Utilization of a 3D Mouse and Motion Scaling for Operation Enhancement of a Telemanipulator

Giovanny Limin, Ping-Lang Yen

***Abstract*—Better operational transparency in a teleoperation system plays key role to the performance of completing designated tasks. In this paper we proposed a teleoperation strategy using a 3D mouse as the control interface at the local site to control robot manipulator at the remote site. A decoupling strategy has been implemented for the 3D mouse that takes into account the observed intention-output mismatch caused by the coupling effect. Error-based variable weights were also used to incorporate information about the remote environment for motion scaling. The experimental results on tele-pressing a button showed that the teleoperation performance could be effectively improved by the proposed method for button pressing task.**

Index Terms— Teleoperation, Motion Scaling, 3D mouse

I. Introduction

Teleoperation systems enable remote handling of tasks that would be dangerous or inaccessible for humans to perform directly, harnessing benefits of human-robot collaboration by leveraging human intelligence and judgement capabilities with the strength and precise execution of the robot. Originally developed for nuclear waste disposal, teleoperation technologies have since been widely adopted for space exploration, deep-sea missions, and remote maintenance [1][2].

In this study, the primary application of our teleoperation system is in industrial surveillance robot, initially designed for tasks such as button-pressing and valve-turning, where the critical nature of these tasks makes human-in-the-loop control preferable [3]. However, as noted by previous research [1], teleoperation for remote maintenance presents serious disadvantages, including high expenses and the need for comprehensive operator training. These factors pose significant barriers to widespread adoption in the foreseeable future.

To address these issues, we propose a teleoperation strategy using a 3D mouse as the control interface for the industrial maintenance robot manipulator. The 3D mouse (also known as the SpaceMouse® product line by 3DConnexion [4]) is commercially marketed as an accessory device for 3D computer graphics software. Despite its primary user in graphics, the six-degrees-of-freedom (6DOF) control it offers is highly suitable for our control tasks while being more cost-effective compared to other interfaces [5]. Additionally, it also has shorter learning curve due to its straightforward mechanism and ergonomic design. Its compact size and small motion range help to reduce fatigue in extended use and provide more flexibility in the operator's workspace. These factors make 3D mouse a promising interface for industrial maintenance teleoperation with high scalability potential.

However, using a 3D mouse for robot control requires further adjustments to ensure smoother and more intuitive control. The first challenge is control axes coupling, which can complicate robot control. Although this inherent coupling behavior may be advantageous for performing multiple interactions simultaneously in 3D CAD software, it is less suited for control tasks requiring high precision and accuracy. Studies have shown that separated DOF control is more effective for such tasks [6][7]. Therefore, we implemented a decoupling strategy for the 3D mouse that takes into account the observed intention-output mismatch caused by the coupling effect.

Another challenge is the coordinate frame mismatch, which can affect the intuitiveness of the control interface. As highlighted in [8], the discrepancies between operator viewpoint, alignment of input device, and the local coordinate frame of the robot can lead to higher mental workload and reduced efficiency. Therefore, aligning the control frame with the user's perspective is crucial for effective teleoperation. To achieve this goal, we align the 3D mouse control axis in respect to the eye-in-hand camera's perspective to provide a more intuitive and task-suited control experience.

To further improve the task performance, we propose a simple motion scaling strategy. Error-based variable weights are used to incorporate information about the remote environment for dynamic velocity scaling. Unlike previous motion scaling solutions that primarily used distance-based approaches where motion direction is insignificant [9][11], our direction-aware solution also function as a form of fault diagnosis [2] and navigation guidance. These environment-adapted control approaches are highly beneficial particularly for systems with time delays to increase robustness and fidelity [10]. Under time delay, direct control strategies tend to fall into an experimental 'move and wait' approach [2][9], leading to inefficiencies and potential overshooting. This issue is also relevant to tasks like button pressing, where accidental activation is undesirable. The proposed solution aims to minimize this trial-and-error behavior and reduce fatigue from continuous one-to-one human commands, allowing operators to focus on their intended goals while the robotic system translates their command to expected execution response, ultimately enhances the overall teleoperation experience by providing intuitive user interaction with remote environment.

