A MODULAR HANDHELD DATA ACQUISITION DEVICE FOR LEARNING FROM DEMONSTRATION APPLICATIONS

Jelmer Hofstra

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF BIOMECHANICAL ENGINEERING

EXAMINATION COMMITTEE
Dr. E.H.F Asseldonk
Dr. Ir. M. Vlutters
Dr. Ir. E.C. Dertien

UNIVERSITY OF TWENTE.

20-11-2025

A Modular Handheld Data Acquisition Device for Learning from Demonstration Applications

Jelmer Hofstra
Nakama robotics Lab , University of Twente

November 20, 2025

Abstract – Labor shortages across industries are increasing the demand for robots and skilled robot programmers. Learning from Demonstration (LfD) offers a promising approach to address this challenge, allowing robots to acquire skills by observing human demonstrations. Existing handheld devices for LfD are often task-specific and limited to particular learning phases. This work presents a modular handheld data acquisition device for LfD, designed to support multiple tasks and learning methods with a single platform. The device features a robotic flange to enable quick swapping of end-effectors and their associated sensors, providing flexible data collection across different tasks. An ArUco marker cube facilitates basic pose and position tracking, while ROS2 integration supports data communication and system control. Validation experiments demonstrated reliable performance, and user evaluations confirmed high ease of use and comfort. By offering a versatile and user-friendly platform, this modular device reduces the need for multiple task-specific handheld devices and broadens its applicability in LfD research and practical applications. Future work will focus on mapping the acquired demonstrations onto specific robots while accounting for kinematic constraints and workspace limitations, enabling seamless task reproduction.

Keywords — Learning from Demonstration, Handheld Device, ROS2, Modular Robotics, Human–Robot Interaction

1 Introduction

Global markets are increasingly facing labor shortages, often driven by an aging population and declining birth rates. These are projected to reduce workforce participation from 65.8% in 2005 to 61.4% by 2050 [1]. To address this growing challenge, extensive research across various sectors, from healthcare to construction, has focused on robotics as a key solution [2, 3, 4].

This trend is also a primary driver of the rapid expansion of the robotics market, which is growing at approximately 16 percent annually and is expected to reach $165.2 billion by 2029 [5]. The largest segment of this market is the service robotics sector. Its growth has been tremendous, rising from $36.2 billion in 2022 to an expected $103.3 billion by 2026 [6]. This growth is particularly evident in industries such as healthcare, production, and manufacturing, where the interactions between a robot and the objects and persons in its environment are crucial. [7, 8].

As robots become more prevalent, the need for skilled coders to program various types of robots increases. This is especially the case for robots that operate in unstructured and dynamic environments. Highly dynamic environments requires a more robust and adaptable programming approach. The traditional method of programming robots, often referred to as hard-coding, is time-consuming and highly task-specific. To address this growing problem, Learning from Demonstration (LfD) has emerged as a promising approach [9].

LfD is a robotics approach in which a robot acquires new skills by observing demonstrations, thereby reducing the need for explicit programming [10, 11]. In LfD, a human demonstrator typically performs a sequence of actions that the robot records and later generalizes into executable policies.

One common method within LfD is kinesthetic teaching, where the teacher physically guides the robot's joints to illustrate the task.

Because LfD leverages intuitive human demonstrations rather than robotics domain-specific knowledge, it enables individuals without technical expertise to teach robots effectively. However, since the robot directly learns from the demonstrations, inaccuracies in the input will be reproduced in the execution, underscoring the importance of precise and reliable demonstrations.

During the learning phase, the recorded demonstrations are processed and encoded into a representation that can later be reproduced or adapted by the robot. This process can be implemented using various techniques, such as Artificial Neural Networks (ANNs), Hidden Markov Models (HMMs), Dynamic Movement Primitives (DMPs), Gaussian Mixture Regression (GMR), or diffusion-based imitation learning methods [12, 13, 14]. The choice of learning mechanism is highly task-specific, as different methods offer distinct advantages depending on the complexity, variability, and constraints of the demonstrated task.

Since the quality of the learned representation strongly depends on the naturalness and accuracy of the demonstrations, the way demonstrations are provided becomes equally important. To enable natural task demonstrations, handheld devices can be employed. Their main advantage is that the teacher can illustrate tasks more intuitively compared to directly manipulating the robot. Furthermore, no physical robot is required during the demonstration phase, reducing both cost and safety concerns, and allowing demonstrations to be conducted in diverse environments. However, handheld devices also introduce certain drawbacks. Since no physical

robot is present, the demonstrator may be unaware of the robot's physical constraints, such as workspace limits or maximum accelerations. Consequently, the collected data must be mapped before execution on the robot.

This mapping process involves several steps: first, aligning the origins of the coordinate systems; second, accounting for robotic constraints. These include geometric constraints (e.g., maximum workspace volume), kinematic constraints (e.g., handling singularities in robots with many degrees of freedom), and dynamic constraints (e.g., maximum velocities and accelerations) [15, 16].

When examining handheld devices for demonstration purposes, many devices are specifically designed for particular tasks. For instance, Hengtai Dai et al. [17] developed an exoskeleton-like handheld device for data acquisition, with movements subsequently learned using Gaussian Mixture Regression (GMR). The device employs a fiducial marker for motion tracking, an IMU for orientation, and a displacement sensor in the gripper, enabling it to capture position, orientation, and grasping data over time. While this design is well suited for simple pick-and-place experiments, it is insufficient for more complex or dynamic environments that require vision-based feedback or force-based interactions.

Cheng Chi et al. [18] proposed a handheld device equipped with an internal camera oriented toward the gripper. Demonstration data were processed using Diffusion Policy, a diffusion-based imitation learning framework. This design is specialized for visuomotor learning and supports more complex pick-and-place experiments in dynamic environments compared to the device of Hengtai Dai et al. [17]. However, it lacks the capability to perform force-based tasks, and since no explicit position data are collected, the system is unnecessarily complex for less dynamic or simpler tasks.

The handheld device developed by Hsien-Chung Lin et al. [19] tracks position using an external camera and measures forces at the end of its cylinder-shaped body. Demonstration data were processed using Gaussian Mixture Regression (GMR) combined with Dynamic Time Warping (DTW). This design excels in force-based tasks, such as sanding or peg-in-hole experiments, due to the integrated force sensor. However, it is not well suited for pick-and-place operations, as it lacks a gripper, and it is unsuitable for dynamic environments because it does not include an internal camera.

These examples illustrate that current handheld devices are often task-specific, limiting their applicability across a range of different tasks. To address this limitation, it is more efficient to develop a modular device for LfD data acquisition. Such a device would allow different sensors and end-effectors to be easily added or removed based on the task. The objective of this work is to design and validate a modular handheld data acquisition device that can accommodate interchangeable end-effectors for multiple LfD pipelines, thereby minimizing the reliance on task-specific handheld devices.

2 Design

This section outlines the overall design process. First, the stakeholders are identified. Based on their needs, functions and requirements are derived. From these, a new handheld device is designed. Finally, the design is elaborated in terms of both mechanical and software aspects.

2.1 Stakeholders

The primary stakeholder for this work is the Nakama Robotics Lab [20] at the University of Twente, the Netherlands. The lab focuses on Learning from Demonstration (LfD), where collaboration between humans and robots is central. Its aim is to develop robotic platforms capable of performing a wide variety of tasks, which requires a reliable, user-friendly, and modular handheld device to support diverse experiments.

A second stakeholder is the broader scientific community, which benefits from the system's modularity and adaptability for research beyond the scope of the Nakama Robotics Lab. Taken together, these stakeholders emphasize the importance of usability and modularity, which have been central to the design decisions in this work.

2.2 Design Requirements

Building on the needs of the identified stakeholders, the requirements of the system are defined. These design considerations aim to ensure the development of a fully functional platform for LfD, with a strong emphasis on modularity to support a wide range of tasks. To guide the specification of these requirements, three representative LfD tasks are introduced, as outlined below:

1. **Orientation and Position-Based Task**
   This task evaluates the device's capability to accurately track movement and orientation. It is relevant for applications requiring precise trajectory following, such as spray painting, and is conducted without an end effector to isolate position and orientation performance.
2. **Force-Based Task**
   This task investigates scenarios in which position and orientation information alone are insufficient, and force feedback is necessary. It is applicable to tasks involving direct interaction with objects, where accurate force measurement is critical for safe and effective execution.
3. **Vision-Based Task**
   This task focuses on learning from demonstrations in which actions are guided by visual input. The system extracts critical information, such as object positions, states, and spatial relationships, from images or video frames to reproduce or generalize demonstrated behavior.

In addition to modularity and the ability to acquire sensor data for different LfD tasks, further requirements were formulated to guide both the design and validation process. These requirements, both technical and functional, are derived from the three representative tasks and the stakeholders and are summarized in Table 1. Each requirement is stated together with a rationale explaining its necessity and the method of validation.

