MoCap-based teleoperation and stylus teaching by demonstration for skill-based collaborative robotics

Shruti Chakraborty[1], Adib Rebbouh[1], Camille Vindolet[1], Baptiste Gradoussoff[1*]

[1] *Université Paris-Saclay, CEA, LIST, F-91120 Palaiseau, France*
**baptiste.gradoussoff@cea.fr*

Keywords: Collaborative robotics, Teaching by demonstration, Motion capture, Teleoperation, Virtual guides

Abstract

Low-volume, high-mix manufacturing environments pose significant challenges for automation due to their high variability in tasks and products, and their reliance on operator observation and expertise. Conventional industrial robotic solutions excel at deterministic and repetitive tasks, but fail to provide adequate solutions for such use-cases. And while collaborative robotics typically aim at natural interaction with operators, they mostly still rely on conventional robot programming, leading to the same limitations for quick deployment and task adaptation. These methods are labour-intensive and require skilled experts to set-up and maintain which is often too costly for use-cases requiring frequent adaptation.

To address these limitations, we propose a skill-based collaborative robotics toolbox that enables intuitive and flexible robot use through teaching by demonstration. We leverage motion-capture technology to perform intuitive and low-cost teaching. Through teleoperation-based or stylus-based demonstrations, it allows operators to effortlessly teach, parametrize and trigger robot skills, on demand, by simply teaching the robot it where the requested skill should be applied. To further improve ease of teaching, we leverage the use of user assistances such as virtual guides and data post-processing. We validate our approach through an industrial case-study involving repetitive screwing tasks, demonstrating improved setup time, task adaptability, and human-robot collaboration efficiency.

Our results indicate that motion-capture-assisted demonstration offers a viable solution for flexible and very cost-effective interactive robotics in dynamic manufacturing settings.

1. Introduction

Modern automation in manufacturing, specifically involving collaborative robots (or cobots), faces a challenge to generalise to small batches that require frequent revisions and adaptive processes. This creates a need for fast, efficient and intuitive programming tools [1]. Traditional programming tools for industrial robots, such as offline programming or teach pendants, are too time consuming and expensive to be employed when dealing with variable environment settings, as they require significant time for robot-cell setup, robot reprogramming and worker training. Our work focuses on simplifying the programming workflow to make human-robot collaboration intuitive and accessible to non-roboticists in an industrial setting. We propose an approach which focus on simplicity and efficiency, by developing an intuitive Motion Capture (MoCap) based solution that allows operators to easily teach points to configure and trigger task-specific robotic skills.

1.1. State of the art

With recent advances in generative AI, there has been large interest in training end-to-end foundation models for developing adaptable general-purpose robots [2][3][4]. Despite some promising results, end-to-end approaches lack robustness and explainability compared to classical control approaches, as the robot behaviour cannot be guaranteed. This is a major hurdle for industrial applications, where determinism is mandatory for persons and process safety.

Skill-based learning is another approach, which allow teaching via deterministic or probabilistic description [5][6][7]. In skill-based systems, the low-level motion primitives constitute blocks that can be combined into skills, further combined into tasks, linking the low level to the high-level description [8][9][10][11][12]. This allows non-expert users to program the robot, only focusing on the skill/task level without worrying about low level implementation [8][10]. The skills can be parameterised manually, via a Graphical User Interface (GUI), through human-robot interactions [8] or via external vision or force sensors [12].

Common skill teaching methods include kinesthetic teaching / hand guiding; teleoperation [1]; visual observation (which allows robots to learn by perceiving human actions and the environment through vision systems) [5]; and Multimodal Guiding (which combines multiple communication channels such as speech, gestures, visual cues, and haptic interaction for a more natural and robust instruction process) [9]. Teleoperation systems yield higher-quality demonstrations than hand-guiding but require slightly more teaching time [1].

Robot teleoperation is a well-studied area of research, and has vast applications especially in nuclear activities [13]. High performance force feedback teleoperation can be achieved either with industrial arms equipped with external force/torque sensor, or specialised arms designed for high mechanical transparency, that exploit motor current for force estimation and feedback [13]. For the best performance, these solutions rely on costly high-fidelity haptic devices called "master"

https://doi.org/10.1049/icp.2025.3686

arms, such as the Haption Virtuose 6D [14], or the Force Dimension Omega/Sigma [15]. They also rely on passivity-based control algorithms, which require high-frequency real-time control and communication frameworks such as TAO2000 [16], CORTEX [17], or ROS2 [18] to provide stable teleoperation. Master arms are also limited in movement amplitude, which can hinder teaching in some situations, such as gesture or trajectory teaching.