While previous studies mentioned the usage of 3D mouse control for robotic tasks [12][13][14][15], they often lack detailed implementation designs. Our study addresses this gap by providing a task-specific implementation with a motion scaling framework. The current focus of this study is on optimizing the 3D mouse control of the robot's end effector to improve operator's efficiency, accuracy, and intuitiveness. This approach aligns with the dual goals of teleoperation systems:

Manuscript received 16 September 2024; revised 4 October 2024; accepted 5 October; Date of publication 17 November 2024. This work was financially supported by Delta Electronics Inc.

All authors are with Department of Biomechatronics Engineering at National Taiwan University, Taipei 106319, Taiwan (Corresponding author: Ping-Lang Yen, e-mail: plyen@ntu.edu.tw).

stability (the control loop should remain stable regardless of the behaviour of the human operator or environmental perturbations) and telepresence (the human operator should feel if they are present on the remote side) [2]. To evaluate the performance of our solution, we conducted a user study. Various quantitative and qualitative metrics were used to assess task efficiency, task accuracy, and user perception and preferences.

II. Methods

The 3D mouse generates a (6×1) signal denote by $\boldsymbol{s_0} = [x, y, z, r_x, r_y, r_z]^T$ that contains translational and rotational input about the device's axes. The process of translating the 3D mouse signal into velocity commands for the robot manipulator's end effector involves several steps: pre-processing, coordinate frame matching, control axis decoupling, dead-zone and sensitivity curve remapping, and dynamic weighting.

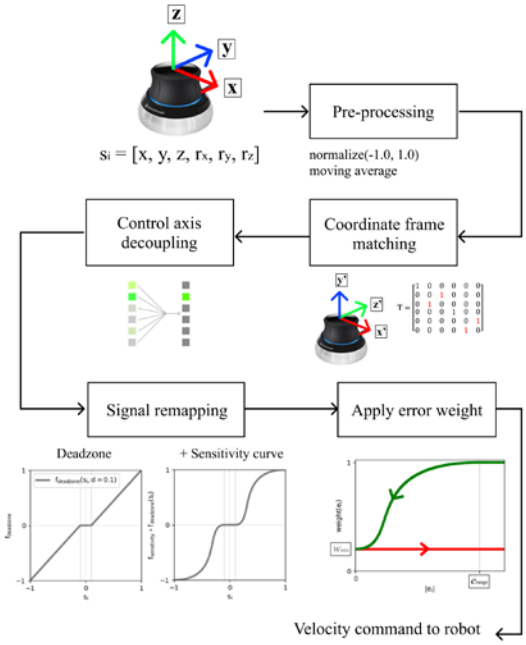


Figure 1. Motion Processing

A. *Pre-processing*

Given the adequate quality of the signal from the 3D mouse, minimal signal pre-processing is needed. The signal is smoothed using a moving average filter of to discard small noise from insignificant fluctuations, and normalized to a range of [-1,1].

B. *Coordinate Frame Matching*

To align the control axes, a (6×6) transformation matrix **T** is applied. The default coordinate frame mapping without modification is an identity matrix.

$$\boldsymbol{s}_{\text{aligned}} = \mathbf{T} \cdot \boldsymbol{s} \tag{1}$$

In our study, we align the control axes of 3D mouse with the coordinate frame of the eye-in-hand camera visual feedback instead of the end-effector coordinate frame to provide intuitive control and reduce the operator's mental load from doing spatial transformation of their intended movement command with input device coordinate mapping.

C. *Control Axis Decoupling*

To provide decoupled control, we assume the axis with the largest signal represents the operator's intended axis of movement. This can be described by introducing a (1×6) decoupling factor matrix **D** where the element corresponding to the maximum signal *i* is set to 1, and the rest are set to 0.

