The International Journal of Advanced Manufacturing Technology manuscript No.
(will be inserted by the editor)

Gesture-based human-robot interaction for human assistance in manufacturing

Pedro Neto · Miguel Simao · Nuno Mendes · Mohammad Safeea

Received: date / Accepted: date

Abstract The paradigm for robot usage has changed in the last few years, from a scenario in which robots work isolated to a scenario where robots collaborate with human beings, exploiting and combining the best abilities of robots and humans. The development and acceptance of collaborative robots is highly dependent on reliable and intuitive human-robot interaction (HRI) in the factory floor. This paper proposes a gesture-based HRI framework in which a robot assists a human co-worker delivering tools and parts, and holding objects to/for an assembly operation. Wearable sensors, inertial measurement units (IMUs), are used to capture the human upper body gestures. Captured data are segmented in static and dynamic blocks recurring to an unsupervised sliding window approach. Static and dynamic data blocks feed an artificial neural network (ANN) for static, dynamic and composed gesture classification. For the HRI interface we propose a parameterization robotic task manager (PRTM), in which according to the system speech and visual feedback the co-worker selects/validates robot options using gestures. Experiments in an assembly operation demonstrated the efficiency of the proposed solution.

Keywords Human-Robot Interaction · Collaborative Robotics · Gesture Recognition · Intuitive Interfaces

1 Introduction

Collaborative robots are increasingly present in manufacturing domain, sharing the same workspace and collaborating with human co-workers.This collaborative scenario allows to exploit the best abilities of robots (accuracy,

P. Neto
Department of Mechanical Engineering, University of Coimbra, Coimbra, Portugal
Tel.: +351 239 790 700
Fax: +351 239 790 700
E-mail: pedro.neto@dem.uc.pt

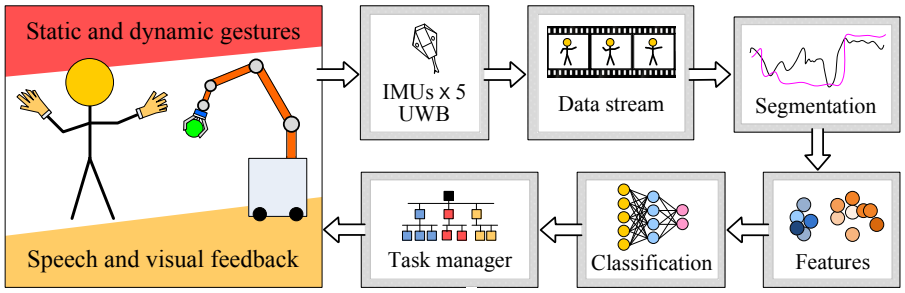

Fig. 1 Overview of the proposed gesture-based HRI framework.

repetitive work, etc.) and humans (cognition, management, etc.) [1][2]. The development and acceptance of collaborative robots in industry is highly dependent on reliable and intuitive human-robot interaction (HRI) interfaces [3], i.e., making robots accessible to human beings without major skills in robotics. Collaborative robots and humans have to understand each other and interact in an intuitive way, creating a co-working partnership. This will allow a greater presence of collaborative robots in industrial companies which are struggling to have ever more flexible production due to consumer demand for customized products [4]. For example, a human-robot collaborative platform for constructing panels from preimpregnated carbon fibre fabrics in which the human and robot share the workspace promoting situation awareness, danger perception and enrichment of communication [5].

Instructing and programming an industrial robot by the traditional teaching method (text and teach pendant based methods) is a tedious and time-consuming task that requires technical expertise [6]. In addition, these modes of robot interfacing are hard to justify for flexible production where the need for robot re-configuration is constant. Recently, human-robot interfaces based in robot hand-guiding (kinesthetic teaching) and haptic interfaces demonstrated to be intuitive to use by humans without deep skills in robotics [7]. In addition, advanced and natural HRI interfaces such as human gestures and speech still lack in reliability in industrial/unstructured environment [8]. An interesting study reports the impact of human-robot interfaces to intuitively teach a robot to recognize objects [9].The study demonstrated that the smartphone interface allows non-expert users to intuitively interact with the robot, with a good usability and user's experience when compared to a gesture-based interface. The efficiency of a conventional keyboard and a gesture-based interface in controlling the display/camera of a robot is presented in [10]. The gesture-based interface allowed smoother and more continuous control of the platform, while the keyboard provided superior performance in terms of task completion time, ease of use, and workload.

Making an analogy with the way humans interact and teach each other, allows us to understand the importance of gesture-based HRI. Static gestures are human postures in which the human is static (small motion like body shaking can occur) and dynamic gestures are represented by a dynamic be-

haviour of part of the human body (normally the arms). Gestures can be used as an interface to teleoperate a robot, allowing to setup robot configurations and combine with other interfaces such as kinesthetic interface and speech. For instance, a human co-worker can point to indicate a grasping position to the robot, use a dynamic gesture to move the robot to a given position and use a static gesture to stop the robot [11,12]. This scenario allows the human co-worker to focus on the process task and not in the robot programming [13].

Fig. 1 illustrates the proposed framework. Static and dynamic gesture data are acquired from upper body IMUs, segmented by motion, and different ANNs are employed to classify static and dynamic gestures. Recognized gesture patterns are used to teleoperate/instruct a collaborative robot in a process conducted by a parameterization robotic task manager (PRTM) algorithm. The system provides visual and speech feedback to the human co-worker, indicating to the user what gesture was recognized, or if no gesture was recognized.

Depending on the industrial domain and the company itself, the shop floor presents restrictions to the technologies used in the manufacturing processes. The implementation of human-robot collaborative manufacturing processes is today a main challenge for industry. Beyond the related human factors, the advanced human-robot interfaces (gestures, speech, hybrid, etc.) are constrained by the shop floor conditions. In noisy environments the human-human verbal communication is difficult to achieve or prohibitive in some cases, especially when the workers are using earplugs. In this scenario speech interfaces are not efficient and gesture interfaces are a valid alternative. On the other hand, confined spaces hamper the use of arm gestures. In these conditions, the design of the collaborative robotic system has to be adapted according to the specific manufacturing conditions.

In this study we assume that the shop floor environment is noisy and not confined in space, so that gestures are used to interface with the robot. Our proposed approach brings benefits and it is practically relevant in the context of flexible production in small lot sizes [8,14], namely:

1. The human co-worker and robot work in parallel, while the robot is ready to assist the human when required;
2. The use of the robot reduces the exposition of the human co-worker to poor ergonomic conditions and possible injuries (through hand-guiding the robot can be adjusted online to the human body dimensions);
3. The use of the robot reduces error in production since the work plan is strictly followed and managed by the PRTM;
4. The robot assists the human in complex tasks that cannot be fully automated, reducing the cycle time;
5. The introduction of the collaborative robot improves the quality of some tasks when compared with human labour;
6. The collaborative robot allows to reduce drastically the setup time for a new product or variant of a product. This is critical in small lot production.

This work was developed according to the needs of the project ColRobot[1], which intends the development of a collaborative robot for assembly operations in automotive and spacecraft industry. The robot should be able to assist workers, acting as a third hand, by delivering parts and tools for the assembly process.

Section II presents the segmentation by motion process. Section III details the proposed classifiers and the feature dimensionality reduction and regularization. The robot task manager is presented in section IV, while experiments and results are shown in section V. Finally, the conclusion and directions for future work are in section VI.

1.1 Challenges, Proposed Approach and Contributions

The problems and challenges to address in collaborative HRI are multiple. Especial attention has to be devoted to the reliability of the existing interfaces, the accuracy of gesture classification in continuous and real-time, and the interface with the robot. This is especially important in a situation where a wrong classification of a gesture may lead to accidents/collisions. The HRI interface has to be prepared to manage this situation, having validation procedures and hardware capable to ensure safety in all circumstances. In presence of an unstructured/industrial environment, several challenges can be pointed out:

1. Achieve high gesture recognition rates (close to 100%) and assure the generalization capability in respect to untrained samples and new users (user independent). The appearance of false positives and false negatives should be reduced to a minimum;
2. Combine and fuse sensor data to better describe the human behavior (hand, arms and body in general) with accuracy, no occlusions and independently from environment conditions (light, magnetic fields, etc.). Selection of proper gesture features according to each specific sensor;
3. Intuitive and modular interfacing with robot, ensuring the management and coordination of human and robot actions. The human co-worker has to receive feedback in anticipation related with future robot actions.

In this paper we propose a gesture-based HRI framework in which a collaborative robot acts as a "third hand" by delivering to the human shared workplace tools, parts, and holding work pieces while the human co-worker performs assembly working operations on it. This framework was tested in an standard manufacturing assembly operation.

The proposed gesture-based HRI, Fig. 1, relies on IMUs to capture human upper body gestures and a ultra wideband (UWB) positioning system to have an indication of the relative position between human and robot. Static and dynamic segments are obtained automatically with a sliding-window motion

[1] https://www.colrobot.eu/

detection method. Static segments will input the classification of static gestures (SGs) and dynamic segments will input the classification of dynamic gestures (DGs) after up- or down-sampling of gesture frames using bicubic interpolation. To avoid false positives/negatives, we implemented what we call composed gestures, which combine SGs and DGs in a given sequence. We proved that ANNs are reliable to classify both SGs, DGs and consequently the composed gestures. A PRTM correlates the classified gestures with actual commands to be sent to a robot and automatic speech and visual feedback for the co-worker.

