

## Article

# Human–Robot Interaction Using Learning from Demonstrations and a Wearable Glove with Multiple Sensors

Rajmeet Singh <sup>1</sup>, Saeed Mozaffari <sup>1</sup>, Masoud Akhshik <sup>1</sup>, Mohammed Jalal Ahamed <sup>1</sup>, Simon Rondeau-Gagné <sup>2</sup> and Shahpour Alirezaee <sup>1,\*</sup>

<sup>1</sup> Mechanical, Automotive, and Material Engineering Department, University of Windsor, Windsor, ON N9B 3P4, Canada; rsbhourji@uwindsor.ca (R.S.); saeed.mozaffari@uwindsor.ca (S.M.); akhshik@uwindsor.ca (M.A.); m.ahamed@uwindsor.ca (M.J.A.)

<sup>2</sup> Department of Chemistry and Biochemistry, University of Windsor, Windsor, ON N9B 3P4, Canada; simon.rondeau-gagne@uwindsor.ca

\* Correspondence: s.alirezaee@uwindsor.ca

**Abstract:** Human–robot interaction is of the utmost importance as it enables seamless collaboration and communication between humans and robots, leading to enhanced productivity and efficiency. It involves gathering data from humans, transmitting the data to a robot for execution, and providing feedback to the human. To perform complex tasks, such as robotic grasping and manipulation, which require both human intelligence and robotic capabilities, effective interaction modes are required. To address this issue, we use a wearable glove to collect relevant data from a human demonstrator for improved human–robot interaction. Accelerometer, pressure, and flexi sensors were embedded in the wearable glove to measure motion and force information for handling objects of different sizes, materials, and conditions. A machine learning algorithm is proposed to recognize grasp orientation and position, based on the multi-sensor fusion method.

**Keywords:** robotic grasping; human–robot interaction; inertia; pressure; flexi sensors; wearable devices; learning from demonstration



**Citation:** Singh, R.; Mozaffari, S.; Akhshik, M.; Ahamed, M.J.; Rondeau-Gagné, S.; Alirezaee, S. Human–Robot Interaction Using Learning from Demonstrations and a Wearable Glove with Multiple Sensors. *Sensors* **2023**, *23*, 9780. <https://doi.org/10.3390/s23249780>

Academic Editors: Daniel Galan, Ramon A. Suarez Fernandez and Francisco Javier Badesa

Received: 13 November 2023

Revised: 7 December 2023

Accepted: 8 December 2023

Published: 12 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Industrial robots have been used in a variety of applications, ranging from manufacturing to healthcare settings. These highly capable machines are designed to perform repetitive tasks with precision and efficiency, enhancing productivity and reducing human labor in industries. While traditionally programmed for specific tasks, the evolving needs of modern industries have led to the demand for collaborative robots (cobots) that can work alongside humans, thus emphasizing the importance of human–robot interaction (HRI) [1]. Collaborative robots are designed to safely and effectively collaborate with human operators, enhancing productivity and flexibility [2]. HRI applications have gained significant attention across various domains. In manufacturing and industrial settings, robots collaborate with humans, assisting in tasks, such as assembly, material handling, and quality control, improving productivity and worker safety [3]. Additionally, robots are employed in healthcare centers, for patient care, rehabilitation, and surgical assistance, enhancing medical procedures and reducing the physical strain on healthcare professionals [4]. In the service industry, robots function as receptionists, guides, or companions, delivering personalized assistance and improving customer experiences [5]. HRI also finds applications in education, entertainment, and social interactions, where robots serve as tutors, performers, or companions, fostering engagement and enriching human experiences [6].

Learning from demonstration (LfD) is a special case of HRI, which enables robots to learn new skills and behaviors through human guidance, enhancing the capabilities of industrial robots, enabling them to interact with humans and optimizing their performance [7]. The rapid development of sensor technologies and integration of artificial intelligence and

machine learning techniques have opened up new possibilities in LfD [8]. Demonstration techniques can be divided into the following main groups [9]: kinesthetic teaching, teleoperation demonstration, learning from observations, and sensor-based methods.

The kinesthetic teaching method involves physical guidance, where a human physically moves the robot's limbs or manipulates its body to demonstrate the desired behavior. The robot learns by recording and reproducing the demonstrated movements. Since the human teacher directly guides the robot, the kinematic boundaries of the robot, including joint limits and workspace, are considered during the demonstration. In other words, the real-time feedback provided by the human teacher during kinesthetic teaching allows for immediate corrections and refinements to the robot's movements, facilitating faster learning and performance improvements. However, kinesthetic teaching requires a significant dependency on human guidance, which makes it time-consuming and labor-intensive. The effect of the human teacher's proficiency on their demonstration data and kinesthetic teaching was investigated in [10]. Scaling up kinesthetic teaching to complex or multi-step tasks can also become challenging, as the complexity increases, and it becomes more difficult for a novice human teacher to provide precise physical guidance.

In teleoperation demonstrations, robots are controlled remotely by the human operator via a control interface, such as a joystick. The operator's inputs are transmitted to the robot in real-time, enabling the robot to mimic the operator's movements and actions. Unlike kinesthetic teaching, in which the human teacher moves individual segments by hand, teleoperation demonstration requires an input device for robot movement. The impact of the input device on the performance of teleoperation was studied in [11]. On the negative side, teleoperation may not be suitable for tasks that require high levels of autonomy or decision making, as the robot relies heavily on the operator's guidance. It is shown that teleoperation is slower and more challenging in comparison with kinesthetic teaching [11]. An overview of teleoperation techniques with a focus on skill learning and their application to complex manipulation tasks is presented in [12]. Key technologies, such as manipulation skill learning, multimodal interfacing for teleoperation, and telerobotic control, are discussed.

Unlike other learning methods that rely on explicit demonstrations or interactions, learning from observations leverages existing data to extract patterns, make inferences, and acquire knowledge. In this approach, the robot typically receives a set of observations in the form of images or videos. Then, machine learning algorithms are employed to analyze the data and extract meaningful patterns or relationships. In [8], an LfD system based on computer vision was developed to observe human actions and deep learning to perceive the demonstrator's actions and manipulated objects. Deep neural networks were utilized for object detection, object classification, pose estimation, and action recognition. The proposed vision-based LfD method was tested on an industrial robot, FANUC LR Mate 200iC, to pick and place objects with different shapes. The experimental results showed 95% accuracy. The demonstrator's intention was recognized by analyzing 3D human skeletons captured by RGB-D videos [13]. It was shown that modeling the conditional probability of interaction between humans and robots in different environments can lead to faster and more accurate HRI. However, the presence of occlusion poses several problems for vision-based HRI systems. It can lead to incomplete or ambiguous visual information, making it challenging to accurately track and recognize objects or individuals.

Although the above demonstration techniques have greatly facilitated the interaction between humans and robots, they cannot completely meet the demands of HRI in dynamic and complex environments. Recently, wearable sensing technology has been applied in HRI. Wearable sensing technology is an example of sensor-based methods, which rely on various kinds of sensors to perceive users' status [14]. Wearable devices are usually equipped with inertial, bending, electromyography (EMG), and tactile sensors to capture position, orientation, motion, and pressure. A data glove is a type of wearable device commonly utilized in HRI to capture and transmit real-time action postures of the human hand to the robot. The most employed approach in detecting finger bending involves the integration of

flex sensors that are attached to the finger joint positions on the hand [15]. A wearable glove system based on 12 nine-axis inertial and magnetic unit (IMMU) sensors was proposed for hand function assessment [16]. The sensory system captured the acceleration, angular velocity, and geomagnetic orientation of human hand movements. In [17], a wearable data glove was designed to sense human handover intention information based on fuzzy rules. The glove contains six six-axis IMUs, one located on the back of the hand and the rest on the second section of each finger.