$$\boldsymbol{s}_{\text{decoupled}} = \mathbf{D} \cdot \boldsymbol{s}$$

$$\mathrm{D}_i = \begin{cases} 1 & \text{if } i = \arg\max_j |s_j| \\ 0 & \text{otherwise} \end{cases}$$

$$\forall\, i, j \in \{x, y, z, r_x, r_y, r_z\} \tag{2}$$

However, it was observed that users frequently mis-trigger the 3D mouse's z-axis signal during operation. To minimize unintended movements, an additional layer of verification is applied when the z-axis signal is found as maximum. This verification involves checking whether another axis has a significant signal magnitude above a predefined threshold ratio $\boldsymbol{\varepsilon} \in (0,1)$. If another axis meets this condition, the z-axis is discarded as the intended axis, and the other axis with second maximum signal is selected instead. Therefore, lower threshold $\boldsymbol{\varepsilon}$ means it will be more likely to pick the second-maximum signal rather than z-axis. This strategy is based on the observation that when users intend to move along the z-axis, signals from other axes are minimal due to the mechanical device design. In our case, the threshold $\boldsymbol{\varepsilon}$ was set to 0.75 for translation signal (x, y) and 0.80 for rotation signal (r_x, r_y, r_z), which were derived from preliminary data of signal behaviors in simple one DOF motions.

The adjusted decoupling factor **D'** can be described as:

$$\mathrm{D'}_i = \begin{cases} 1, & \text{if } i = \arg\max_j |s_j| \text{ and } (i \neq z \text{ or } \frac{\max_{k \neq z}}{|s_z|} < \varepsilon_k) \\ 1, & \text{if } i = \arg\max_{k \neq z} |s_k| \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

D. *Dead-zone and Sensitivity Curve Mapping*

Following common practice in game controllers or other mainstream input devices [15] [16], we implemented dead-zone and sensitive curve mapping to the 3D signal.

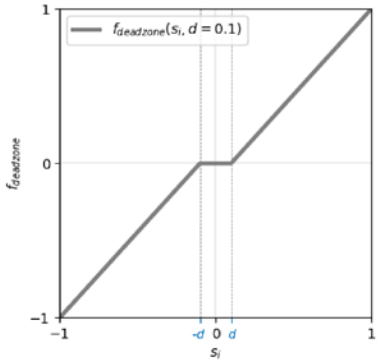


Figure 2. Dead-zone function plot described in Equation (3)

The dead-zone is implemented to filter out small signal from accidental touches within input zone $[-d, d]$:

$$f(s_i, d) = \begin{cases} 0 & if\ |s_i| < d\ or\ s_i = 0 \\ \frac{s_i - \text{sign}(s_i)\, d}{1 - d} & \text{otherwise} \end{cases} \tag{4}$$

Since the 3D mouse signal is used as velocity control and not position control, our initial design employed a simple step function with binary states (on/off) based on a trigger threshold. To smooth out this function, a cubic-Bezier curve was implemented, preserving similar characteristics while offering smoother transitions. The cubic-Bezier curve is a common easing function in animation and computer graphics [17], described by the equation below, with $\mathbf{P_0}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}$ as predefined control points:

$$B(t, \mathbf{P_0}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}) = (1-t)^3\mathbf{P_0} + 3(1-t)^2 t\mathbf{P_1} + 3(1-t)^2\mathbf{P_2} + t^3\mathbf{P_3} \quad (5)$$

For more efficient runtime, precomputed value pairs were stored during initialization, which accessed via key indexing with *O*(1) complexity. The control points used in this study was $\mathbf{P_0} = (0,0), \mathbf{P_1} = (0.4,0), \mathbf{P_2} = (0,1), \mathbf{P_3} = (1,1)$ to achieve ease-in-out effect with half-peak occurring at 27.5% of the input range (See Fig 3).

E. Error-based Motion Scaling Weights

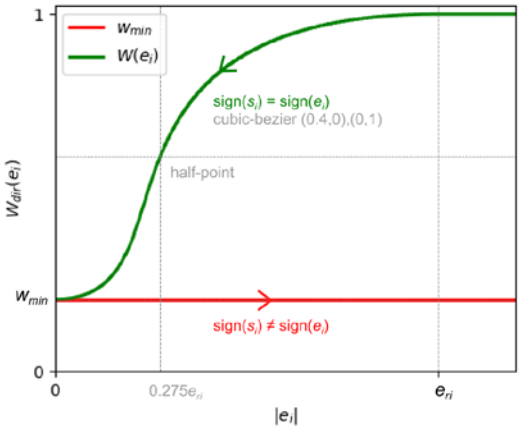


Figure 3. Dynamic scale weight plot described in Equations (6)-(8)