Inspired by the way humans interact with a phone auto attendant (digital receptionist) in which computer speech feedback indicates to the human the phone number to select according to the desired service (navigate in the menus), our proposed gesture-based HRI interface works in a similar way. The PRTM uses computer speech and visual feedback to indicate the options available to the human co-worker (for example bring a tool, a part or holding a part by setup a kinesthetic teaching mode) and the human uses gestures to select and validate the existing options. This is a modular solution (other functionalities can be added), intuitive (the co-workers does not have to remember a large number of gestures), and flexible (adapted to different scenarios, users and robots). The PRTM can be customized to run with speech recognition commands or robot touch commands instead gestures. Due to the advances in speech recognition in the last two decades it is expected that such a solution will work with a high level of reliability in silent environments. Nevertheless, the use of automatic speech feedback (using headphones) combined with visual feedback (using a monitor installed in the robotic cell) to the human demonstrated to be effective. The feedback information is redundant so that when the level of noise is too high the human co-worker can follow the information in the monitor screen. Both audio and visual feedback provides information about robot state, the next task of the sequence and if the task ended.

The human co-worker is free to move in the workspace, which may conduct to the appearance/classification of gesture false positives (human behaviors are unexpectedly classified as gestures). To avoid this scenario, since the UWB provides human positional data, gesture classification is only activated when the human is in a specific place in front of the robot (other places may be defined). In addition, the classifiers only act when the PRTM is expecting a given gesture during a parameterization phase. The human co-worker selects from the available library what gestures associate to the robot actions managed by the PRTM, customizing the human-robot interface.

The experiments performed in an assembly operation demonstrated the following contributions:

1. The proposed unsupervised segmentation allows to detect all static and dynamic motion blocks, i.e., when a given static or dynamic gesture starts and ends;

2. Gesture recognition accuracy is relatively high (90% - 100%) for a library of 8 SGs and 4 DGs. These results were obtained in continuous, real-time and with seven different subjects (user independent);
3. A good generalization can be achieved with respect to untrained samples and new subjects using the system;
4. The PRTM demonstrated efficiency, reliability, and easy to use behaviour. Several users indicated in questionnaires that it is easy to understand the speech and visual instructions to select robot tasks and use the robot as a "tool", without skills in robot programming.

1.2 Related Work

Collaborative robotics is an emerging and multidisciplinary research field, in which gesture-based HRI is an active research topic. Gestures are a meaningful part of human communication, sometimes providing information that is hard to convey in speech [15]. Gestures can be categorized according to their functionality [16]. Communicative gestures provide information that is hard to convey in speech, for example command gestures [17], pointing [18], gestures to represent meaningful, objects or actions, and mimicking gestures [19, 17]. Gestures have been proven to be one of the most effective and natural mechanisms for reliable HRI, promoting a natural interaction process. In the context of HRI, they have been used for robot teleoperation, and to coordinate the interaction process and cooperation activities between human and robot. As stated in [7], a gesture-based robotic task generally consists of individual actions, operations, and gestures that are arranged in a hierarchical order. Also, there is not necessarily a one-on-one relationship between gestures and actions, one gesture can encode several actions. Therefore, a hierarchical chain of gestures is required to perform a certain task. For example, the user can point to an object in order to select it, but the action to be taken in respect to that object is unknown to the system. The actions can be picking up the object, painting it, welding it or inspecting it, among others.

Recognized human gestures and actions can be applied to define robot motion directions [20] and to coordinate the interaction process and cooperation activities [21]. Some authors discuss what gestures are the most effective in improving human robot interaction processes [22, 23].

Some gestures, although not all, can be defined by their spatial trajectory. This is particularly true for pantomimic gestures [19], which are often used to demonstrate a certain motion to be done, e.g., a circle. Burke and Lasenby focused with success on using Principal Component Analysis (PCA) and Bayesian filtering to classify these time series. In [24], Shao and Li propose the use of an estimation of integral invariants – line integrals of a class of kernel functions along a motion trajectory – to measure the similarity between trajectories. They also propose boosting the classification using machine learning methods such as Hidden Markov Models (HMM) and Support Vector Machine (SVM).

Gesture spotting, either static or dynamic, is an active area of research with many possible applications. The problem becomes more challenging when gestures are recognized in real-time [13]. The difficulty is that gestures typically appear within a continuous stream of motion. Temporal gesture segmentation is the problem of determining when a gesture starts and ends in a continuous stream of data. Segmentation should also decrease the number of classifications performed, reducing the processing load and enhancing the real-time characteristic of a system. When the segmentation is incorrect the recognition is more likely to fail [25]. Analyzing continuous image streams is a challenge to solve spatial and temporal segmentation [26].

The input features for gesture recognition are normally the hand/arm/body position, orientation and motion [27], often captured from vision sensors. However, it is difficult to construct reliable features from only vision sensing due to occlusions, varying light conditions and free movement of the user in the scene [28,17]. With this in mind, several approaches to gesture recognition rely on wearable sensors such as data gloves, magnetic tracking sensors, inertial measurement units (IMUs), Electromyography (EMGs), etc. In fact, these interaction technologies have been proven to provide reliable features in unstructured environments. Nevertheless, they also place an added burden on the user since they are wearable. Data from commercial off-the-shelf devices like a smartwatch can be used to recognize gestures and for defining velocity commands for a robot in an intuitive way [29].

Researchers have used various methods such as HMM, ANN, SVM, Dynamic Time Warping (DTW), deep learning, among other techniques, to recognize gesture patterns. HMMs can be used to find time dependencies in skeletal features extracted from image and depth data (RGB-D) with a combination of Deep Belief Networks (DBNs) and 3D Convolutional Neural Networks (CNNs) [30]. Deep learning combined with recurrent networks demonstrated state of the art performance in the classification of human activities from wearable sensing [31]. ANNs demonstrated superior performance in the classification of high number of gesture patterns, for example an accuracy of 99% for a library of 10 dynamic gestures and 96% for 30 static gestures [13]. Field et al. used a Gaussian Mixture Model (GMM) to classify human's body postures (gestures) with previous unsupervised temporal clustering [32]. A Gaussian temporal smoothing kernel is incorporated into a Hidden-State Conditional Random Fields (HCRF) formulation to capture long-range dependencies and make the system less sensitive to input noise data [33].

Hand detection is critical for reliable gesture classification. This problem has been approached using wearable and vision sensing. Recent studies report interesting results in hand detection and gesture classification from RGB-D video using deep learning [34]. Boosting methods, based on ensembles of weak classifiers, allow multi-class hand detection [35]. A gesture-based interface based on EMG and IMU sensing report the classification of 16 discrete hand gestures which are mapped to robot commands [12]. This was materialized in point-to-goal commands and a virtual joystick for robot teleoperation. A challenging study deals with a situation in which users need to manually

control the robot but both hands are not available (when users are holding tools or objects in their hands) [23]. In this scenario, hand, body and elbow gestures are recognized and used to control primitive robot motions. Gestures can also be used to specify the relevant action parameters (e.g. on which object to apply the action) [36]. The study refers that according to the experiments with 24 people the system is intuitive to program the robot, even for a robotics novice [36]. The required HRI reliability and efficiency can be achieved through a multimodal interactive process, for example combining gestures and speech [37]. Multimodal interaction has been used to interact with multiple unmanned aerial vehicles from sparse and incomplete instructions [38].

Gesture recognition associated to HRI is today an important research topic. However, it faces important challenges such as the large amount of training data required for gesture classification (especially for deep learning) and problems related with appearance of false positives and false negatives in on-line classification. Moreover, many studies approach gesture-based HRI in an isolated fashion and not as an integrated framework that includes segmentation, classification and the interface with the robot.

2 Segmentation

The segmentation of continuous data streams in static and dynamic blocks depends on several factors: (1) interaction technologies, (2) classification method (supervised or unsupervised), (3) if gestures are static, dynamic or both, (4) if the inter-gesture transitions (IGT) were previously trained or not, among other factors. Another problem is related with the difficulty to eliminate the appearance of false positives and false negatives. In the context of gesture segmentation, it can be stated that false negatives are more costly than false positives since they divide the data representing a dynamic gesture into two sections, corrupting the meaning of that gesture. False positives are more easily accommodated by the classifier, which can report that the pattern is not a trained gesture.

Real-time segmentation relies on the comparison of the current state (frame) $\mathbf{f}_i$ with the previous states, $\{\mathbf{f}_{i-1}, \ldots, \mathbf{f}_{i-\eta}\}$. We propose a method to segment a continuous data stream into dynamic and static segments in an unsupervised fashion, i.e. without previous training or knowledge of gestures and the sequence, unsegmented and unbounded [25]. The method detailed in [25] was partially implemented and customized to the specific sensor data used in this study (input data, sliding window size and thresholds). We propose establishing a feasible (optimal or not) single threshold for each motion feature using a genetic algorithm (GA) – because the performance function is non linear and non smooth – fed by a set of calibration data. The GA parameters were obtained by manual search. Gesture patterns with sudden inversions of movement direction are analyzed recurring to the available velocities and accelerations. The proposed method deals with upper body gesture motion patterns vary-

ing in scale, rate of occurrence and different kinematic constraints. A sliding window addresses the problem of spatio-temporal variability.

