Multimodal sensor-based human-robot collaboration in assembly tasks

James Male
University of Bath
Bath, UK
jjm53@bath.ac.uk

Gorkem Anil Al
University of Bath
Bath, UK
gga31@bath.ac.uk

Arya Shabani
University of Bath
Bath, UK
as4940@bath.ac.uk

Uriel Martinez-Hernandez
University of Bath
Bath, UK
u.martinez@bath.ac.uk

***Abstract*—This work presents a framework for Human-Robot Collaboration (HRC) in assembly tasks that uses multimodal sensors, perception and control methods. First, vision sensing is employed for user identification to determine the collaborative task to be performed. Second, assembly actions and hand gestures are recognised using wearable inertial measurement units (IMUs) and convolutional neural networks (CNN) to identify when robot collaboration is needed and bring the next object to the user for assembly. If collaboration is not required, then the robot performs a solo task. Third, the robot arm uses time domain features from tactile sensors to detect when an object has been touched and grasped for handover actions in the assembly process. These multimodal sensors and computational modules are integrated in a layered control architecture for HRC collaborative assembly tasks. The proposed framework is validated in real-time using a Universal Robot arm (UR3) to collaborate with humans for assembling two types of objects 1) a box and 2) a small chair, and to work on a solo task of moving a stack of Lego blocks when collaboration with the user is not needed. The experiments show that the robot is capable of sensing and perceiving the state of the surrounding environment using multimodal sensors and computational methods to act and collaborate with humans to complete assembly tasks successfully.**

***Index Terms*—vision and touch sensing, wearable sensing, human-robot collaboration, assembly tasks**

I. Introduction

The trend of mass customisation in manufacturing has led to companies having flexible production methods where human-robot collaboration (HRC) has become a leading approach [1]. HRC systems need to be capable of sensing the surrounding environment and perceiving the state of the user to perform a natural task flow in robotic assembly tasks whilst maintaining the safety of the operator [2]. These high-level processes can be achieved reliably by the use of multimodal sensor data, advanced computational methods and multilayer control architectures. These components have been used for research and development of safe and interactive work environments between humans and robots using a variety of sensing and robotic technologies [3].

In human-robot collaborative tasks, the robot can perform collaborative actions by perceiving its surrounding environment, but also by explicit commands from the operator. Understanding the actions performed by the human allows the robot to predict the next actions required for a synchronised and efficient collaboration with the user [4]. This prediction process has been investigated in a variety of assembly tasks

This work was supported by The Engineering and Physical Sciences Research Council (EPSRC), The Republic of Turkey Ministry of National Education and the Centre for Autonomous Robotics (CENTAUR).

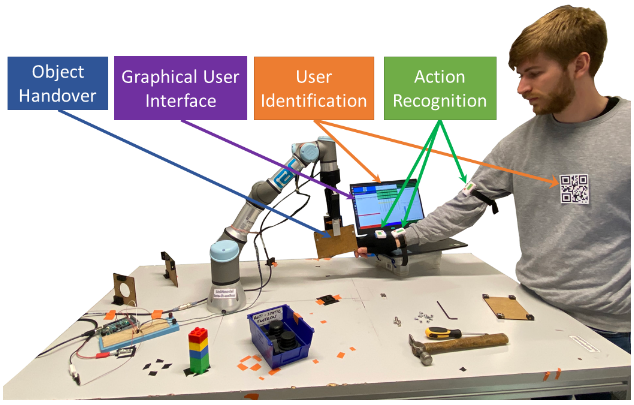

Fig. 1. Overview of collaborative system setup.

such as PCB assembly and legs onto a table [4], [5]. Explicit robot control to perform actions required by the user has been explored with the recognition of arm and hand gestures for a variety of collaborative assembly tasks [6]–[10]. User identification, based on face recognition and visual authentication methods, has been used to explicitly control the robot to personalise the speed and type of collaborative actions in manufacturing environments [11], [12].

Collaborative tasks require object handover actions, where the robot detects and grasps objects to bring them to the user. Handover actions, which involve physical contact and close interaction with objects and the human operator, require tactile sensors and perception methods [13]. This sensing modality allows robots to interact with objects and humans safely. Handover actions have been studied with a variety of tactile technologies and data processing methods in assembly tasks and human-robot interaction scenarios [14]–[16].