A simple dynamic scaling factor, based on pose error, is implemented to assist the operator during approaching tasks. The underlying principle is that as the current pose approaches the desired pose or target, greater precision is needed, demanding for smaller movements. Large movements pose a risk of overshooting, especially under time delay conditions where operators need to exert more effort to estimate their control input with delayed feedback. Therefore, the weight factor decreases with decreasing error, as described in equation (6) with pose error $\boldsymbol{e}$, error range threshold $\boldsymbol{e_r}$ where the scaling start to take effects (in our case, is set to 30mm for translation and 10 degree for rotation), and the cubic-Bezier function *B* for smoother deceleration curve.

$$W(e_i) = \begin{cases} w_{\min} + (1 - w_{\min})\, B(e_i/e_r) & \text{if } |e_i| \le e_{r_i} \\ 1 & \text{if } |e_i| > e_{r_i} \end{cases} \quad (6)$$

It is crucial to set a minimum weight $w_{\min}$ corresponding to $e = 0$ to ensure the system does not get stuck as it approaches the target pose (in our case, $w_{\min}$ is set to 0.2 to allow visible movement with sufficient precision). Additionally, the error weight is direction-sensitive. If the motion control input is directed towards increasing error (moving away from the desired pose), the scaling factor will be set to the minimum scale to act as artificial resistance to guide the operator.

$$W_{\text{dir}}(e_i, s_i) = \begin{cases} W(e_i) & \text{if } \text{sign}(s_i) = \text{sign}(e_i) \\ w_{\min} & \text{if } \text{sign}(s_i) \neq \text{sign}(e_i) \end{cases} \quad (7)$$

$$\boldsymbol{s}_{\text{error weighted}} = W_{dir}(\boldsymbol{e}) \cdot \boldsymbol{s} \quad (8)$$

To calculate the pose error, assuming the target end-effector pose ${}^{TCP}_{base}\mathbf{T_t} = \{x_t, y_t, z_t, \theta_{x_t}, \theta_{y_t}, \theta_{z_t}\}$ and current measured pose ${}^{TCP}_{base}\mathbf{T_m} = \{x_m, y_m, z_m, \theta_{x_m}, \theta_{y_m}, \theta_{z_m}\}$ are known, the translational component of pose error is calculated as below:

$$e_{\text{trans}} = \begin{pmatrix} x_m - x_t \\ y_m - y_t \\ z_m - z_t \end{pmatrix} \quad (9)$$

The rotational component of the error requires few more steps:

$$\mathbf{R} = \mathbf{R}_m^{-1} \cdot \mathbf{R_t} \quad (10)$$

$$\mathbf{R}_t = \mathbf{R_z}(\theta_{z_t})\mathbf{R_y}(\theta_{y_t})\mathbf{R_z}(\theta_{z_t}) \text{ , } \mathbf{R}_m = \mathbf{R_z}(\theta_{z_m})\mathbf{R_y}(\theta_{y_m})\mathbf{R_z}(\theta_{z_m})$$

where the $\mathbf{R_x}, \mathbf{R_y}, \mathbf{R_z}$ are the elemental rotation matrices. $\mathbf{R}$ represents the relative rotation matrix of $\mathbf{T}_{\text{m}}$ with respect to $\mathbf{T}_{\text{t}}$. Finally, for Euler angles representation:

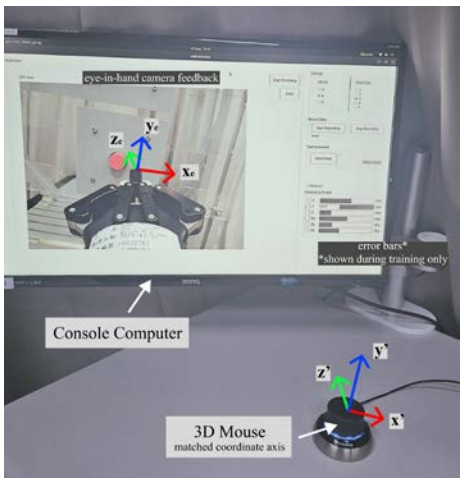
$$e_{\text{rot}} = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}, \begin{matrix} \theta_y = atan2\left(-\mathbf{R}_{31}^2, \sqrt{\mathbf{R}_{11}^2 + \mathbf{R}_{21}^2}\right) \\ \theta_x = atan2\left(\frac{\mathbf{R}_{32}}{\cos\theta_y}, \frac{\mathbf{R}_{33}}{\cos\theta_y}\right) \\ \theta_z = atan2\left(\frac{\mathbf{R}_{21}}{\cos\theta_y}, \frac{\mathbf{R}_{11}}{\cos\theta_y}\right) \end{matrix} \quad (11)$$