We consider that there is motion if there are motion features above the defined thresholds. The threshold is a vector, $\mathbf{t}_0$, with a length equal to the number of motion features chosen, p. The features obtained from a frame are represented by the vector $\mathbf{t}$. The sliding window $\mathbf{T}$ is composed of w consecutive frames of $\mathbf{t}$. At an instant i, the real-time sliding window $\mathbf{T}(i)$ is:

$$\mathbf{T}(i) = \left[\, \mathbf{t}(i-w+1) \cdots \mathbf{t}(i-1)\ \mathbf{t}(i) \,\right] \tag{1}$$

At each instant i, the w sized window slides forward one frame and $\mathbf{T}(i)$ is updated and evaluated. A static frame is only acknowledged as such if none of the motion features exceed the threshold within the sliding window. This way, we guarantee that a motion start is acknowledged with minimal delay (real-time). On the other hand, this also causes a fixed delay on the detection of a gesture end, equal to the size of the window w.

The proposed method to achieve the motion function $m(i)$ relies in the computation of the infinite norm of a vector ϱ that contains feature-wise binary motion functions:

$$m(i) = \begin{cases} 1, & \text{if } \|\boldsymbol{\varrho}\|_\infty \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where vector ϱ, for each instant of time i, is calculated by comparing the sliding window with the threshold vector:

$$\varrho_m = (\max_g |\mathbf{T}_{mg}| \geq k_s \cdot \mathbf{t}_{0_m}),\ m = 1, \ldots, p,\quad g = 1, \ldots, w \tag{3}$$

in which k_s represents a user-defined threshold sensitivity factor and $\mathbf{t}_{0_m}$ the vector of thresholds of the motion features. $\mathbf{t}_{0_m}$ is determined by an initial calibration process in which two sets of data with equal length/time are acquired: static samples (recorded with the user performing a static pose) and motion samples (recorded with the user performing slow movements that activate the selected motion features). These data are used to estimate the segmentation error caused by an arbitrary threshold vector, which is then optimized by a GA with variables bounded by their maximum and minimum value in these data, a population size of 100 and mutation rate of 0.075. The sensitivity factor is then adjusted online for each user when needed by trial and error according to the human body shaking behaviour and the speed a dynamic gesture is performed, especially for gestures with sudden inversions of movement direction.

In an ideal system, the absence of movement would be defined by null differences of the system variables between frames. Therefore, the simplest set of features that can be used for this method is the frame differences, $\triangle\mathbf{f}$, that at an instant i is given by:

$$\Delta\mathbf{f}\,(i) = \mathbf{f}\,(i) - \mathbf{f}\,(i-1) \tag{4}$$

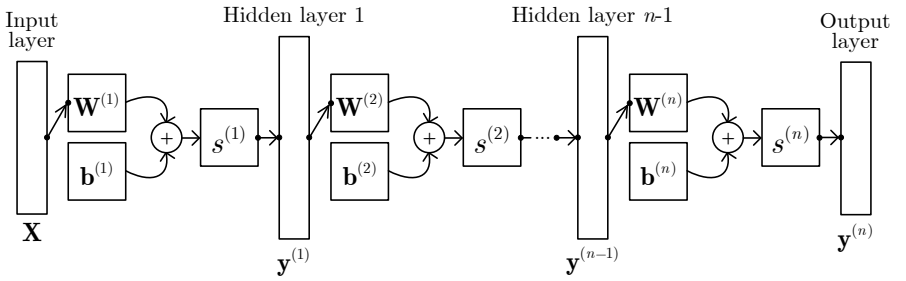

Fig. 2 Architecture of a feed-forward MLNN with n layers.

However, these features do not yield consistently reliable results. For example, if we consider as input a position in Cartesian coordinates, this approach performs poorly, since the differences would be relative to the coordinated axis. A motion pattern with a direction oblique to an axis would have lower coordinate differences compared to a pattern parallel to an axis with similar speed, thus producing different results. This issue can be solved by replacing the three coordinate differences with the respective Euclidean length, directly acquired from the IMUs angular velocity $\omega(i)$.

$$\|\boldsymbol{\omega}(i)\| = \sqrt{\omega_x(i)^2 + \omega_y(i)^2 + \omega_z(i)^2}, \quad i \in \mathbb{R}^+ \tag{5}$$

In the presence of gesture patterns with sudden inversions of direction false negatives are very detrimental to the classifier accuracy. The proposed solution is adding an extra motion feature, the acceleration, $a(i)$. The acceleration is at its highest when an inversion of direction occurs, which solves the low velocity problem. This feature does not cause false positives in a static gesture and deals successfully with the inversions of movement on dynamic gestures. The accelerations are directly acquired from the IMUs.

In summary, the features for segmentation by motion are the IMUs parameters representing motion, namely the accelerations and angular velocity. They are organized in a feature vector $\mathbf{t}$:

$$\mathbf{t}(i) = \left[\, \omega_1(i)\ a_1(i) \cdots \omega_u(i)\ a_u(i) \,\right]^T \tag{6}$$

where $\omega_u(i)$ is the angular velocity for IMU number u, and $a_u(i)$ is the acceleration for IMU number u.

2.1 Multi-Layer Neural Networks

A two-hidden-layer Multi-Layer Neural Network (MLNN) is proposed, Fig. 2. The state $y^{(q+1)}$ of each layer $(q+1)$ is defined by the state $y^{(q)}$ of the previous layer (q):

$$\mathbf{y}^{(q+1)} = f^{(q+1)}(\mathbf{y}^{(q)}) = s^{(q+1)}\left(\mathbf{b}^{(q+1)} + \mathbf{W}^{(q+1)}\mathbf{y}^{(q)}\right) \tag{7}$$

where s is the transfer function, $\mathbf{b}$ is the biases vector and $\mathbf{W}$ is the weight matrix. The estimation of $\mathbf{b}$ and $\mathbf{W}$ is obtained by training the network with samples of which we know the classification result *a priori* (training samples). Given a set of training samples $\mathbf{X}$ with known target classes $\mathbf{tg}$ (supervised learning), the objective is obtaining weights and biases that optimize a performance parameter E, e.g., the squared error $E = (t - y)^2$. The optimization is very often done with a gradient descent method in conjunction with the backward propagation of errors, method called Backpropagation (BP). Specifically, we used the Scaled Conjugate Gradient (SCG) BP method [39] which has the benefits of not requiring user-dependent parameters and of being fast to converge.

The performance function used was cross-entropy, $E_{ce} = -\mathbf{tg} \cdot \log \mathbf{y}$, which heavily penalizes very inaccurate outputs ($y \sim 0$) and penalizes very little fairly accurate classifications ($y \sim 1$). This is valid assuming a *softmax* transfer function was used on the last layer. A log-sigmoid function is also often used $s_{logsig}(x) = 1/1+e^{-x},\ s \in [0, 1]$.

BP is an iterative method that relies on the initialization (often done randomly) of the weight and bias vector, $\tilde{w}_1$ ($k = 1$). The next step is determining the search direction $\tilde{p}_k$ and step size α_k so that $E(\tilde{w}_k + \alpha_k) < E(\tilde{w}_k)$. This leads to the update of $\tilde{w}_{k+1} = \tilde{w}_k + \alpha_k \tilde{p}_k$. If the first derivative $E'(\tilde{w}_k \neq \tilde{0})$, meaning that we are not yet at a minimum/maximum, then a new iteration is made ($k = k + 1$) and a new search direction is found. Else, the process is over and $\tilde{w}_k$ should be returned as the desired minimum. BP variations typically rely on different methods to find $\tilde{p}_k$, determination of α_k or new terms to the weight update equation. This often leads to the introduction of user-defined parameters that have to be determined empirically.

2.2 Feature Dimensionality Reduction and Regularization

For the SG no dimensionality reduction is proposed, since the feature space is still small. To solve the issue of undetermined feature size of the DG, we propose re-sampling with bicubic interpolation. It allows to transform a DG sample $\mathbf{X}^{(i)},\ i \in i^D,\ \mathbf{X} \in \mathbb{M}^{d \times \eta}$, which has a variable number of frames η, into a fixed-dimension sample $\mathbf{X}',\ \mathbf{X}' \in \mathbb{M}^{d \times \eta'}$. Usually $\eta' \geq \eta$, being η' arbitrarily defined as the maximum η in all the samples i so that $i \in i^D$. So, although in almost every case the proposed transformation is up-sampling the sample, it is also valid for new cases where $\eta' < \eta$, effectively down-sampling the sample.

$$\begin{aligned} interp: \quad & \Re^{d \times \eta} \quad \rightarrow \Re^{d \times \eta'} \\ & \mathbf{X} \qquad\ \rightarrow \mathbf{X}' \end{aligned} \tag{8}$$

3 Robotic Task Manager

The gesture recognition acts in parallel with the called Parametrization Robotic Task Manager (PRTM), which is used to parametrize and manage robotic tasks