This paper proposes an experimental HRI system to grab different objects with a robotic hand. Human demonstrators grasp objects with a wearable glove equipped with an accelerometer, flex sensor, and pressure sensor. The data glove provides useful information to train machine learning algorithms to mimic human grasping actions. In addition, an HRI module is evaluated by a universal robot. To evaluate the effectiveness of this sensor-based HRI system, different objects with different shapes were grasped by the robotic hand.

## 2. Methodology

In this section, we will provide a detailed description of the proposed data glove, encompassing its hardware and software components, as well as the sensors utilized. Additionally, we will delve into the calibration process of the sensors, the coordinate systems employed, and the placement of the sensors on the glove.

### 2.1. System Hardware and Software

The proposed HRI system consists of MEMS sensors, a microprocessor, a host computer, and a robotic hand. Figure 1 illustrates the communication flow between the hardware. The ADXL335 MEMS (microelectromechanical system) inertial sensor has been selected as the sensing module for capturing finger motion information. It is a compact, slim, low-power accelerometer that provides a fully integrated 3-axis accelerometer with voltage output that is conditioned for signal processing. It features  $\pm 3$  g tri-axis accelerometer readings. The second sensor is the flex sensor that is employed to detect the bending of the first finger segment. We also used SPX-14687 sensor, which is a pressure sensor working based on the principle of proximity. Table 1 shows the specifications of each sensor. The raw sensor data of multiple ADXL335, flex, and SPX-14687 are sampled at 100 Hz with I2C interface by using low-power processor RP2040. The incorporation of flexible GPIO (General Purpose Input/Output) in our system enables the connection of various digital interfaces, including SPI (Serial Peripheral Interface) Master/slave, TWI (Two-Wire Interface) Master, and UART (Universal Asynchronous Receiver–Transmitter). These GPIOs provide the flexibility to connect and communicate with external devices or modules that utilize these digital interfaces. The host computer receives the data transmitted by the processor through the serial port and subsequently performs the necessary calculations. The robot hand receives action commands to execute the specific applications.

Figure 2a illustrates a system schematic of the data glove, depicting the arrangement of sensors and the communication network. In Figure 2b, the prototype of the data glove is presented, showcasing various versions and the interface with a robotic hand. In the initial version of the data glove, flex sensors were employed to measure finger bending. To capture the bending values of the finger segments, a rigid printed circuit board (PCB) was designed. In the second version, a significant improvement was made by developing a compact and flexible PCB. This flexible PCB design allowed for a more ergonomic and comfortable fit on the hand, while still effectively recording the bending data of the fingers. In the third version of the data glove, additional sensors were incorporated to further enhance the accuracy and precision of data capture, particularly during grasping actions. In addition to the flex sensors, an accelerometer and pressure sensors were integrated into the glove. The accelerometer measured the hand's orientation and movement in three-dimensional space, providing additional information about hand motion. The pressure sensors detected the pressure or force applied by the fingers during grasping actions, enabling more detailed and comprehensive data capture.

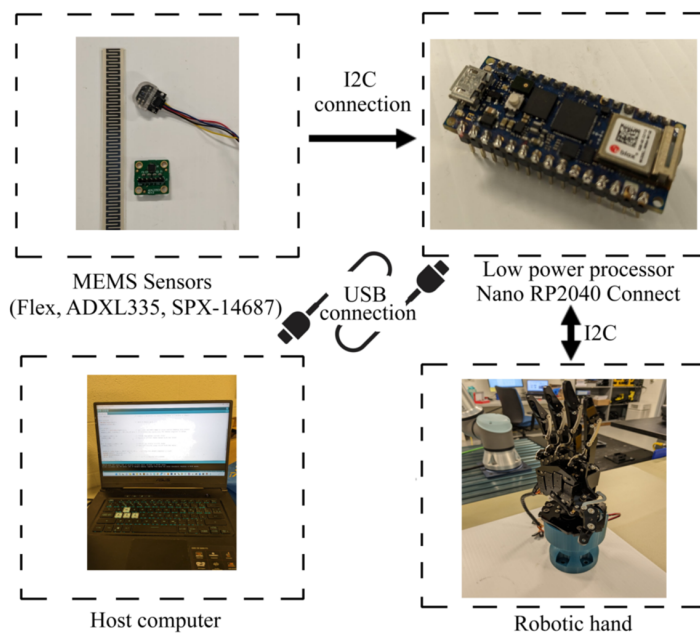


Figure 1. The communication flow of hardware system.

Table 1. The specifications of sensors.

Units	ADXL335	Flex Seonsor	SPX-14687
Dimensions	3 axis	1 axis	1 axis
Power supply	1.8–3.6 V	1.8–3.6 V	2.5–3.6 V
Operating temperature	−40–+85 °C	−35–+80 °C	−40–+85 °C
Dynamic range	±3 g	-----	-----
Cross-axis sensitivity	±1%	-----	-----
Accuracy	±1%	±8%	±10%

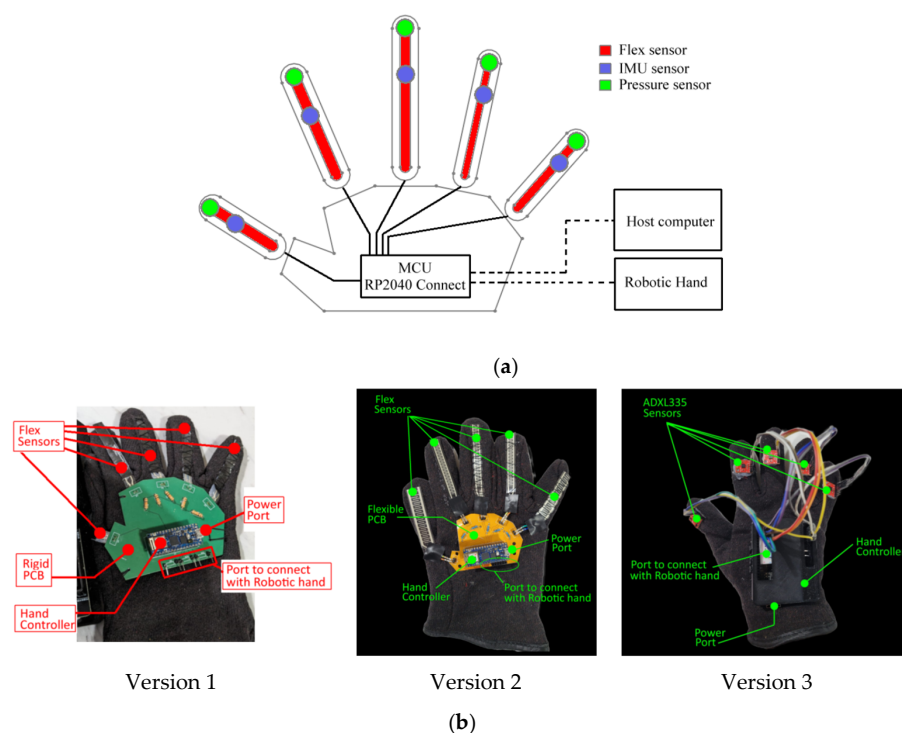
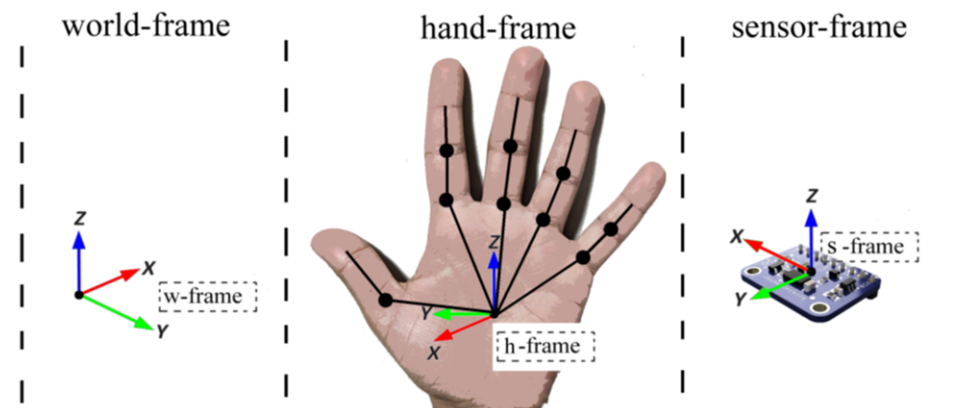


Figure 2. Structure of data glove; (a) schematic diagram of data glove. (b) different prototypes.