Robots need to be capable of processing data from multiple sensors at different levels of abstraction for better perception, learning and interaction with objects and humans [17]. This process has been investigated using layered control architectures in robotic applications such as context-aware scenarios, assistive robots and collaborative tasks [18], [19].

In this work, we present a framework for multimodal sensor-based human-robot collaboration in assembly tasks. This framework uses data from wearable inertial measurement units (IMU), tactile and vision sensors. These sensing modalities, together with Convolutional Neural Networks (CNN), authentication methods and time domain features, are employed for human action recognition (HAR), hand gesture recognition, bi-directional handover actions and user identification. These processes are implemented in a control architecture composed of perception, reasoning and control layers employed by a Universal Robot (UR3) arm to collaborate with the human operator in assembly tasks. This control architecture also allows the robot arm to perform a solo task when no collaborative actions are required by the user.

The multimodal sensor-based framework is validated with the collaborative assembly of two types of objects: 1) a box and 2) a small chair. The robot arm identifies the user and determines the required assembly task and set of collaborative actions. The robot also recognises the actions performed by the user during the assembly process, which involve screw driver, Allen key, hand screw and hammer. Explicit robot control actions are also tested with wave, forwards, backwards, left, right and stop hand gestures. The capability of the robot arm for handover actions is also tested, bringing objects to the user and grasping objects placed on the robot by the user. The processes for user identification, gesture recognition and handover actions improve the complexity of the collaborative scenario from our previous work in [20]. The experiments show that the robot arm is capable of collaborating with the human to perform different sets of actions needed to complete different assembly tasks. Overall, the results demonstrate that the proposed multimodal sensor-based framework has the potential for reliable control of robotic platforms for collaborative tasks with humans.

II. METHODS

A. Overview of System Architecture

The modules for sensing, perception, reasoning, control and memory are implemented with ROS. These modules have been extended from our previous work on action recognition in [20]. Communication between modules is managed with ROS messages and a PostgreSQL database containing tables of key information. This includes the list of actions required for each assembly task, action duration, an episodic memory of all completed actions, and the real-time predictions of how long actions have left and when the next robot collaborative action is required.

B. User identification using vision

A user identification step is included to allow the collaborative system to adapt its behaviour to the current user. This can be used as a security feature to ensure only eligible persons use the system while also offering improved robot behaviour personalised to each user. In this work, user identification allows automated selection of which assembly task the robot should perform based on which user is recognised.

The person is identified using QR code recognition. The user wears a QR code attached to their chest which encodes their name. During system initialisation, the user is instructed to look at the webcam on the control laptop allowing the QR code to be read. For demonstration purposes, each user name is preallocated with the task that they are to perform (either assembling a box or chair). Then, the robot loads the list of actions for that task from the relevant PostgreSQL database.

C. Action recognition using wearable sensors

HAR is used as the key form of environment perception to allow the robot to interact with the user. In this process, the robot must be able to track at what stage through the assembly task the user is in order to predict what actions to perform next. Hand gesture recognition is also implemented to allow natural and explicit communication with the robot when specific collaborative actions are required by the user.

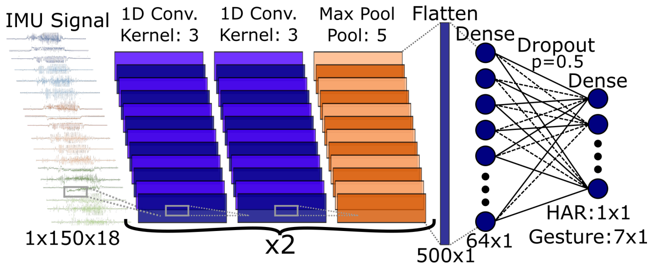

Fig. 2. CNN architecture for HAR and gesture recognition

Bodyworn IMU sensors are used for action and gesture recognition. IMU sensors are immune to occlusion and lighting conditions while also allowing tracking of fine motions, which are key requirements for use in industrial environments. Three Shimmer3 IMU sensors are placed on the user; one on the hand, wrist and arm. The 3-axis accelerometer and gyroscope signals from each sensor are streamed over Bluetooth to the user control laptop at 50 Hz. A 3 s window of data are collected, updated every 0.5 s, and used for both the HAR and gesture recognition methods.