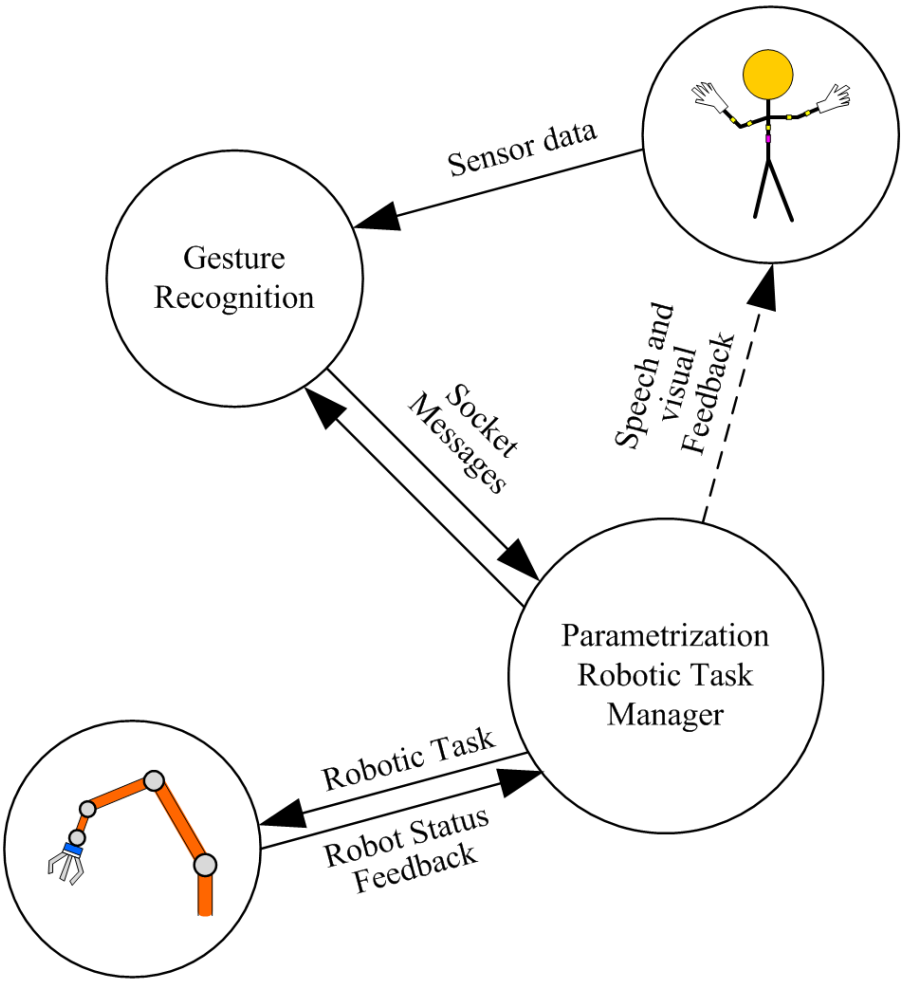

Fig. 3 Control architecture highlighting the central role of the PRTM. The PRTM receives information from the gesture recognition system and sends commands to the robot. In addition, the PRTM manages the feedback provided to the human co-worker.

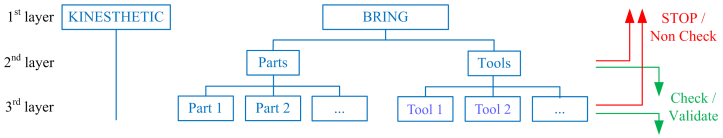

Fig. 4 The three layers of the proposed PRTM. The BRING and KINESTHETIC options are in the first layer. For the BRING option we have in the second layer two options to select PARTS and TOOLS. In the third layer we have all the parts and tools available to be selected.

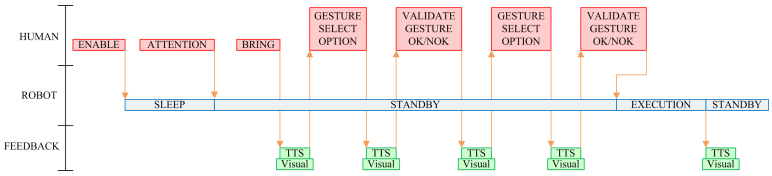

Fig. 5 BRING architecture with the role of the human co-worker, robot and feedback.

with the human co-worker in the loop, Fig. 3. Additionally, PRTM is used to provide speech feedback to the user through computer text-to-speech (TTS), and visual feedback using a monitor. The gesture recognition has implemented the methods presented in previous sections such as data sensory acquisition, raw data processing, segmentation, and static and dynamic gesture classification. The communication between PRTM and the gesture classification module is achieved by using sockets TCP/IP. The PRTM communicates with the robot through Ethernet.

When a gesture (static or dynamic) is recognized a socket message is sent to the PRTM with information about the recognized gesture. It works as a phone auto attendant providing options to the human (speech feedback) which selects the intended robot service using gestures. The proposed PRTM includes in the first layer 2 options, BRING and KINESTHETIC, Fig. 4. The BRING option refers to the ability of the robot to deliver parts, tools, and consumables to the human co-worker, while the KINESTHETIC is related with the operation mode in which the co-worker can physically guide the robot to the desired poses in space to teach a specific task or to hold a part while he/she is working on it. In the second layer, and for the BRING option, the user can select Tools or Parts, with different possibilities in each one (third layer). The BRING functionalities and operation actions related with the human co-worker, robot and user feedback are detailed in Fig. 5. The robot poses were previously defined using teach-in programming, i.e., moving the robot end-effector to the target poses and saving them.

The interactive process starts with the user performing a gesture called “Attention”. This gesture makes the system to know that the user wants to perform a given robotic task parametrization. The speech and visual feedback informs the human user about the selection options in the first layer. The user has few seconds (a predefined time) to perform a “Select” gesture to select the desired option. After this process, the PRTM through images and text displayed in the monitor and TTS asks the user to validate the selected option with a “Validation” gesture. If validated the PRTM goes to the next layer, if not validated the system continues in the current layer. If the user does not perform the “Select” gesture during the predefined time period, the PRTM continues with the other options within the layer. The procedure is repeated until the user selects one of the options or until the PRTM through TTS repeats all of the options three times. The process is similar for the second and third layer. In the third layer the PRTM sends a socket message to the robot to perform the parametrized task. If required, at any moment the user

can perform the "Stop" gesture so that the system returns to initial layer and the robot stops.

The above interactive process consumes a significant amount of time. In response to this problem, the PRTM can be setup with the pre-established sequence of operations so that the human intervention resumes to accept or not the PRTM suggestions in some critical points of the task being performed. The pros and cons of this mode of operation are discussed in the Experiments and Results section.

4 Experiments and Results

4.1 Setup and Data Acquisition

Five IMUs and a UWB positioning system were used to capture the human upper body shape and position in space, respectively, Fig. 6. The collaborative robot is a KUKA iiwa with 7 DOF equiped with the Sunrise controller.

The 5 IMUs (Technaid Tech-MCS) are composed by 3 axis acelerometers, magnetometers and gyroscope. The IMUs are synchronized in the Technaid Tech-MCS HUB and an extended Kalman filter is applied to fuse sensor data to estimate IMUs orientation Euler angles α, β and γ. In Bluetooth connection mode and for 5 IMUs the system outputs data at 25 Hz. These data will be the input for the gesture recognition module.

The UWB (ELIKO KIO) provides the relative position of the human co-worker in relation to the robot. This information is used to define if the human is close to the robot. If the human is at less than 1 meter from the robot the interactive mode is valid. The UWB tag is in the human's pocket and the 4 anchors are installed in the working room.

The sensors are connected to a computer running MATLAB. Sensor data are captured and stored in buffers. A script reads the newest samples from the buffers and processes them. The stream of data is segmented by the motion-threshold method detailed in section II, Eq. (6), considering a sensitivity factor of 3.0, and with the following segmentation features related with the 5 IMUs:

$$\mathbf{t} = [\omega_1 \ a_1 \ \omega_2 \ a_2 \ ... \ \omega_5 \ a_5]^T \tag{9}$$

Concerning the classification features, a full frame of data from the IMUs is represented by $\mathbf{f}$, Eq. (10), namely the IMUs accelerations and Euler angles in a total of 30 DOF. These features represent almost all representative data from IMUs and were selected by manual search. A binary segmentation variable m, Eq. (2), represents whether the frame belongs to a dynamic segment or not.

$$\mathbf{f} = [a_{x1} \ a_{y1} \ a_{z1} \ ... \ a_{z5} \ \alpha_1 \ \beta_1 \ \gamma_1 \ ... \ \gamma_5 \ m] \tag{10}$$

Where a_{xh}, a_{yh} and a_{zh} represent the accelerations (including gravity effect) from IMUh with $h = 1, ..., 5$ along the coordinated axis of IMUh. The Euler angles α_h, β_h and γ_h are relative to IMUh with $h = 1, ..., 5$. The frames $\mathbf{f}$ are

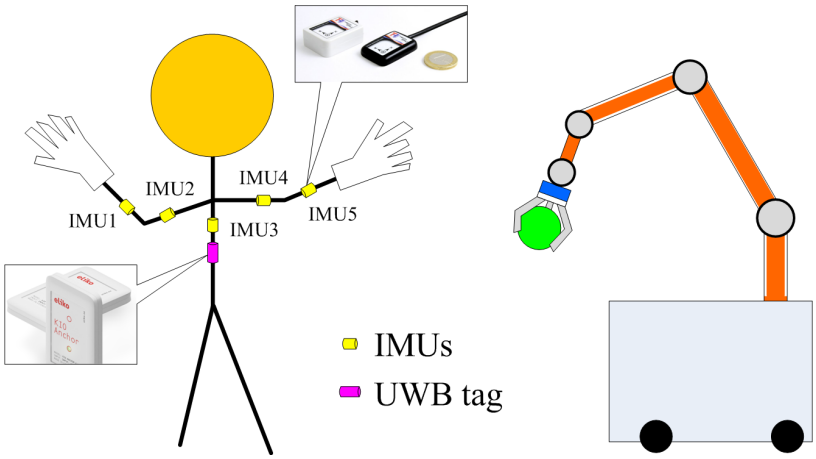

Fig. 6 Wearable sensors applied for the proposed HRI interface, 5 IMUs and a UWB tag.

arranged in static and dynamic samples according to the segmentation output m.