III. Experiment

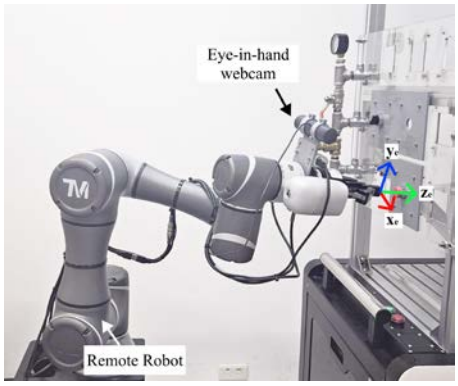
The experiment was conducted using a teleoperated robotic system running on ROS2 (ROS2 Humble-Hawksbill) architecture. The hardware setup included a TM robot (TM5M-700), controlled by a remote side computer (NVIDIA Jetson AGX Orin Developer Kit). A webcam was mounted on the robot's flange to provide visual feedback, ensuring the end effector tip was visible. The 3D mouse signal was used as the scaling factor [0, 1] of the end effector's velocity, which is set as 20 mm/s for translational elements and 4 deg/s for rotational elements. A 750ms delay was introduced when sending the velocity command script to the robot.

A. Task

The primary focus of this experiment was to evaluate the effectiveness of the 3D mouse and proposed control in moving the robot's end effector to a desired pose. Participants were given a scenario to approach a button without pressing it. A screenshot of the desired pose's eye-in-hand's camera view was presented side by side with the live visual feedback, allowing participants to compare and reproduce the image of desired pose by controlling the robot's end effector. The desired end effector pose was defined as being positioned in front of the button's centroid, oriented normal to the button surface, and not touching the button, distanced at 1 cm.



(a)



(b)

Figure 4. Button-reaching teleoperation experiment setup
(a) Controller side. (b) Remote robot side

B. Procedure

Participants were first given an untimed practice chance to familiarize themselves with the 3D mouse. Once they felt ready, they were instructed to move the robot to the desired pose from three different starting points for each control mode.

- Default mode: The signal from 3D mouse only went through preprocessing and coordinate frame mapping.
- Proposed mode: Axis decoupling, signal remapping (dead-zone and sensitivity curve), and error-based motion scaling weights was applied to account for the remote environment's information and operator's information.

When switching to different control mode, participants were given additional practice time to get familiarized with the control mode and figure out the differences.

Motion data from the 3D mouse and robot was recorded to capture task performance metrics. Upon completing all tasks, participants filled out a questionnaire.

C. Evaluation Metrics

Both task performance metrics from motion data and user questionnaire metrics are used in this experiment to evaluate the system's effectiveness.

To validate the task efficiency among control modes, we compared the following metrics:

- Task completion time: total time to complete the task.
- Error convergence time: time to reach steady-state error from the start of leaving the initial static error.
- Error range: quantifies the range of motion space occupied during the task.
- Total variation: integral of absolute slopes, quantifying the overall smoothness of the motion.

For task accuracy, we compared the following metrics:

- Final errors: the residual errors at the end of the task.
- Number of accidental button presses: number of times the button was unintentionally pressed during the task.

To account for the operator's subjective measures, we used the following metrics:

- NASA Task Load Index (TLX) score: a widely-used measure to assess perceived workload [18].
- Likert scale ratings of precision and usability: perceived precision and usability of control modes.
- Mode preference: preferred control mode indicated by participants in the questionnaire.

IV. Results and Discussion

Ten participants (5 females, 5 males, aged 22-37) were recruited for preliminary experiment. The end-effector's offset to the target in relation to time was evaluated from the motion data captured during trials for evaluating task efficiency and accuracy metrics, and the participant's answer to the post-test questionnaire was evaluated for user's subjective measures.