## 2.2. Coordinate System

In our system design, there are three coordinate systems that interact through mutual transformations: the world frame (w-frame), the hand frame (h-frame), and the sensor frame (s-frame), as depicted in Figure 3. The definitions of these three frames are as follows.

- *w-frame* is defined with the Earth as the reference for describing human posture. In this study, we adopt the local Earth frame as the w-frame, where the X-axis aligns with the north direction, the Y-axis aligns with the east direction, and the Z-axis is perpendicular to the ground plane, pointing towards the center of the Earth.
- *s-frame* represents the coordinate system defined by the manufacturer during the sensor's design. In our study, s-frame is determined by the ADXL335 inertial sensor. The specific definitions of these coordinate systems are illustrated in Figure 3.
- *h-frame* is primarily employed to characterize the spatial posture of the hand and individual finger segments. In this paper, we consider the palm as the root node and establish five kinematic chains extending to each finger based on the hand's skeletal structure (Figure 3). The segments are connected through the joint points, and each joint is associated with its respective frame to represent the spatial posture of the segment.



**Figure 3.** Definitions of the three frames.

## 2.3. Sensor Placement

The placement of sensors plays a crucial role in the development of the data glove. Figure 4a depicts the structure of the hand joints. The joints in the human hand's index finger, middle finger, ring finger, and little finger can be classified into three main categories: distal interphalangeal joint (DIP), proximal interphalangeal joint (PIP), and metacarpophalangeal joint (MP). However, the thumb differs in having only two joints: interphalangeal joint (IP) and metacarpophalangeal joint (MP). In the work by Fei et al. [16], they positioned 12 IMU sensors on the hand phalanges, while an additional IMU sensor was placed on the back of the hand to estimate joint angles and hand gestures. In this research paper, we find the optimum position of sensors by conducting grasping experiment using objects with different shapes and sizes, as shown in Figure 4b–d. Based on the data obtained from these experiments, we proposed placing accelerometer sensors on the second segment of fingers, specifically between the PIP joint and DIP joint. This segment is primarily responsible for strong grasping actions. Therefore, we use 5 accelerometer sensors and flex sensors for capturing grasping actions, reducing the number of sensors compared to the previous work [16]. The flex, accelerometer, and pressure sensor placement for single finger is depicted in Figure 5.

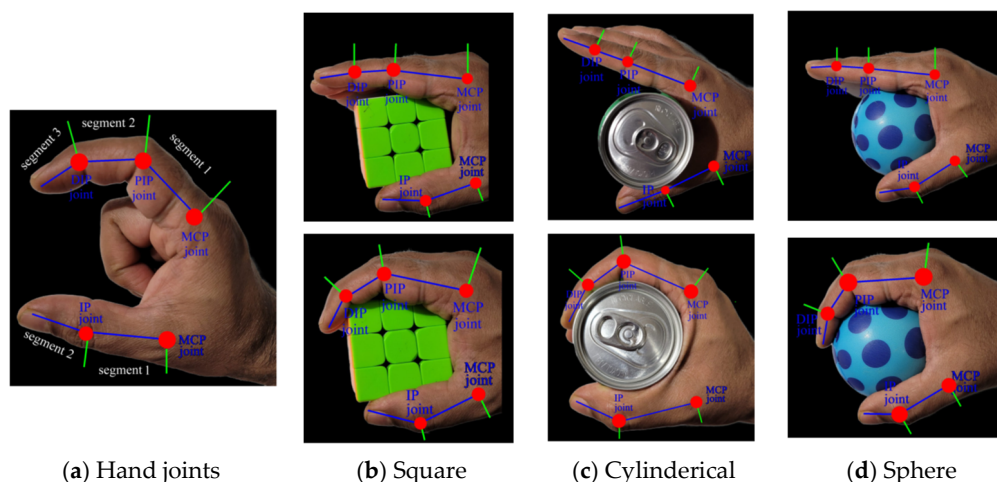


Figure 4. (a) Structure of hand joints, (b–d) grasping action for different objects.

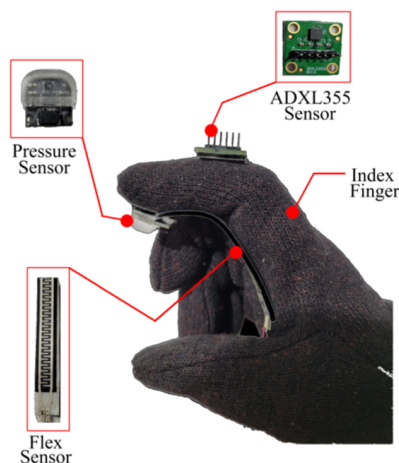


Figure 5. Sensor’s placement for index finger.

### 2.4. Sensor Calibration

The accelerometer may encounter several errors, including zero deviation error, scale factor error, and non-orthogonal error. These errors contribute to the overall accelerometer error model, which can be represented as shown in reference [17].

$$\begin{bmatrix} a_{x\_c} \\ a_{y\_c} \\ a_{z\_c} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{12} & S_{22} & S_{23} \\ S_{13} & S_{23} & S_{33} \end{bmatrix} \times \begin{bmatrix} a_{x\_raw} - b_x \\ a_{y\_raw} - b_y \\ a_{z\_raw} - b_z \end{bmatrix} \tag{1}$$

In Equation (1),  $a_{x\_raw}, a_{y\_raw}, a_{z\_raw}$  show the raw data obtained from the accelerometer and  $a_{x\_c}, a_{y\_c}, a_{z\_c}$  represent the calibrated acceleration data. The terms  $b_x, b_y, b_z$  correspond to the bias correction values, while  $S_{ij} (i = 1, 2, 3; j = 1, 2, 3)$  represents the scale factor and nonorthogonality correction factor values. The calibration error, denoted as  $e(S_{11}, \dots, S_{33}, b_x, \dots, b_z)$ , is defined as the discrepancy between the squared sum of the calibrated acceleration and the squared gravitational acceleration:

$$e(S_{11}, \dots, S_{33}, b_x, \dots, b_z) = a^2_{x\_c} + a^2_{y\_c} + a^2_{z\_c} - g^2 \tag{2}$$