1) Action Recognition: The HAR method uses a CNN based classifier method updated from [20]. A separate classifier is used to recognise each of four assembly actions versus null: screw driver, Allen key, hand screw and hammer. By using separate classifiers, different actions could easily be added to the system in future works without having to retrain the entire recognition process. Other benefits are improved real-time accuracy given the expected action sequence can be used to isolate which classifier is relevant, and allowing each action to have a different, optimal recognition method, illustrated here with different numbers of training epochs.

The structure of the CNN classifiers used in this work is shown in Fig. 2. The 18 channel, 150 sample long (3 s at 50 Hz) IMU data are first normalised to mean of zero and unit standard deviation based on training data. The signal is then processed by two sets of two 1D convolution layers and a 1D max-pooling followed by a fully connected layer, dropout and output unit. All convolution kernels have length of 3 with 100 filters, max-pooling window lengths of 5, 64 fully connected units and 0.5 dropout. ReLU and sigmoid activation functions are used for internal layer and output units, respectively. The Adam optimiser with learning rate of 0.001 and batch size of 128 is used. Each action is trained with an optimised number of epochs as follows: screw driver 4, Allen key 7, hand screw 23 and hammer 18.

2) Gesture Recognition: Gesture recognition allows for natural and explicit communication with the robotic system, even in noisy manufacturing environments where other communication modalities such as speech is less reliable. A CNN classifier is used to recognise 7 hand gestures: wave, forwards, backwards, left, right, stop and null. The structure of the classifier is similar to that of the HAR classifiers (Fig. 2), though the output layer now has 7 units with softmax activation. The Adam optimiser is again used with batch size of 128, though learning rate set to 0.0001 for 95 epochs.

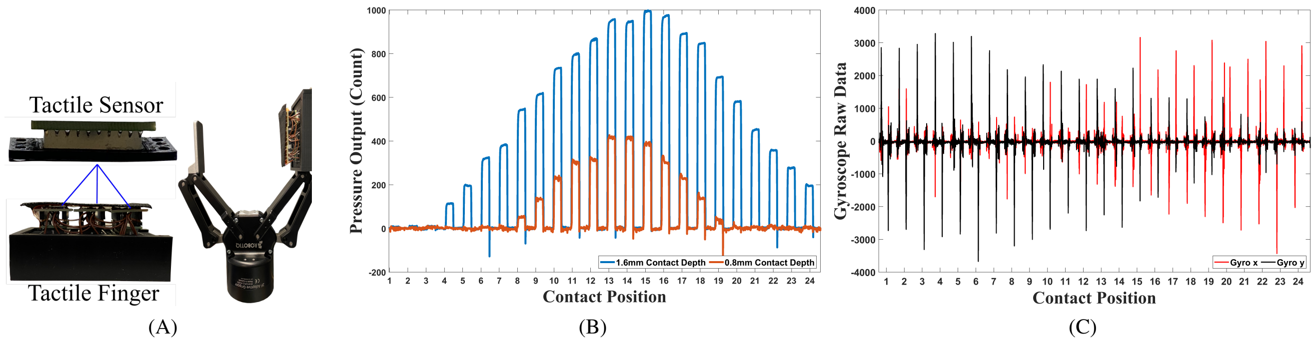

Fig. 3. (a) The tactile sensor is fabricated in a sandwich form, with three sensors used to design a tactile finger. The tactile finger is mounted to the 2F-140 gripper for handover (b) Pressure data collected from 24 contact positions at 0.8 mm and 1.6 mm contacted depth. (c) The change of gyroscope data on x and y axes at 1.6 mm contact depth along 24 contact points.

