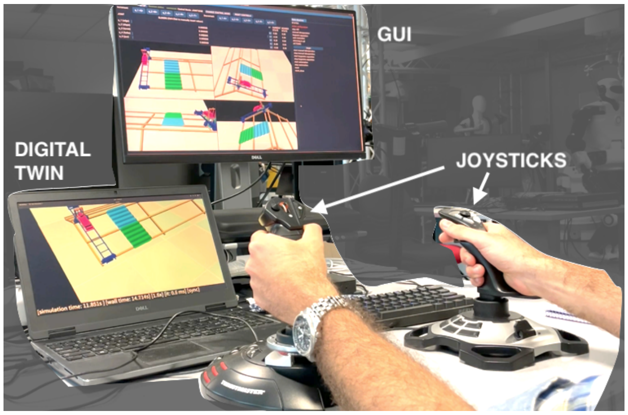

Fig. 2: Our prototype teleoperation solution: the operator can manually move the robot with two joysticks while looking at the camera feed (cameras embedded on the robot); or, with keyboard and mouse interacting with a Graphical User Interface, enabling more high-level commands. A digital twin can be used as a "third camera", to compensate for the missing visual from the cameras and to provide the operator with a higher situation awareness.

a system to teleoperate RODDE on roofs composed of several modules: a graphical user interface, a digital twin, a controller for the actuators, and the integration of input and sensing peripherals (e.g. joysticks and cameras); 2) We present insights and lessons learned in the pilot study about key elements of the robotics solution, which are the human operator interface and the camera feed / digital twin mixture for improved situational awareness.

II. OVERVIEW OF THE TELEOPERATION SYSTEM

Our teleoperation solution is shown in Fig. 2. It is thought with the end user in mind: she/he is a worker expert in asbestos removal operations and general construction works; for this application, she/he will be inside a site office, not necessarily next to the building to be treated; inside the office, she/he will have a computer and teleoperation devices, she/he will not be able to quit the office and will rely on visual feedback from the cameras to see the robot in action. The network communication in several construction and building sites is not ideal, so the system has to be designed with an imperfect communication network in mind. For this reason, closing the control loop on the operator's side does not seem a viable solution [9]. The operator should communicate commands to the robot, receiving enough information from onboard sensors to correctly perceive the robot's status and environment. A comprehensive view of the robot's status on the roof is not available, and using drones to provide such a view is not possible because of the further risk of spreading airborne asbestos fibers.

Similarly to what is developed for robots teleoperated in other areas [7], [10], [3], we created a system combining a Graphical User Interface (GUI), teleoperation devices (joysticks) and a digital twin. Using a Virtual Reality system was considered not suitable for workers [11].

The robot is controlled in two modalities: a direct control modality by teleoperation, either joint by joint, or by Cartesian position on the roof. Buttons and sliders can be used to give the input, and also the keyboard/mouse or joysticks used in flight simulators or video games/game consoles. To directly teleoperate the robot, the operator will use two joysticks: this solution was quickly identified after pilot studies as the closest man-machine interface to the one used by workers on construction sites for operating machines. The joysticks are configured to control the robot axis by axis, in a way that is friendly to operators.

The GUI informs the operator about the status of the robot, its configuration (joints and positions of each relevant link in the space and on the roof), and lets the operator configure the behavior and the tasks that the robot must execute. The GUI is mostly thought with interactive control and supervision in mind, although it also enables direct commands to the robot at a low level. The GUI also offers a number of high-level features, such as optimized robot maneuvers for executing critical tasks (e.g., automatic transportation of tiles to a drop location), and high-level "programming" of action sequences to automatize the tile removing procedures, which enable the operator to become a "supervisor" of the robot's work with different degrees of autonomy.

Visual feedback from the robot's cameras is shown in dedicated windows. To increase the number of point of views on the robot and improve the situational awareness, a digital twin aligned with the robot's on the roof is shown in a side window with a bird's view that can be changed by the operator according to their needs.

A. *Software architecture*

The robot's computer runs the Robot Operating Systems (ROS) to control the robot and acquire data from the sensors. It receives the command information from the operator's computer, running the GUI and the Digital Twin. In our pilot experiments, the robot was simulatedin the operator's PC.