4.2 Gesture Data Set

According to the functionalities to be achieved and industry feedback, a dataset of continuous static/dynamic gestures is used. It contains 8 SG, Fig. 7, and 4 DG, Fig. 8. These gestures are composed by upper body arm data captured from IMUs, Eq. (10). Industry feedback was provided by production engineers from automotive sector and by two shop floor workers that experienced the system. They indicated that the number of gestures to be memorized by the robot co-workers should be relatively small, the co-workers should be able to customize each gesture to a given robot functionality, error in gesture classification should not cause a safety problem or to be detrimental to the work being done, and the co-workers should have feedback about the process (for example they need to know if the robot is moving or is waiting for a command). They selected these gestures from a library of possible gestures we provided. To avoid false positives/negatives, we implemented what we call composed gestures, which are a mix of the SGs and DGs mentioned above. The composition of a composed gesture can be customized by each different user according to the following rules: (1) the composed gesture begins with a static pose with the beginning of a selected dynamic gesture B-DG, (2) a DG, (3) a static pose with the end of the dynamic gesture E-DG, (4) an inter-gesture transition (IGT), and (5) a SG. Three examples of composed gestures are detailed in Fig. 9. However, several other combinations may be selected/customized by each different user.

The training samples for SGs $S^{(\mathbf{i}^S)}$ and DGs $S^{(\mathbf{i}^D)}$ were obtained from two different subjects, subjects A and B (60 samples each subject for each gesture

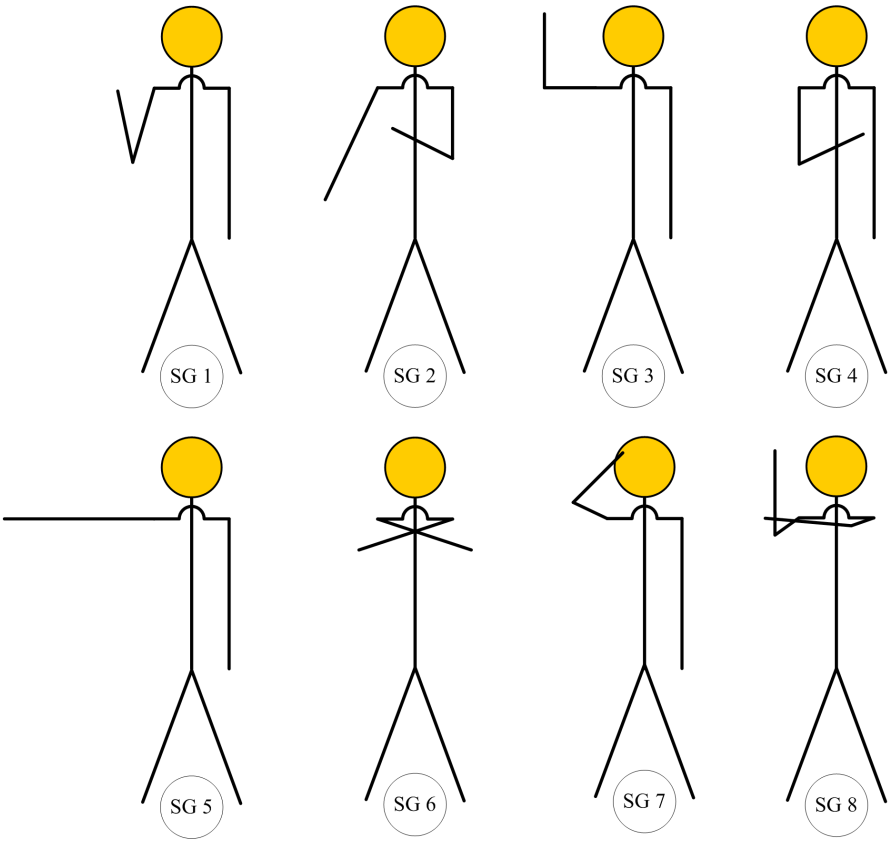

Fig. 7 Representation of the 8 SGs.

(8 SGs and 4 DGs) in a total of 720 trained patterns). These two subjects participated in the development of the proposed framework.

4.3 Features

For the SG, $m = 0$, the features for classification are all the elements of $\mathbf{f}$, excluding m. The notation for the ith-SG feature vector is $\mathbf{z'}^S \in \mathbb{R}^{30}$:

$$\mathbf{z'}^S = (a_{x1}\ a_{y1}\ a_{z1}\ ...\ a_{z5}\ \alpha_1\ \beta_1\ \gamma_1\ ...\ \gamma_5) \tag{11}$$

For DG, the features will be derived from $\mathbf{f}$, Eq. (11), namely the unit vectors representing the human arms orientation. Each arm will be described by two rigid links, with a unit vector representing each, arm $\mathbf{o}_{\mathrm{a}}(i) = (o_{ax}, o_{ay}, o_{az})_i$ and forearm $\mathbf{o}_{\mathrm{f}}(i) = (o_{fx}, o_{fy}, o_{fz})_i$. From the Euler angles of each IMU we can define the spherical joints between each two sensors, such that we get three

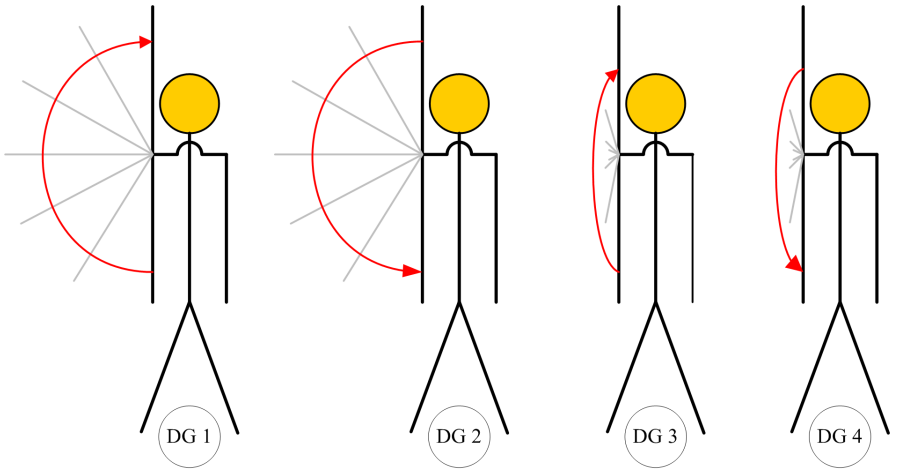

Fig. 8 Representation of the 4 DGs.

orthogonal rotation angles between each two sensors. From these we can construct the direct kinematics for each arm of the human body and obtain the unit vectors, Fig. 10. The notation for the ith-DG feature vector is $\mathbf{z}'^{D} \in \mathbb{R}^{12}$:

$$\mathbf{z}'^{D} = (\mathbf{o}_{\mathrm{a1}}\ \mathbf{o}_{\mathrm{a2}}\ \mathbf{o}_{\mathrm{f1}}\ \mathbf{o}_{\mathrm{f2}}) \tag{12}$$

Each DG, including gestures in the same class, normally have a variable number of frames. For classification purposes, we need to establish a fixed dimension for all DGs, recurring to bicubic interpolation as detailed in previous section. Given a sample $\mathbf{X}^{(i)} : i \in i^{D}$ with η frames ($\mathbf{X}^{(i)} \in \mathbb{M}^{12\times\eta}$), the objective is to resample it to a fixed size η'. The value for η' can be chosen arbitrarily but higher values have a detrimental effect on the classification accuracy. For that reason, η' should have an upper bound such that $\eta' \leq \eta,\ \forall\eta|\mathbf{X}^{(\mathbf{i}^{D})} \in \mathbb{M}^{12\times\eta}$. For the proposed gesture dataset, the gesture length varies between 42 and 68 frames. Therefore, we choose the lowest η of the the DG samples, $\eta' = 42$. Applying the bicubic interpolation, the result is a matrix $\mathbf{Z} \in \mathbb{R}^{12\times42}$. Fig. 11 shows an example of gesture data before and after compression and regularization for DG 2 (length reduced from 48 to 42 frames). It is visible that the data significance is maintained. By concatenating every frame vertically, $\mathbf{Z}$ is transformed into a vector $\mathbf{z} \in \mathbb{R}^{504}$:

$$\mathbf{z}^{(i)} = \mathrm{concat}(\mathbf{Z}^{(i)}) = \begin{pmatrix} \mathbf{Z}^{(i)}_{\bullet 1} \\ \vdots \\ \mathbf{Z}^{(i)}_{\bullet 42} \end{pmatrix} \tag{13}$$

The last feature processing step is feature scaling. It is essential for achieving smaller training times and better network performance with less samples. It harmonizes the values of different features so that all of them fall within