D. Tactile finger for grasping and handover

Grasping and handover objects are performed in a controlled environment, with predetermined picking location, grasping strategy and handover location. For these processes, a tactile finger is mounted in the Robotiq gripper of the UR3 robot arm. The tactile finger is composed of three soft multimodal tactile sensors capable of providing accelerometer, gyroscope and barometric pressure data (Fig. 3A). Each soft tactile sensor can provide pressure output at 24 contact locations with 1 mm intervals along the sensor surface (Fig. 3B), with the highest sensitivity at the centre of the sensor. The gyroscope response in x and y axes for applied contact at the different locations on the sensor is shown in Fig. 3C. When pressure output from any of three sensors in the tactile finger exceeds the predefined threshold value of 500, the gripper is controlled to perform a grasping motion for posterior handover actions in collaborative tasks [16]. A linear filter is also used on the gyroscope and pressure signals of the tactile sensor to ensure robust grasping and handover actions. This approach reduces the effect of vibrations generated from assembly actions performed by the user such as hammering.

The optimal weights of the linear filter are obtained using tactile datasets recorded in a handover scenario where the robot grasps an object and the operator attempts to pull the object multiple times within 30 s. This data collection process is repeated for each object used in the assembly of the box and chair. Tactile data are also recorded while the robot grasps an object and the human performs hammering actions. All the data are processed using 0.5 s sliding windows to extract time domain features such as standard deviation (STD), mean absolute value (MAV), slope sign change (SSC) and waveform length (WL) [21]. These features are then used to compute the optimal weights of the filter as follows:

$$d = XW \tag{1}$$

where $d = [d(1), .., d(n)]^T$ is an n-by-1 vector of zeros and ones representing the occurrence of handovers for n discrete time indexes. X is the n-by-m matrix of observations, where m is the total number of extracted features for gyroscope in three axes and the pressure signal:

$$\begin{bmatrix} x_1(1) & ... & x_1(m) \\ x_2(1) & ... & x_2(m) \\ \vdots & ... & \vdots \\ x_n(1) & ... & x_n(m) \end{bmatrix}$$

where $W = [w(1), ..., w(m)]^T$ is the m-by-1 vector of filter coefficients. The Wiener solution to equation 1 is as follows:

$$W_o = (X^T X)^{-1} X^T d \tag{2}$$

Since in practice this solution is perturbed with a series of statistical and computational inaccuracies, W_o represents weights approximated to the theoretical optimum [22]. Hence, as we assumed any exogenous disturbances are bounded we found a general empirical value ϵ for detection of handover in the assembly tasks as follows:

$$\begin{cases} XW_o > \epsilon, & \text{handover} \\ XW_o < \epsilon, & \text{do nothing} \end{cases}$$

E. Task Status Reasoning and Robot Control

Task status reasoning allows the system to track what actions have been finished and how long until future actions are completed. This is combined with the gesture recognition commands for efficient robot control allowing an adaptable and intuitive collaborative experience.

Each assembly task has a predefined list of actions the user and robot must complete in sequence, with each action type having a default time taken to complete it. During the task, the HAR module updates the current action prediction

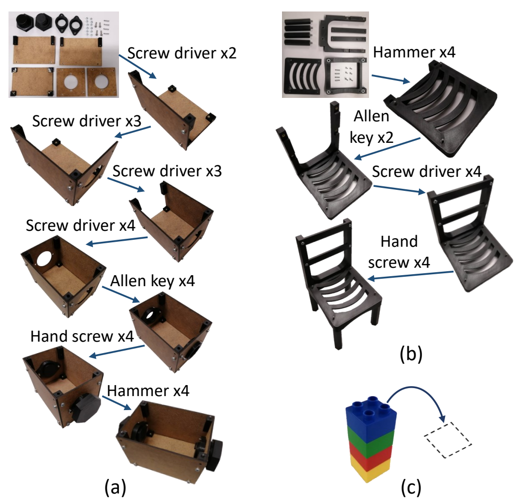

Fig. 4. Assembly tasks used in system demonstrations with user actions shown. (a) Box assembly task (b) Chair assembly task (c) Lego stack moved between positions for solo robot task

(a) (b) (c) (d) (e) (f)

Fig. 5. Sequence of collaborative steps. (a) Wave (b) Forwards (c) Part handover (d) User assembly (e) Robot solo task (f) Assembly handover

every 0.5 s. If the action matches the previous action, the current episode end time is updated. If the action changes, then the completed action is saved to the episodic memory and compared against the expected action from the action list. If it matches, then the current action number is moved on and the next action start time is set as the current time. If the current action expected is a robot action, the episodic memory is checked to observe if the robot has completed the action or not. If it has been completed, then the current action is moved on, otherwise, the user is instructed to wait for the robot to complete it.