Motion Capture (MoCap) teleoperation, using position tracking hardware, and VR/AR teleoperation (using an added VR/AR headset) offer a favourable trade-off: much lower cost, complete freedom of movement and lower constraints on robot compatibility, at the price of loss of force feedback. We propose to compensate for this loss with basic vibro-tactile feedback and added user assistances such as virtual guides and collision avoidance. Recent literature leverages consumer-grade VR/AR hardware (e.g., HTC Vive, HoloLens) for teleoperation [19][20]. Although ergonomic and easy to integrate, these solutions suffer from communication instability and latency issues, limiting teleoperation to slow tasks.

1.2. Proposed approach

We propose to combine the advantages of skill-based programming (for high-level robot control), conventional robot control (for determinism), and motion-capture teaching (for ease of use), to achieve a very cost effective and efficient way to deploy collaborative robots in high mix / low volume industrial environments.

With the Skill-based paradigm, literature focus mainly on skill automation, robot programming and task sequencing. Our approach differentiates itself by focusing on interactive use, to provide a skilled cobotic assistance to the operator. We focus on a single skill execution workflow, which allows quick teaching for on-the-fly skills execution. The operator simply selects a predefined skill, teaches the points of application of the skill, then asks the robot to execute it. It makes the approach highly versatile, reliable, fast and easy to use for the non-experts.

In this article, we use cost-effective and high-performance MoCap Hardware to develop and evaluate our skill-based MoCap-teaching toolbox. It provides two teaching modalities (Teleoperation and Stylus), user assistances to enhance ergonomics and teaching efficiency, and use-case specific skills.

2. Technical Implementation

2.1. Architecture

We selected a Motion-Capture system from A.R.T. (AR-Tracking) for its precision, reasonable price, and ease of deployment [21]. The ART SmartTrack3 camera claims sub-millimetric precision at high frequency (150Hz to 240Hz), within a 2 meters range. Integration is also facilitated, as the camera embeds its own controller. The setup includes the Smarttrack3 camera, and the Flystick2+ controller, as illustrated in Fig 1., which shows a similar setup with Doosan robot.

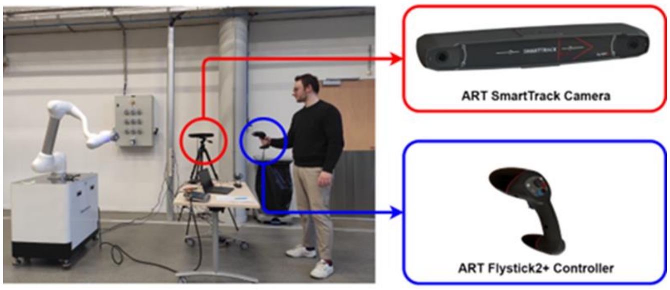

Fig. 1 - Example of MoCap Teleoperation setup

The full architecture detailing the interactions between the used hardware and developed software is presented in Fig. 2.

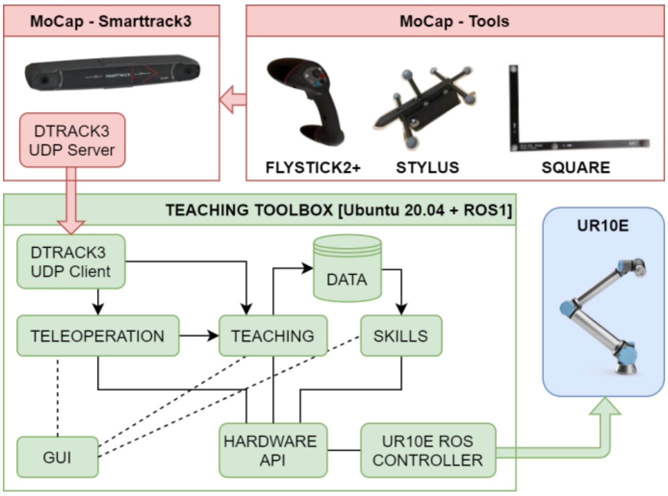

Fig. 2 - Software architecture

The operator interacts with the teaching hardware and the GUI to teleoperate the robotic arm with the Flystick, or teach directly with the stylus. The calibration square is used to define the teleoperation frame of reference. The "Hardware API" gives a standard interface to communicate with the drivers of various robots to allow transferability of skills and teleoperation features.