Gauss–Newton’s method is applied to estimate the unknown calibration parameters  $e(S_{11}, \dots, S_{33}, b_x, \dots, b_z)$ . The iteration of calibration error can be represented by the following equation:

$$e_{k+1} = e_k + \gamma_k d_k \tag{3}$$

where  $\gamma_k$  is the damping control factor, which is used to control the convergence speed of the iteration algorithm. A larger  $\gamma_k$  value means a faster convergence speed but lower accuracy.  $d_k$  is defined as the iterative direction:

$$d_k = \left( J^T J^{-1} \right) \left( J^T (-e) \right) \quad (4)$$

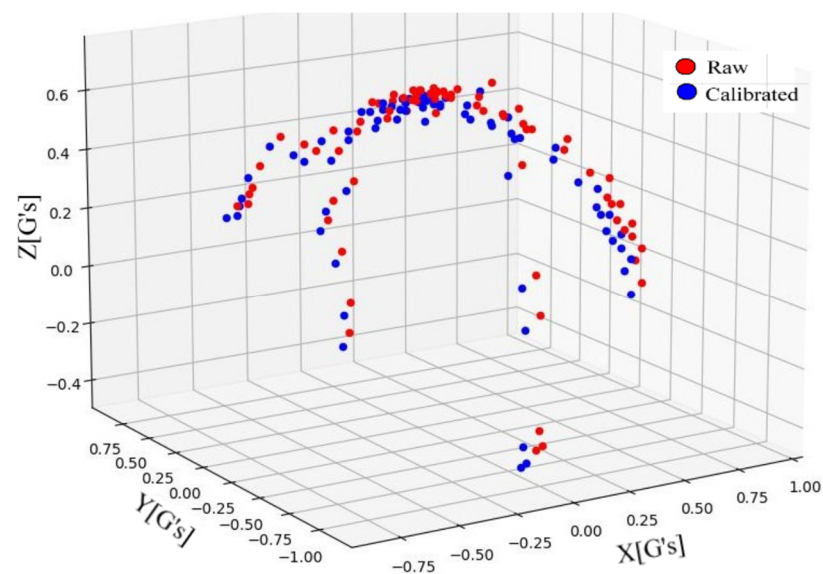
where matrix  $J$  is defined as the Jacobian matrix of the calibration error:

$$J(e) = \frac{\delta e}{\delta x} = \begin{bmatrix} \frac{\delta e}{\delta s_{11}} & \cdots & \frac{\delta e}{\delta b_z} \\ \vdots & \ddots & \vdots \\ \frac{\delta e}{\delta s_{11}} & \cdots & \frac{\delta e}{\delta b_z} \end{bmatrix} \quad (5)$$

When the iteration time reaches its maximum or the convergence condition is satisfied, the unknown calibration parameters can be estimated. With  $\zeta$  defined as the convergence threshold, the convergences condition can be represented as follows:

$$|e(s_{11}, \dots, s_{33}, b_x, \dots, b_z)| < \zeta \quad (6)$$

The Arduino code is utilized to capture the raw data of the accelerometer by moving the ADXL335 sensor along the X-Y-Z directions. Subsequently, a Python code is developed to estimate the bias correction values, as well as the scale factor and nonorthogonality correction factor values. Figure 6 illustrates the comparisons of normalized acceleration data before and after calibration. As can be seen, the accuracy of the local gravitational acceleration is enhanced after calibration.



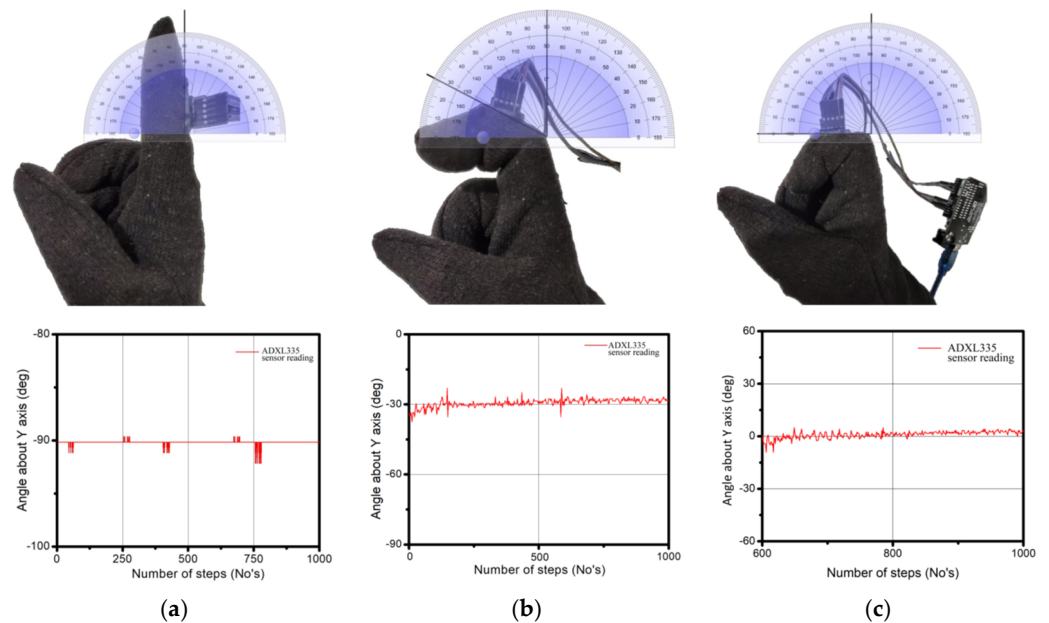
**Figure 6.** Comparisons of accelerometer data distribution before and after calibration.

### 2.5. Experimental Results of Sensor Calibration

To verify the accuracy of the joint angles obtained from the data glove accelerometer (ADXL335), a manual drawing protector is employed to measure the angles. The validation process is depicted in Figure 7. It is worth noting that the angle values are represented as negative due to the utilization of a second quadrant system.

Table 2 presents a comparison of the measurement results obtained from two sources: the drawing protector and the data glove accelerometer. Around 20 samples were collected for 90- and 30-degree positions. The angles recorded correspond to the index finger when held in two distinct positions. It is important to note that only the rotation of the Y-axis of

the accelerometer was considered for this specific experiment. The average error rate was less than 2% and the maximum deviation was nearly 1.55 degrees.



**Figure 7.** Comparisons of the measured angle between data glove accelerometer and drawing protector: (a)  $-90$ -degree, (b)  $-30$  degree, and (c)  $0$  degrees.

**Table 2.** Measurement results of drawing protector and data glove accelerometer.

Angle from drawing protector (deg)	$-90$	$-28.5$
Average angle from data glove (deg)	$-88.45$	$-28.12$
Error (degree)	$1.55$	$0.38$
Error %	$1.75\%$	$1.35\%$

### 3. Human–Robot interaction Using Data Glove

To verify the functionality of the developed data glove, real-time human–robot interaction is performed. In this research paper, a robotic hand is attached to the end of a universal robot (UR5) to replicate the motions performed by the human demonstrator. The robotic hand employed in this study is equipped with five servo motors to control each finger. It is connected to the end of the universal robot (UR5) with 3D printed mounting and operated through the data glove. Figure 8 shows the components in our HRI facilitated by the data glove.

Moreover, this paper proposes a learning-from-demonstration method to develop a machine learning model based on the sensor data from the data glove during various grasping operations. This approach aims to enhance the understanding and capabilities of HRI through machine learning techniques.

This study concentrates on objects with different shapes and sizes. Figure 9 illustrates a flow diagram of the machine learning model proposed in this paper, which incorporates a learning-from-demonstration approach. The objects are categorized into three sets: rectangle, cylindrical, and spherical (Figure 10). The experimentation begins by capturing data from sensors mounted on the data glove when grasping different objects. Each sensor's data was recorded with a sample length of 10,000 ms. This means that for every sensor, a continuous stream of measurements was recorded for a duration of 10 s. The ADXL335 sensor data was collected at a frequency of 7 Hz. A total of 1550 samples were collected as raw data for each object. After recording the raw data from the sensors, we proceeded to split them into training and testing sets using an 80:20 ratio.