Table (1) *Functional and Technical Requirements with Corresponding Validation Criteria*

Technical Requirements		
Requirement	**Explanation**	**Validation Criteria**
Precision and Accuracy	The device should collect data with high accuracy to ensure reliable task replication by the robot.	**V1.** Experimental validation of measurement accuracy and precision to confirm suitability for task specific data collection.
Sensor Synchronization	All additional sensors must be time-synchronized to ensure consistent data capture and reliable robot execution.	**V2.** Experimental validation of sensor temporal alignment to confirm whether additional time-correction is required during data processing.
Robotic System Integration	The device must be compatible with standard robot end-effectors and sensors, supporting both mechanical and software interfaces.	**V3.** The handheld device supports standard robot communication protocols and mechanical interfaces.
Cable Management	Cables must be properly routed and secured to ensure safe and reliable operation.	**V4.** Cables must be strain-relieved and secured to prevent tension during demonstrations. **V5.** Cables must not interfere with task execution.

Functional Requirements		
Requirement	**Explanation**	**Validation Criteria**
Modularity	The device must allow easy attachment and removal of sensors or other components, enabling use in a wide range of tasks.	**V6.** The device includes a standardized robotic flange supporting modular attachments. **V7.** The device's modular configuration is validated by successfully executing at least two distinct LfD data acquisition tasks using different end-effector setups.
Ergonomics	The device must be comfortable to use for prolonged sessions and suitable for a wide range of users.	**V8.** The device weight must not exceed $0.5\,\mathrm{kg}$. **V9.** The operator must be able to perform a demonstration without discomfort.
Ease of Use	The device should be intuitive to operate by different users with minimal training.	**V10.** Setup time for the device must be under 2 minutes. **V11.** Operators can change the end-effector independently within 2 minutes. **V12.** Clear visual indicators must show when recording starts and stops.
Portability	The device should be easy to transport for demonstrations and testing in different environments.	**V13.** The complete setup must fit in a standard $25\,\mathrm{L}$ backpack (excluding end-effectors).

2.3 Mechanical Design

This section presents the technical aspects of the handheld device, with particular emphasis on the key components that ensure its functionality and accuracy. An overview of the device and its elements is shown in Figure 1. The complete SolidWorks design files are provided in Appendix B.

Figure (1) *The designed modular handheld proxy device device for LFD, with an ArUco marker cube, a flange designed for robotics, internal cable routing, and a cable retainer to secure the cables of possible added sensors.*

2.3.1 ArUco Marker Cube

Accurate position tracking is critical for this application, as most tasks depend on precise position and orientation data. Several methods were considered, including GPS, IMUs, and vision-based systems [21, 22]. However, as summarized in Table 2, the requirements for accuracy, portability, and ease of implementation make many of these approaches impractical for the present design.

IMUs provide reliable orientation data and can estimate position through double integration of acceleration. However, this approach introduces significant drift errors, even after filtering. While corrections using known reference points or advanced filtering techniques have been proposed [21], these methods are not feasible in this context due to the absence of reference positions during motion.

Table (2) *Comparison of position tracking methods*

Method	Accuracy	Portability	Ease of Implementation	Suitability for This Design
GPS	Low (indoors)	High	Moderate	Not suitable due to low accuracy, particularly in indoor environments.
IMU	Low	High	Moderate	Not suitable due to accumulated position errors from noisy acceleration data.
Vision-Based (Reflective markers)	Very High	Low	Low	Not suitable due to excellent accuracy, but poor portability and high infrastructure requirements.
Vision-Based (ArUco)	High	Moderate	Moderate	Best suitable due to accurate and robust position/orientation tracking, even during rotation.

For orientation tracking, an IMU can still be easily incorporated and remains a valid option.

GPS is another common solution, but its accuracy is insufficient for indoor environments, rendering it unsuitable for this application.

Vision-based tracking offers higher precision and is widely implemented through optical motion capture (MoCap) systems. Although MoCap provides highly accurate position and orientation data, it requires multiple calibrated cameras in a controlled environment, which compromises portability and scalability.

To balance accuracy and practicality, a lightweight vision-based solution using fiducial markers was adopted. ArUco markers were chosen due to their widespread use and robustness in similar applications [17, 18, 22, 23]. Tracking accuracy depends on several factors, including camera quality, marker illumination, and marker size. Wang et al. [24] report that with a marker size of approximately 12 cm, a relative measurement error of around 2.5% of the measured distance can be achieved. An ArUco cube was mounted on top of the handheld device, ensuring visibility from multiple angles to a single external camera. This configuration enables robust tracking of both position and orientation during motion while maintaining compactness and portability.

For orientation tracking, the same ArUco markers can be utilized. While IMUs remain the gold standard for orientation estimation, distinct markers on each cube face provide a practical alternative. Chen et al. [25] reported an orientation accuracy of approximately 0.1 rad, which is sufficient for some tasks. For applications requiring higher accuracy, an IMU could be integrated with the end effector to achieve the desired precision.

2.3.2 Modularity

For modularity, a robotic flange was added to the device. The flange is based on the ISO 9409-1 standard (Manipulating Industrial Robots – Mechanical Interfaces) [26], which is commonly used on robotic arms such as the Franka Emika Panda and the KUKA LBR iiSY cobot [27, 28]. This standard ensures that different end-effectors can be attached in the same way as on their industrial robotic counterparts.

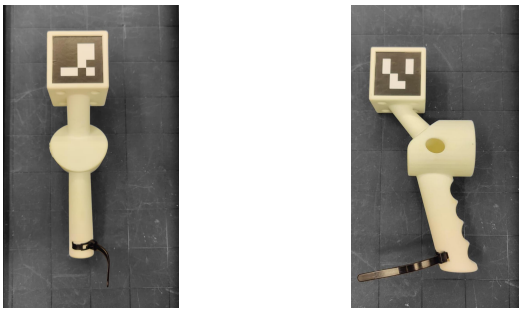
To facilitate the attachment of various end effectors, helicoils were inserted into the flange holes after 3D printing the device.

2.3.3 Handle

The handle features an ergonomic cut-out with four arcs for the fingers, making it easier to grasp securely. In most cases, the center of mass of the device shifts toward the end-effector, particularly when sensors or additional components are attached. To compensate for this weight distribution, the handle is designed with a slight 10% tilt, which improves control and handling of the device [29].

2.3.4 Cabling

Internal routing of the cabling is designed to prevent obstruction of the ArUco markers, which could occur if the flange faces upwards. A cutout is provided to secure the cables using a tie-wrap, visible in Figure 2. Initially, a swivel was considered; however, due to the relatively large diameter of certain cables, such as Ethernet, a sufficiently sized swivel would not fit within the handle design. The tie-wrap ensures that the cables remain fixed during measurements, reducing noise. Additionally, it prevents cable tension and potential damage to attached peripherals.



(a) *Back view.* **(b)** *Side view.*

Figure (2) *Back and side views of the handheld device. The hole for internal cable routing is visible in Figure 2a. At the bottom of both images, a tie-wrap can be seen, which is used to tighten the cables during demonstrations.*

2.4 Software design

The software architecture consists of two main components: data acquisition and data pre-processing of position and pose information, as most learning methods rely on these data types. Both components are described in detail in the following paragraphs. All source code is available in the GitHub repository, referenced in Appendix B.

2.4.1 Data Acquisition

To ensure modularity in data acquisition, Robot Operating System 2 (ROS2) is used for data logging. The use of ROS2 nodes allows sensors with similar functionality to be easily exchanged, requiring only minor modifications within the corresponding node while leaving the rest of the processing chain unaffected. In addition, ROS2 provides a common communication interface, meaning that even completely different types of sensors can interact with the system in a uniform way. Moreover, ROS2 benefits from a large and growing community, which facilitates component reuse and simplifies system extensions.

Each sensor has a dedicated ROS2 node. For the standalone device (no end-effector) this is a node for the external camera for position tracking. This node publishes images of the handheld device over time, tracking the ArUco cube, along with the camera's orientation and, when available, an associated depth map. Other sensor nodes publish data specific to their function, for example, wrench data in the case of a force sensor. All sensor data is collected by a dedicated ROS2 listener node, which subscribes to all active publishers nodes. The listener then records the sensordata in a ROS2 bag (`.db3` file).

Storing data in ROS2 bag files provides several advantages. First, it enables replay and offline processing, allowing future data points to be incorporated into filtering pipelines to improve accuracy. Second, bag files employ efficient storage techniques, making them well suited for handling large volumes of information, which is particularly important when working with high-frequency sensors.

2.4.2 Data Pre-processing

Data pre-processing is performed prior to the actual learning phase in LfD. Since the learning phase varies depending on the chosen technique, there is no universal processing pipeline. However, certain initial steps are required to make the data suitable for further processing. These steps are described below.

Because pre-processing is sensor-specific, only the pre-processing of the standalone handheld device (position and orientation tracking with the ArUco cube) is described here. The complete pre-processing pipeline for the ArUco cube is shown in Figure 3.

First, the data are extracted from the recorded bag file. The position and orientation of the detected ArUco cube face are then computed using the OpenCV ArUco detector, including the orientation, pixel coordinates of the marker plane, and the ArUco ID. The pixel coordinates of the plane's center are used to estimate the distance from the camera to the center of the face. If the external camera provides a depth map topic, this depth map is used; otherwise, the intrinsic camera parameters, together with the OpenCV toolbox, are used to compute the distance based on a pinhole camera model.