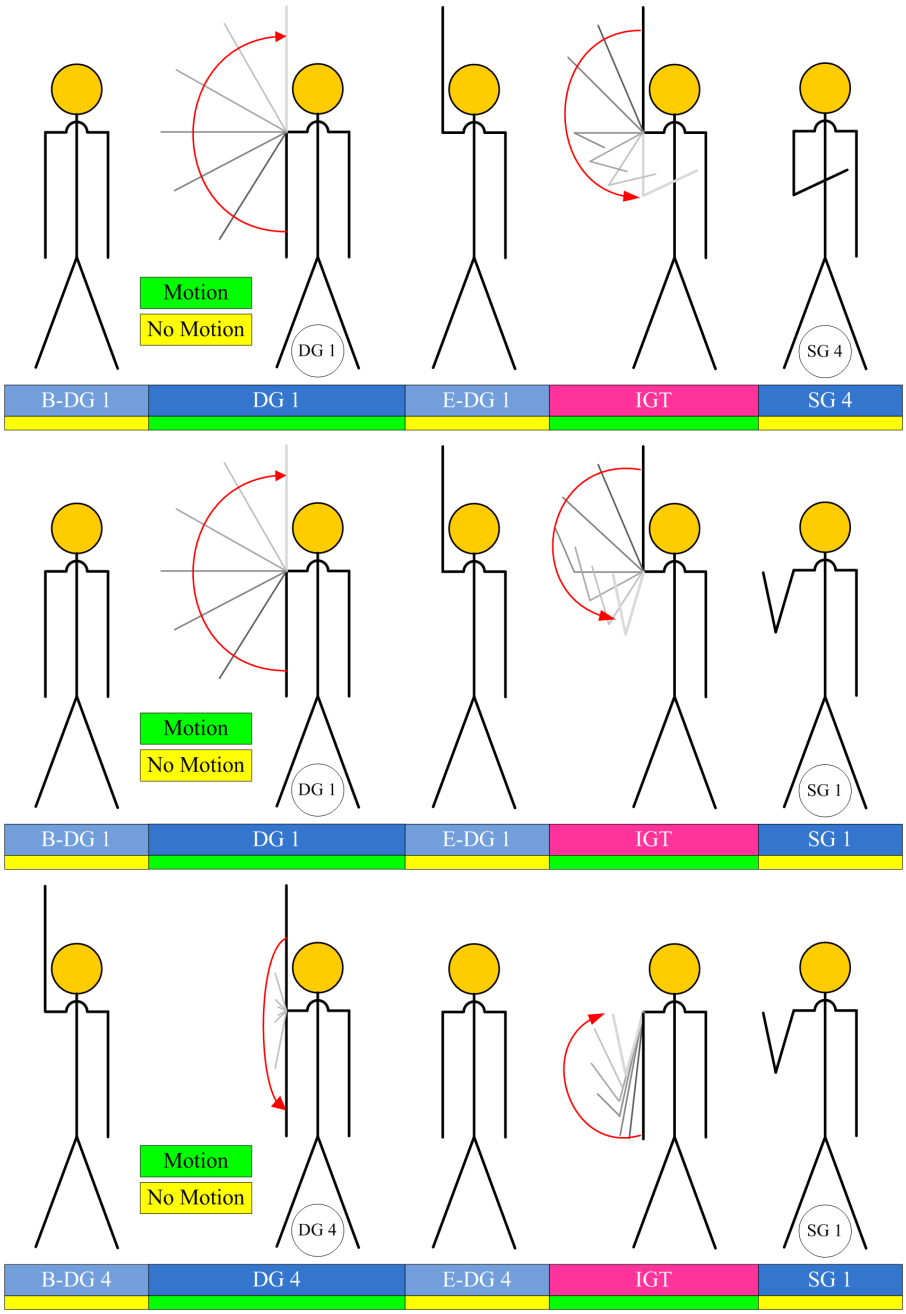

Fig. 9 Example of 3 composed gestures. B-DG indicates the beginning of a DG, E-DG indicates the end of a DG and IGT the inter-gesture transition between gestures.

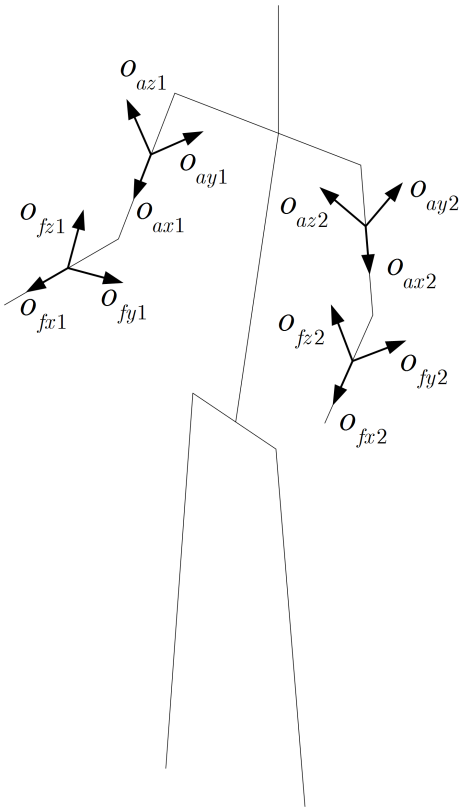

Fig. 10 Human arms described by 2 unit vectors each (representing orientation of arm and forearm).

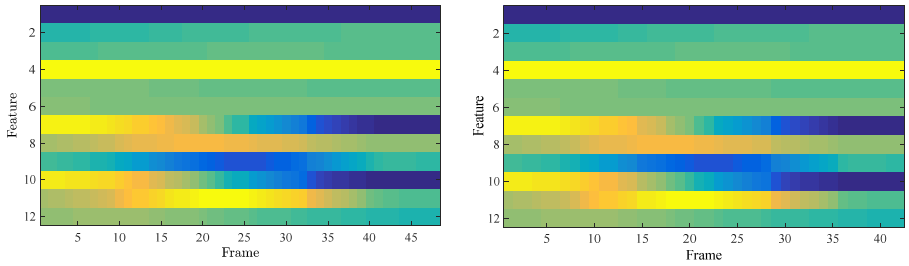

Fig. 11 DG 2 gesture data before (at left) and after compression and regularization (at right).

the same range. This is especially important when some features have distinct orders of magnitude. Applying linear rescaling, l:

$$l(\mathbf{x}) = \frac{2\mathbf{x} - \widehat{\mathbf{X}}^T}{\widehat{\mathbf{X}}^T} \tag{14}$$

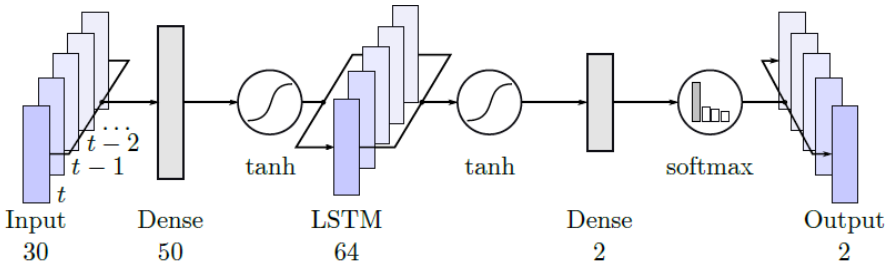

Fig. 12 LSTM network architecture.

where $\widehat{\ }$ is the max+min operator defined in Eq. (15). $\mathbf{X}^T = \left(\cup \mathbf{z}^{(i)} : \ i \in i^T\right)$ is the set of unscaled features of the training set. This operator is valid both for static and dynamic gestures but the sample subsets used should be exclusive.

$$\widehat{\mathbf{X}}_i = \max \mathbf{X}_{i\bullet} + \min \mathbf{X}_{i\bullet}, \quad i = 1, ..., d \tag{15}$$

4.4 Results and Discussion: Gesture Recognition

Experiments were conducted to verify the performance and effectiveness of the proposed framework. It was tested by two subjects (subject A and B) that contributed to the development of the system and created the gesture training data set, and five subjects (subject C, subject D, subject E, subject F and subject G) that are not robotics experts and are using the system for the first time. Subjects F and G are automotive plant workers with 25-30 years old and with expertise in the assembly of components for gear boxes. For the testing dataset, each subject performed each SG 60 times (for the 8 SGs we have a total of 480 testing patterns for each subject) and each DG 60 times (for the 4 DGs we have a total of 240 testing patterns for each subject).

The proposed solution for gesture segmentation aims to accurately divide a continuous data stream in static and dynamic segments. The conducted experiments consisted in the analysis of samples containing sequences of static and dynamic behaviours. For each subject, ten composed gestures were analysed, each with 2 motion blocks and 3 static blocks, Fig. 9.

Segmentation performance depends largely on the size of the sliding window. The segmentation accuracy was measured for different sliding window sizes. Considering small sliding windows, there is excessive segmentation (over-segmentation), leading to low accuracy. The best results were achieved for a window size of 20.

The proposed unsupervised solution was compared with two supervised methods, a one-class feed-forward neural network (ANN) and a Long Short-Term Memory (LSTM) network, Fig. 12. For both networks, inputs are the sliding window data and a single output neuron outputting a motion index. They were trained with the same calibration data applied in the unsupervised

Table 1 Segmentation error (%) comparing the unsupervised proposed solution with two supervised methods.