B. *The robot*

Isotop Etancheité developed RODDE (Fig. 1), a robot prototype that moves on rails on the roof. Mostly made of linear actuators, it has a rotational base and its end-effector consists of two forks that close as a clutch to hold a tile. These add up to a total of five joints, four prismatic and one rotational: the first two prismatic joints are rails used to move on the surface of the roof, then there is the rotational joint followed by the last two prismatic joints that compose the end-effector. The surface in which the robot is moving might be inclined, depending on the slope of the roof, and thus particular attention must be put when it is carrying a heavy tile, as the tile can easily slip if the robot moves too fast. Once a tile is grasped, the robot rotates toward the top or the bottom of the roof, where a tile-dropping location is set on a transportation trolley.

To contain the costs, the robot is equipped with limited sensing: in particular, only two standard cameras are mounted on its structure.

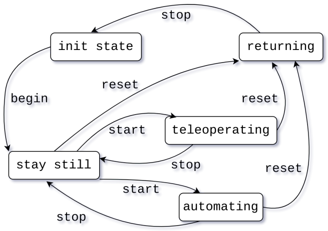

Fig. 3: State machine showing the basic control principle of the robot.

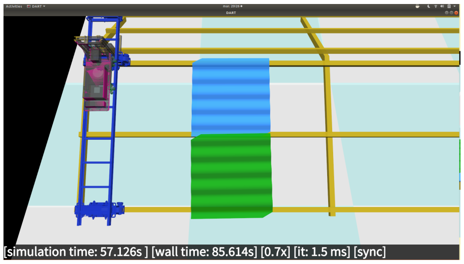

Fig. 4: Robot simulation in dartsim.

The robot's actuators only allow position control. We implemented a ROS controller that acts as a medium between the operator's commands and the real robot. It enables two modes of control: first, direct joint position control; second, Cartesian control of the end-effector, implemented as an Inverse Kinematics control module solving a quadratic programming problem with the constraint of the robot's dynamics, extracted from the URDF (Unified Robot Description Format) description of the robot. This latter is built based on the CAD model of the robot. We used the inria_wbc [12] whole-body robot control framework to implement the controller since the RODDE robot is assimilated to a floating base manipulator because its base moves in the $x - y$ dimensions on the roof surface and the base movement can be used to generate suitable Cartesian trajectories at the level of the forks (useful when the tile must be grabbed), subject to to the usual constraints of joint and velocity limits. The whole-body control is implemented with a stack-of-tasks approach: the first task, with high priority, is the desired end-effector pose, while the second, with low priority, is the default joint configuration which serves as a regularization task.

The robot's behavior is set by a *StateMachine* controlled by the operator, organized as a graph, as shown in Fig. 3. It allows transitions between different states (nodes) only by executing the corresponding actions (edges) determined by the operator in the GUI.

C. Digital Twin

The digital twin is basically a "virtual copy" of the robot in a reconstruction of its environment that is an approximation of reality. Ideally, it should behave as close as possible to the real robot in its real environment, and its status should be aligned with the real world using the robot's sensory feedback. In our case, we use physical engines to simulate the status and the dynamics of the system, which consists of three elements (see Fig. 4): the robot, the roof and the tiles. For each of these three elements we designed URDF models, which were used in robot_dart [13], a wrapper of the DART (Dynamics Automation and Robotics Toolkit) physical engine [14] that we use in other robots [15], [16]. Since the environment around the robot is bound to change, we also implemented a programmatic way of creating the roof of the building starting from a set of parameters (height, width, tiles dimensions, etc.).

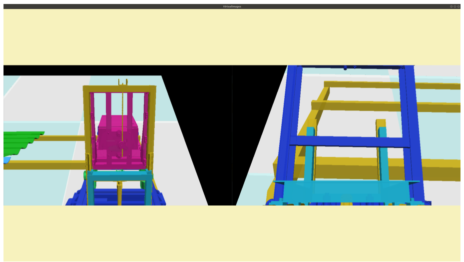

Fig. 5: Virtual cameras in the digital twin.

The digital twin can be used by the human operator to have a comprehensive overview of the robot on the roof, together with the status of the tiles: thanks to a bird's eye view that can be easily changed with a keyboard and mouse, the operators can improve their situational awareness and get a better idea of the orientation and position of the entire robot moving on the roof. As such, the digital twin provides the environmental camera that the real world cannot have.

Additionally, in the absence of the real robot, the digital twin can be used more classically as a system simulator: the operator's commands can be sent to the simulated robot's ROS controller, which provides the possibility to test controllers and teleoperation devices rapidly and safely. This simulator also offers the possibility to spawn virtual cameras in the simulation at specific positions and orientations (Fig. 5): this is used to simulate and stream virtual cameras to the operator's GUI. It can also be leveraged to simulate different locations for the cameras, which is useful to guide the choice of where to optimally place the cameras on the real platform.

