



## A Novel Stepper Motor Haptic Interface for Efficient Robotic Task Programming

Abdelghani Daoudi<sup>\*ID</sup>, Mohamed Z Amrani<sup>ID</sup>, Nouara Achour<sup>ID</sup>

LRPE Laboratory, University of Sciences and Technology Houari Boumediene, Algiers 16111, Algeria

Corresponding Author Email: [adaoudi@usthb.dz](mailto:adaoudi@usthb.dz)

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.570512>

### ABSTRACT

**Received:** 26 July 2024

**Revised:** 29 September 2024

**Accepted:** 10 October 2024

**Available online:** 28 October 2024

#### Keywords:

virtual reality, industrial robot, robotics learning task, haptic interface, stepper motor

Programming RRR-type serial arm robots, especially for pick-and-place tasks, is time-consuming and intricate, often requiring the robots' temporary removal from service for programming and testing. This study proposes a haptic interface replicating the mechanical structure of RRR-type robots, integrating position sensors and stepper motors at each joint. This interface communicates with a virtual environment governed by the robot's mathematical model and task space data, allowing users to intuitively manipulate the virtual robot and record trajectories. Operational safety is ensured through joint locking to prevent collisions and avoid singular points. Comparative evaluations show significant reductions in computation time and enhanced learning efficiency compared to traditional methods. The evaluation includes measuring task completion times, demonstrates faster task accomplishment with the haptic interface. Beyond streamlining robotic arm programming, the approach prioritizes user-friendliness and operational efficiency, validated through experiments on both lab platforms and real robots. Additionally, the System Usability Scale (SUS) was employed to assess user satisfaction, with results indicating high user approval. These findings underscore the effectiveness of the haptic interface in advancing robotic trajectory learning and its potential for broader applications in robotics research and development.

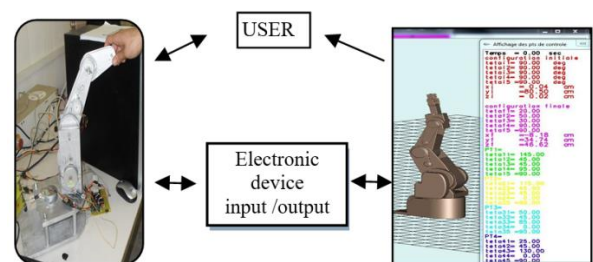
## 1. INTRODUCTION

The use of virtual environments offers a significant advantage by providing one or more users with an immersive sensory-motor activity within a digitally created artificial world. This artificial world can be imaginary, symbolic, or a simulation of the real world based on a mathematical model [1]. To interact with this virtual environment, users employ various types of interfaces, including VR headsets, motion controllers, haptic interfaces, tracking cameras, motion platforms, motion sensors, and motion capture systems. Among these, haptic interfaces are particularly suitable for robotic applications [2, 3]. Programming robotic tasks is a prominent application of virtual reality and haptic interfaces [4, 5].

Robotics has revolutionized the industry with its flexibility, allowing the same robot to perform a wide range of tasks. In the realm of robotic task learning, various methods have been employed. Initially, simulations based on the geometric model of the robot were used [6, 7]. However, this offline method did not provide adequate visualization of the task in space. Command boxes that control the robot in joint or task space were developed, necessitating several trials on the robot and consequently significant production downtime.

With the advent of virtual reality techniques [8], users can program their robots more efficiently by leveraging 3D visualization. Several robotic simulation platforms [9-11] have been developed, enabling users to visualize tasks and refine the learning process. The use of haptic interfaces with

these platforms enhances immersion in the virtual environment [12].



**Figure 1.** Block diagram of the platform

In this article, we present a haptic interface that we designed and developed to facilitate the programming of robotic tasks. The mechanical architecture of the interface is designed to resemble that of industrial robots, providing users with a better immersive experience during the learning phase. The article is structured into three sections. The first section reviews related works in robotic task learning using virtual reality with and without haptic interfaces. The second section describes our platform, which comprises three components: the mechanical design of the haptic interface resembling 3R manipulator robots, the data acquisition and control device, and the virtual environment, which includes the robot, its environment, mathematical model, and stops. The third section presents the evaluation of this robotic task learning platform, conducted

with 20 users. The synoptic of Figure 1 shows the operational principle of our platform.

## 2. RELATED WORKS

Numerous studies have explored the integration of virtual environments and haptic interfaces in robotic task programming. Early works focused primarily on virtual environments without the inclusion of haptic interfaces, while more recent studies have integrated haptic feedback to enhance user interaction and immersion.

We begin this section by presenting various works that address task programming in robotics using only virtual reality tools. We will explore the contribution of virtual reality in terms of understanding and the significant time savings compared to traditional methods.

For instance, Darmoul et al. [13] developed a virtual environment for a robotic cell to validate its design and plan the implementation of the actual robotic cell. They demonstrated that the semi-immersive environment provided users with a sense of presence in the digital environment and allowed them to work with digital objects at a 1:1 scale, similar to the real world. They concluded that the application of virtual environments is beneficial for teaching and training purposes.