A. Task Efficiency

TABLE I. Task Efficiency Metrics

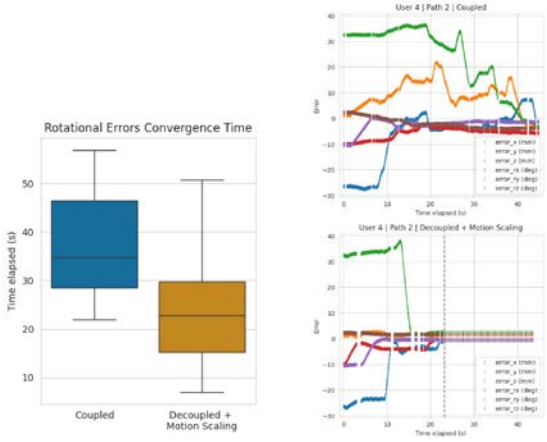
mean ± std	**Control Modes**	
	Default	Decoupled + Motion Scaling
Task completion time (s)	59.92 ± 41.96	**46.31 ± 20.95**
Convergence time (trans.) (s)	30.00 ± 36.90	**25.78 ± 19.32**
Convergence time (rot.) (s)	40.74 ± 40.74	**19.80 ± 16.87**
Error range (trans.) (mm)	19.46 ±5.92	**14.43 ± 2.60**
Error range (rot.) (deg)	6.09 ± 3.87	**4.00 ± 2.96**
Total variation (trans.) (mm)	64.43 ± 47.98	**31.03 ± 10.73**
Total variation (rot.) (deg)	13.74 ± 12.41	**5.68 ± 3.77**

The proposed method (decoupled + motion scaling) showed significant improvements in convergence time for rotational errors. Post hoc Tukey's HSD test revealed the improvement from the proposed mode to the default mode is statistically significant ($p < 0.05$) for rotational error convergence times t_{rx} ($p = 0.020$), t_{ry} ($p = 0.005$), and t_{rz} ($p = 0.028$).

This improvement supports the goal of assisting the operator during control, especially since controlling angles is notably harder than translation with limited feedback. During the experiment, the only cue available to participants was matching the picture from the eye-in-hand camera to the image of the

desired pose. Translating the perspective difference to angle error is a challenging task for humans in general.

Lower values in the task efficiency metrics (Table I) indicate better control efficiency, which can also be translated to reduced trial-and-error approaches and more confident control. This corresponds to shorter motion paths (total variation) and less motion space (error range). The improvement effect is illustrated in the error/time plot example (Fig 5), which is consistently observed across trials.



(a) (b)

Figure 5. Reduced trial-and-errors
(a) Rotational errors convergence time. (b) Error-time plot

B. Task Accuracy

TABLE II. TASK ACCURACY METRICS

mean ± std	**Control Modes**	
	Default	Decoupled + Motion Scaling
Accidental button press rate	0.5 ± 0.4	**0.2 ± 0.3**
Final pose error {*x*} (mm)	1.1 ± 1.2	**0.9 ± 1.2**
Final pose error {*y*} (mm)	4.2 ± 3.2	**3.6 ± 2.8**
Final pose error {*z*} (mm)	8.7 ± 8.6	**6.7 ± 7.0**
Final pose error {*Rx*} (deg)	2.6 ± 1.8	**0.9 ± 1.1**
Final pose error {*Ry*} (deg)	5.5 ± 5.4	**4.0 ± 4.6**
Final pose error {*Rz*} (deg)	1.5 ± 1.2	**0.7 ± 0.7**

Although the task accuracy results in Table II showed little significance between the control modes, they reveal how operators struggled with alignment on certain axes, likely due to limitations in the visual feedback of the current system. During the experiment, operators had notable difficulties inferring position in the *z*-axis (depth) from the image, which explain higher residual pose error in the *z*-axis and similar rate of accidental button presses.