2.2. Teaching and skill execution workflow

Our MoCap teaching toolbox provides motion-capture teaching modalities, user assistances, and skill automation. The general workflow consists of a development phase, where the roboticist programs high-level, configurable skills (not discussed here), and a usage phase, where the operator simply uses available skills and configures them via teaching by demonstration of points or trajectories.

The operator workflow, illustrated by Fig. 3, revolves around the idea of improving ease of teaching through user assistances. The cycle is the following:

- Selection of adequate teaching modality: depending on the needs, the operator can use teleoperation or stylus teaching.
- Teaching of points to trigger user assistances: virtual guides (to limit robot degrees of freedom in teleoperation) and post-processing functions (such as orientation

alignment) typically need a frame of reference, which can be taught by touching 3 points (O, X, Y) on the surface of the desired object.

- Trigger of desired user assistance.
- Teaching of the actual points needed for skill execution.
- Skill configuration and execution.

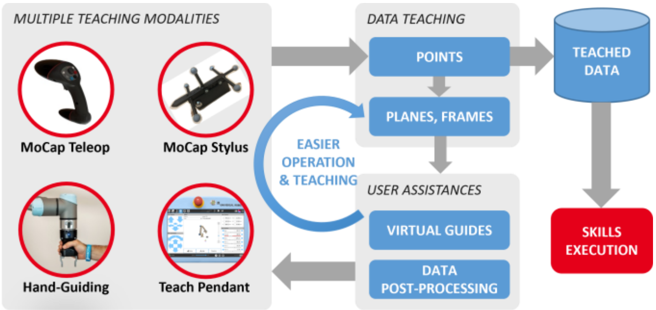

Fig. 3 - MoCap teaching toolbox conceptual workflow

2.3. *Teleoperation coupling*

Teleoperation based on motion-capture is a unilateral coupling: there is no force feedback from the robot to the master device. This has a dual impact on the user experience. Firstly, it increases risks of hardware degradation when the robot is interacting with the environment. To address this limitation, it is mandatory to use a force-compliant robot controller, to ensure progressive and limited interaction forces between the robot and the environment. Secondly, the lack of force feedback implies that there is no enforced position synchronization between robot and master device, which can lead to major tracking error if the robot cannot follow the user movements dynamics. To ensure satisfactory unilateral teleoperation behaviour, we evaluated several control approaches, illustrated by next figure.

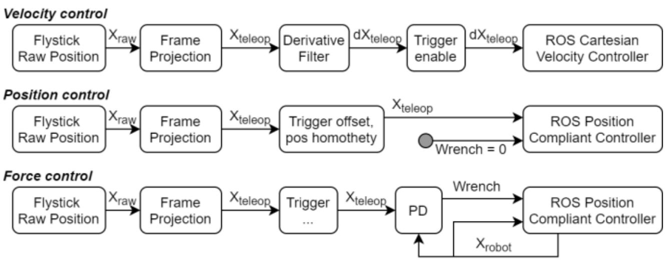

Fig. 4 - Tested robot control approaches

In the velocity-control strategy, the raw MoCap position is derived and filtered, and the cartesian velocity is sent to the robot controller. This approach has a couple advantages. First, it avoids position overshoots and perceived lag, especially when the user performs oscillatory movements. In case of position drift due to limited robot dynamics, the robot simply goes slower, but will not try to compensate the drift afterwards. Second, it makes physical interactions safer, as a maintained motion is needed on the master side in order for the robot to exert forces onto the environment. When the user stops the movement, velocity command is nulled, thus nulling the interaction forces as well. However, having to maintain velocity in order to maintain contact forces is also a constraint, which favors position control.

The position-control strategy, based on the FZI ROS compliant controller, which can take position and/or wrench inputs [22]. Here, the raw MoCap position is filtered and projected in the correct teleoperation frames, then sent to the robot controller. Wrench command is null. Position command is reset to stop the robot when the user releases the trigger. This approach ensures compliant behaviour, exact position homothety between master and slave, and safe interaction as well, since interaction forces are reset when the user releases the trigger (position command is reset to current robot pose when trigger is released).