Similarly, Loreto-Gómez et al. [14] conducted a formal evaluation comparing two teaching methods. The first method used traditional PowerPoint presentations and oral explanations to teach theoretical concepts and solve exercises for group A. In the second method, the professor added 3D simulations to teach the same concepts to group B. The results showed that students in group B performed significantly better, indicating that the use of 3D simulators improves academic performance and provides a valuable teaching tool for robotics.

In another study, Togias et al. [15] proposed a method based on simple operations for designing processes and controlling industrial robots using virtual reality. Their primary goal was to reduce the time and effort required for reprogramming the robot without the physical presence of an operator. They successfully reduced both the effort and time needed for task programming.

Further building on the advantages of VR, Monetti et al. [16] studied the impact of using a virtual robot for object manipulation compared to using a real ABB IRB 120 robot. They evaluated execution time and success rates to estimate programming efficiency. The results showed that students using virtual reality needed less time to complete tasks and achieved higher success rates. Most participants found the VR protocol very useful for familiarizing themselves with the real robot, highlighting the benefits of VR training over conventional methods.

Building on these results, Wolfartsberger et al. [17] investigated whether VR-supported training improves learning success compared to traditional workplace training. Their results indicated that VR training applications had a positive and significant impact on learning outcomes.

Several studies have also explored the integration of haptic interfaces in VR simulators for robotic programming.

In the second part, we present various works that use virtual reality coupled with a haptic interface. This approach combines the use of a haptic interface and an immersive environment, allowing the user to save even more time during the robot programming phase.

Aron et al. [18] developed a robotic task programming

platform using a virtual environment (RobotStudio), a haptic interface, and an ABB IRB 1600 robot. They concluded that this approach reduces the required skill levels for programmers, shortens programming times, and provides a natural user interface for performing tasks as in the real world.

In a related effort, Hurtado et al. [12] proposed a system that immerses users in a VR simulation of a real robot programming environment, enhancing the experience through tactile (haptic) and 3D visual feedback.

An additional study conducted by Glamnikand and Šafarič [19] developed an application for controlling a KUKA KR5 robot using the OMNI haptic interface, allowing real-time remote control of the robot.

Another research effort by Gonzalez-Badillo et al. [20] created a VR platform with a haptic interface for planning and evaluating assembly processes of two parts.

Additional advancements were made by Marzszalik [21] designed an application for calibrating and aligning the joint values of the OMNI haptic interface with a manipulator robot, demonstrating minimal errors between the real and simulated robot trajectories.

Further exploration in this area was carried out by Crespo et al. [22] developed a virtual reality simulator using Unity3D to simulate the behavior of the Mitsubishi Movemaster RV-M1 robot, improving student's understanding of the robot's operation in a virtual environment. It was designed to reduce the student's learning curve by displaying a complete virtual environment where the three-dimensional model of the robotic arm could be visualized and programmed according to the real environment. The evaluation showed that 53% of participants found it faster, 20% found it more attractive, 80% thought the system allowed for better understanding of the robot's functioning, 60% found it easy to introduce a robot sequence into the virtual application, 70% executed the sequence with the same precision in the real model, and 100% believed the system could be a beneficial learning tool.

Dhivin et al. [23] proposed a bilateral controller for a robotic manipulator using a haptic device. It allows the operator to control the 5-DOF robot remotely using a Phantom Omni master device, improving control in dynamic environments.

Knopp et al. [24] introduced a novel approach using the KUKA LBR robot as a haptic interface in a VR environment for simulating invasive surgical interventions, providing significant force feedback.

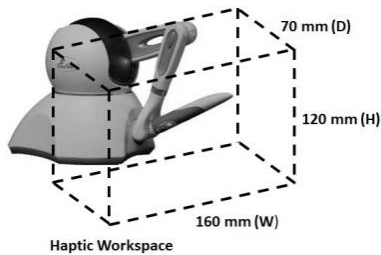
Complementing these findings, Andersson and Syberfeldt [25] conducted a study to determine if using virtual reality with appropriate feedback could provide an effective platform for training and familiarization. They used haptic feedback from VR controllers to simulate physical interactions with collaborative robots. Their results showed that participants considered the moving haptic feedback as the most appropriate representation.

Despite significant progress, challenges remain in achieving high levels of user immersion and realistic interaction in robotic task programming using virtual environments and haptic interfaces. Most works rely on the OMNI interface, which, while precise, has limitations due to its restricted joint angle range and workspace volume. The base joint ranges from  $-50^\circ$  to  $+55^\circ$ , the second joint ranges from  $0^\circ$  to  $105^\circ$ , and the third joint's range depends on the second joint's values (Figure 2) [26]. Additionally, the OMNI's geometry does not resemble that of manipulator robots, and its workspace volume does not reflect that of a real robot (Figure 3). Furthermore, using the OMNI requires mapping between the OMNI's joint

angles and the real robot's angles [21].