At each 0.5 s update step, the future action timing predictions are also updated. Each action in the task is iterated over with time left for completed actions set to 0 s. The time left for the current action is set as the default time for that action minus the time since the action started. Future actions in the sequence add the previous actions' time left to the default duration for that action. Each of these timing predictions is then uploaded to the future action predictions database table.

The robot control module is responsible for deciding which action the robot should undertake next. Given the action completion time estimates in the future action predictions table, the time until the next collaborative action is required can be found. The robot aims to be ready to handover/receive the part in collaborative actions at the exact time the user is ready. The robot action is therefore started when the time left for the preceding user action is equal to or less than the time the robot action takes. If there is time to complete an action on the robot's solo task before the collaborative action is required, then the robot arm performs one action on that task first. If neither of those conditions is met, then the robot waits in the home position until the collaborative action should be started.

III. Experiments and Results

A. Task description and process flow

Two collaborative assembly tasks and a robot only solo task are used to test the system. The tasks show how the methods used are adaptable to any similar collaborative task where relevant action recognition methods can be provided. The two collaborative tasks are the assembly of a box and small chair shown in Fig. 4(a) and 4(b), respectively. Each task requires the robot to bring the next part or box of parts to the user at the right time where a handover will take place. At the end of the task the user hands the completed assembly back over to the robot, removing it from the workspace. The robot solo task is to move a stack of Lego blocks from one position to another and back again continuously (Fig. 4(c)). This task is used for illustration purposes to force the robot to choose between which task to perform an action on next.

The handover location is determined by considering the safety, physical comfort of the operator and the UR3 robot workspace. To improve handover robustness, the tactile sensor is only monitored when the robot is stopped in the handover position and no human action or gesture is detected. These conditions help prevent undesired grasps/releases from the gripper in the event the user or robot is not ready.

Key collaborative steps are shown in Fig. 5 with overall module interaction process flow shown in Fig. 6. The system initialises without knowing who the user is or which task is being performed. When ready, the user performs wave actions which are recognised by the gesture recognition module, enabling the person recognition step. The QR code on the user is read to provide their name, and the task that user will perform is inferred from the preallocation of tasks to users. The system loads in the task specific actions from the PostgreSQL database. The user gestures forwards to start the task, at which point the system begins predicting how long each future action will take and the robot brings the first part required. The system progresses through the task continuously updating action prediction, task status reasoning and action selection as described in Section II-E.

Throughout the task, gestures can be used for added levels of control over the robot. The initial wave and forwards gestures allow the robot to be sure of when to start. Throughout the task, the stop gesture can be used to force the robot to wait at the end of whichever action it

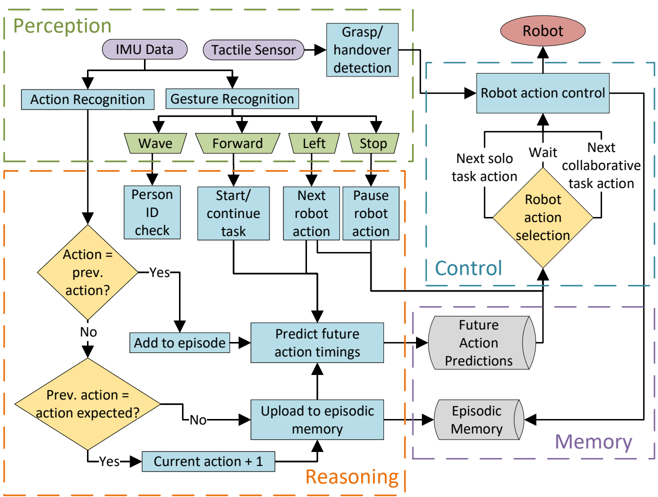

Fig. 6. Overview of process flow

TABLE I
OFFLINE HAR AND GESTURE RECOGNITION CLASSIFIER RESULTS. MEAN AND STD. DEV. FROM 10 FOLD LOOCV SHOWN.