Once the position and orientation of one cube face are known, the cube's center can be calculated. This is achieved by creating a vector perpendicular to the face, pointing toward the cube's center, and adding it to the face's center position. The cube's orientation is then derived from the face's orientation, combined with its corresponding ArUco ID, to obtain the global orientation of the handheld device.

Full automation of this process is not feasible due to inherent ambiguities with ArUco markers [30]. For example, when a marker is perfectly perpendicular to the camera, OpenCV may fail to produce a valid solution, often resulting in a reversed Z-axis orientation (pointing toward the camera). Attempts to automate correction using information from previous frames were also unsuccessful, as the software cannot reliably detect markers under poor lighting conditions or during rapid motion, which can cause pixel blurring and hinder accurate back-tracing. To address these challenges, an interactive display was developed. This tool allows users to step through all image frames containing an ArUco marker and manually correct orientations to ensure a right-handed coordinate frame. To minimize manual corrections, the operator selects the first correctly oriented frame, and subsequent frames are automatically proposed based on the previous orientation. The display also enables the removal of false positives, such as markers not part of the device, and facilitates rapid selection of start and end frames for a sequence. An example of this display is shown in Figure 4.

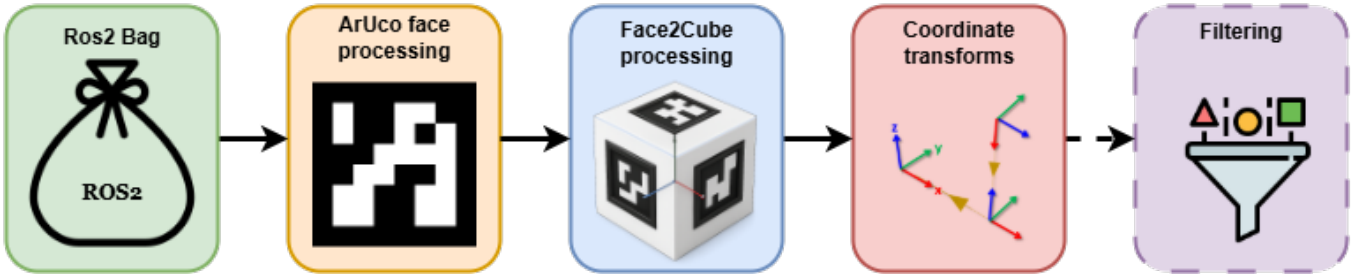


Figure (3) *The data pre-processing processing pipeline from the ROS2 bag data to the optional filtering stage. The steps include processing the ArUco face, followed by the face-to-cube conversion and coordinate transformation, and, if needed, filtering the data.*

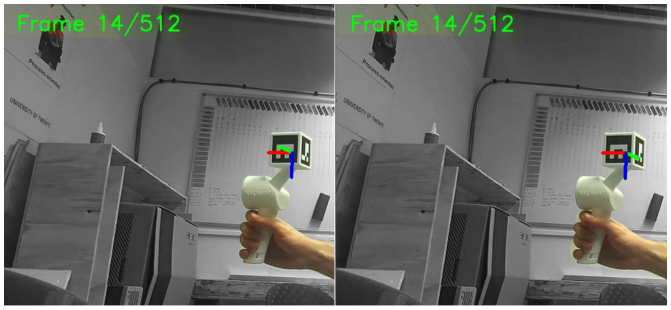


Figure (4) *Interactive coordinate frame correction interface. Left: incorrectly assigned left-handed frame displayed on the ArUco cube; Right: corrected right-handed frame. The interface allows users to easily switch between coordinate frames (from left to right). The current frame index (e.g., 14/512) is shown in the top-left corner of each view.*

After determining the cube's center position and orientation, a series of coordinate frame transformations is applied to represent the device's motion consistently. First, the camera frame is transformed into a fixed world frame using the camera's orientation (T_1). This world frame serves as a stable reference for all subsequent measurements. Next, the world frame is transformed into the marker cube frame at its first detection (T_2), which is defined as the origin of the motion sequence, setting the initial position to $(0, 0, 0)$. By establishing the cube frame relative to the first sighting, all subsequent movements of the device are expressed relative to this initial reference point, which is particularly convenient for mapping motions to the robot. Finally, a transformation from the cube frame to the end-effector frame is applied. This last transformation is end-effector specific and is therefore not shown in the figure, as different end effectors define their coordinate frames differently.

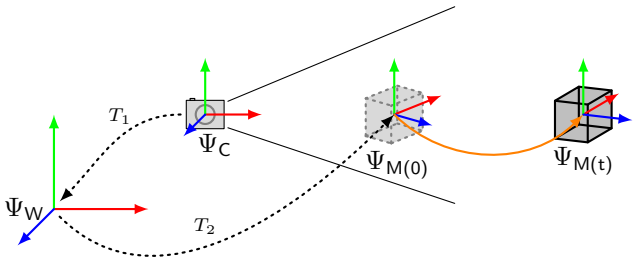


Figure (5) *Visual representation of the frame (Ψ) transformations $T1$ and $T2$. The camera (c), world (w), and marker cube (M) are shown at and after $t = 0$, with the orange arrow indicating the cube's movement.*

The final optional pre-processing step is filtering, which smooths the data. A low-pass filter can be applied to position data to reduce sudden jumps between consecutive frames, resulting in a more natural spline. For orientation data, a moving average filter can be applied using spherical linear interpolation (SLERP) on quaternions.

3 Design Evaluation

The design evaluation of the handheld device is divided into three parts. First, the modularity of the device is assessed (**V2 & V8**). Second, the accuracy of the standalone handheld device (**V1**) is evaluated. With the use of a questionair **V9, V10, V11 & V12** are validated. Finally, the remaining functional and technical requirements (**V3, V4, V5, V6 & V13**) are validated through practical use and observation.

3.1 Materials

The materials used for the evaluation are listed below:

XYZ-table
This is a specialized table that allows selective locking or unlocking of specific degrees of freedom (translations along the X, Y, and Z axes, and rotations around these axes, for a total of six DOF).

External Camera
For basic tracking of position and orientation, the ZED2 camera from Stereolabs [31] is used as the external camera. This device features a dual-lens system capable of capturing depth information. Additionally, it includes an IMU to determine the camera's orientation in space. Integration with the system is achieved using the ROS 2 node developed by Stereolabs [32]. Video capture is configured at 30 FPS with a 2K resolution.

Internal Camera
For the internal camera, mounted at the end effector, the ZED Mini from Stereolabs is used [33]. This device also features a dual-lens setup and includes an IMU. The same ROS 2 node developed by Stereolabs is used to interface with this device. Video capture is configured at 15 FPS with a 1080p resolution.

Force Sensor
For wrench tracking, the SenseOne force sensor (EtherCAT variant) from Botasystems is used [34]. This sensor captures force and torque data at frequencies of up to 1000 Hz. A custom ROS2 node was developed based on the standard Python script provided by Botasystems [35]. The data were filtered internally by the force sensor using a finite impulse response (FIR) filter with a window length of 512 samples, designed to remove high-frequency noise, while the additional high-speed and chopping filters were disabled. With this configuration, the effective system sampling rate is approximately 136 Hz, meaning that data are acquired and published at this frequency.

Desktop
A single computer was used to control and collect data from all sensors as well as to perform data processing. The computer was equipped with multiple ports, including at least two USB 3.0 ports and an Ethernet port. The system was configured with ROS2 and the ZED SDK (for the ZED cameras), along with Python 3 and the necessary Python packages: OpenCV for reading and analyzing ArUco markers and generating video files; NumPy, Pandas, and SciPy for data processing and calculations; and time, struct, os, json, and openpyxl for data organization and management.

3.2 Modularity assessment

To assess the modularity of the device, its ability to record data was evaluated using the three representative tasks outlined in Section 2.2. During these tests, different end-effectors were attached to the handheld device, and it was verified whether data acquisition could be successfully performed in each configuration. For the orientation- and position-based tasks, no end-effector was attached; for the force-based task, the SenseOne force sensor was attached; and for the vision-based task, a ZED Mini camera was attached as the end-effector. The objective of these tests was not to optimize performance, but to confirm that the system functioned correctly in each configuration.

In addition, sensor synchronization was evaluated through a touch experiment using the force sensor and the internal camera, with the objective of verifying whether the timestamps of the different data streams were properly aligned and whether a time-warping step would be necessary. For this experiment, the XYZ table was configured with two translational axes and all three rotational axes locked, allowing motion along only a single translational axis away from the camera. The device was then pushed against the mechanical end-stop of the table, after which the sensor data were compared with the device's positional measurements to determine the exact moment of contact. The complete setup is shown in Figure 6.