**Figure 2.** Dependency of  $\theta_3$  on  $\theta_2$



**Figure 3.** Workspace of the OMNI

While previous studies have made significant strides in integrating virtual reality and haptic interfaces for robotic task programming, many still face limitations, particularly in terms of hardware flexibility, precision, and user immersion. Most notably, the OMNI interface, widely used in past research, is constrained by its limited joint angles and workspace, reducing its applicability to complex robotic tasks. Our work builds on these foundations by addressing these hardware limitations. We developed a novel haptic interface that replicates the mechanical structure of RRR-type robots, incorporating position sensors and stepper motors to improve accuracy and user interaction. By offering real-time feedback and eliminating the need for angle mapping, our solution enhances both operational efficiency and user immersion. This work, therefore, represents a significant advancement in haptic-enabled robotic programming, paving the way for more intuitive and effective task learning in virtual environments.

### 3. METHODOLOGY

Our work aims to design a haptic interface that has the same mechanical structure as 3R-type robots for robotic task programming. This design eliminates the need for angle mapping and the use of reverse kinematics, thus avoiding issues with singularities and multiple solutions. It significantly enhances user immersion by providing more natural interaction. In previous work [27, 28], we utilized direct current (DC) motors to generate haptic feedback in the form of variable force. While DC motors are effective for producing variable forces, they are limited in torque, making them unsuitable for applications requiring high torque generation.

In this study, we need to generate substantial torques to simulate blocking and prevent any collisions with the environment or between the robot's body parts. Therefore, we have opted to use stepper motors due to their superior holding torque capabilities. When the user reaches a limit or a collision point, a command is sent to the stepper motors to lock them in place, providing the necessary resistance to simulate blocking. This choice simplifies the design while ensuring that the

stepper motors deliver the significant torque required for accurate and realistic haptic feedback during locking.

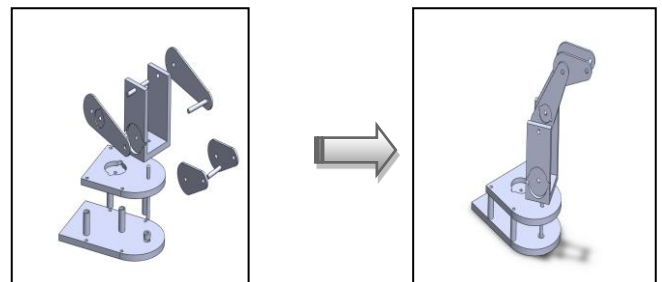
### 3.1 Research design

This research follows an experimental design, focusing on developing, implementing, and testing a customized haptic interface for robotic task programming. The hypothesis is that the customized haptic interface will provide higher torque and better blocking capabilities than the OMNI interface, leading to greater user immersion and efficiency.

The experimental setup includes a virtual environment created with 3DSMax and Visual Studio using OpenGL, and a haptic interface built with stepper motors and position sensors, connected via a Microchip 18F4450 microcontroller. Participants will use both interfaces to perform robotic programming tasks, with data collected on torque, blocking capability, task completion time, and user feedback. Statistical methods will be used to compare the performance of the two interfaces.

### 3.2 Mechanical design

We focused on the manipulator's first three joints, which are responsible for the tool's positioning. This design simulates the behavior of 6R robots, enhancing transparency and user immersion. The articulated arm was initially designed using SolidWorks (Figure 4), allowing a detailed preview of each part, including dimensions and constraints. This process enables optimization before physical realization. The articulated arm consists of a chassis incorporating the first degree of freedom (the base) and two articulated segments forming the other two degrees of freedom. The structure elements are designed to minimize weight and increase strength, inspired by the human forearm. The primary material used is wood.



**Figure 4.** Appearance of pre-assembly and final assembly of the articulated arm

The primary material used is wood, chosen for its lightweight and time-saving benefits during the design phase. Wood prototyping offers distinct advantages, particularly in rapid mechanical design iterations. It allows for quick and easy modifications without requiring specialized tools, making it ideal for testing different configurations of motors, sensors, and gears. In contrast to 3D printing, which involves longer lead times from design in SolidWorks to printing, wood enables faster construction and adjustment. This flexibility significantly reduces the overall prototyping time. Additionally, the cost-effectiveness of wood, combined with its ease of handling, makes it a practical choice for building and refining prototypes. Several prototypes were created using wood, allowing us to seamlessly integrate the components

while optimizing the design for performance.

To incorporate the sensors and motors into our structure, we used gears between the motor axes and the sensor axes (Figure 5). These gears were designed to achieve better precision in sensor readings and improved torque for the motors. The sensors are potentiometers, and a voltage/angle identification procedure was performed for each joint. The motors are stepper motors, used for blocking the joints. The stepper motors used are model 55SI-25DAYA (nominal voltage=12 volts, nominal current=330mA, coil resistance=36Ω, holding torque=1350g/cm, and 48 steps per revolution). We used 5KΩ multi-turn potentiometers.

This assembly forms the mechanical part of our interface (Figure 6).