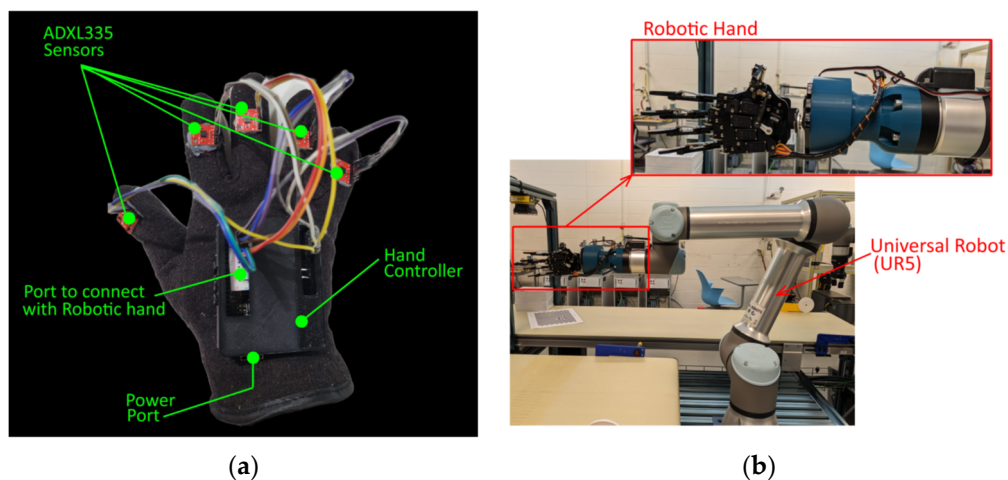


Figure 8. Human–robot interaction components: (a) data glove, and (b) robotic hand with UR5 robot.

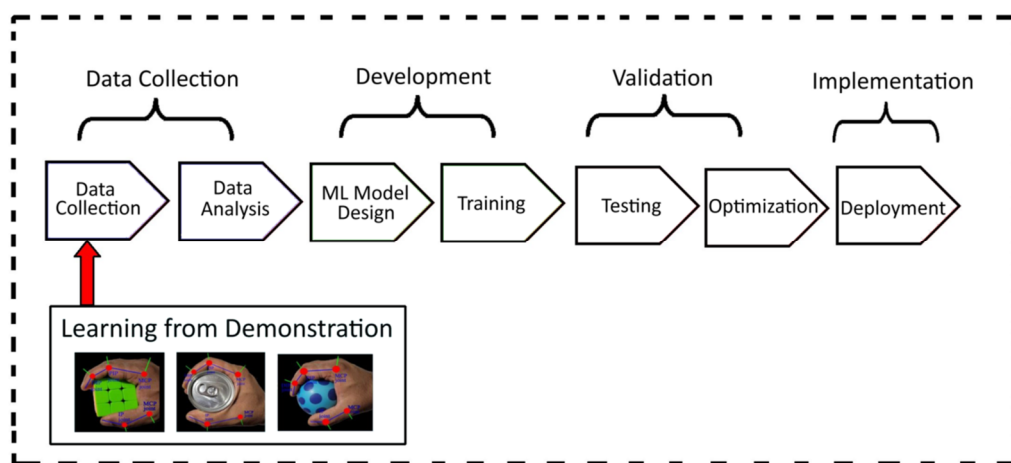


Figure 9. Flow diagram of ML model based on learning from demonstration.

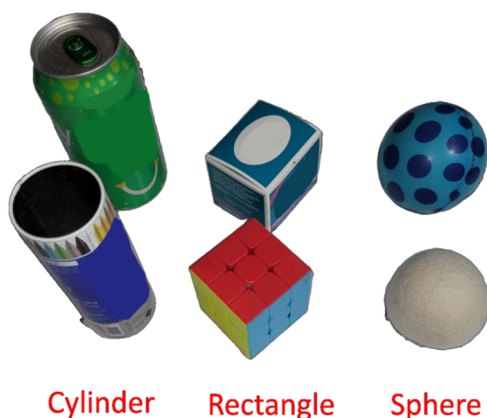
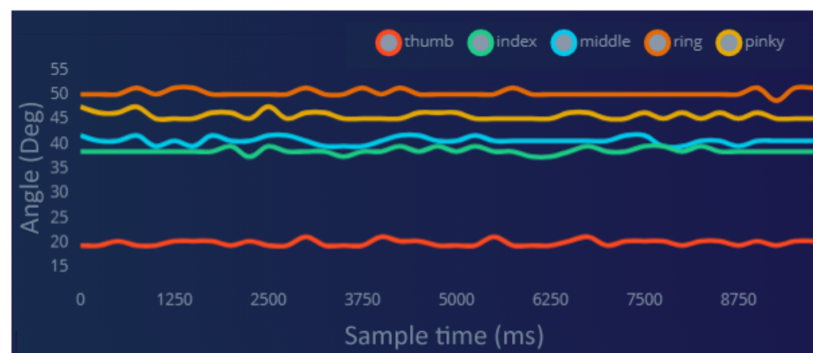


Figure 10. Everyday items for data capture.

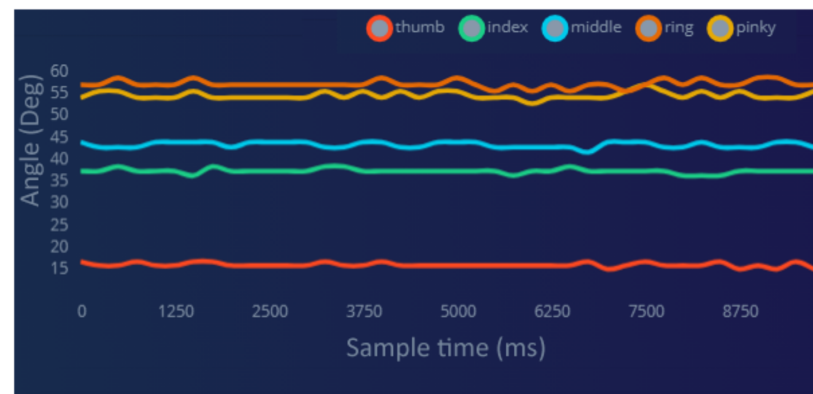
During the data analysis step, the sensor data was fused before being used as input for the machine learning (ML) model. The TinyML model was developed using a preprocessing feature extraction method in the development phase [18]. Following the training phase, the ML model was tested on the test dataset. To optimize the performance of the ML model, the number of training cycles was increased to 250, and the learning rate was set to 0.0005. Finally, the developed model was deployed on a real robotic hand apparatus for grasping objects of various shapes and sizes.

### 3.1. Data Collection

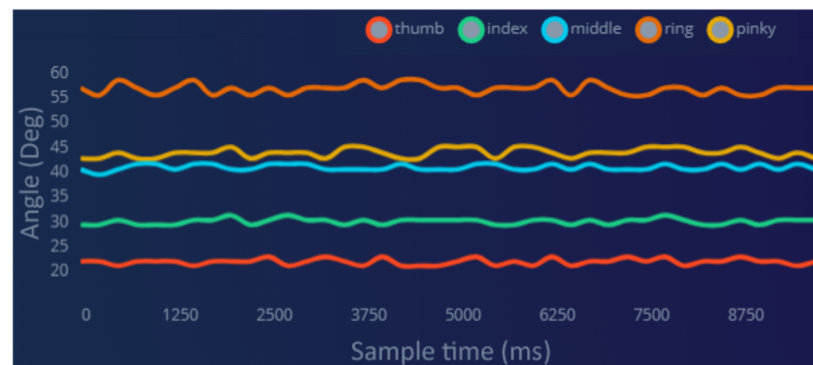
The human demonstrator grabbed each item for 10 s. The accelerometer was utilized to acquire hand data in this study, with a frequency of 7 Hz. The sensors were positioned on the second segment of each finger. Figure 11 illustrates a few samples of the data acquired by the data glove for grasping different objects. As illustrated in Figure 11a, the thumb angle for grasping rectangle-shaped objects ranges from 18 to 21 degrees, while the angles for the ring and pinky fingers vary between 45 and 51 degrees. Similarly, for grasping spherical-shaped objects (Figure 11b), the thumb angle ranges from 15 to 17 degrees, and the angles for the ring and pinky fingers vary between 55 and 58 degrees. When it comes to grasping cylindrical-shaped objects, as shown in Figure 11c, the thumb, ring, and pinky finger angles vary between 20–22 degrees, 55–58 degrees, and 43–45 degrees, respectively. Notably, the index finger angle, as shown in Figure 11c, differs significantly from the angles observed when grasping rectangular- and spherical-shaped objects, ranging from 29 to 30 degrees.