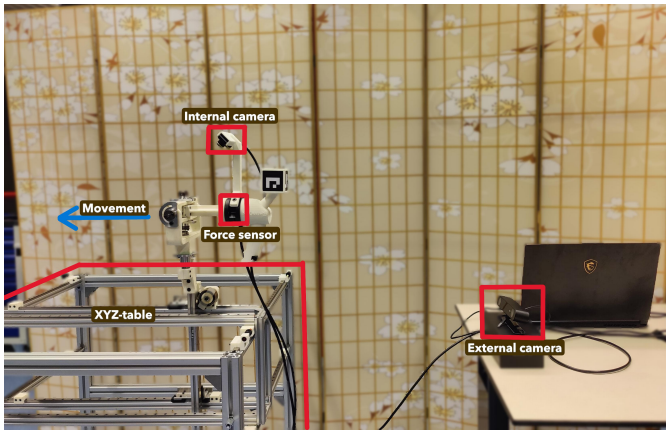


Figure (6) *Set-up for the touch experiment, highlighting in red the XYZ-table, force sensor, internal camera, and external camera, and in blue the permitted movement direction.*

The different data streams were subsequently plotted for analysis. For the force sensor, no additional processing was required, as a distinct peak clearly indicated the moment the end-stop was reached. In contrast, the camera-based position estimation required post-processing, since no static end position was observable due to measurement noise, which is approximately 1% of the measured depth [36]. It was assumed that the highest recorded position corresponds to the maximum deviation, from which the minimal position was calculated. Using this information, a mean value was derived, and, in combination with interpolation of the data, an estimated point of contact was determined. For the internal camera, no automated processing was applied; instead, each frame was manually inspected to determine the moment of contact. Finally, the identified contact times from all three sensors were compared against each other to assess sensor synchronization.

3.3 Accuracy Assessment

To evaluate the accuracy of the standalone device, a series of tests were performed. Positioning accuracy was assessed by moving the device along a square trajectory in 3D space, with the device mounted on an XYZ-table. In this setup, rotations and translation along the Z-axis (up–down movement) were locked, while only the X- and Y-axes were free to move. This allowed execution of a precise square trajectory measuring 40 cm by 40 cm, as shown in Figure 8.

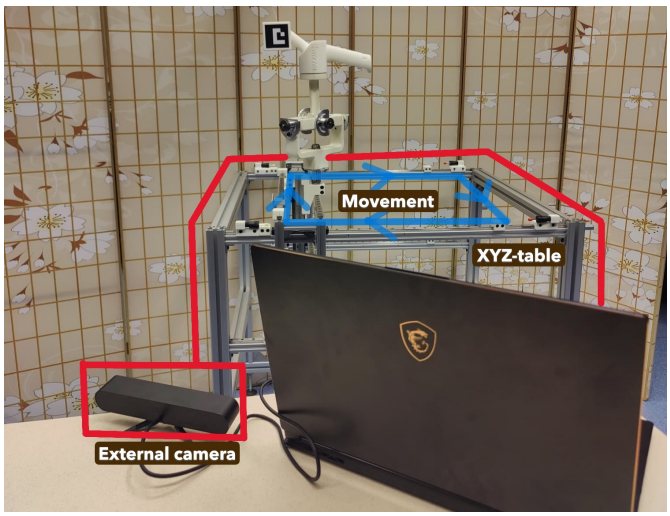


Figure (8) *Set-up for the accuracy experiment of the standalone device, comparing the pinhole camera model with the depth map. The figure highlights the XYZ-table and external camera in red, and the permitted movement directions in blue.*

Depth estimation obtained with the OpenCV ArUco toolbox was compared against the depth map generated directly by the camera. Accuracy was evaluated at two distances (0.5 m and 1.0 m) and at three cube orientations (0°, 22.5°, and 45° relative to the camera). Each trajectory was repeated twice to verify measurement consistency.

Positional accuracy was quantified by calculating the distance between each recorded point and the known ground truth of the cube. Since the XYZ-table was used, no user-induced demonstration errors were expected.

Orientation accuracy was evaluated by mounting an IMU on the device and using it as the ground truth. Recordings were taken over time with varying orientations, and the angular difference between the detected orientation of the ArUco cube and the IMU measurements was analyzed.

3.4 Questionnaire

To investigate the validation criteria associated with the use of the device, particularly ergonomics and ease of use, a questionnaire was developed and combined with a set of simple demonstration tasks. Two tasks were selected: one requiring fine and precise motion control, and another involving a broader, less precise objective.

Before the first task, participants were asked to assemble the complete setup (**V10**) while being timed. The starting point included an already-booted computer and a camera placed nearby.

Participants were then instructed to write their name on a whiteboard using a pre-attached marker as the end effector (a fine and precise task). After completing this task, participants removed the end effector and attached the device to the XYZ-table; this process was timed to assess **V11**. A second demonstration was subsequently performed using the XYZ-table (a broad-objective task). Finally, participants completed a short questionnaire based on a five-point Likert scale (provided in Appendix C), which included questions related to comfort and ease of use (**V5, V9, V11,** and **V12**).

Data collected from the timed tasks and questionnaire responses were used to evaluate the device's ergonomics, usability, and setup efficiency. Quantitative data, such as task completion times (**V10, V11**), were analyzed using descriptive statistics to identify performance trends and potential improvements. Qualitative data from the questionnaire (**V5, V9, V11, V12**) were summarized to capture participant feedback regarding comfort, ease of use, and overall user experience.

3.4.1 Participants

A total of five adult volunteers participated in the validation tasks. Participants had varying hand sizes, allowing the device's ergonomics to be evaluated across different user anatomies. All participants were capable of safely operating handheld tools and provided verbal consent prior to the study. No prior experience with robotics or LfD was required, as participation focused solely on assessing ergonomics and ease of use.

4 Results

The following subsections present the outcomes of the evaluation, assessing the device in terms of modularity, accuracy, and compliance with predefined functions and requirements.

4.1 Modularity and Sensor Synchronization

The device successfully gathered data for the three predefined representative tasks and demonstrated the ability to switch between different end-effectors. Figure 7 illustrates the handheld device equipped with various end-effectors corresponding to the different tasks, along with additional examples highlighting the modularity of the design.

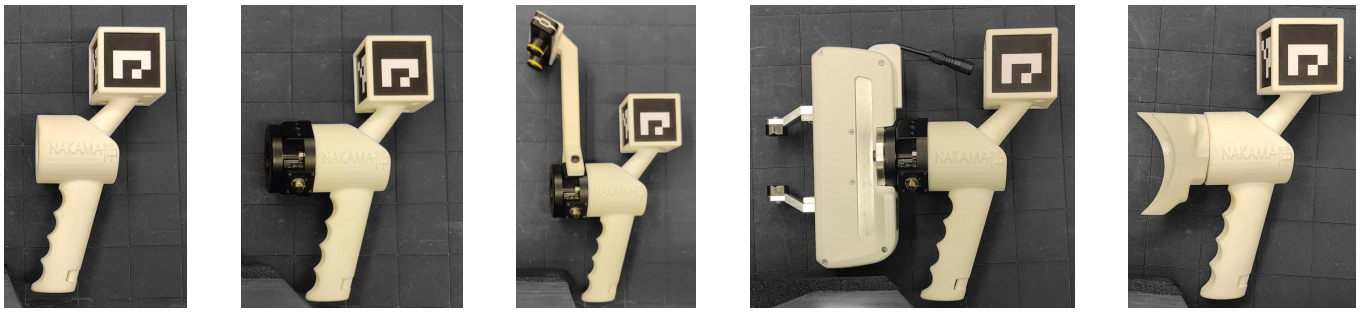
The results of the touch experiment are shown in Figure 14 (Appendix F). For the force sensor, a clear peak indicating the moment of contact is observed at approximately 5.04 seconds. For the external camera (X-position), the contact point is identified between 5.00 and 5.22 seconds, corresponding to the error margins of the ZED camera. By interpolating the position data, the estimated point of contact is 5.09 seconds, closely matching the force sensor measurement. For the internal camera, comparison of images at 5.078 and 5.143 seconds shows visible differences, indicating that the device is still in motion. Images at 5.143 and 5.211 seconds appear identical, suggesting that the device has reached the end of the track. Overall, these observations indicate that the interaction occurs between 5.078 and 5.143 seconds, aligning with the position data but slightly lagging behind the peak detected by the force sensor.

4.2 Accuracy

The ArUco (pinhole camera)-based approach, illustrated in Figure 9, exhibited substantial noise and distortion, preventing the formation of a clearly identifiable square. Consequently, this method was deemed unsuitable for LfD applications. All subsequent validation therefore relied exclusively on depth estimations obtained from the camera's depth map, where the square marker was clearly visible.

Figure 10 summarizes the experimental results for three cube orientations ($0°$, $22.5°$, and $45°$) at two distances from the camera (0.5 m and 1.0 m). For completeness, Figure 12 in Appendix D provides a visual distribution of the recorded data points. Although the designed hybrid system allows erroneous points to be removed, no filtering was applied during validation.

The mean error across all recordings was approximately 0.5 cm. Overall, the error increased with distance from the camera. Both the number and accuracy of recorded points were strongly influenced by the cube's orientation and its distance from the external camera. The most consistent square pattern (lowest absolute error) was obtained at $0°$ and 0.5 m, although results at 1 m were generally more accurate, particularly at the bottom-right corner of the square. Point density varied considerably between measurements despite identical conditions on the XYZ-table.