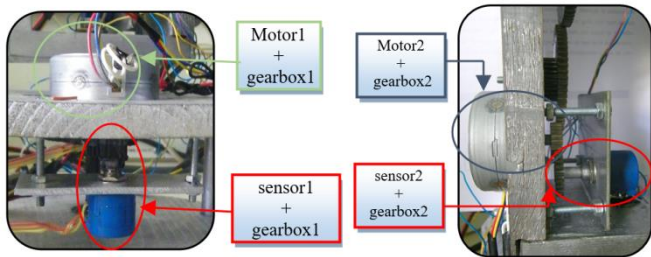


Figure 5. Motors and sensors integrated into the mechanical structure



Figure 6. Mechanical structure final assembly

### 3.3 Electronic design

This section outlines the architecture of the electronic board (inputs/outputs) of our interface, designed around the Microchip PIC 18F4550. The architecture consists of three main components: sensor data acquisition via analog inputs (Port A of the PIC 18F4550), motor control for the articulated arm (haptic interface) via digital outputs (Port B of the PIC 18F4550), and bidirectional communication with the virtual environment via a USB port. The PIC 18F4550 was chosen for its easy-to-program USB module for data transmission. No data processing is performed at the PIC 18F4550 level.

Our goal is to simulate virtual fixture, for which we use stepper motors to create a blocking sensation. The blocking command is generated at the PC level (virtual environment + geometric model) and transmitted to the PIC. The PIC then sets high levels on B0, B1, and B2 (Port B). These signals are sent to three drivers based on H-bridge L298 circuits, which set all phases (A, B, C, D) of the stepper motors to the same

12V potential. This results in an instant blocking effect, thereby simulating a virtual stop.

### 3.4 Virtual environment design

Creating a virtual simulator is essential for real-time communication and interaction between the articulated arm and the virtual environment. To achieve real-time performance, our application was optimized to minimize data transmission time between the acquisition card and the virtual robot. The robot and its virtual environment were designed using 3DSMax and exported to the Visual Studio platform via Okino Polytrans. Visual Studio, using the OpenGL library, was then used to generate the scene and the virtual robot.

#### Components and Process

**3D Modeling:** The robot was meticulously designed using 3DSMax to ensure accurate and realistic representations (Figure 7).

**Exporting Models:** The 3D models were exported from 3DSMax to Visual Studio using Okino Polytrans to maintain model integrity and fidelity.

**Scene Generation:** The Visual Studio environment, leveraging the OpenGL library, was used to create the virtual scene and integrate the virtual robot with its environment (Figure 8).

**Real-time Interaction:** The system was optimized for real-time interaction, ensuring that the articulated arm's movements are accurately mirrored in the virtual environment with minimal latency.

By optimizing the data transmission process and leveraging powerful design tools, our virtual environment provides a robust and responsive platform for users to interact with the haptic interface and the virtual robot, enhancing both immersion and operational efficiency.



Figure 7. Robot in 3DSMax

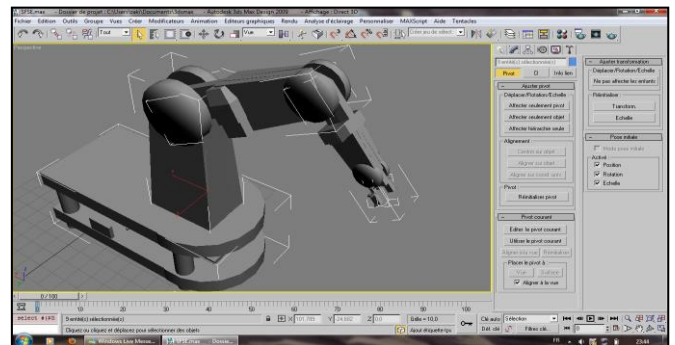


Figure 8. Robot and its environment

To ensure that the virtual robot behaves identically to the real robot, we employed the direct geometric model,

developed using the Denavit-Hartenberg method [29].

We integrated the direct geometric model equations (Eqs. (1)-(3)) into the virtual robot, including its joint and Cartesian (end-effector) stops, to stay within the robot's working area. This model allows us to know the end-effector's position in real-time and can also be used to simulate any other 3R-type robot by adjusting the segment lengths ( $l_1$ ,  $l_2$ , and  $l_3$ ) and stops. The real-time Cartesian positions ( $X$ ,  $Y$ , and  $Z$ ) of the end-effector are obtained as follows:

$$X = \cos(\theta_1) * [l_3 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_2)] \quad (1)$$

$$Y = \sin(\theta_1) * [l_3 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_2)] \quad (2)$$

$$Z = l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2) + l_1 \quad (3)$$

### 3.5 Operation

Our application's operation is based on the synoptic shown in Figure 9. During a task programming scenario, the user manipulates the haptic interface to vary the joint angles. These angle values are sent to the virtual environment and processed by the robot's direct kinematic model. The user can visualize the virtual robot's movements in real-time and observe the position of the robot's end-effector.

This scenario includes a loop involving the user interface, the virtual environment, and the direct kinematic model until the desired position is reached. Once the user achieves the desired position, they save the corresponding joint values.

When the user manipulates the haptic interface, if they reach a joint or Cartesian stop, the application (virtual environment/kinematic model) will inform them by blocking the relevant joint(s) or displaying a visual alert. The user must then reverse their movement to remain within the operational limits.