The force-control strategy, also based on FZI controller, uses the wrench control input instead of position [22]. MoCap Position is sent to an external PD corrector, which outputs wrench commands for the robot. The position command is a copy of the current robot pose. This setup could be useful if implemented in a hard real-time environment, as it allows to shape the wrench command as needed (for example, limit wrench to max values). However, it didn't provide sufficient performance in our case, due to the python non-RT execution context: teleoperation was working, but the limited value in our PD corrector led to major position drift.

We chose the position control implementation, as it provides good overall performance and safe interaction, while being the only one without major drawbacks.

The ART "calibration square" provides the teleoperation frame of reference, and must be aligned approximately (± 10°) with the orientation of the robot base frame. During teleoperation and stylus-based teaching, the teaching devices must be in the field of view (FOV) of the camera. For safety, the robot motion is paused while the Flystick is out of the camera FOV.

2.4. *Teaching*

Two MoCap teaching modalities have been implemented and tested.

The **teleoperation teaching modality** is based on the previously discussed teleoperation implementation. One of the buttons on the Flystick is mapped to the "teach point" feature, and triggers a snapshot of the robot cartesian pose when pressed. Here, the precision of the MoCap system is not involved, since we record the actual robot pose.

The **stylus teaching modality** is using a dedicated measurement stylus, with a calibrated tooltip. This workflow is quite different, as the teaching part is fully decorrelated from the robot part. The steps are as follows:

1. Task frame mocap calibration. The operator teaches 3 points previously marked within the task environment, to define the task frame.
2. Teaching. All points are expressed in the task frame previously defined.
3. Task frame robot calibration. The operator equips the robot with a calibration tip, and teaches the 3 points using either MoCap teleoperation, hand-guiding, or the robot teach pendant.

4. Skill execution, using the taught points projected into the robot base frame.

With this modality, the precision is the direct result of the MoCap system precision, cumulated with both calibrations and robot precision. It is thus expected to have a lower precision compared with teleoperation teaching, but on the other side, stylus teaching is easier, and doesn't rely on robot performance for teaching.

2.5. User assistances

In order to improve the ease, efficiency and precision of teaching, each teaching modality can benefit from dedicated user assistances.

For teleoperation teaching, 6DOF remote control is challenging, especially when trying to perform precision movements. We implemented several user assistances. The first one, basic but effective, is the locking of rotations or translations. It reduces the control to 3DOF, and does not need any configuration, so it can easily be mapped on a dedicated button on the Flystick, and used on the fly by the operator. The second one is selectable position homothety. We implemented 3 levels of homothety (1:1, 1:2, 1:4 ratios), also mapped on a dedicated button, to enable robot fine positioning. The third one is virtual guides. For the needs of our use-case, we implemented a "plane" guide, which enforces the end-effector orthogonality to a given plane. This feature is powerful, but it needs more configuration to setup. The workflow is as follows:

1. Using teleoperation, operator teaches 3 points [O, X, Y] to define the plane frame. The goal is to identify the normal Z to the plane, so any tool/end-effector can be used for this step.
2. Operator moves the robot at the required distance from the plane
3. Operator activates the guide with the taught frame.
4. Robot automatically rotates to plane perpendicular.
5. Robot locks rotations Rx, Ry, Rz and translation Tz.
6. Operator moves the robot along Tx, Ty axis of the plane frame.

For stylus teaching, the previous assistances do not apply. Here, one of the main difficulties is the teaching of orientations: it's very easy to precisely teach [x, y, z] coordinates with the stylus pinpoint, but for orientations, the operator does not have much physical cues. We implemented a dedicated "align orientation" feature, which post-processes taught points to enforce their orientation along a given frame of reference. The workflow is similar to the one for the "plane" guide, the operator has to teach the plane frame to align points orientation along the z axis of this frame.

2.6. Skills

Skills aim at providing the operator with high level robotic functions that correspond to common tasks specific to his work, such as drilling, soldering, screwing, inserting skills. We developed a skills framework which is part of the toolbox, and aims at simplifying the development of new skills. This part is not discussed in this paper, as we focus on the operator workflow here.

3. Experimental Validation and Results

The effectiveness of our setup is determined by the global precision of the skill teaching and execution pipeline, as well as the user experience and comfort provided by the teaching modalities. To evaluate this, we chose an industrial screwing use case to deploy and test our toolbox.