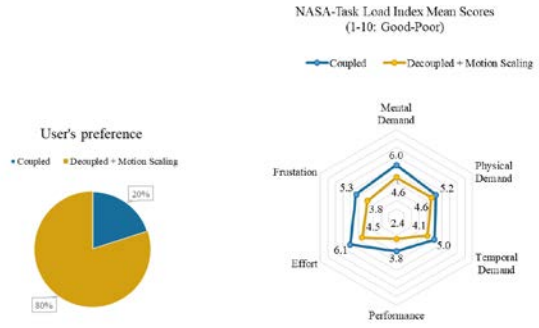
Similarly, determining the R_y alignment from the camera was challenging because the camera was slightly tilted and not parallel to the end effector. This required participants to rely on perspective cues and mentally perform spatial transformations, explaining the higher pose error in R_y compared to other rotational component errors.

This error in Ry also influenced the visual feedback of the end effector's position relative to the button, creating a skewed image on the y-axis. Consequently, this could mislead participants by believing that the y position was correct based on the distorted image, explaining the higher pose error in *y*.

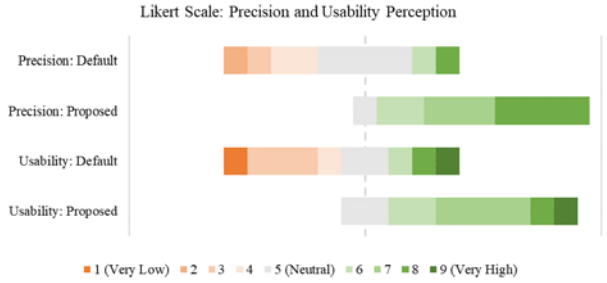
C. User Perception and Preferences

II. QUESTIONNAIRE RESULTS

1-10 (Low/High) (Good/Poor)	**Control Modes**	
	Default	Decoupled + Motion Scaling
Mental Demand	6.0 ± 1.6	**4.6 ± 1.7**
Physical Demand	5.2 ± 2.0	**4.6 ± 1.6**
Temporal Demand	5.0 ± 1.4	**4.1 ± 1.3**
Performance (1-10: Good-Poor)	3.8 ± 1.8	**2.4 ± 1.2**
Effort	6.1 ± 1.7	**4.5 ± 1.2**
Frustation Level	5.3 ± 2.1	**3.8 ± 1.5**
Total Task Load Index Score	31.4	**24.0**



(a) (b)



(c)

Figure 6. Questionnaire results
(a) User preferences. (b) Task-Load Index (TLX): the lower the better.
(c) User's perception of precision and usability (Default: coupled, Proposed: decoupled + motion scaling)

The questionnaire results indicated a better perception of decoupled modes, shown by the lower Task Load Index (TLX) scores in Table III. Eight out of ten participants selected the proposed mode (decoupled + motion scaling) over the default mode. Two participants who preferred default mode as their first preference perceiving it as faster due to simultaneous motions in multiple DOF, and higher speed suits more to their liking. However, the NASA TLX results showed that default mode had the highest workload, particularly in mental demand, effort, and frustration, which can be translated as a more challenging control experience.

All participants were satisfied with the translation mapping that aligned well with the image from camera's visual feedback

and the positioning of 3D mouse. However, for the rotational motion, participants with experience in robotics control preferred the rotational mapping to respect the end effector's base. They suggested that the R_z axis on the 3D mouse should map to the R_z velocity command of the end effector, resulting in visual feedback image rotation about R_y, and vice versa.

D. Future Directions

Proposed mode generally offered better overall control in terms of task efficiency along with high user preference, making it promising for further development and refinement. Based on observations from accuracy metrics and user feedback, future experiments should address the following areas to provide fair evaluation of control modes:

Enhanced Feedback Cue Design: The limited feedback from a single visual cue highlights the need for a more comprehensive environment sensing feedback design. This could involve adding additional live feeds from other camera views or incorporating more sensors and GUI indicators to provide operators with a clearer understanding of the operational context.

Task Contextualization: The influence of task context on user preference is significant, as participants may prefer different control settings depending on specific scenarios. Therefore, the control strategy, including speed parameters should be carefully tailored according to the expected tasks to optimize performance and user satisfaction.

V. CONCLUSION

In this paper we have demonstrated a teleoperation manipulator for pressing button controlled by a 3D mouse at the local site by human operator. The position accuracy and task completion time can be effectively improved by incorporating dynamic scaling weighting to the leader follower controller. More investigation could be performed on various task characteristics to optimize the scaling functions for better usability.