The direct kinematic model ensures that the virtual robot behaves identically to the real robot, allowing for accurate simulation and task programming. The integration of the model provides real-time feedback on the end-effector's position, which is crucial for precise task execution and learning.

## 4.EVALUATION

Evaluating a VR application involves considering various factors, such as user domain knowledge, feedback devices, and the virtual environment. It is essential to have a reliable platform for accurate evaluation, and the questionnaire must be targeted effectively.

Lordache [7] developed a VR application evaluation tool structured across multiple dimensions. The first dimension pertains to the ergonomics of feedback interfaces (haptic feedback), the second to the high realism of the virtual environment, and the third to the pedagogical effectiveness of the tested applications.

Ellis et al. [30] concluded that there are significant differences in user evaluations of VR applications, emphasizing the importance of designing resources that meet the needs of different user profiles.

Monetti et al. [16] used two approaches: one with a real robot and the other with a virtual robot. Execution time and success rates were used to estimate learning efficiency.

In our study, we conducted tests with a panel of users comprising master's students in robotics and robotics technicians working in the industry. The users were randomly selected, each in their own group, without establishing specific criteria. We thought this would give more credibility to our assessment. The panel included 4 female and 6 male subjects with an average age of 23 years.

In selecting the panel: we considered two criteria.

The first is that second-year master's students in automation and systems (a program offered at our institution) have a strong theoretical background but very little practical experience. This is relevant to understand the advantages brought by the use of our platform in the educational field and its relevance for teaching robotics, particularly robotic tasks.

The second criterion is that robotics technicians have good practical skills in the field of robotics since they come from industry, but their theoretical knowledge is rather vague. Therefore, this will allow us to evaluate our platform in another area, which is the learning of tasks in the industrial environment

This will allow us to determine whether it is well-suited for students with a strong theoretical background as well as for robotics technicians with solid practical experience, and to evaluate the potential of our platform in terms of time savings for programming robotic tasks.

### 4.1 Evaluation protocol

The users participated in a robotics course covering direct and inverse kinematics, workspace, and joint space. They were provided with a computer running our virtual environment and tasked with finding the Cartesian coordinates of three points in the task space of the tool. Users followed two different protocols:

#### 1. Virtual environment without haptic interface

a. Users relied solely on visualizing the virtual environment to input the joint values  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$ . This process was repeated until the desired position was reached. The advantage of this protocol was the 3D visualization of the robot and its environment, aiding users in selecting the joint values.

b. We did not use the inverse kinematic model to avoid users having to choose between multiple solution sets for  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$ .

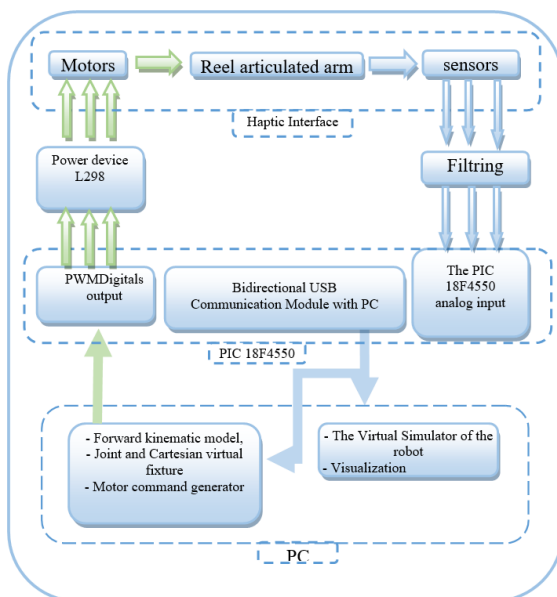


Figure 9. General operating flowchart of our platform

## 2.Virtual environment with haptic interface

a. Users utilized the haptic interface within the same virtual environment, allowing for full immersion in the virtual world. Movements imposed on the haptic interface, causing variations in angles  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$ , resulted in real-time animations of the virtual robot in the task space.

b. This method helped users find the  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  values corresponding to the desired end-effector position. The haptic interface automatically introduced the angular values of its joints, displaying the X, Y, and Z coordinates of the end-effector in the task space.

Users were then asked to propose three intermediate points between the previously found points, making a total of six points.

After completing both protocols, users answered a series of questions.

### 4.2 Evaluation of tests

This procedure allowed us to evaluate both methods and determine which was faster, more practical, more effective, and provided better immersion in the virtual environment.

The evaluation involved a questionnaire divided into two parts. The first part assessed user satisfaction with or without the haptic interface. The second part evaluated the relevance of using the haptic interface. A final quantitative assessment was based on the task completion times with and without the haptic interface.

#### Questions for evaluating satisfaction:

- Did you enjoy the part without the haptic interface?
- Did you enjoy the part with the haptic interface?
- How did you find the handling without the haptic interface?
- How did you find the handling with the haptic interface?
- Did you find the exercise easy without the haptic interface?
- Did you find the exercise easy with the haptic interface?
- Did the interface spark interest?

#### Questions for evaluating relevance: (forme)

If you had to program a robot for a given task, would you use the interface?