3.1. Use-case and experimental setup

The task consists on screwing several hundreds of bolts, for the assembly of flat parts, in a context of very low volume series. The task is currently done by hand, as standard robotic solutions did not fit the cost-effectiveness and ease of use required here.

The experimental setup involves a mock-up of the task to be performed, a UR10e robot, a prototype electric screwdriver controlled via the robot I/O, an A.R.T MoCap system composed of a Smarttrack3 camera, a Flystick 2+ controller, a calibration square, and a prototype active stylus. The whole system is controlled via a Ubuntu 20.04 supervision computer hosting ROS for the robot control, and our skill-based MoCap-teaching toolbox. The setup time is minimal: the Smarttrack camera is simply put on a tripod at a convenient position to ensure adequate field of view for continuous tracking of the teaching devices, then connected to the supervision computer network. Camera intrinsic calibration was performed prior to the tests using dedicated calibration wand and in-built intrinsic calibration software.

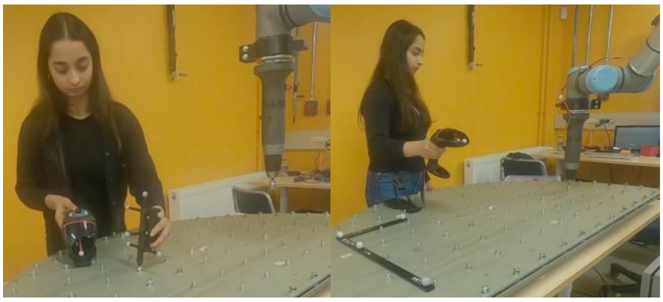

Fig. 5 - Left: Stylus teaching; Right: Teleop teaching

3.1.1. Screwing task experiments

To validate its practical use, the whole teaching and skill execution pipeline was evaluated on the screwing use case, on a subset of 70 bolts equally distributed on the task-space. Following the teaching procedure detailed in section 2.4, we tested both Teleop MoCap and Stylus MoCap teaching.

In both cases, after their respective initial calibration steps, the teaching phase started with teaching the task-plane frame, needed to enable the user assistances. Next, the positions of all 70 screwing points were taught. Once the teaching phase completed, the "screw" skill was executed on each of the taught points. The skill takes the list of taught screw positions and for each point generates a cartesian trajectory motion to position the robot on the bolts and power the screwdriver to perform the screwing task. The skill handles the robot and screwdriver motions, without particular optimization or searching strategy for socket insertion into the nut.

For teleoperation teaching, we used "plane alignment" and "orientation lock" assistances to improve ease of teaching and accuracy. The screwing task was successful, with 70 bolts engaged, and 69 tightened to the desired torque.

For stylus teaching, we used "points orientation alignment" post-processing function, to ensure that all points were correctly perpendicular to the reference plane. All 70 bolts were correctly engaged and tightened.

These first experiments show that both MoCap teaching modalities provide sufficient accuracy for performing this kind of "medium-precision" task.

3.2. System Precision

The overall system precision, for a full teach and skill execution cycle, depends on the teaching modality: stylus teaching combines MoCap calibration & tracking, user, calibration, and robot precision; while teleoperation teaching relies solely on user precision and robot repeatability.

3.2.1. MoCap camera noise

MoCap measurement noise was evaluated by positioning a single marker at a given distance from camera, and recording variations of positions returned by DTrack3 software over 10 seconds. Noise is larger along the depth Z-axis of the camera, but overall, it is very low and stays self-contained except at the very range limit of the camera. Thus, it has no relevant impact on the overall system precision.

Dist. from camera	X, Y camera axis	Z camera axis
1m	± 0.05 mm	± 0.1mm
1.5m	± 0.07 mm	± 0.2mm
2m (max range)	± 0.1 mm	± 0.5mm

Table 1 - Noise amplitude on a single marker position, for various camera distances

3.2.2. MoCap camera precision

The camera precision has direct impact on stylus MoCap teaching precision. It was evaluated by performing differential measurements of poses of 20mm markers, to compare the physical distance between two markers, and the equivalent computed distance from the MoCap measurements. The following table shows a few examples obtained after calibration of the SmartTrack, along different camera axis.