	Accuracy		F1 Score	
	Mean	Std.	Mean	Std.
Screw driver	0.89	0.05	0.68	0.14
Allen Key	0.89	0.03	0.67	0.15
Hammer	0.98	0.01	0.57	0.24
Hand Screw	0.87	0.04	0.16	0.07
Gestures	0.91	0.05	0.66	0.13

is completing and the forwards gesture used to restart it. The left gesture can be used to force the robot to perform the next collaborative action in the event the system fails to correctly track human actions and thus fails to bring a part at the right time. All actions preceding the next collaborative robot action are labelled as completed, while the robot is instructed to begin the next collaborative action. A graphical user interface (GUI) provides the user feedback on the system status including the current action and gesture predictions, action status predictions and text feedback instructing the user which action to perform next.

B. Action recognition in offline mode

A dataset of IMU signals has previously been collected for offline training of the HAR and gesture recognition methods. Ten participants completed the box assembly task and each of the gestures for 10 s twice over while IMU data are recorded and current action is labelled in real-time. This leads to 12,458 s of data overall. The data are split into 3 s windows with 50 % overlap where the target class for each window is the most common label within the window. For each of the HAR classifiers, all data windows corresponding to actions other than the target action are added to the null class set. Model validation is performed using Leave One Out Cross Validation (LOOCV), where an entire participant's data are used for testing in each of the 10 folds.

The results from offline testing can be seen in Table I, along with confusion matrices in Fig. 7 and Fig. 8. These show that screw driver, Allen key and hammer actions are well recognised, while the hand screw action has many false negatives due to the less definite motion of the action. The gesture recognition results show all gestures recognised adequately, though false null predictions provide the most common source of error.

C. Online results from assembly tasks

Real-time experiments are conducted with 3 participants, each performing 2 trials of each assembly type (box and chair). This gives a total of 54 robot-to-human and 12 human-to-robot handovers. For robot-to-human handovers, 1.9 % failed with the robot erroneously dropping the part, while 16.7 % of human-to-robot handovers failed with the robot grasping while an object was not present. Handover duration, measured from the initial time both robot and human have contact with a part to the time it is grasped/released by the robot indicates the fluidity of handover. After removing 1 erroneous data point, a mean of 2.25 s (SD 1.77) is found, heavily skewed towards shorter handovers (Fig. 9). The longer duration handovers are largely due to the additional

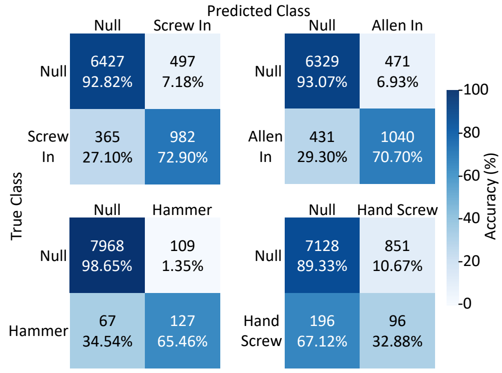

Fig. 7. Confusion matrices from offline HAR classifiers training

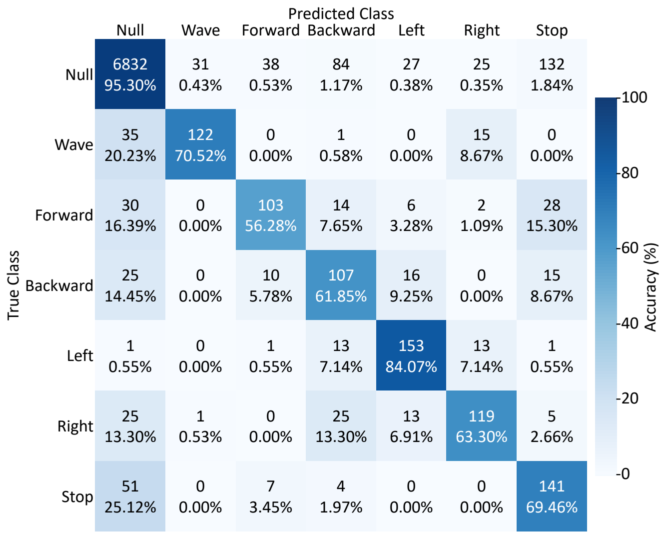

Fig. 8. Confusion matrix from offline gesture classifier training

checks, such as no HAR prediction, incorrectly blocking the part release.