(a)



(b)



(c)

**Figure 11.** Raw data acquired by data glove: (a) rectangle objects, (b) spherical objects, and (c) cylindrical objects.

### 3.2. Data Analysis and Feature Extraction

In this paper, the spectral analysis is utilized for data analysis and feature extraction. Low-pass and high-pass filters are applied to filter out unwanted frequencies. Spectral analysis is a great tool for analyzing repetitive motion, such as data from accelerometers in our case. It extracts the features based on frequency and power characteristics of a signal over time. The wavelet transformation is often preferred over Fourier transformations in certain applications due to its advantages in examining specific frequencies and reducing computational requirements.

We used the following wavelet transform to extract the features.

$$F(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t) \psi^* \left( \frac{t - \tau}{s} \right) dt \quad (7)$$

where  $\tau$  is scaling factor and  $s$  represents time shift factor. Figure 12 depicts the feature extraction from the accelerometer data using wavelet transform for a sample time of 5 s.

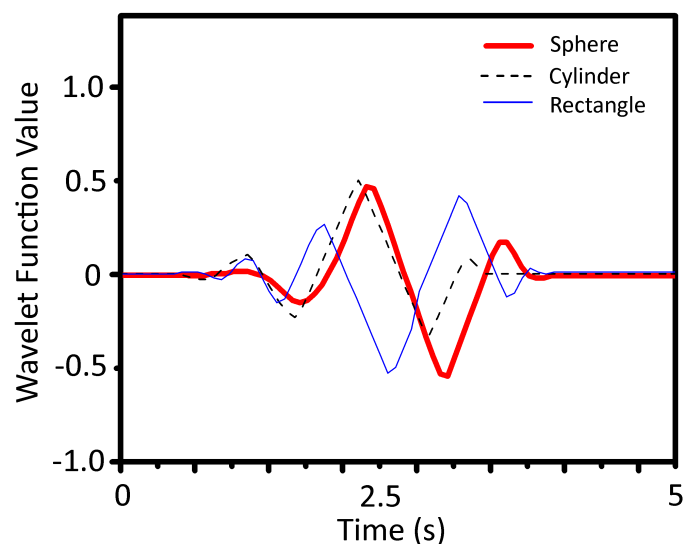
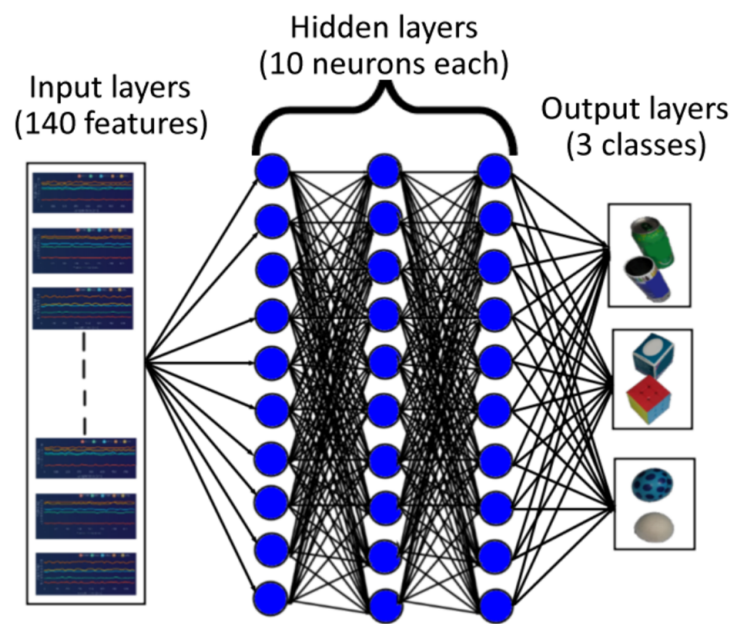


Figure 12. Feature extraction using wavelet transform.

### 3.3. Neural Network Classifier

A neural network classifier takes some input data and outputs a probability score that indicates how likely it is that the input data belongs to a particular class. The neural network is composed of multiple layers, each containing numerous interconnected neurons. The network's output is compared to the correct answers, and based on the results, the weights of the connections between neurons are modified. This iterative process is repeated until the network effectively learns to predict the correct answers for the training data [18]. Figure 13 shows the neural network architecture employed in this paper. We used a shallow feed-forward neural network (NN) composed of three layers. The input layer consists of 140 neurons, which are determined by the dimensions in the input data. The hidden layer is composed of layers, each having 10 neurons with hyperbolic tangent activation functions. The output layer has three neurons with the softmax transfer function, which corresponds to the number of classes (cylinder, rectangle, sphere). For training the NN, we use the gradient descend method and the Adam optimizer from the *tensorflow.keras.optimizers* library. A batch size of 32 is considered for the training process. Table 3 shows the neural network classifier setting parameters. The training was performed on a personal computer with an Intel® Core™ i7 processor with 8 GB of RAM memory.



**Figure 13.** Neural network architecture.

**Table 3.** Neural network classifier parameter values.

Parameters	Value
Number of training cycles	300
Learning rate	0.0005
Validation set size	20

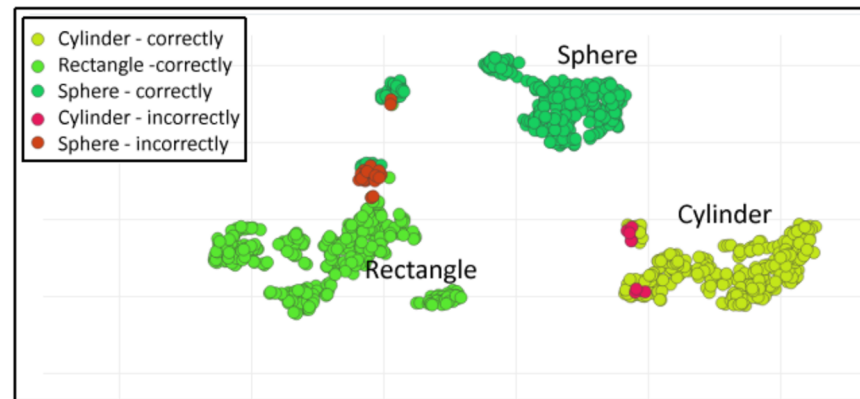
To evaluate our model performance, we used a confusion matrix, which tabulates a summary of the correct and incorrect responses generated by the model when presented with a dataset. The labels on the side of the matrix represent the actual labels present in each sample, while the labels on the top represent the predicted labels produced by the model. The confusion matrix is illustrated in Figure 14. The overall accuracy of the trained model is 95.7%, which is acceptable for our application. The spatial distribution of input features is shown in Figure 15. Green-color items are classified correctly, whereas items in red color are misclassified. Hence, the wavelet-based function proposed in Section 3.2 demonstrates its effectiveness in effectively distinguishing between the different classes.

	Cylinder	Rectangle	Sphere
Cylinder	96.2%	1.9%	1.9%
Rectangle	0%	100%	0%
Sphere	0%	8.3%	91.7%
F1 Score	0.98	0.95	0.95

**Figure 14.** Confusion matrix.

By analyzing the results of grasping, it was observed that accelerometer sensors, which capture the dynamic movements and orientations of the grasped objects, can effectively differentiate between spheres and rectangles or cylinders. This is due to fact that the rolling motion of a sphere during a grasp generates distinctive accelerometer patterns compared to the linear movements associated with grasping rectangles or cylinders. Moreover, we noticed that grasps on spheres resulted in more evenly distributed pressure across the contact area, while grasps on rectangles and cylinders exhibit higher pressure around the edges or corners due to the need for more precise finger positioning. Additionally, we

observed that flexi sensors placed on the finger joints could capture the degree of finger flexion during grasping. Experiments showed that grasps on rectangles require more pronounced flexion in the fingers to achieve a stable grip, while grasps on spheres involve more extended finger positions.