D. Input peripherals

We use a pair of aeronautical joysticks that, in coordination with a mouse and keyboard, bestow the user with the ability to control the robot at various levels: both low-level control (controlling the robot joints with the joysticks) and high-level interaction with the automation panel on the GUI that requires a mouse and keyboard.

E. GUI

In a teleoperation system, the design of an intuitive and efficient GUI is a critical aspect: the goal is to enable a high level of performance in the tasks by the operator, which requires carefully choosing the information that is displayed to the user. The GUI should give the ability to the operator

ACTIONS:
begin
start manual teleoperation
start automation

CURRENT STATE:
initial configuration
stay still
manual teleoperation
auto execution
returning to q0

Gui state: RUNNING
Roscore status: connected
Control Mode: JOINTWISE
End-effector frame indicator status: EE non-vis

JOINT	SLIDERS	CURRENT STATE	DESIRED STATE	INCREMENT	DECREMENT	DIRECT VALUES MODIFICATIONS			
q_1 (edge)	2.917456	2.92	2.92	q_1	q_1	MIN VALUE	MAX VALUE		
q_2 (slope)	-1.911231	-1.91	-1.91	q_2	q_2	MIN VALUE	MAX VALUE		
q_3 (turret)	1.005092	2.04	1.01	q_3	q_3	0	-90	-180	-270
q_4 (forks)	0.161367	0.16	0.16	q_4	q_4	MIN VALUE	MAX VALUE		
q_5 (jaw)	0.021245	0.02	0.02	q_5	q_5	MIN VALUE	MAX VALUE		

KeyPoint Creation : (insert name)
Name :
KeyPoints list:
k1
k2

Current KeyPoint:
q_1: 0.000000
q_2: 0.000000
q_3: 0.000000
q_4: 0.000000
q_5: 0.000000
Delete KeyPoint:

Action Creation: (insert name)
Name :
select starting keypoint
select ending keypoint
Duration (s): 0.000000
Actions list:
a1

Current action:
Name : a1
Initial KeyPoint:
k1
Final KeyPoint:
k2
Duration : 67.000000
Insert action into current Sequence:
as last
Delete Action:

Sequence Creation: (insert name)
Name :
Select Sequence:
s1
Current Sequence:
Name : s1
Actions:
a1
Add action to the sequence:
select an action
as last
Insert this sequence into current playlist:
as last
Remove action from this sequence:
last
Delete Sequence:

Playlist Creation: (insert name)
Name :
Select Playlist:
p1
Current Playlist :
Name : p1
Sequences:
s1
Add sequence to the playlist:
select a sequence
as last
Remove sequence from this playlist:
last
Delete Playlist:

States:
Errors:

Fig. 6: Teleoperation Interface (GUI).

to handle routine as well as complex teleoperation tasks, informing on the status of the robot.

Our GUI was implemented with *Dear ImGui* [17]. It was composed of multiple windows: those for displaying the stream from the robot's cameras (or the virtual cameras, in the case of connecting to a simulated system), together with a main window containing the teleoperation interface (Fig. 6).

The first objective of the main window is to inform on the state of the robotic system (e.g. joint coordinates, robot connection status, eventual error messages) and to display statistics on the evolution of the dynamic system through a series of plots embedded in the window.

The second objective is to inform about the current session, which requires to link with the *State Machine*: the robot can be idle in a *init state*, and switch to the *teleoperating* mode; the user can command the robot to go back to its initial position (*returning*) or to follow the execution of an automated sequence of actions (*automating*).

Creating automated sequences is the third objective of the GUI. The automation functionalities are based on the following entities: a *Keypoint* that represents a robot configuration associated to a location; an *Action*, which requires a starting and an ending *Keypoint*;*Actions* are grouped in *Sequences*, and *Sequences* are grouped in *Playlists*. The *Automator* class contains references to all these entities in the form of associative maps. Typical Keypoints are the center of the side of the tiles, where the robot's forks are supposed to grab the tile, the initial position of the robot on the roof and the tile-dropping location at the border of the roof.

Finally, the GUI offers a simple interface for direct teleoperation of the robot, at the joint level and the Cartesian level (only for the end-effector and the base). Using sliders, the operator can choose how to change the robot's configuration. To smoothly move the robot between joint configurations, the GUI generates polynomial trajectories and sends them as references to the robot's ROS controller.