Do you think the interface enhances understanding and visualization?

Do you think the interface helped you immerse yourself in the robot's environment?

Do you prefer using the haptic interface or not?

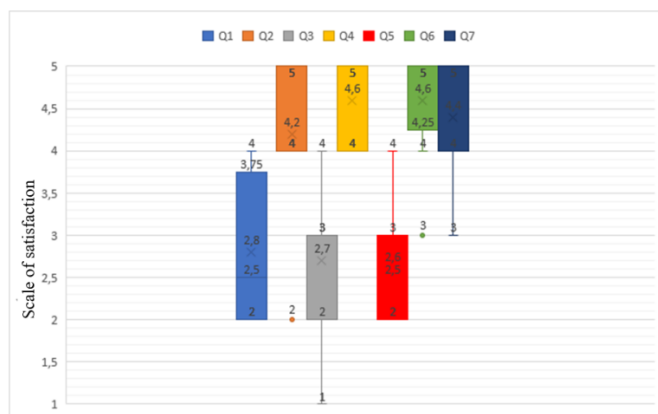


Figure 10. Boxplot of responses related to satisfaction for the first seven questions

## 4.3 Results

A scale of 1 to 5 was provided to the users, with 1 indicating not at all satisfied and 5 indicating very satisfied. The results of the responses to questions 1, 2, 3, 4, 5, 6, and 7 are illustrated in the following graphs (Figure 10).

**Evaluation of the Relevance Criterion:** This evaluation presents the users' responses to questions 8, 9, 10, and 11, which demonstrate the users' adherence to the use of the haptic interface as shown in Figure 11.

**Evaluation of the Speed Criterion (Time Efficiency):** This evaluation shows the time taken by users to complete the two exercises as illustrated in Figure 12 and Table 1.

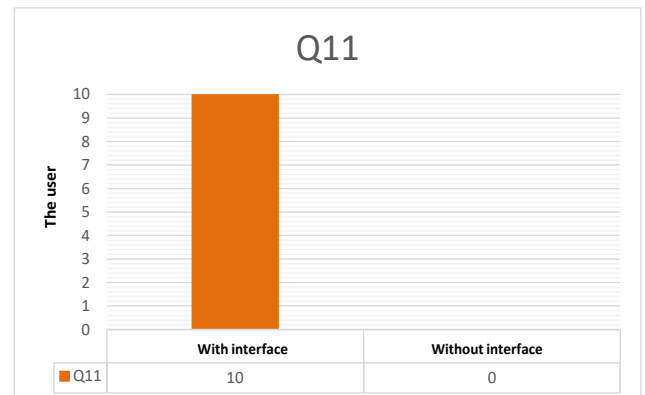
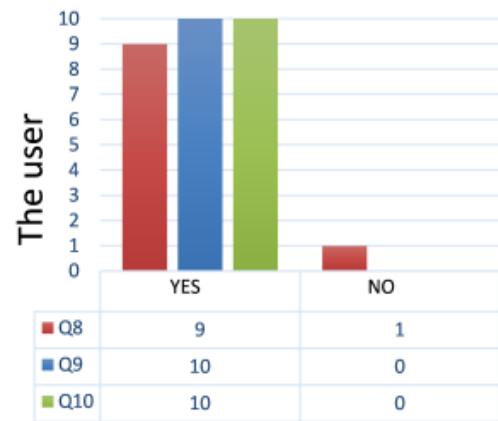


Figure 11. Responses to questions 8, 9, 10 and 11

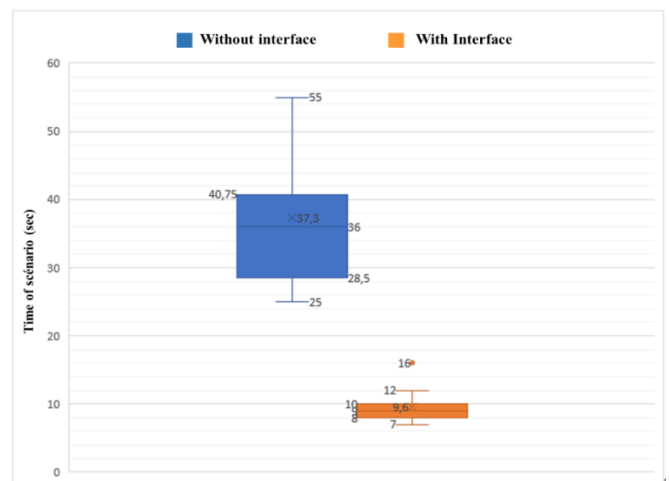


Figure 12. Boxplot of the time taken by each user for the test with and without the interface

**Table 1.** Execution time for each user during the test, with and without the interface (time in seconds)

	Without Interface (sec)	With Interface (sec)
User 1	41	16
User 2	34	9
User 3	27	8
User 4	55	7
User 5	55	8
User 6	30	9
User 7	28	10
User 8	38	10
User 9	40	12
User 10	25	7

#### 4.4 Discussion

The boxplot of the results from the first seven questions (Figure 10) shows that users are relatively satisfied with the haptic interface and find it easier to use for the given exercise compared to the part without the interface.