Subject	A	B	C	D	E	F	G
Proposed solution (unsupervised)	0%	0%	8%	4%	10%	7%	9%
ANN (supervised)	0%	0%	6%	4%	6%	2%	8%
LSTM (supervised)	0%	2%	8%	4%	10%	4%	8%

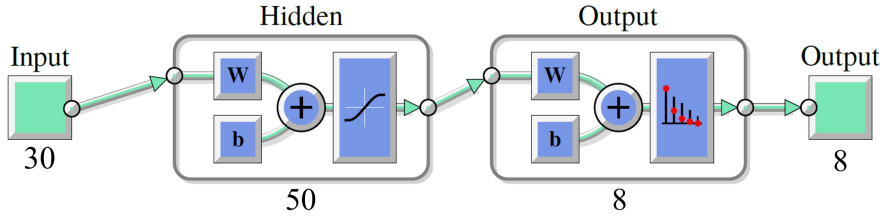

Fig. 13 ANN architecture for SG classification.

method to achieve an optimal sliding window size (data from subject A and subject B).

Table 1 shows the segmentation error results. Results indicate that the segmentation error for the supervised methods (ANN and LSTM) is identical to the proposed unsupervised solution. For subjects A and B the segmentation error is almost zero, justified by the fact that they tested a system calibrated/trained with data they produced. The error detected for the other subjects (C, D, E, F and G) is mainly due to oversegmentation. Generally, oversegmentation occurs in the IGT phase and is not critical for the classification. The proposed unsupervised method is effective, especially if calibrated (threshold parameters) with data from the user.

Concerning SG classification, $\S^{(i)}$, $i \in \mathbf{i}^S$, 60 samples from subject A and B were used for the training set ($i \in \mathbf{i}^{ST}$) and 60 samples from subjects A, B, C, D, E, F and G for the validation set ($i \in \mathbf{i}^{SV}$). The validation set is not used for optimization purposes. The loss of the network on this set is monitored during optimization, which is halted when this loss stops decreasing in order to prevent overfitting.

The MLNN architecture for SG classification, Fig. 13, has 30 neurons in the input layer, which is the size of the SG feature vector (Eq. 11). Also, it is composed by one hidden layer with 50 neurons, having the hyperbolic tangent as the transfer function. The output layer has 8 nodes, the number of classes, with the *softmax* function as transfer function.

The accuracy results, Table 2, indicate an overall classification accuracy on the testing set for subject A of 99.0% (475/480) and for subject B of 98.50% (473/480). For subjects C, D, E, F and G that did not train the system the accuracy was reduced, Table 2. SG1 was mistaken with SG3, which are very similar gestures if the user is not positioning the right arm correctly.

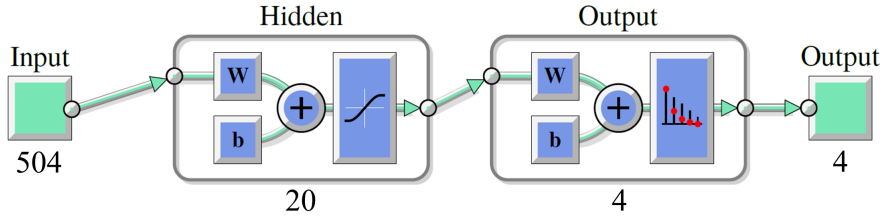

Fig. 14 ANN architecture used for DG classification.

Table 2 Classification accuracy for both static and dynamic gestures.

Subject	A	B	C	D	E	F	G
SGs (proposed ANN)	99.0%	98.5%	94.2%	93.5%	89.6%	95.0%	90.1%
DGs (proposed ANN)	99.5%	99.0%	95.8%	92.5%	94.6%	96.9%	95.4%
SGs (SVM)	98.6%	97.6%	92.4%	88.3%	84.3%	89.5%	87.5%
DGs (SVM)	98.2%	97.4%	92.1%	88.7%	91.1%	92.1%	88.1%

For DG classification $\S(i) : \ i \in \mathbf{i}^D$, the training set is composed by 60 samples from subject A and B ($i \in \mathbf{i}^{DT}$) and 60 samples from subjects A, B, C, D, E, F and G for the validation set ($i \in \mathbf{i}^{DV}$). The network architecture, Fig. 14, has 504 input neurons, one hidden layer with 20 neurons and the output layer has 4 output neurons, the transfer function is the hyperbolic tangent in the first layer and the *softmax* function in the last layer.

The gesture classification accuracy, Table 2, shows a good accuracy for subjects A and B. Even for subject C, D, E, F and G that did not train the system the accuracy is relatively high. These good results are due to the relatively small number of DG classes. It should be noted that the model was not trained with data from subject C, D,E, F and G and no calibration was performed.

For the composed gestures, the accuracy is directly related with the accuracy of the SGs and DGs.

Deep learning algorithms require a large number of training data, being more suitable for the classification of images and sequences of images. The results we obtained with the proposed ANN-based classification solution are satisfactory, especially considering that we have few training data from wearable sensors and only from two subjects. Nevertheless, the results are acceptable for subjects C, D, E, F and G, and excellent for the subject A and B. In this context, we compared the proposed MLNN method with a common classification method, SVM. The SVM was not optimized. We tried different SVM methods recurring to the MATLAB Classification Learner, obtaining the best results with the Medium Gaussian SVM with a Gaussian kernel function. Results indicate that SVM method presents interesting results but compares unfavourably with proposed MLNN method, Table 2. The results for subjects F and G are in line, or even better, when compared with the results for the other three subjects that did not train the system. This can be justified by

Fig. 15 Human-robot collaborative process. In this use case the robot delivers tools and parts to the human co-worker (top and middle) and the robot holds the workpiece while the co-worker is working on it (bottom). For better ergonomics the co-worker adjusts the workpiece position and orientation through robot hand-guiding. The monitor that provides visual feedback to the user is also represented, indicating the task being performed, the next task and asking for human intervention id required in each moment.

the fact that these workers are relatively young (average age of 25 years old) and familiarized with information and communications technologies (ICT).

4.5 Results and Discussion: Robot Interface

The collaborative robot acts as a "third hand" by assisting the human co-worker in an assembly operation by delivering to the human shared workplace tools, parts, and holding work pieces, Fig. 15. After a gesture is recognized it serves as input for the PRTM that interfaces with the robot and provides speech and visual feedback to the human co-worker (section IV), Fig. 3.

The framework was tested by the seven subjects mentioned above. Subjects C, D, E, F and G received a 15 minutes introduction to the system by subjects A and B that contributed to the system development and created the gesture dataset. From the library of 8 SGs and 4 DGs the seven subjects chose the gestures that best suited them to associate with the PRTM commands: "attention", "select", "validation", "stop", "abort" and "initialize" (according to the functionalities detailed in section IV). Finally, subjects C, D,E, F and G were briefed on the assembly sequence and components involved.

The complete assembly task is composed of subtasks: manipulation of parts, tools, consumables, holding actions and screw. Some tasks are more suited to be executed by humans, others by robots, and others by the collaborative work between human and robot. When requested by the human co-worker (using gestures), the robot has to deliver to the human workplace

the parts, consumables (screws and washers) and tools for the assembly process. The parts and tools are placed in know fixed positions. Moreover, the human can setup the robot in kinesthetic precision mode [40] to manually guide it to hold workpieces while tightening the elements, Fig. 15. Although the gestures recognition rate is high, the occurrence of false positives and negatives was analysed. Our experiments demonstrated that if a given gesture is wrongly classified the "validation" procedure allows the user to know from the speech and visual feedback that it happened, so that he/she can adjust the interactive process.

The collaborative activities may present the risk of potential collisions between human and robot. From the UWB positional data, when a threshold separation distance is reached the robot stops. The estimation of the separation distance contemplates the velocity and reach of both robot and human (dimensions of the human upper limbs), and the UWB error (about 15 cm). In our experiments we considered a separation distance of 1 meter. This is valid when the robot is delivering the tools and consumables to the human co-worker. The robot is also performing these actions with a velocity according to safety standards so that this stop operation is not mandatory. For the kinesthetic teaching the separation distance is not considered. During the interactive process, the reached target points can be saved and used in future robot operations. The impedance controlled robot compensates positioning inaccuracies, i.e., the co-worker can physically interact with the robot (kinesthetic mode) to adjust positioning.

On average, the time that passes between the recognition of a gesture and the completion of the associate PRTM/robot command is about 1 second. If the setup of the PRTM is taken into account, with the selection of the desired options, it takes more than 5 seconds.

The seven subjects filled a questionnaire about the proposed interface, resulting in the following main conclusions:

1. The gesture-based interface is intuitive but delays the interactive process. It can be complemented with a tablet to select some robot options faster;
2. It was considered by all the subjects that the "validation" procedure slows the interactive process. The subjects F and G indicated that this is discouraging from an industrial point of view. Nevertheless, they indicated that the problem is attenuated when we setup a given sequence in the PRTM avoinding the validations;
3. The shop floor workers (subject F and G) indicated that the main concerns they have are the safety (emergency buttons recommended) and the need to make the interactive process as simple as possible. They adapted easily to the system but pointed that this can be a difficult task for older workers. At this stage we can assume that mainly these systems have to be operated by young workers familiarized with basic ICT technologies;
4. Operating a version of the PRTM without all the validations proved to be faster. Nevertheless, the system presents lower flexibility, i.e., requires an

initial setup of the task sequence so that the human intervention resumes to accept or not the PRTM suggestions with the NEXT command;
5. The composed gestures are more complex to perform compared to SGs and DGs. Nevertheless, they are more reliable than SGs and DGs;
6. The automatic speech and visual feedback is considered essential for a correct understanding of the interactive process, complementing each other;
7. The subjects that were not familiarized with the system (subjects C, D, E, F and G) considered that working with the robot without fences present some degree of danger (they did not feel totally safe). The industry workers indicate the need of one or several emergency buttons placed close to the robotic arm;
8. All subjects reported that the proposed interface allows the human co-worker to abstract from the robot programming, save time in collecting parts and tools for the assembly process, and have better ergonomic conditions by adjusting the robot as desired. The ergonomics factor was reinforced from subjects F and G from industry.