F. Pilot study

The pilot study consisted of three different validation tests of the system, with volunteer participants from the lab as well as staff of Isotop Etancheité, at Inria's premises and their headquarters. Due to the limited number of participants and their non-homogenous participation across the tests, we can only report on qualitative results. The pilot experiments were focused on testing: the display of information in the graphical user interface; the use of cameras, including the use of the digital twin for visual feedback; the devices and systems for direct teleoperation (joysticks, game pad, keyboard and mouse); the user experience in controlling the robot, setting the key points and sequences for automatizing the operations.

III. INSIGHTS FROM THE PILOT STUDY

A few insights gathered from the pilot studies are summarized below.

A. Digital twin: a useful tool for robot design

Building real robotic prototypes is expensive and time-consuming, so it is important to test all the possible ideas and requirements of a platform before launching the production. In this sense, the digital twin is extremely useful at the design stage. The robot's CAD model and few other information about its mechatronics are sufficient to have a physical simulation of the platform in a simplified reproduction of the environment, which enables the development of the robot planning and control system and preliminary tests of the robot's teleoperation. Our solution was based on robot_dart [13] for the simulator and inria_wbc [12] for the controller, two software pieces that were already used in our team to control different articulated robots such as Talos [15] and Franka [16]. The cost of including another robot is low when compared to the advantage of re-using existing working code; the success should encourage the development of new concepts of construction robots.

One interesting outcome of the simulation was to find an optimal positioning of the cameras on the robot: we tested different camera configurations attached to the robotic structure, to optimize the operator's field of view during the operation. With a limited budget for the cameras to be mounted on the real platform, deciding where to put cameras can be critical to enable visual feedback to the operators.

B. Situation awareness and cameras

Overall, the feedback from the participants highlighted a lack of situation awareness, and the "third view" of the digital twin in bird's eye view helped.

Teleoperating the robot using only two cameras was often very difficult or impossible without many trials and errors. The expert operators had a good idea of the robot's configuration, yet the camera feed was often not enough to act, because of the limited field of view: it does not provide enough "situation awareness". For this reason, many participants in the test trials teleoperated while looking at the digital twin: they reported it was useful to have a better idea of where the robot was on the roof and to align the forks pick the tile. To do it, the robot's forks should be open and aligned with the wavy pattern of the tile to ensure a robust grasp. Laboratory tests with the real robot prototype confirmed that this alignment is important to prevent slipping or tile oscillations. Unfortunately, it was difficult to identify the precise alignment of the robot in front of the tile without some training with the camera feedback alone. Of course, our pilot experiments were conducted with the digital twin only and it is difficult to assess if this was a limiting factor and predict if such issues will persist once the teleoperation is performed directly on the real platform.

Another solution to increase situation awareness could be to increase the number of cameras onboard, which could also enable in the long term to fully automatize the robot's tasks, as the experience of autonomous cars shows [18]. However, too many camera streams may be a problem for remote teleoperation, because of the limited bandwidth preventing high-resolution streaming and the extra cognitive effort that is required for users to switch among views. Although [19] reports that they streamed one 4K and four 2.5K cameras simultaneously with a latency of only 100ms, there are no details on the communication network. The networks that can be set up in many construction sites may not have good bandwidth and latency, which are critical for teleoperation [9]. Downgrading images could be a solution if more cameras are kept. Other perception devices could be used. For example, HEAP, a construction robot for excavation [19], uses LiDAR for perception because "they outperform camera-based sensors in heavy dust environments", but the perception feed is used for autonomous operations and not necessarily for visual feedback to human operators.

C. Joysticks

Preliminary tests with expert operators from the construction business highlighted that their preferred interface for teleoperating the robot was the joysticks. This means they preferred to be in direct teleoperation mode, rather than sending high-level commands to the robot through the interface or supervising the robot's autonomous activity. This result is in line with another experimental study we conducted in the past on a teleoperation setup with the Kinova Jaco arm [20], where we found that participants preferred to use joysticks than graphical interfaces even if they made more errors with the hand-held devices. Among the motivation for this preference, the participants reported that they felt more in control, they felt they could improve their performance over time (expected improvement), had fewer attention switches and could focus exclusively on the visual feedback (looking at the robot). The operators even asked us to implement the same mapping they had on existing hand-held devices equipped with joysticks that they already used on the construction sites to control other machines. The mapping was not necessarily ideal for our application, but it greatly facilitated the use of the teleoperation solution and reduced the time for training.