(a) **(b)** **(c)** **(d)** **(e)**

Figure (7) *The designed handheld device with different end-effectors: (a) standalone device without an end-effector, used for the position- and orientation-based task; (b) device with a force-sensor end-effector, used for the force-based task; (c) device with a gripper-mounted camera end-effector, used for the vision-based task; (d) and (e) examples of additional end-effectors demonstrating modularity.*

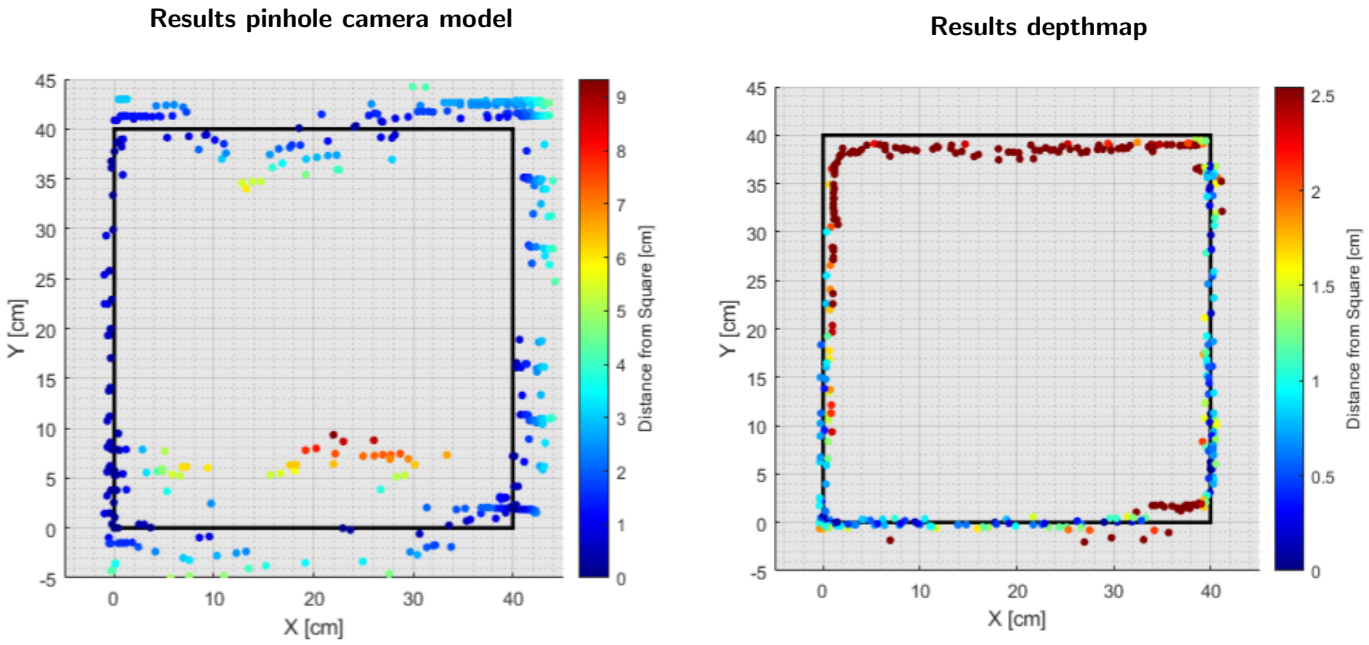


Figure (9) *Results of the square experiment. The left figure shows the results using the pinhole camera model while the right figure uses the depth map. The camera origin is positioned along the negative Y axis at a distance of 1 meter. Data points are colored according to their maximal deviation from the ideal square trajectory, which is plotted in black.*

The main loss of points occurred during X-axis (sideways) movements, where video inspection revealed that the ArUco cube remained visible but became distorted under faster motion, preventing reliable detection by the recognition software. As expected, the largest deviations were observed at the furthest distance, consistent with the manufacturer's reported error margin of approximately 1% of the measured distance [36].

The angular misalignment between the IMU and the cube in the world frame is shown in Figure 11 as the magnitude of the rotation-vector error. The overall root mean square (RMS) error is 36.3°, indicating a substantial misalignment. Contributions along the X, Y, and Z axes are detailed in Appendix E. The largest source of error occurs along the Z-axis, with an RMS of 31.2°, followed by the X-axis (15.2°) and Y-axis (11.5°).

During the experiment, multiple movements were performed about all axes, resulting in the Z-axis pointing in various directions. Consequently, it is unclear which specific factors contribute most to this angular discrepancy.

4.3 Questionnaire

The results of the questionnaire are summarized in Table 3. Overall, the device scored high in terms of ease of use and comfort. Minor discomfort was reported by 1–2 participants, primarily related to the specific end-effector rather than the device itself. One participant noted that the grip was uncomfortable during the writing task. Overall, demonstrations could be performed without major discomfort (**V9**).

The average setup time of the device was 17 seconds (range: 8–27 seconds), well within the 2-minute target (**V10**). Changing the end-effector proved more challenging due to its complexity, making tightening and loosening slower.

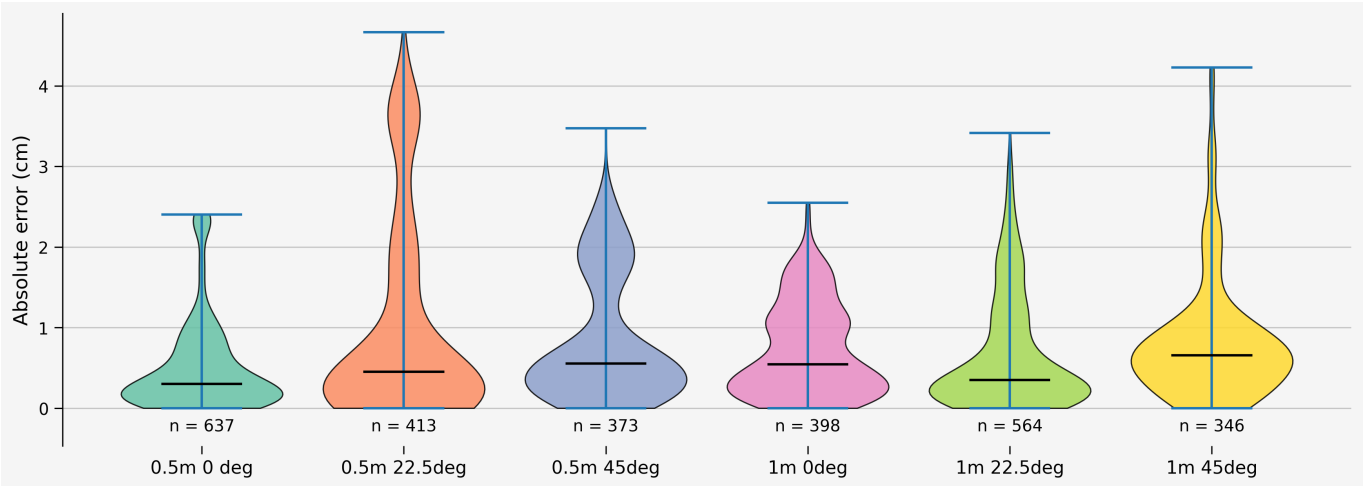


Figure (10) *Violin plot illustrating the distribution of errors in the square experiment across varying cube orientations and distances from the camera. The width of each violin represents the density of data points corresponding to each error value, highlighting regions with higher or lower frequency of occurrences*

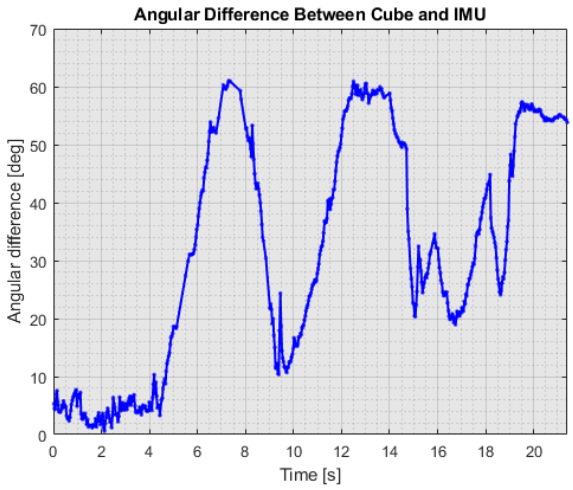


Figure (11) *Total angular difference between the IMU and the cube in the world frame over time. The curve represents the magnitude of the rotation-vector error (degrees), showing the overall orientation misalignment.*

The average time for this task was 3 minutes and 15 seconds (range: 2–6.5 minutes), exceeding the 2-minute target (**V11**). Overall, participants were able to intuitively remove and attach end-effectors, but fastening required additional time. For the majority of participants, the visual indicators (print statements in the terminal) were clear and easy to understand (**V12**).

4.4 Remaining Validation

The remaining requirements concerning ergonomics, ease of use, portability, and cable management were also evaluated.