Analysing the amount of time the user and robot wait for each other shows the efficiency of the robot action planning. The robot aims to deliver a part to the user at the exact time it is required, thus the user and robot wait times while in the handover position should approach 0 s. The average user and robot wait times per user and task type are shown in Table II, with a histogram of times shown in Fig. 10. Both robot and user wait times are heavily skewed towards 0 s,

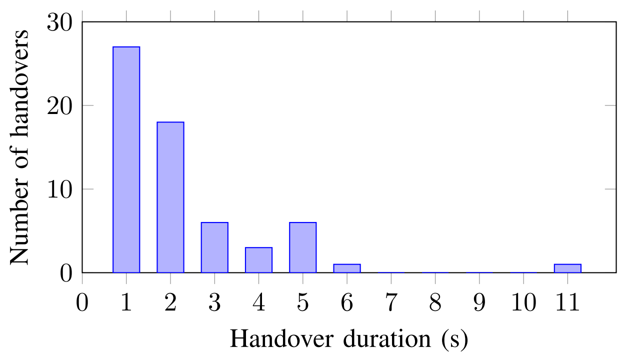

Fig. 9. Bincount of handover durations from robot and user both touching part to grasp/release.

TABLE II
AVERAGE WAIT TIMES FOR HANDOVER TO BEGIN FOR USER AND ROBOT.

	User Wait (s)			Robot Wait (s)		
	Box	Chair	Overall	Box	Chair	Overall
User 1	6.75	5.90	6.36	6.42	8.80	7.50
User 2	3.50	0.60	2.18	0.75	8.20	4.14
User 3	0.67	0.00	0.36	16.00	20.90	18.23
Overall	3.64	2.17	2.97	7.72	12.63	9.96

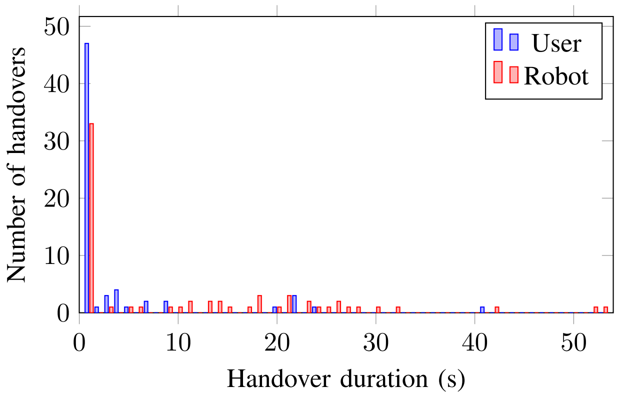

Fig. 10. Waiting times for handover to begin for user and robot.

with a range of large wait times inflating the mean. This is largely due to different users and tasks having varying durations to complete actions, making prediction of when a handover will occur difficult with the simple fixed time per action approach used. The occurrences where the robot waits a very long time are often due to the user fumbling a part, quickly adding large amounts of time to the action, or incorrect perception of a 'left' gesture, causing the robot to jump ahead to the next collaborative action early. The occurrences where a user waits a long time for the robot are mainly due to the action recognition failing to keep track of the assembly state. The user must then perform a next action command, wait for the robot to finish its solo task action before moving onto the collaborative action. The average rate of next action gestures per handover operation is 10.6 %.

The robot is instructed to wait in the home position when it has excess time before the next collaborative action but not enough to complete an action on the solo task. Removing occurrences where the robot has completed all actions on its solo task gives an average idle time of 34.3 s and 3.67 s per trial for the box and chair tasks, respectively. Overall, the average trial completion times are 405 s and 260 s for the box and chair assembly tasks, respectively.