The results illustrated in the previous Figure 11 show a predominance of affirmative responses, indicating that users have better understanding and visualization due to the ease of immersion in the robot environment for a given task with the haptic interface.

The boxplot of the times with and without the haptic interface (Figure 12) shows that the haptic interface reduces the time for understanding, visualizing, and solving the exercise by four times. However, these results are subject to limitations such as:

- It was not possible to verify if the results obtained were influenced by the way we explained the concepts and methods for solving the exercises.
- It was not possible to verify if the poor performance of some students was due to their deficits in spatial visualization.
- We noticed that the group of technicians adapted to using the platform much more quickly, which is due to their practical experience. The student group took a bit longer to adapt, as it was new for them, but in the end, the results were almost identical in terms of evaluations.
- The technicians were quicker in learning the task, but once the students became familiar with the platform, their results improved.

#### 4.5 System Usability Scale

The System Usability Scale (SUS) [31] was applied to our platform. We submitted the 10 questions to 10 users. The overall average score of the users is 73.25 (min=62.5 to max=82.5). This score reflects a good appreciation of our platform.

#### 4.6 Evaluation's conclusion

The haptic interface achieved the expected results due to the ease of use it offered and the time savings obtained compared to the method without the interface. We also note that the user panel was generally satisfied with what the interface provided.

### 5. CONCLUSIONS

In this study, we developed a comprehensive robotic task learning platform integrating a haptic interface with a virtual

environment, significantly enhancing task learning efficiency and effectiveness. Our evaluation demonstrated that the haptic interface facilitates learning, even for users with limited robotics knowledge, by providing an immersive experience that improves understanding of robotic movements. Notably, tasks were completed four times faster compared to a non-haptic platform, highlighting substantial time efficiency. Users reported high satisfaction and engagement, emphasizing the system's ease of use and enhanced interaction. To further this research, we are developing a more advanced interface with six degrees of freedom instead of three. With the experience gained from our work, we will use 3D printing technology in its design. This new interface will be integrated into virtual simulation platforms that have a rich library of industrial robot manipulators, such as V-REP, Robot Studio, and Rviz. This will enable more comprehensive task simulations and broaden the scope of applications to a wide range of robots.

Future improvements will focus on refining user interface design to enhance intuitiveness and accessibility. Overall, our study underscores the potential of haptic interfaces in transforming robotic task programming and learning, contributing significantly to the field of human-robot interaction.

### REFERENCES

- [1] Fuchs, P., Moreau, G., Arnaldi, B., Guitton, P. (2006). *Le traité de la Réalité Virtuelle*, volume 1: L'homme et l'environnement virtuel. Collection sciences mathématiques et informatiques. Les Presses de l'Ecole des Mines. <http://www.pressesdesmines.com/produit/le-traite-de-la-realite-virtuelle-volume-1-l-homme-et-l-environnement-virtuel/>.
- [2] Gosselin, F. (2005). *Optimisation des interfaces haptiques: Problèmes, méthodes, applications*. 17ème Congrès Français de Mécanique, pp. 1-12.
- [3] Casiez, G. (2004). *Interface Contribution à l'étude des interfaces haptiques Le DigiHaptic: Un périphérique haptique de bureau a degrés de liberté séparés*. Thèse de doctorat, Lille. Phd thesis. USTL Lille 1. <https://gery.casiez.net/publications/TheseCasiez.pdf>.
- [4] Preusche, C., Hirzinger, G. (2007). Haptics in telerobotics. *The Visual Computer*, 23(4): 273-284. <https://doi.org/10.1007/s00371-007-0101-3>
- [5] Preusche, C. (2006). Haptics and telerobotics. In *IST 2006 – Haptex Workshop*, Helsinki, Finland. <https://elib.dlr.de/46898/>.
- [6] Žlajpah, L. (2008). Simulation in robotics. *Mathematics and Computers in Simulation*, 79(4): 879-897. <https://doi.org/10.1016/j.matcom.2008.02.017>
- [7] Iordache, D.D. (2020). Evaluation of virtual reality-based applications in education. In *Conference proceedings of eLearning and Software for Education (eLSE)*, pp. 499-504.
- [8] Huang, X., Zhang, P. (2009). A general robot simulation system for education and research. In *2009 International Conference on Computational Intelligence and Software Engineering*, pp. 1-4. <https://doi.org/10.1109/CISE.2009.5364871>
- [9] RobotStudio 5, ABB, <http://www.robotstudio.com/rs5/>.
- [10] Rohmer, E., Singh, S.P., Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent*