The handheld structure includes an integrated routing option with a tie-wrap to secure cables and prevent strain (**V4**). When trimmed, the tie-wrap did not interfere with task execution (**V5**). During inspection, it was noted that some rigid cables, such as the EtherCAT cable of the SenseOne force sensor, did not fit optimally through the provided opening. This limitation can be addressed by slightly widening the routing channel. For other sensors, the current setup was sufficient, and the cables could be securely fastened using tie wraps.

The handheld device has a total weight of approximately 250 grams (less than 0.5 kg; **V8**), allowing it to be comfortably operated with one hand for extended periods.

Regarding portability, the device, together with the external camera and a laptop, fits into a 25-liter backpack (**V13**), leaving sufficient space for additional end-effectors and accessories. This makes the system suitable for mobile use across different laboratory and field environments.

Table (3) *Questionnaire Responses for Ergonomics and Ease of Use Evaluation (5 Participants)*

Statement	P1	P2	P3	P4	P5	Comments
The device felt comfortable to hold during the demonstration.	5	5	4	5	5	Grip slightly uncomfortable during writing task (P3)
The weight and balance of the device felt manageable throughout use.	5	5	5	5	5	None
I could set up the device quickly and easily.	4	5	4	5	5	Setup very quick for all participants
Setting up the device felt intuitive and straightforward.	4	5	4	5	5	Minor guidance needed for first step
I was able to change the end-effector independently without difficulty.	4	5	4	5	5	Fastening slower for some participants
The process of changing components was intuitive.	4	5	4	5	5	Minor difficulties with complex end-effector
The visual indicators for recording start/stop were clear and easy to understand.	3	4	5	5	5	None
Overall, I found the device easy and comfortable to use.	4	5	5	5	5	Positive overall experience
Open-ended question: Suggestions for improvements?	An improved ergonomic feel would be nice, as well as an adjustable handle angle for certain tasks(P2).					

Scale: 1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree

5 Discussion

The primary goal of this paper was to design and validate a modular handheld data acquisition device capable of supporting multiple LfD pipelines through the use of interchangeable end-effectors. The developed prototype demonstrates the ability to integrate multiple sensing modalities, such as force, vision, and motion tracking, within a single device. Validation experiments confirmed the system's viability, showing sufficient measurement accuracy, synchronization, and ergonomic usability for simple demonstration tasks. Furthermore, by incorporating a standardized robotic flange and ROS 2 compatibility, the device ensures seamless integration with robotic interfaces, thereby facilitating future research and development in LfD and human–robot interaction.

The validation experiments provided valuable insights into the device's performance and confirmed its ability to support data acquisition for a variety of LfD tasks. During the position validation, some recordings were repeated, and faulty trials were excluded, most often due to the cube being detected only partway through a recording. Despite these exclusions, the square trajectory provided a clear and well-defined baseline for assessing positional accuracy. Future work could extend this validation to more complex, fluid trajectories that better reflect real-world applications.

Another important aspect of performance concerns how the device captures motion over time. In this study, individual data points were compared to the square path without considering their temporal sequence. As a result, dynamic characteristics such as lag or overshoot were not fully assessed. While this simplification ensured clarity in the baseline evaluation, incorporating temporal dynamics in future studies would allow a more comprehensive assessment of motion accuracy. Additionally, while faulty points could have been manually removed using the visualization tool, this option was deliberately avoided to provide insight into the device's baseline accuracy under realistic, unprocessed conditions.

The orientation validation also yielded valuable insights. Although the observed orientation error was relatively high, the device remains suitable for tasks in which precise orientation tracking is not critical. For applications requiring higher orientation accuracy, an inertial measurement unit (IMU) could be integrated into the end-effector or included in future design iterations to enhance rotational precision. Improved orientation tracking would also contribute to more accurate position estimation, as the device's orientation is used to determine the cube's center point. A dedicated follow-up study could investigate the influence of orientation accuracy on overall positional performance.

In addition to position and orientation accuracy, synchronization across sensors is critical for reliable data acquisition. This was evaluated through the touch experiment. Inspection of the external camera data (used for ArUco detection) revealed slight deformation of the XYZ-table at the moment of contact, preventing the captured images from appearing fully static and potentially influencing the synchronization results. Nevertheless, measured synchronization was well within 100 ms in the worst case, which is generally sufficient for LfD tasks.

Further experiments are recommended to determine a more precise measurement of the synchronization delay, and future research could investigate the impact of this temporal offset on data quality and task performance, particularly for higher-speed or precision-critical demonstrations.

Placing these findings in context, direct comparisons to other LfD devices are not straightforward. Most studies present complete LfD pipelines, including task reproduction phases where accuracy is validated, typically based on task completion rather than continuous accuracy over time. Since many of these systems also rely on ArUco markers, comparable positional accuracy is expected for the present device [17, 19]. Importantly, the handheld device successfully acquired data for a wide range of LfD tasks, demonstrating its modularity and adaptability. While the device currently lacks buttons or controls to actuate end-effectors such as grippers, its modular design allows straightforward extension of functionality in future versions.

Building on these strengths, several technical improvements could further enhance system robustness and reliability. One recurring challenge in ArUco cube tracking is pose ambiguity, in which the Z-axis may invert, effectively switching from a right- to a left-handed coordinate system. Future research could investigate alternative fiducial systems, such as AprilTags [37], or explore more complex 3D marker structures that guarantee a unique solution. Related work [25] has demonstrated the potential of such approaches, while highlighting the challenges of reliably detecting all marker points. Addressing this limitation remains an open and promising research problem and could ultimately eliminate the need for manual intervention via the interactive display.

Optimizing marker detection itself represents another avenue to improve accuracy and robustness. A trade-off exists between recognition accuracy and practical usability: larger markers improve reliability but reduce handheld usability. Systematic studies on marker size, camera resolution, and frame rate will be key to balancing these factors. Additionally, environmental illumination strongly affects marker detection, suggesting that future work should analyze performance under varied lighting conditions.

While these improvements would enhance data acquisition translating captured trajectories onto robots for task reproduction represents an equally important challenge. Mapping between the handheld device and robot coordinate systems must account for workspace limitations and kinematic constraints. These mappings are robot-specific, as different robots operate with varying maximum velocities and workspace limits. Common approaches include scaling and normalization of trajectories to fit within the robot's reachable workspace. Feature-based extraction is a promising strategy, where key aspects of the recorded motion,such as waypoints, contact points, or orientation cues, are identified and adapted according to the parameters of the target robot.

Additionally, inverse kinematics must be solved, as most robots have multiple degrees of freedom, resulting in multiple valid joint configurations that achieve the same end-effector pose. The goal is to achieve smooth, collision-free motions while respecting joint limits. Trajectory smoothing combined with optimization can further refine motion paths.

For instance, the Franka Emika Panda requires 1000 Hz input [27], which exceeds the current capture rate of the device. Simple interpolation of recorded points can bridge this gap, but these steps highlight that additional processing is necessary before practical implementation. Addressing these considerations represents a crucial step toward integrating the device into complete LfD frameworks and enabling its use across a broader range of tasks and robotic platforms.

Taken together, these results provide a foundation for both technical refinement and broader integration into LfD frameworks. Overall, the handheld device demonstrated its feasibility as a modular, ROS2-based platform for LfD data acquisition. While several technical challenges remain, the results confirm the device's potential and highlight clear directions for refinement, making it a promising tool for both research and applied settings.

6 Conclusion

This paper presented a modular handheld data acquisition device for LfD applications, designed to overcome the limitations of task-specific devices. By leveraging modularity, a standardized robotic flange, and ROS2 integration, the system supports a wide range of LfD tasks and pipelines. Validation experiments demonstrated the device's ability to collect data for different kinds of LfD tasks, highlighting its flexibility and applicability in structured settings.

Although certain tasks, such as pick-and-place operations requiring gripper actuation, remain challenging, the platform provides a robust and versatile foundation for advancing research in LfD and human–robot interaction. Future work will focus on refining the design, extending its sensing and end-effector capabilities, and enabling direct mapping of demonstrated trajectories onto robotic platforms to support deployment in more complex, dynamic, and unstructured scenarios.

References

[1] David E. Bloom et al. *Population Aging: Facts, Challenges, and Responses*. Tech. rep. PGDA Working Paper No. 71. Program on the Global Demography of Aging, 2011. URL: https://www.hsph.harvard.edu/pgda/WorkingPapers/2011/PGDA_WP_71.pdf.

[2] Ali Ahmad Malik, Tariq Masood, and Alexander Brem. "Intelligent Humanoids in Manufacturing to Address Worker Shortage and Skill Gaps: Case of Tesla Optimus". In: *arXiv preprint arXiv:2304.04949* (2023). URL: https://arxiv.org/abs/2304.04949.

[3] Karen Eggleston, Yong Suk Lee, and Toshiaki Iizuka. *Robots and Labor in the Service Sector: Evidence from Nursing Homes*. Tech. rep. 28322. National Bureau of Economic Research, 2021. DOI: 10.3386/w28322. URL: https://www.nber.org/papers/w28322.

[4] Muhammad Ali Musarat. "Substitution of Workforce with Robotics in the Construction Industry: A Wise or Witless Approach". In: *Automation in Construction* 156 (2024), p. 104214. DOI: 10.1016/j.autcon.2024.104214. URL: https://www.sciencedirect.com/science/article/pii/S2199853124002142.