REFERENCES

[1] J. Luo et al., "Combined perception, control, and learning for teleoperation: Key Technologies, applications, and challenges," *Cognitive Computation and Systems*, vol. 2, no. 2, 19 May 2020, pp. 33–43.

[2] S. Lichiardopol, "A survey on teleoperation," Technische Universiteit Eindhoven, Eindhoven, Netherlands, DCT rapporten; Vol. 2007.155, 2007.

[3] R. Parasuraman et al., "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 3, May 2000, pp. 286–297.

[4] "SpaceMouse Compact," *3Dconnexion*. Munich, Germany, Apr. 2023. [Online] Available: https://3dconnexion.com/dk/product/spacemouse-compact/. Accessed 29 May 2024.

[5] P. Wu et al., "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," arXiv preprint *arXiv:2309.13037*, 2023.

[6] G. Casiez and C. Chaillou, "Effects of dof separation on elastic devices for the navigation in 3d virtual environments with force feedback," *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, World Haptics Conference. IEEE, 2005.

[7] V. R. Garate, S. Gholami, and A. Ajoudani, "A scalable framework for multi-robot tele-impedance control." *IEEE Transactions on Robotics* 37.6, 2021, pp. 2052-2066.

[8] L. M. Hiatt and Reid Simmons, "Coordinate frames in robotic teleoperation," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006.

[9] F. Richter, R. K. Orosco, and M. C. Yip, "Motion scaling solutions for improved performance in high delay surgical teleoperation," *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[10] C. Passenberg, A. Peer, and M. Buss. "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems," *Mechatronics* 20.7 (2010): 787-801.

[11] S. E. Everett and R. V. Dubey, "Human-machine cooperative telerobotics using uncertain sensor or model data," *Proc. 1998 IEEE International Conference on Robotics and Automation* (Cat. No. 98CH36146). Vol. 2. IEEE, 1998.

[12] G. Hirzinger et al., "Advances in robotics: The DLR experience," *The International Journal of Robotics Research* 18.11 (1999): 1064-1087.

[13] Y. Kim et al., "Telerobotic neurovascular interventions with magnetic manipulation," *Science Robotics* 7.65 (2022): eabg9907.

[14] S. Baklouti et al., "On the improvement of ROS-based control for teleoperated yaskawa robots," *Applied Sciences* 11.16 (2021): 7190.

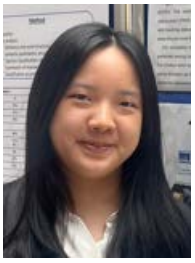
[15] V. Dhat, N. Walker, and M. Cakmak. "Using 3D Mice to Control Robot Manipulators." *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. 2024.

[16] "Joystick Input and Using Deadbands", *Mimir Games*, Nov. 2022, [Online]. Available: http://www.mimirgames.com/articles/games/joystick-input-and-using-deadbands/. Accessed 29 May 2024.

[17] "CSS: Cascading Style Sheets: MDN." *MDN Web Docs*, [Online] Available: https://developer.mozilla.org/en-US/docs/Web/CSS/easing-function. Accessed 9 June 2024.

[18] S. G. Hart, "NASA-task load index (NASA-TLX); 20 years later," *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 50. No. 9. Sage CA: Los Angeles, CA: Sage publications, 2006.

[19] Giovanny Limin, and Ping-Lang Yen, "Enhanced Teleoperated Manipulator Control with 3D Mouse and Dynamic Motion Scaling," International conference on Advanced Robotics and Intelligent Systems, 2024.



Giovanny Limin received the B.S. degree in Biomechatronics Engineering Department from National Taiwan University, Taiwan (R. O. C.), in 2024. Her current research interests and publications are in the areas of Human-Computer Interaction.

Ping-Lang Yen was born in 1966. He Received B.S. degree from Dept. of Power Mechanical Engineering, National Tsing-Hua University, M.S. degree from Dept. of Mechanical Engineering, National Taiwan University, and PhD. degree from Dept. of Mechanical Engineering, Imperial College London, UK in 1998, 1990 and 1996 respectively. Currently, he is a professor in Department of Biomechatronics Engineering, National Taiwan University. He is also a co-founder of Point Robotics Medtech Inc. since 2016.