**Figure 15.** Spatial distribution of input features.

### 3.4. Model Testing

The trained model undergoes testing using the test dataset to validate its performance. Table 4 shows the prediction results for the test dataset. The overall accuracy achieved during testing is 96.22%. To further improve the model's accuracy, the samples that were initially part of the test dataset can be moved back to the training dataset, and the model can be retrained with these augmented training data.

**Table 4.** Test dataset validation.

Sample Name	Expected Lables	Time Length	Accuracy %	Results
cylinder.4dmd	cylinder	10 s	97%	36 cylinder, 1 other
rectangle.4dmf	rectangle	10 s	89%	33 cylinder, 4 others
sphere.4dme	sphere	10 s	100%	37 sphere
sphere.4dmt	sphere	10 s	94%	35 sphere, 2 others
cylinder.4dmdh	cylinder	10 s	100%	37 cylinder

Metric parameters, such as accuracy, precision, and recall, play a crucial role in evaluating the performance of classification models in machine learning. Accuracy measures the overall correctness of the model's predictions. Precision indicates the model's accuracy when it predicts the target class, while recall measures the model's ability to correctly identify all instances of the target class. Equations (8)–(10) represent the formulas for calculating accuracy, precision, and recall. These metrics provide valuable insights into the effectiveness and reliability of a classification model.

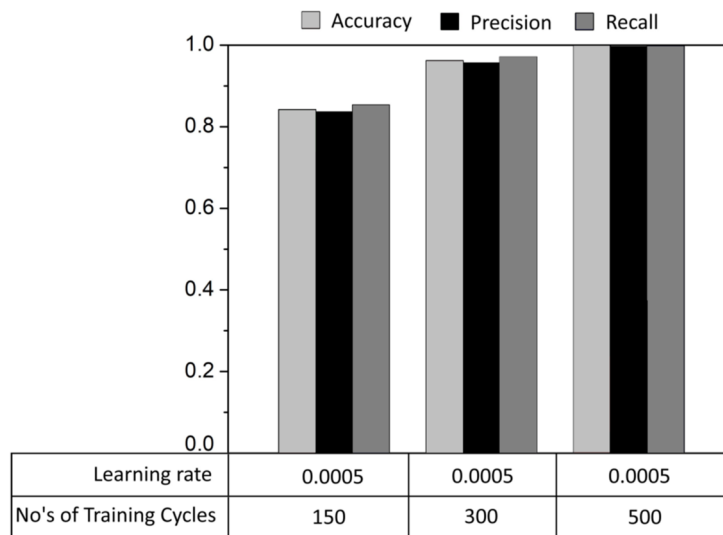
$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (8)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (9)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (10)$$

Figure 16 illustrates the performance metrics of the proposed model, considering different training settings by varying the number of training cycles. In this comparison, the learning rate value was kept constant at 0.0005. It can be seen that when the number of training samples is 150, the overall accuracy of the model reaches 84%. However, if the

number of training samples is increased to 500, the model exhibits near-perfect accuracy, indicating overfitting. To address this issue, we set the number of training samples to 300, resulting in improved overall accuracy, precision, and recall values.



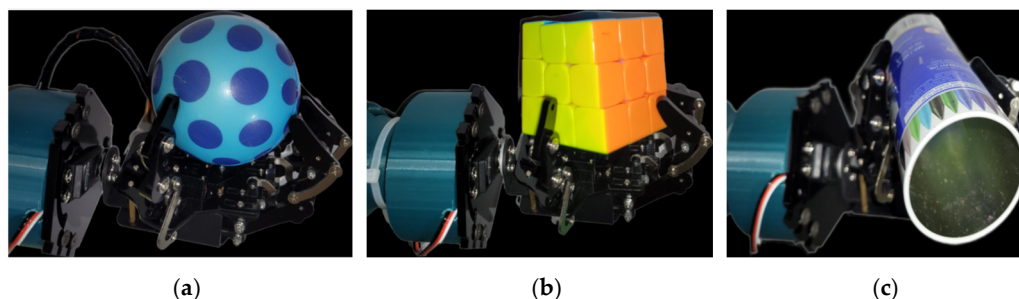
**Figure 16.** Performance matrix.

### 3.5. Experimentation Setup

The proposed HRI system is validated using a real robot arm with an attached robotic hand, as depicted in Figure 8b. In this study, the object class type (sphere, cylinder, rectangle) is manually provided as input to the model, and the model predicts the corresponding movements (angle in degrees) of the robotic hand's fingers, which are mapped into servo angles ranging from 0 to 90 degrees. The average bending angles of servo joints of robotic hand fingers for grasping the objects are shown in Table 5. Figure 17 showcases the results of the robotic grasping of objects with different shapes and sizes.

**Table 5.** Bending angles of servo joints.

Objects Shape	Thumb (deg)	Index Finger (deg)	Middle Finger (deg)	Ring Finger (deg)	Pinky Finger (deg)
Sphere	14.2	36.46	34.42	40.13	42.12
Rectangle	18.59	39.62	42.93	55.70	52.81
Cylinder	22.45	32.43	44.52	57.20	45.19



**Figure 17.** Grasping the objects with different shape and size: (a) sphere, (b) rectangle, and (c) cylinder.

The performance of the proposed data glove is compared to other state-of-the-art data glove prototypes in Table 6. The results indicate that data gloves utilizing optical fiber sensors and IMU sensors outperform in measuring joint angles. However, IMU sensors generally offer a lower cost compared to optical fiber sensors. In our proposed design,

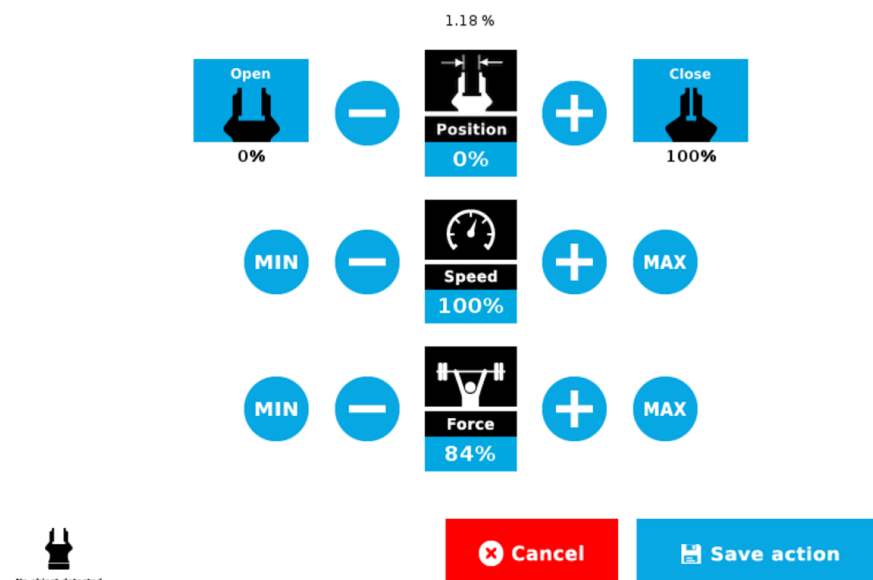
we incorporate five IMU sensors (accelerometers) along with flex sensors to monitor the proximal interphalangeal (PIP) joint angle for each finger. Additionally, an optional IMU sensor is included to track wrist roll motion.