	Distance between two markers [mm]			
Main axis	**measured**	**theoretical**	**error**	**error %**
Z	725,4	724	+1,4	+0,20%
X	555,2	555	+0,2	+0,03%
X+Z	664,1	662,5	+1,4	+0,24%
X+Z	651,6	651,61	+0,0	+0,00%
Z	1268,9	1266,7	+2,2	+0,17%

Table 2 - Differential error between two markers

On paper, the max differential error is the cumulation of max error for both points, leading to an estimated point precision of ±1.1mm based on these measurements. However, given the sensitivity to camera calibration, more in-depth measurements should be performed to provide exact values. We consider from our various experiments that one can conservatively expect a better than ± 2mm precision for individual markers measurement, across the whole range of the SmartTrack3.

3.2.3. Overall precision for stylus teaching modality

The stylus teaching involves the most elements into the resulting precision. To provide a representative evaluation of its overall precision, we performed the following experiment. We equipped the robot screwdriver with a marker pen, and made a simple "point" skill for the robot to write a point on paper at each taught point. We printed a series of 35 millimetric targets on A4 paper, and conducted the measurements in the 4 main areas of the task space, for a total of 140 points of measurement (cf. next figure). Each target centre was taught precisely with the stylus, and precision was measured as the distance between target centre and the point printed by the robot. Mean error is 1.7mm, Max error is 3mm.

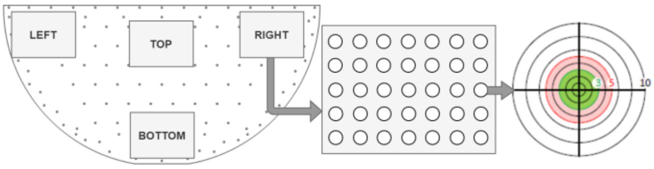

Fig. 6 - Stylus teaching error measurement. 4 A4 sheets containing each 35 targets with millimetric scaling.

Zone	Left	Top	Bottom	Right
Mean error	1.52 mm	2,5 mm	1,28 mm	1,5 mm
Max error	3 mm	3 mm	2 mm	2 mm

Table 3 - Overall skill precision for stylus teaching modality

3.2.4. Overall precision estimation and comparison

The following table provides a synthesis of our estimation of overall precision, for various teaching modalities.

	Stylus MoCap	Teleop MoCap	Hand guiding	Teach pendant
Teaching	< 1 mm	< 2 mm	< 5mm	< 1 mm
Measure	< 2 mm	0 mm	0 mm	0 mm
Calibration	< 2 mm	0 mm	0 mm	0 mm
Global	< 3 mm	< 2 mm	< 5 mm	< 1 mm

Table 4 - Comparison of overall teaching precision

Precision is decomposed into 3 parts. **Teaching** relates to the operator precision. With stylus and teach pendant, it's easy to achieve sub millimetric teaching precision. With teleoperation, provided the teleoperation system is smooth enough, the teaching precision depends mostly on the point of view of the operator (close-up or remote view, direct or through a camera). As an estimation from that one can expect typically 1 to 3 mm teaching precision. With hand guiding, precision relies heavily on the cobot transparency. Using a UR10e, it is quite difficult to fine-tune position below 5mm. Of course, all these values are indicative, and are highly dependent to the effort and time allowed on the teaching. **Measure** relates to the precision of the measurement equipment. For teleoperation, hand guiding and teach pendant, there is no loss of precision, as we use direct robot measurements. **Calibration** relates to the need of one or several calibrations for skill replay. Again, it's only needed for stylus teaching.

3.3. *Teaching duration*

We measured teaching duration for a sequence of 30* bolts, for each modality, with 3 different operators. (* 5 bolts for teach pendant, to limit test duration). Here, operator 3 was not familiar to the teleoperation setup, explaining the noticeable difference with operators 1 and 2. Overall, for all operators, Stylus teaching is much faster and easier. Although Teleop and hand guiding show similar timings, operators reported Teleop method to be physically much less strenuous. Teach pendant was found to be the most time-consuming and mentally fatiguing to use.

Teaching time per bolt	Stylus MoCap	Teleop MoCap	Hand guiding	Teach pendant
Operator 1	2.26s	6.1s	7.2s	26s
Operator 2	2.3s	5.1s	5.5s	32s
Operator 3	1.3s	9.0s	5.9s	31s
Mean	2,0s	6,7s	6,2s	29,7s

Table 5 - Teaching time comparison

3.4. *Teaching ergonomics*

Teaching ergonomics was evaluated by 3 operators, comparing the 4 teaching modalities together. All had similar conclusions, synthetized in the following table. Notably, stylus MoCap is overall the most comfortable approach, followed by teleoperation. Hand guiding is natural as well, but physically very demanding, as adjusting precisely the robot position requires several trials, and as it becomes harder when operator has to bend over to access further points. Teach pendant is very cumbersome to use, and prone to error in movements directions, leading to high risk of collision, and high mental load.