- Robots and Systems, pp. 1321-1326. <https://doi.org/10.1109/IROS.2013.6696520>
- [11] Raz, T. (1989). Graphics robot simulator for teaching introductory robotics. *IEEE Transactions on Education*, 32(2): 153-159. <https://doi.org/10.1109/13.28048>
- [12] Hurtado, C.V., Valerio, A.R., Sanchez, L.R. (2010). Virtual reality robotics system for education and training. In 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference, pp. 162-167. <https://doi.org/10.1109/CERMA.2010.98>
- [13] Darmoul, S., Abidi, M.H., Ahmad, A., Al-Ahmari, A.M., Darwish, S.M., Hussein, H.M. (2015). Virtual reality for manufacturing: A robotic cell case study. In 2015 International Conference on Industrial Engineering and Operations Management (IEOM), pp. 1-7. <https://doi.org/10.1109/IEOM.2015.7093880>
- [14] Loreto-Gómez, G., Rodríguez-Arce, J., González-García, S., Montaña-Serrano, V.M. (2019). Analysing the effect of the use of 3D simulations on the performance of engineering students in a robotics course: Findings from a pilot study. *The International Journal of Electrical Engineering & Education*, 56(2): 163-178. <https://doi.org/10.1177/0020720918790113>
- [15] Togias, T., Gkourmelos, C., Angelakis, P., Michalos, G., Makris, S. (2021). Virtual reality environment for industrial robot control and path design. *Procedia CIRP*, 100: 133-138. <https://doi.org/10.1016/j.procir.2021.05.021>
- [16] Monetti, F.M., de Giorgio, A., Yu, H., Maffei, A., Romero, M. (2022). An experimental study of the impact of virtual reality training on manufacturing operators on industrial robotic tasks. *Procedia CIRP*, 106: 33-38. <https://doi.org/10.1016/j.procir.2022.02.151>
- [17] Wolfartsberger, J., Zimmermann, R., Obermeier, G., Niedermayr, D. (2023). Analyzing the potential of virtual reality-supported training for industrial assembly tasks. *Computers in Industry*, 147: 103838. <https://doi.org/10.1016/j.compind.2022.103838>
- [18] Aron, C., Marius, I., Cojanu, C., Mogan, G. (2008). Programming of robots using virtual reality technologies. *Product Engineering: Tools and Methods Based on Virtual Reality*, 555-563. [https://doi.org/10.1007/978-1-4020-8200-9\\_30](https://doi.org/10.1007/978-1-4020-8200-9_30)
- [19] Glamnik, A., Šafarič, R. (2012). Control of KUKA KR 5 robot with a haptic device. In 2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV), pp. 1-7. <https://doi.org/10.1109/REV.2012.6293116>
- [20] Gonzalez-Badillo, G., Medellin-Castillo, H., Lim, T., Ritchie, J., Garbaya, S. (2014). The development of a physics and constraint-based haptic virtual assembly system. *Assembly Automation*, 34(1): 41-55. <https://doi.org/10.1108/AA-03-2013-023>
- [21] Marszalik, D. (2014). Application of haptic omni device to determination of the set point trajectory. In Proceedings of the 2014 15th International Carpathian Control Conference (ICCC), pp. 332-335. <https://doi.org/10.1109/CarpathianCC.2014.6843622>
- [22] Crespo, R., García, R., Quiroz, S. (2015). Virtual reality simulator for robotics learning. In 2015 International Conference on interactive collaborative and blended learning (ICBL), pp. 61-65. <https://doi.org/10.1109/ICBL.2015.7387635>
- [23] Dhivin, D., Jose, J., Bhavani, R.R. (2017). Bilateral tele-haptic interface for controlling a robotic manipulator. In 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), pp. 1614-1620. <https://doi.org/10.1109/ICICT1.2017.8342812>
- [24] Knopp, S., Lorenz, M., Pelliccia, L., Klimant, P. (2018). Using industrial robots as haptic devices for VR-training. In 2018 IEEE conference on virtual reality and 3D user interfaces (VR), pp. 607-608. <https://doi.org/10.1109/VR.2018.8446614>
- [25] Andersson, M., Syberfeldt, A. (2024). Improved interaction with collaborative robots-evaluation of event-specific haptic feedback in virtual reality. *Procedia Computer Science*, 232: 1055-1064. <https://doi.org/10.1016/j.procs.2024.01.104>
- [26] San Martín, J., Triviño, G. (2006). A study of the Manipulability of the PHANToMTM OMNITM Haptic Interface. Workshop on Virtual Reality Interactions and Physical Simulations.
- [27] Achour, N., Daoudi, A. (2012). An haptic interface design to virtual environments. In 2012 24th Chinese Control and Decision Conference (CCDC), pp. 1473-1476. <https://doi.org/10.1109/CCDC.2012.6243006>
- [28] Daoudi, A., Chibani, D. (2019). A haptic interface design for robotics teaching task. In 2019 International Conference on Advanced Electrical Engineering (ICAEE), pp. 1-5. <https://doi.org/10.1109/ICAEE47123.2019.9014725>
- [29] Fu, K.S., Gonzalez, R.C., Lee, C.G., Freeman, H. (1987). *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill.
- [30] Ellis, R.C.T., Dickinson, I., Green, M., Smith, M.B. (2006). The implementation and evaluation of an undergraduate virtual reality surveying application. In: BEECON 2006 Built Environment Education Conference, London, UK.
- [31] Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan; B. Thomas; B. A. Weerdmeester; A. L. McClelland (eds.). *Usability Evaluation in Industry*. London: Taylor and Francis.