IV. CONCLUSIONS

This work presented an approach for human-robot collaboration using human action recognition, hand gesture recognition, tactile feedback handovers and user personalisation. Collaborative trials on two assembly tasks demonstrated interaction fluency with reliable handovers and action recognition leading to reliable action selection. Future work will focus on investigating methods for more robust tactile sensing perception to remove unintended grasping/release actions, updating the task status reasoning to be more adaptable to different users and tasks, and faster handover processes by predicting pregrasp phase with the wearable sensors.

REFERENCES

[1] S. El Zaatari, M. Marei, W. Li, and Z. Usman, "Cobot programming for collaborative industrial tasks: An overview," *Robotics and Autonomous Systems*, vol. 116, pp. 162–180, 2019.

[2] M. L. Nicora, R. Ambrosetti, G. J. Wiens, and I. Fassi, "Human–robot collaboration in smart manufacturing: Robot reactive behavior intelligence," *Journal of Manufacturing Science and Engineering*, vol. 143, no. 3, 2021.

[3] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[4] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco, "Prediction of human activity patterns for human–robot collaborative assembly tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3934–3942, 2018.

[5] K. Darvish, E. Simetti, F. Mastrogiovanni, and G. Casalino, "A hierarchical architecture for human–robot cooperation processes," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 567–586, 2020.

[6] M. Al-Amin, R. Qin, W. Tao, D. Doell, R. Lingard, Z. Yin, and M. C. Leu, "Fusing and refining convolutional neural network models for assembly action recognition in smart manufacturing," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, p. 0954406220931547, 2020.

[7] P. Neto, M. Simão, N. Mendes, and M. Safeea, "Gesture-based human-robot interaction for human assistance in manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 101, no. 1, pp. 119–135, 2019.

[8] 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.

[9] G. A. Al, P. Estrela, and U. Martinez-Hernandez, "Towards an intuitive human-robot interaction based on hand gesture recognition and proximity sensors," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2020, pp. 330–335.

[10] X. Wen and H. Chen, "3d long-term recurrent convolutional networks for human sub-assembly recognition in human-robot collaboration," *Assembly Automation*, 2020.

[11] I. El Makrini, K. Merckaert, D. Lefeber, and B. Vanderborght, "Design of a collaborative architecture for human-robot assembly tasks." in *IROS*, 2017, pp. 1624–1629.

[12] G. Horváth, C. Kardos, A. Kovács, B. Pataki, and S. Tóth, "Worker assistance suite for efficient human–robot collaboration," in *Advanced Human-Robot Collaboration in Manufacturing*. Springer, 2021, pp. 375–394.

[13] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kulić, "Object handovers: a review for robotics," *IEEE Transactions on Robotics*, 2021.

[14] J. Zhang, H. Liu, Q. Chang, L. Wang, and R. X. Gao, "Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly," *CIRP annals*, vol. 69, no. 1, pp. 9–12, 2020.

[15] K.-H. Tang, C.-F. Ho, J. Mehlich, and S.-T. Chen, "Assessment of handover prediction models in estimation of cycle times for manual assembly tasks in a human–robot collaborative environment," *Applied Sciences*, vol. 10, no. 2, p. 556, 2020.

[16] G. A. Al, B. E. A. Oussallam, and U. Martinez-Hernandez, "A single-chip multimodal tactile sensor for a robotic gripper," in *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021, pp. 364–369.

[17] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.

[18] C. Di Napoli and S. Rossi, "A layered architecture for socially assistive robotics as a service," in *2019 IEEE international conference on systems, man and cybernetics (SMC)*. IEEE, 2019, pp. 352–357.

[19] U. Martinez-Hernandez, A. Damianou, D. Camilleri, L. W. Boorman, N. Lawrence, and T. J. Prescott, "An integrated probabilistic framework for robot perception, learning and memory," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016, pp. 1796–1801.

[20] J. Male and U. Martinez-Hernandez, "Collaborative architecture for human-robot assembly tasks using multimodal sensors," in *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021, pp. 1024–1029.

[21] M. Zecca, S. Micera, M. C. Carrozza, and M. C. Carrozza, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Reviews™ in Biomedical Engineering*, vol. 30, no. 4-6, pp. 459–485, 2002.

[22] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 5th ed. Switzerland: Springer Nature, 2020.