[5] BCC Research. *Global Robotics Market Size, Share and Growth Analysis Report*. Accessed: 2025-04-04. 2024. URL: https://www.bccresearch.com/market-research/engineering/robotics.html.

[6] Rae Yule Kim. "Anthropomorphism and Human-Robot Interaction". In: *Communications of the ACM* 67.2 (Jan. 2024), pp. 80–85. ISSN: 0001-0782. DOI: 10.1145/3624716. URL: https://doi.org/10.1145/3624716.

[7] Lea Bolz et al. *Growth Dynamics in Industrial Robotics*. Tech. rep. McKinsey & Company, July 2019. URL: https://www.mckinsey.com/industries/industrials-and-electronics/our-insights/growth-dynamics-in-industrial-robotics.

[8] Grand View Research. *Medical Service Robots Market Size, Share & Trends Analysis Report By Product and Region, 2025–2030*. Tech. rep. Grand View Research, Apr. 2024. URL: https://www.grandviewresearch.com/industry-analysis/medical-service-robots-market-report.

[9] Harish Ravichandar et al. "Recent Advances in Robot Learning from Demonstration". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 297–330. ISSN: 2573-5144. DOI: 10.1146/annurev-control-100819-063206. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-100819-063206.

[10] Arturo Daniel Sosa-Ceron, Hugo Gustavo Gonzalez-Hernandez, and Jorge Antonio Reyes-Avendaño. "Learning from Demonstrations in Human–Robot Collaborative Scenarios: A Survey". In: *Robotics* 11.6 (2022), p. 126. ISSN: 2218-6581. DOI: 10.3390/robotics11060126. URL: https://www.mdpi.com/2218-6581/11/6/126.

[11] Brenna D. Argall et al. "A Survey of Robot Learning from Demonstration". In: *Robotics and Autonomous Systems* 57.5 (2009), pp. 469–483. DOI: 10.1016/j.robot.2008.10.024.

[12] Ahmed Hussein et al. "Imitation Learning: A Survey of Learning Methods". In: *ACM Computing Surveys* 50.2 (Apr. 2017), 21:1–21:35. ISSN: 0360-0300. DOI: 10.1145/3054912. URL: https://doi.org/10.1145/3054912.

[13] Sylvain Calinon. "Mixture Models for the Analysis, Edition, and Synthesis of Continuous Time Series". In: *Mixture Models and Applications*. Ed. by Nizar Bouguila and Wentao Fan. Cham: Springer International Publishing, 2020, pp. 39–57. ISBN: 978-3-030-23876-6. DOI: 10.1007/978-3-030-23876-6_3. URL: https://doi.org/10.1007/978-3-030-23876-6_3.

[14] Cheng Chi et al. *Diffusion Policy: Visuomotor Policy Learning via Action Diffusion*. 2024. arXiv: 2303.04137 [cs.RO]. URL: https://arxiv.org/abs/2303.04137.

[15] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. 1st ed. Cambridge, UK: Cambridge University Press, 2017. ISBN: 9781107156302.

[16] John J. Craig. *Introduction to Robotics: Mechanics and Control*. 3rd ed. Chapter 2: Spatial Descriptions and Transformations — essential for understanding frame alignment in robotics. Upper Saddle River, NJ: Pearson Prentice Hall, 2005. Chap. 2. ISBN: 978-0-201-54361-2.

[17] Hengtai Dai et al. "A Gripper-like Exoskeleton Design for Robot Grasping Demonstration". In: *Actuators* 12.1 (2023), p. 39. ISSN: 2076-0825. DOI: 10.3390/act12010039. URL: https://www.mdpi.com/2076-0825/12/1/39.

[18] Cheng Chi et al. "Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots". In: *Proceedings of Robotics: Science and Systems (RSS)*. 2024.

[19] Hsien-Chung Lin et al. "Robot Learning from Human Demonstration with Remote Lead Through Teaching". In: *2016 European Control Conference (ECC)*. 2016, pp. 388–394. DOI: 10.1109/ECC.2016.7810316.

[20] Nakama Robotics Lab. *Nakama Robotics Lab*. https://www.utwente.nl/en/et/be/research/nakama_robotics_lab/. Accessed: 2025-08-23. 2025.

[21] Olarn Wongwirat and Chutchai Chaiyarat. "A Position Tracking Experiment of Mobile Robot with Inertial Measurement Unit (IMU)". In: *Proceedings of the International Conference on Control, Automation and Systems (ICCAS)*. 2010, pp. 304–308. DOI: 10.1109/ICCAS.2010.5670227.

[22] Darin Tsui et al. "An Optical Tracking Approach to Computer-Assisted Surgical Navigation via Stereoscopic Vision". In: *ASME 2023 32nd Conference on Information Storage and Processing Systems*. Aug. 2023. DOI: 10.1115/ISPS2023-111020.

[23] Arie Bagus Hendra Wicaksana, Ronny Mardiyanto, and Astria Nur Irfansyah. "Drone Position Tracking System Based on Object Detection and ARUCO Marker for Autonomous Navigation Applications". In: *2024 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. July 2024, pp. 131–135. DOI: 10.1109/ISITIA63062.2024.10668046.

[24] Renpei Wang et al. "The Effect of ArUco Marker Size, Number, and Distribution on the Localization Performance of Fixed-Point Targets". In: *2023 6th International Conference on Robotics, Control and Automation Engineering (RCAE)*. 2023, pp. 118–123. DOI: 10.1109/RCAE59706.2023.10398770.

[25] Lingling Chen et al. "Trajectory-Based Alignment for Optical See-Through HMD Calibration". In: *Multimedia Tools and Applications* 83 (2024), pp. 1–26. DOI: 10.1007/s11042-024-18252-6.

[26] *Industrial automation systems and integration - Mechanical interfaces for industrial end-effectors - Part 1: Plain reduction modules*. Geneva, Switzerland, 2000. URL: https://www.iso.org/standard/36578.html.

[27] Franka Robotics. *Franka Research 3 Product Manual*. Accessed: 2025-02-11. Sept. 2024. URL: https://franka.de/documents.

[28] KUKA. *LBR iisy Cobot Product Page*. 2025. URL: https://www.kuka.com/en-be/products/robotics-systems/industrial-robots/lbr-iisy-cobot.

[29] Jia-Hua Lin, Raymond W. McGorry, and Chien-Chi Chang. "Effects of handle orientation and between-handle distance on bi-manual isometric push strength". In: *Applied Ergonomics* 43.4 (2012), pp. 664–670. ISSN: 0003-6870. DOI: https://doi.org/10.1016/j.apergo.2011.10.004. URL: https://www.sciencedirect.com/science/article/pii/S0003687011001645.

[30] OpenCV contributors. *ArUco Documentation*. Section on pose estimation ambiguity with 4 coplanar points. OpenCV. 2025.

[31] Stereolabs. *ZED2: Stereo AI Camera*. 2025. URL: https://www.stereolabs.com/products/zed-2.

[32] Stereolabs. *zed-ros2-wrapper: ZED ROS2 Wrapper for Stereolabs Cameras*. https://github.com/stereolabs/zed-ros2-wrapper. Accessed: 2025-06-08. 2025.

[33] StereoLabs. *ZED Mini*. Accessed: 2025-02-11. 2025. URL: https://www.stereolabs.com/en-nl/store/products/zed-mini.

[34] Bota Systems. *SensOne Force and Torque Sensor*. Accessed: 2025-02-11. 2025. URL: https://www.botasys.com/force-torque-sensors/sensone.

[35] Botasense. *Python3 EtherCAT Script for Industrial Communication*. https://github.com/orgs/BOTAsys/repositories. Accessed: 2025-04-06. 2025.

[36] Stereolabs. *Depth Sensing*. https://www.stereolabs.com/docs/depth-sensing. Accessed: 2025-07-15.

[37] John Wang and Edwin Olson. "AprilTag 2: Efficient and Robust Fiducial Detection". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.

Appendices

A declaration of AI

During the preparation of this work, I utilized AI-based tools, including ChatGPT, Deepseek, and Qwen, for spell checking and enhancing academic tone. Additionally, I used Autopilot for code generation. After employing these tools, I thoroughly reviewed and edited all content as necessary, taking full responsibility for the final outcome.

B Github repository

The code for this project, as well as the Solidworks file are available here: Handheld_lfd on GitHub.

C Questionnaire

Table (4) *Questionnaire for Ergonomics and Ease of Use Evaluation*

Statement	**1**	**2**	**3**	**4**	**5**	**Comments**
The device felt comfortable to hold during the demonstration.						
The weight and balance of the device felt manageable throughout use.						
I could set up the device quickly and easily.						
Setting up the device felt intuitive and straightforward.						
I was able to change the end-effector independently without difficulty.						
The process of changing components was intuitive.						
The visual indicators for recording start/stop were clear and easy to understand.						
Overall, I found the device easy and comfortable to use.						
Open-ended question: Do you have any suggestions for improvements for the device?						