The task completion time was analysed for the presented assembly use case. The task completion time of the collaborative robotic solution (eliminating the validation procedures) is about 1.4 times longer than when performed by the human worker alone. The collaborative robotic solution is not yet attractive from an economic perspective and needs further research. This result is according to similar studies that report that the collaborative robotic solutions are more costly in terms of cycle time than the manual processes [41]. Nevertheless, the system demonstrated to be intuitive to use and with better ergonomics for the human.

5 Conclusion and Future Work

This paper presented a novel gesture-based HRI framework for collaborative robots. The robot assists a human co-worker by delivering tools and parts, and holding objects to/for an assembly operation. It can be concluded that the proposed solution accurately classifies static and dynamic gestures, trained with a relatively small number of patterns, and with an accuracy of about 98% for a library of 8 SGs and 4 DGs. These results were obtained having IMUs data as input, unsupervised segmentation by motion and a MLNN as classifier. The proposed parameterization robotic task manager (PRTM) demonstrated intuitiveness and reliability managing the recognised gestures with robot action control and speech/visual feedback.

Future work will be dedicated to testing the proposed solution with other interaction technologies (vision) and adapt the PRTM to be easier to setup a novel assembly task. In addition we will perform more tests with industry workers.

Acknowledgement

This work was supported in part by the Portuguese Foundation for Science and Technology (FCT) project COBOTIS (PTDC/EME-EME/32595/2017), and the Portugal 2020 project DM4Manufacturing POCI-01-0145-FEDER- 016418 by UE/FEDER through the program COMPETE2020.

References

1. G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood, "The grand challenges of science robotics," *Science Robotics*, vol. 3, no. 14, 2018.
2. L. Johannsmeier and S. Haddadin, "A hierarchical human robot interaction planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, Jan 2017.
3. B. Sadrfaridpour and Y. Wang, "Collaborative assembly in hybrid manufacturing cells: An integrated framework for human-robot interaction," *IEEE Transactions on Automation Science and Engineering*, vol. PP, no. 99, pp. 1–15, 2017.
4. K. Kaipa, C. Morato, J. Liu, and S. Gupta, "Human-robot collaboration for bin-picking tasks to support low-volume assemblies." Robotics Science and Systems Conference, 2014.
5. E. Matsas, G.-C. Vosniakos, and D. Batras, "Effectiveness and acceptability of a virtual environment for assessing human–robot collaboration in manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 92, no. 9, pp. 3903–3917, Oct 2017.
6. I. E. Makrini, K. Merckaert, D. Lefeber, and B. Vanderborght, "Design of a collaborative architecture for human-robot assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 1624–1629.
7. T. Ende, S. Haddadin, S. Parusel, T. Wüsthoff, M. Hassenzahl, and A. Albu-Schäffer, "A human-centered approach to robot gesture based communication within collaborative working processes." IROS 2011, 25-30 Sept. 2011, San Francisco, California.
8. S. Sheikholeslami, A. Moon, and E. A. Croft, "Cooperative gestures for industry: Exploring the efficacy of robot hand configurations in expression of instructional gestures for human-robot interaction," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 699–720, 2017.
9. P. Rouanet, P. Oudeyer, F. Danieau, and D. Filliat, "The impact of human?robot interfaces on the learning of visual objects," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 525–541, April 2013.
10. S. Radmard, A. J. Moon, and E. A. Croft, "Interface design and usability analysis for a robotic telepresence platform," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug 2015, pp. 511–516.
11. M. Simao, P. Neto, and O. Gibaru, "Natural control of an industrial robot using hand gesture recognition with neural networks," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 5322–5327.
12. M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, "Gesture-based robot control with variable autonomy from the JPL BioSleeve," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, may 2013, pp. 1160–1165.
13. P. Neto, D. Pereira, J. N. Pires, and a. P. Moreira, "Real-time and continuous hand gesture spotting: An approach based on artificial neural networks," *2013 IEEE International Conference on Robotics and Automation*, pp. 178–183, 2013.
14. B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry intuitive human-robot communication from human observation," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2013, pp. 349–356.
15. S. Goldin-Meadow, "The role of gesture in communication and thinking," *Trends in Cognitive Sciences*, vol. 3, no. 11, pp. 419 – 429, 1999.

16. R. S. Feldman, *Fundamentals of Nonverbal Behavior*. Cambridge University Press, 1991.
17. S. Waldherr, R. Romero, and S. Thrun, "A Gesture Based Interface for Human-Robot Interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151–173, sep 2000.
18. C. P. Quintero, R. T. Fomena, A. Shademan, N. Wolleb, T. Dick, and M. Jagersand, "Sepo: Selecting by pointing as an intuitive human-robot command interface," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 1166–1171.
19. M. Burke and J. Lasenby, "Pantomimic gestures for human–robot interaction," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1225–1237, Oct 2015.
20. Y. Okuno, T. Kanda, M. Imai, H. Ishiguro, and N. Hagita, "Providing route directions: Design of robot's utterance, gesture, and timing," pp. 53–60, 2009.
21. M. Salem, S. Kopp, I. Wachsmuth, K. Rohlfing, and F. Joublin, "Generation and Evaluation of Communicative Robot Gesture," *International Journal of Social Robotics*, vol. 4, no. 2, pp. 201–217, feb 2012.
22. C.-M. Huang and B. Mutlu, "Modeling and evaluating narrative gestures for humanlike robots," in *In Proceedings of Robotics: Science and Systems*, 2013, p. 15.
23. M. Wongphati, H. Osawa, and M. Imai, "Gestures for manually controlling a helping hand robot," *International Journal of Social Robotics*, vol. 7, no. 5, pp. 731–742, 2015.
24. Z. Shao and Y. Li, "Integral invariants for space motion trajectory matching and recognition," *Pattern Recognition*, vol. 48, no. 8, pp. 2418 – 2432, 2015.
25. M. A. Simao, P. Neto, and O. Gibaru, "Unsupervised gesture segmentation by motion detection of a real-time data stream," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2016.
26. J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation." *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1685–99, sep 2009.
27. R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming." *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 462–77, mar 2010.
28. B. Burger, I. Ferrané, F. Lerasle, and G. Infantes, "Two-handed gesture recognition and fusion with speech to command a robot," *Autonomous Robots*, vol. 32, no. 2, pp. 129–147, dec 2011.
29. V. Villani, L. Sabattini, G. Riggio, C. Secchi, M. Minelli, and C. Fantuzzi, "A natural infrastructure less human robot interaction system," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1640–1647, July 2017.
30. D. Wu, L. Pigou, P. J. Kindermans, N. LE, L. Shao, J. Dambre, and J. M. Odobez, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2016.
31. F. J. Ordonez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.
32. M. Field, D. Stirling, Z. Pan, M. Ros, and F. Naghdy, "Recognizing human motions through mixture modeling of inertial data," *Pattern Recognition*, vol. 48, no. 8, pp. 2394 – 2406, 2015.
33. Y. Song, D. Demirdjian, and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction," *ACM Transactions on Interactive Intelligent Systems*, vol. 2, no. 1, pp. 1–28, mar 2012.
34. C. Monnier, S. German, and A. Ost, *A Multi-scale Boosted Detector for Efficient and Robust Gesture Recognition*. Springer International Publishing, 2015, pp. 491–502.
35. K. Mei, J. Zhang, G. Li, B. Xi, N. Zheng, and J. Fan, "Training more discriminative multi-class classifiers for hand detection," *Pattern Recognition*, vol. 48, no. 3, pp. 785 – 797, 2015.
36. M. R. Pedersen and V. Krüger, "Gesture-based extraction of robot skill parameters for intuitive robot programming," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 149–163, 2015.

37. S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno, "An extensible architecture for robust multimodal human-robot communication," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 2208–2213.
38. J. Cacace, A. Finzi, V. Lippiello, M. Furci, N. Mimmo, and L. Marconi, "A control architecture for multiple drones operated via multimodal interaction in search rescue mission," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2016, pp. 233–239.
39. M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *NEURAL NETWORKS*, vol. 6, no. 4, pp. 525–533, 1993.
40. M. Safeea, R. Bearee, and P. Neto, *End-Effector Precise Hand-Guiding for Collaborative Robots*. Cham: Springer International Publishing, 2018, pp. 95–101.
41. O. Madsen, S. Bgh, C. Schou, R. S. Andersen, J. S. Damgaard, M. R. Pedersen, and V. Krger, "Integration of mobile manipulators in an industrial production," *Industrial Robot: An International Journal*, vol. 42, no. 1, pp. 11–18, 2015.