	Stylus MoCap	Teleop MoCap	Hand guiding	Teach pendant
Learning curve	Best	Good	Good	Average
Physical load	Good	Best	Worst	Best
Mental load	Best	Average	Good	high
Overall ease	Best	Good	Average	Worst

Table 6 - Ergonomics comparison across teaching modalities

3.5. *Robustness limitations*

Teleoperation, hand-guiding and teach pendant all rely directly on the robot measurements for teaching, and as such, guarantee reachability and repeatability of taught points; whereas stylus teaching doesn't guarantee robot reachability nor the trajectory behaviour of the robot. In the screwing use case, this wasn't an issue since accessibility was ensured easily with robot placement, and the 2D nature of the use-case made robot movements more predictable. However, in the general case, integration of additional path planning would be recommended to ensure robust behaviour of the robot.

3.6. *Low cost*

Regarding hardware, this kind of high-end MoCap setup is very cost-effective (5-10 k€), especially compared to force-feedback teleoperation. Master arms prices typically range from 20-30k€ for low cost / small systems, to over 100k€ for high-end models. However, the main advantage is the overall deployment cost. With automated robotic solutions, costs for deployment and process variants adaptation are usually much higher than the robot hardware itself, due to the complexity of robot programming for full automation. With our approach, which relies on interaction instead of automation, task-specific skills have to be developed, but they are simpler, and there is no process-specific robot programming, which usually takes weeks of development and integration. Deployment time is negligible, training is expected to be minimal; typically a few days for an operator without previous robotic knowledge. Based on Table 5 results, teaching time is expected to be 5 to 10x faster, and performed by operator instead of robotics engineer.

4. Conclusion and future work

The main goal in this paper was to elaborate new innovative and low-cost solutions and workflow, truly accessible to operators, to tackle the challenges of low-volume / high-mix industrial applications, where constant adaptation of the robot behaviour is mandatory, and to evaluate whether the solution precision and ergonomics are sufficient to address practical industrial applications.

Although the proposed solution was demonstrated on a single industrial use case, preliminary results show without ambiguity that Skill-based MoCap teaching can provide sufficient accuracy for many practical use cases, while providing noticeable improvements on teaching efficiency and ergonomics. The resulting workflow is simple and straightforward, and allow to envision many improvements.

Stylus MoCap and teleop MoCap are highly complementary. Stylus MoCap teaching allows effortless and natural teaching, in total independence of the robot. Teleoperation MoCap allows to teach remotely, also without effort, thus without dependency on robot transparency. This allows to address a broader range of industrial robotics applications, by keeping the robot in the cell, while allowing cobot-like teaching modalities. Both solutions differ from hand-guiding by working without direct robot interaction, thus allowing simple, safe, and intuitive teaching on a greater range of robots.

For future work, we plan to further extend operator user surveys and incorporate their feedback to improve user experience. As previously discussed, path planning should be added for improved robustness in more complex applications. Also, teaching is currently limited to point and frame geometric primitives; it shall be extended to include trajectories to allow development of more complex skills based on gesture teaching. Finally, many other post-processing functions could be added to improve efficiency, inspired by drawing software features, such as linear & grid repetition, position alignment, etc.

Acknowledgements

Part of this research was supported by funding from the European Union's Horizon Europe research and innovation programme (project JARVIS, under grant agreement nº 101135708).

References

[1] Dall'Alba, D., & Boriero, F. (2025). Towards an intuitive industrial teaching interface for collaborative robots : Gamepad teleoperation vs. kinesthetic teaching. The International Journal of Advanced Manufacturing Technology, 138(3), 1505-1522. https://doi.org/10.1007/s00170-025-15657-x

[2] Ma, Y., Song, Z., Zhuang, Y., Hao, J., & King, I. (2024). A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*.

[3] Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., ... & Finn, C. (2024). Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

[4] O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., ... & Chen, M. (2024, May). Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6892-6903). IEEE.