**Table 6.** Comparison of proposed data glove.

Publications, Year	Type of Sensor	Number of Sensors	Joint Angle Deviation
Fei et al. [16], 2021	IMMU	12 (one hand)	1.4 deg
Cha et al. [19], 2017	Piezoelectric	19 (one hand)	5 deg
Li et al. [20], 2011	Optical encoder	14 (one hand)	1 deg
Proposed data glove	IMMU + flex	5 + 5 (one hand)	1.55 deg

Although a wearable glove-based HRI offers promising possibilities, it comes with certain limitations and challenges. Integrating multiple sensors into a wearable glove requires accurate calibration. To this end, we calibrated all sensors to ensure accurate and synchronized sensor measurements across different modalities (e.g., accelerometers, force sensors, flex sensors). Also, sensor measurements can be affected by noise or occasional sensor failures, leading to unreliable data. In this paper, we employed the sensor fusion technique to combine information from multiple sensors, improving overall accuracy and robustness. Moreover, proper sensor positioning is essential to capture relevant data and enable accurate interpretation of human hand movements and grasp patterns. By conducting several experiments, we optimized the sensor placement and orientation for improved HRI performance.

For future work, we will incorporate a camera into the HRI system to capture images of objects and automatically detect object classes, enhancing its ability to recognize and manipulate objects. Additionally, we will deploy the flex and pressure sensors to adjust position and force of gripping with an industrial gripper. As can be seen in Figure 18, universal robots need position, speed, and force parameters, which should be set before object grasping.



**Figure 18.** Universal robot interface for grasping.

#### 4. Conclusions

This paper presents a practical human–robot interaction system based on a wearable glove for object grasp application. We embedded three different types of sensors to capture grasp information from human demonstrators to imitate the posture and dynamics of the human hand. Calibration algorithms were implemented to avoid errors caused by the

sensors. After analyzing the collected data, the positions of the accelerometer, pressure, and flexi sensors were determined in the wearable glove to capture grasp information of objects with different sizes and shapes. A three-layer neural network was trained to recognize grasp orientation and position, based on the multi-sensor fusion method. The experimental results showed that an industrial robot can successfully grasp sphere, rectangle, and cylinder objects with the proposed HRI based on learning from demonstration.

**Author Contributions:** Conceptualization, S.A.; methodology, M.J.A. and S.M.; software, R.S.; validation, R.S. and M.A.; formal analysis, M.J.A.; investigation, S.A.; resources, S.M.; data curation, M.A.; writing—original draft preparation, R.S.; writing—review and editing, S.M.; visualization, M.A.; supervision, S.A.; project administration, S.A.; funding acquisition, S.R.-G. and S.A. All authors contributed equally. All authors have read and agreed to the published version of the manuscript.

**Funding:** The University of Windsor, Faculty of Engineering Innovating Sustainability Grant, 2022.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data related to this paper are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bartneck, C.; Belpaeme, T.; Eyssel, F.; Kanda, T.; Keijsers, M.; Sabanovic, S. *Human-Robot Interaction: An Introduction*; Cambridge University Press: Cambridge, UK, 2020.
2. Matthews, P.; Greenspan, S. *Automation and Collaborative Robotics: A Guide to the Future of Work*; Apress: Berkeley, CA, USA, 2020.
3. Jahanmahin, R.; Masoud, S.; Rickli, J.; Djuric, A. Human-robot interactions in manufacturing: A survey of human behavior modeling. *Robot. Comput. Integr. Manuf.* **2022**, *78*, 102404. [[CrossRef](#)]
4. Esterwood, C.; Robert, L.P. A Systematic Review of Human and Robot Personality in Health Care Human-Robot Interaction. *Front. Robot. AI* **2021**, *8*, 306. [[CrossRef](#)] [[PubMed](#)]
5. Triebel, R.; Arras, K.; Alami, R.; Beyer, L.; Breuers, S.; Chatila, R.; Chetouani, M.; Cremers, D.; Evers, V.; Fiore, M.; et al. Spencer: A socially aware service robot for passenger guidance and help in busy airports. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 607–622.
6. Cui, Y.; Song, X.; Hu, Q.; Li, Y.; Sharma, P.; Khapre, S. Human-robot interaction in higher education for predicting student engagement. *Comput. Electr. Eng.* **2022**, *99*, 107827. [[CrossRef](#)]
7. Akhshik, M.; Mozaffari, S.; Singh, R.; Rondeau-Gagné, S.; Alirezaee, S. Pressure Sensor Positioning for Accurate Human Interaction with a Robotic Hand. In Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 13–14 July 2023; pp. 1–4.
8. Elachkar, M.; Mozaffari, S.; Ahmadi, M.; Ahamed, J.; Alirezaee, S. An Experimental Setup for Robot Learning From Human Observation using Deep Neural Networks. In Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 13–14 July 2023; pp. 1–4.
9. Ravichandar, H.; Polydoros, A.S.; Chernova, S.; Billard, A. Recent Advances in Robot Learning from Demonstration. *Annu. Rev. Control. Robot. Auton. Syst.* **2020**, *3*, 297–330. [[CrossRef](#)]
10. Sakr, M.; Freeman, M.; Van der Loos, H.M.; Croft, E. Training Human Teacher to Improve Robot Learning from Demonstration: A Pilot Study on Kinesthetic Teaching. In Proceedings of the 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, 31 August–4 September 2020; pp. 800–806.
11. Fischer, K.; Kirstein, F.; Jensen, L.C.; Krüger, N.; Kukliński, K.; aus der Wieschen, M.V.; Savarimuthu, T.R. A comparison of types of robot control for programming by demonstration. In Proceedings of the 11th ACM/IEEE International Conference Human-Robot Interaction, Christchurch, New Zealand, 7–10 March 2016; pp. 213–220.
12. Si, W.; Wang, N.; Yang, C. A review on manipulation skill acquisition through teleoperation-based learning from demonstration. *Cogn. Comput. Syst.* **2021**, *3*, 1–16. [[CrossRef](#)]
13. Wei, D.; Chen, L.; Zhao, L.; Zhou, H.; Huang, B. A Vision-Based Measure of Environmental Effects on Inferring Human Intention During Human Robot Interaction. *IEEE Sens. J.* **2021**, *22*, 4246–4256. [[CrossRef](#)]
14. Fang, B.; Sun, F.; Liu, H.; Liu, C.; Guo, D. *Wearable Technology for Robotic Manipulation and Learning*; Springer: Singapore, 2020.
15. Ponraj, G.; Ren, H. Sensor fusion of leap motion controller and flex sensors using Kalman filter for human finger tracking. *IEEE Sens. J.* **2018**, *18*, 2042–2049. [[CrossRef](#)]
16. Fei, F.; Sifan, X.; Xiaojian, X.; Changcheng, W.; Dehua, Y.; Kuiying, Y.; Guanglie, Z. Development of a wearable glove system with multiple sensors for hand kinematics assessment. *Micromachines* **2021**, *12*, 362–381. [[CrossRef](#)] [[PubMed](#)]



17. Frosio, I.; Pedersini, F.; Borghese, N.A. Autocalibration of MEMS accelerometers. *IEEE Trans. Instrum. Meas.* **2008**, *58*, 2034–2041. [[CrossRef](#)]
18. Avellaneda, D.; Mendez, D.; Fortino, G. A TinyML Deep Learning Approach for Indoor Tracking of Assets. *Sensors* **2023**, *23*, 1542–1564. [[CrossRef](#)] [[PubMed](#)]
19. Cha, Y.; Seo, J.; Kim, J.-S.; Park, J.-M. Human–computer interface glove using flexible piezoelectric sensors. *Smart Mater. Struct.* **2017**, *26*, 057002. [[CrossRef](#)]
20. Li, K.; Chen, I.M.; Yeo, S.H.; Lim, C.K. Development of finger-motion capturing device based on optical linear encoder. *J. Rehabil. Res. Dev.* **2011**, *48*, 69–82. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.