Scale: 1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree

D Figure of results of square experiment

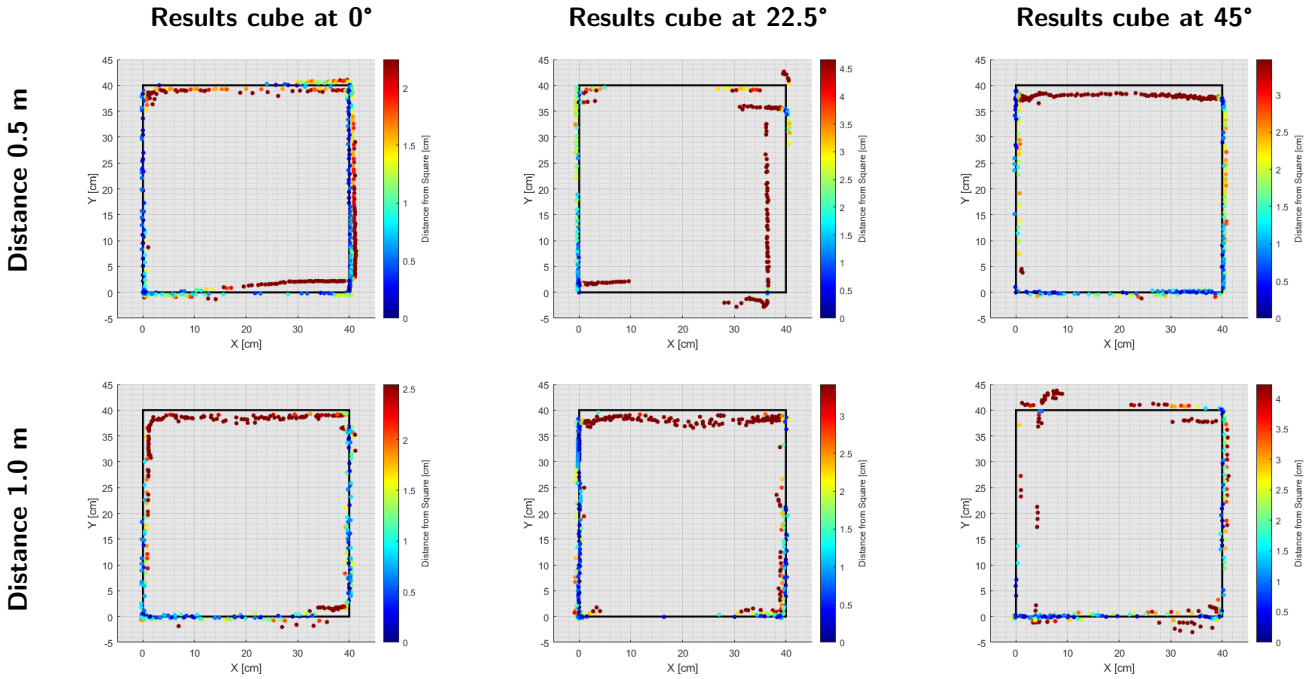


Figure (12) *Visual representation of the square experiment. Each row corresponds to a different distance, and each column corresponds to a different orientation of the ArUco cube relative to the camera. The camera's origin is placed along the negative Y-axis at the corresponding distance. Data points are colored according to their deviation from the ideal square trajectory plotted in black.*

E Results orientation experiment

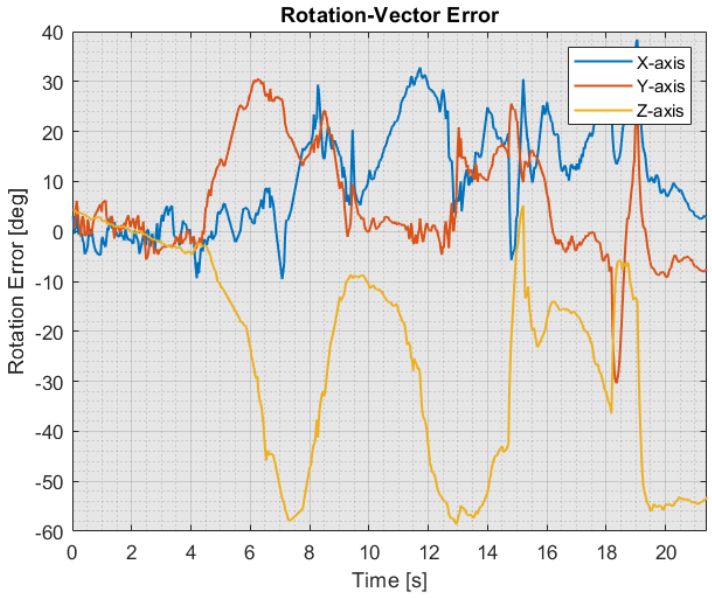


Figure (13) *Orientation errors of the handheld device along the X, Y, and Z axes during the validation experiments. Each axis shows the rotation-vector error compared to the IMU referencel.*

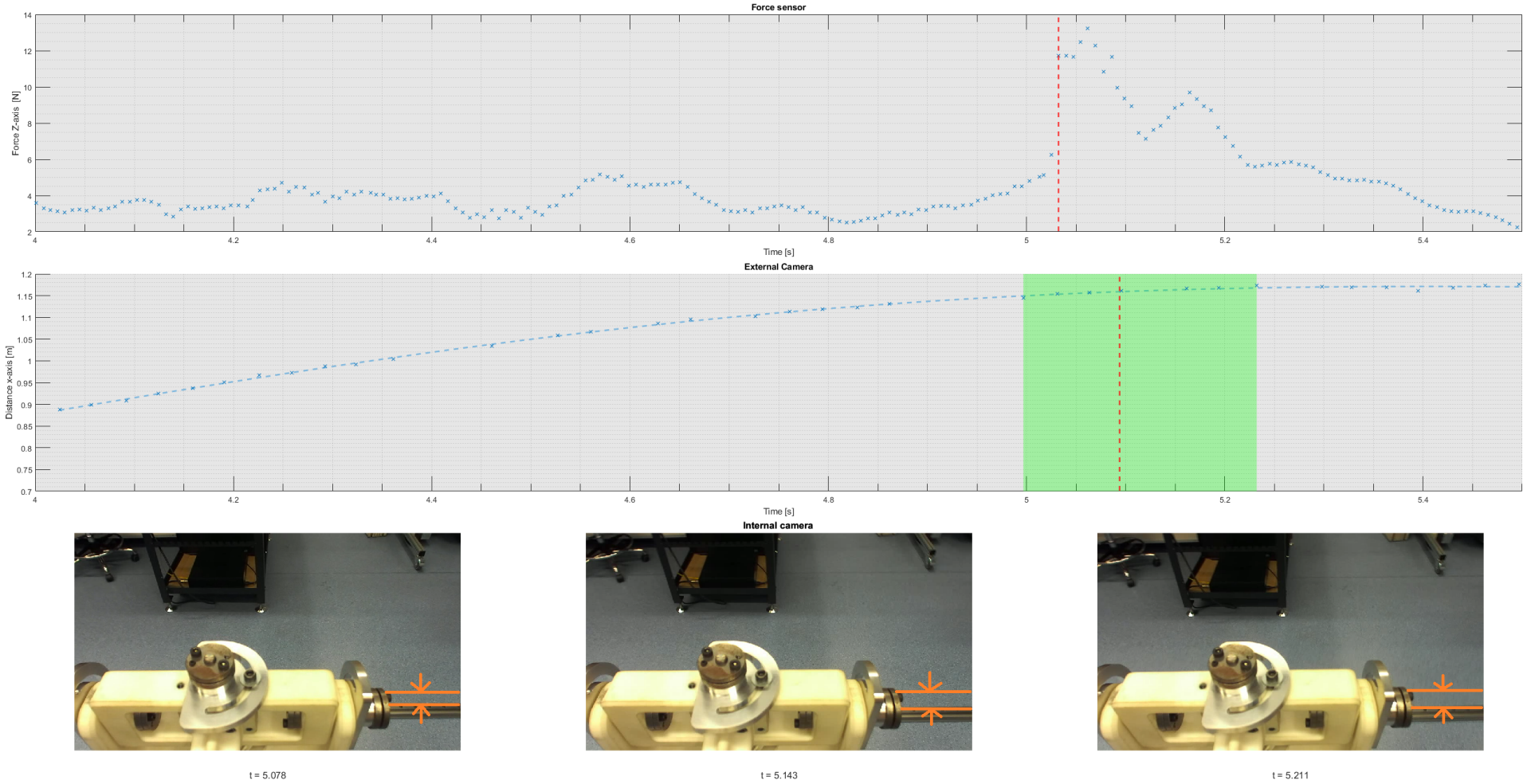


Figure (14) *Results of the touch experiment, with each sensor presented in a separate row. The first row shows the force sensor data, with a red dotted line indicating the moment of contact. The second row displays the external camera data points, including a dashed interpolated line, a green shaded region representing the estimated contact range (from minimum to maximum contact distance), and a red dotted line indicating the estimated point of contact. The third row presents images from the internal camera, oriented toward the end effector, with orange lines highlighting regions where differences between consecutive frames are more easily observed. Corresponding timestamps are displayed below each image.*