[5] Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Survey : Robot Programming by Demonstration. Springer Handbook of Robotics, 1371-1394. https://doi.org/10.1007/978-3-540-30301-5_60

[6] Sosa-Ceron, A. D., Gonzalez-Hernandez, H. G., & Reyes-Avendaño, J. A. (2022). Learning from Demonstrations in Human–Robot Collaborative Scenarios : A Survey. Robotics, 11(6), Article 6. https://doi.org/10.3390/robotics11060126

[7] Wang, W., Li, R., Chen, Y., Diekel, Z. M., & Jia, Y. (2019). Facilitating Human–Robot Collaborative Tasks by Teaching-Learning-Collaboration From Human Demonstrations. IEEE Transactions on Automation Science and Engineering, 16(2), 640-653. https://doi.org/10.1109/TASE.2018.2840345

[8] Schou, C., Andersen, R. S., Chrysostomou, D., Bøgh, S., & Madsen, O. (2018). Skill-based instruction of collaborative robots in industrial settings. Robotics and Computer-Integrated Manufacturing, 53, 72-80. https://doi.org/10.1016/j.rcim.2018.03.008

[9] Akkaladevi, S. C., Propst, M., Hofmann, M., Hiesmair, L., Ikeda, M., Chitturi, N. C., & Pichler, A. (2021). Programming-Free Approaches for Human–Robot Collaboration in Assembly Tasks. In L. Wang, X. V. Wang, J. Váncza, & Z. Kemény (Éds.), Advanced Human-Robot Collaboration in Manufacturing (p. 283-317). Springer International Publishing. https://doi.org/10.1007/978-3-030-69178-3_12

[10] George, P., Cheng, C.-T., Pang, T. Y., & Neville, K. (2023). Task Complexity and the Skills Dilemma in the Programming and Control of Collaborative Robots for Manufacturing. Applied Sciences, 13(7), Article 7. https://doi.org/10.3390/app13074635

[11] Pedersen, M. R., & Krüger, V. (2015). Gesture-Based Extraction of Robot Skill Parameters for Intuitive Robot Programming. Journal of Intelligent & Robotic Systems, 80(1), 149-163. https://doi.org/10.1007/s10846-015-0219-x

[12] Yin, Y., Zheng, P., Li, C., & Wan, K. (2024). Enhancing human-guided robotic assembly : AR-assisted DT for skill-based and low-code programming. Journal of Manufacturing Systems, 74, 676-689. https://doi.org/10.1016/j.jmsy.2024.04.016

[13] Piolain, G., Geffard, F., Coudray, A., Garrec, P., Thro, J.-F., & Perrot, Y. (2010). Dedicated and standard industrial robots used as force-feedback telemaintenance remote devices at the AREVA recycling plant. 2010 1st International Conference on Applied Robotics for the Power Industry, 1-6.

[14] Haption. (s.d.). Virtuose 6D TAO product page. https://www.haption.com/product/virtuose6dtao

[15] Dimension, F. (s.d.). Force Dimension Sigma.7 master haptic interface. https://www.forcedimension.com/products/sigma

[16] Geffard, F., Garrec, P., Piolain, G., Brudieu, M.-A., Thro, J.-F., Coudray, A., & Lelann, E. (2012). TAO2000 V2 computer-assisted force feedback telemanipulators used as maintenance and production tools at the AREVA NC–La Hague fuel recycling plant. Journal of Field Robotics, 161-174.

[17] LIST, C. (s.d.). CORTEX: a software suite for faster and easier robot controller engineering. Retrieved on CEA LIST: https://list.cea.fr/en/page/cortex-a-software-suite-for-faster-and-easier-robot-controller-engineering/

[18] ROS2 (s.d.). The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications. https://github.com/ros2/ros2

[19] Wang, X., Guo, S., Xu, Z., Zhang, Z., Sun, Z., & Xu, Y. (2024). A Robotic Teleoperation System Enhanced by Augmented Reality for Natural Human–Robot Interaction. Cyborg and Bionic Systems, 0098.

[20] Udekwe, D., & Seyyedhasani, H. (2025). Human robot interaction for agricultural Tele-Operation, using virtual Reality: A feasibility study. Computers and Electronics in Agriculture.

[21] AR-Tracking. (s.d.). DTRACK Products. Retrieved on AR-Tracking.com: https://ar-tracking.com/en/product-program/products-connection-software-dtrack

[22] FZI Forschungszentrum Informatik (s.d.). Cartesian controller. Retrieved at: https://github.com/fzi-forschungszentrum-informatik/cartesian_controllers/tree/ros1