Of course, the joysticks were good to teleoperate the robot and demonstrate the removal operations, but the participants only tried the operations on a few tiles (less than ten). In the intended use of this robot, direct teleoperation should be an occasional operation where the operator would intervene in a complex situation; most of the time, the operator should be supervising the robot's autonomous or semi-autonomous operations and hence it should look at the camera feedback and use the GUI, as in other remote robot applications [10], [21].

When confronted with the supervision mode, the participants required more time to understand how to interact with the remote robot and its control interface. The concept of identifying the key points, the sequence and the repetition of operations was not intuitive for the construction workers; other participants (robotics engineers) familiar with those questions understood the principle but had issues when using the interface to configure the operations. We encountered similar issues as in [20], which means more attention switches and cognitive operations for the graphical interface, even if it could enable the operator to automatically perform certain operations without error, such as automatically aligning the robot forks to the roof tile.

This means the GUI and the user experience both need significant improvement. It is unclear if the "sequence programming" concept is intuitive for construction workers: interactive planning and supervised execution are very common in other areas of application of teleoperated solutions, such as in space applications [21], but the construction operators here are not naive to this application and it will probably require extra training and human studies to identify an efficient user experience. For sure, the UX and the graphical display of information will have to be improved. It is known that teleoperation performances can quickly degrade if interfaces do not consider human factors as well [22].

D. Automated operations

One problem that emerged during the tests, and which was not foreseen in advance, was the automation of the robot's trajectories during the transport of the roof tiles, which risks slipping and falling from the roof, with significant health and safety risks (problem of dust and asbestos waste). This problem was addressed by [23] in the case of slip prevention during nonprehensile object transportation. We implemented their solution for planning and control of the robot's twist when the forks are holding the tile and found that slipping could be reduced if the robot had an additional degree of freedom that could change the elevation angle between its horizontal axis and the surface of the roof. Adding this extra joint would be very expensive and drastically change the robot's mechatronics. The retained solution is to significantly limit the speed of the robot during transportation, reducing the speed of rotation of the main axis. This result is consistent with the heuristics found by the mechatronics engineers when developing and testing the prototype in their lab with a tile held by the forks. A better estimation of the friction coefficients between the forks and the sheets could help to automatically find the recommended speed values for the operators when deploying the robotic solution: however, this identification activity cannot be done in simulation, it will have to be done on the robot in a real or analogous situation with real sheets.

Limiting the speed automatically could be a critical element for the safety of the final robotic system. Indeed, during the pilot study, the participants often found that the robot was too slow and manually increased the speed limit using the sliders on the GUI. However, this was the major cause of slipping when the robot was rotating to align with the rail and go back to the dropping location. While these errors were not dramatic since the robot was simulated, they could be with the real robot. Again, in our pilot, the operators were interacting with a simulated robot, which could create the illusion of a game and induce participants to take on risky behaviors [24]. But such behaviors must be anticipated and it could be the role of the robot controller to automatically set new safety constraints that consider the behavior of the operators too. Another possibility, that we explored in the pilot, was to let human operators supervise the robot during the transportation task: once the robot has grabbed the tile with the forks, the operator could use the quick actions in the GUI to command the automatic alignment of the robot and return to the dropping location. The robot's behavior, optimized, was executed successfully, but the participants were inactive during that time and so their level of engagement during the many seconds of this action dropped. Supervised execution should be the goal of our robotics solution, but we wonder if it will not cause disattention in operators in the long term [25]. Future human studies will have to study operators' behavior and assess this risk.

IV. CONCLUSION

In this paper, we report on the preliminary findings of a pilot study about teleoperating a construction robot for asbestos removal on roofs. We discuss the rationale behind our choices for the teleoperation and control interface and report on the insights that we gained after the first test trials with naive and specialized operators. The main take-home message is that while the robot direct teleoperation system is relatively easy to implement with existing software used in labs for other -even more complex- redundant robots, much work must be done to improve the user experience and make it easier for the operators to directly control and supervise the robot's tasks execution. Considering the specific population of end users, future work will focus on improving the human-machine interface and providing better situational awareness.

ACKNOWLEDGMENTS

The authors would like to thank Jean-Baptiste Mouret, Eloise Dalin and Olivier Rochel for their help with the software; Josselin Schumaker for his help with the URDF of the robot; Stanislas de Benoist and the staff of the company Isotop Etancheité for their collaboration in the project.

REFERENCES

[1] [Online]. Available: https://www.santepubliquefrance.fr/determinants-de-sante/exposition-a-des-agents-physiques/amiante

[2] V. Pruks, K.-H. Lee, and J.-H. Ryu, "Shared teleoperation for nuclear plant robotics using interactive virtual guidance generation and shared autonomy approaches," in *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE, 2018, pp. 91–95.

[3] P. Schmaus, D. Leidner, T. Krüger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, "Preliminary insights from the meteron supvis justin space-robotics experiment," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3836–3843, 2018.

[4] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, pp. 1–22, 2023.

[5] K. Jung, B. Chu, S. Park, and D. Hong, "An implementation of a teleoperation system for robotic beam assembly in construction," *Intern. Journ. of Precision Enginering and Manufacturing*, vol. 14, no. 3, pp. 351–358, 2013.

[6] J. S. Lee, Y. Ham, H. Park, and J. Kim, "Challenges, tasks, and opportunities in teleoperation of excavator toward human-in-the-loop construction automation," *Automation in Construction*, vol. 135, p. 104119, 2022.

[7] T. Detert, S. Charaf Eddine, J.-C. Fauroux, T. Haschke, F. Becchi, B. Corves, R. Guzman, F. Herb, B. Linéatte, and D. Martin, "Bots2rec: introducing mobile robotic units on construction sites for asbestos rehabilitation," *Construction Robotics*, vol. 1, no. 1, pp. 29–37, 2017.

[8] [Online]. Available: https://www.inria.fr/en/telemovtop-asbestos-removal-robotics

[9] H. Zhu, M. Sharma, K. Pfeiffer, M. Mezzavilla, J. Shen, S. Rangan, and L. Righetti, "Enabling remote whole-body control with 5g edge computing," in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and systems (IROS)*, pp. 3553–3560.

[10] D. W. Hainsworth, "Teleoperation user interfaces for mining robotics," *Autonomous Robots*, vol. 11, no. 1, pp. 19–28, 2001.

[11] H. Yamada, T. Xinxing, N. Tao, Z. Dingxuan, and A. A. Yusof, "Teleoperation construction robot control system with virtual reality," *IFAC Proceedings Volumes*, vol. 42, no. 16, pp. 639–644, 2009, 9th IFAC Symposium on Robot Control.

[12] [Online]. Available: https://github.com/resibots/inria_wbc

[13] [Online]. Available: https://github.com/resibots/robot_dart

[14] [Online]. Available: http://dartsim.github.io/

[15] E. d'Elia, J.-B. Mouret, J. Kober, and S. Ivaldi, "Automatic tuning and selection of whole-body controllers," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 935–12 941.

[16] L. Vianello, W. Gomes, F. Stulp, A. Aubry, P. Maurice, and S. Ivaldi, "Latent ergonomics maps: Real-time visualization of estimated ergonomics of human movements," *Sensors*, vol. 22, no. 11, p. 3981, 2022.

[17] [Online]. Available: https://github.com/ocornut/imgui

[18] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.

[19] D. Jud, S. Kerscher, M. Wermelinger, E. Jelavic, P. Egli, P. Leemann, G. Hottiger, and M. Hutter, "Heap - the autonomous walking excavator," *Automation in Construction*, vol. 129, p. 103783, 2021.

[20] S. Marichal, A. Malaisé, V. Modugno, O. Dermy, F. Charpillet, and S. Ivaldi, "One-shot evaluation of the control interface of a robotic arm by non-experts," in *Social Robotics: 8th International Conference, ICSR 2016*. Springer, 2016, pp. 458–468.

[21] W. Pryor, B. P. Vagvolgyi, A. Deguet, S. Leonard, L. L. Whitcomb, and P. Kazanzides, "Interactive planning and supervised execution for high-risk, high-latency teleoperation," in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 1857–1864.

[22] J. Y. C. Chen, E. C. Haas, and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1231–1245, 2007.

[23] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, and P. R. Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 569–583, 2022.

[24] C. de Juan-Ripoll, J. L. Soler-Domínguez, J. Guixeres, M. Contero, N. Álvarez Gutiérrez, and M. Alcañiz, "Virtual reality as a new approach for risk taking assessment," *Front. in Psych.*, vol. 9, 2018.

[25] F. Penizzotto, E. Slawinski, and V. Mut, "Method to estimate human inattention in teleoperation of mobile robots," *Journal of Control Engineering and Applied Informatics*, vol. 16, no. 3, pp. 94–105, 2014.