2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Learning-based Adaptive Admittance Controller for Efficient and Safe pHRI in Contact-rich Manufacturing Tasks

Pouya P. Niaz[1], Engin Erzin[2], and Cagatay Basdogan[1]

***Abstract*—This paper proposes an adaptive admittance controller for improving efficiency and safety in physical human-robot interaction (pHRI) tasks in small-batch manufacturing that involve contact with stiff environments, such as drilling, polishing, cutting, etc. We aim to minimize human effort and task completion time while maximizing precision and stability during the contact of the machine tool attached to the robot's end-effector with the workpiece. To this end, a two-layered learning-based human intention recognition mechanism is proposed, utilizing only the kinematic and kinetic data from the robot and two force sensors. A "subtask detector" recognizes the human intent by estimating which phase of the task is being performed, e.g., *Idle*, *Tool-Attachment*, *Driving*, and *Contact*. Simultaneously, a "motion estimator" continuously quantifies intent more precisely during the *Driving* to predict when *Contact* will begin. The controller is adapted online according to the subtask while allowing early adaptation before the *Contact* to maximize precision and safety and prevent potential instabilities. Three sets of pHRI experiments were performed with multiple subjects under various conditions. Spring compression experiments were performed in virtual environments to train the data-driven models and validate the proposed adaptive system, and drilling experiments were performed in the physical world to test the proposed methods' efficacy in real-life scenarios. Experimental results show subtask classification accuracy of 84% and motion estimation R^2 score of 0.96. Furthermore, 57% lower human effort was achieved during *Driving* as well as 53% lower oscillation amplitude at *Contact* as a result of the proposed system.**

***Index Terms*—human-robot interaction, intention recognition, adaptive admittance control, subtask detection, motion estimation, collaborative manufacturing**

I. Introduction

Many pHRI tasks in small-batch manufacturing involving contact of the machine tool with the environment, such as drilling, polishing, cutting, etc., contain different phases (subtasks) with different requirements, rendering an interaction controller, such as an admittance controller, with fixed parameters impractical. For instance, in *"Driving"*, where a human is guiding the robot in free space by hand, low admittance damping values are desired to minimize human effort. By contrast, *"Contact"* requires higher damping to ensure stability [1], [2]. Moreover, *Contact* must be detected before it occurs to smoothly change the damping, which requires estimating human motion intent during *Driving*. Delayed damping adaptation can lead to instability during *Contact*. In contrast, premature adaptation may heighten human effort and extend the duration of *Driving*. This emphasizes the importance of human intention recognition in pHRI; it ensures the seamless transition between task phases and optimizes human-robot interaction by dynamically adjusting control parameters while maintaining stability and efficiency.

Researchers have used various methods to detect human intent to improve task performance in pHRI systems. In controlled settings with predefined constraints, threshold values of velocity and force (or their derivatives) have been used for detecting human intent [3]–[7]. Though easy to implement, such rule-based methods often fail in dynamic settings with changing conditions, which is why data-driven methods have started to replace them, owing to their ability to extract meaningful relationships and latent features. In tasks involving reaching and co-manipulation, minimum-jerk models have been employed to infer human intentions and assist users in completing tasks more efficiently [8], [9]. Complex pHRI tasks, however, rarely follow minimum-jerk profiles. Hidden Markov Models (HMM) have been used in collaborative assembly tasks to produce collision-free movements, detect human intentions, and assist them accordingly [10], [11]. HMM processes the latest state of the system and may be ineffective when long-term dependencies affect future states. In addition, shape-invariant representations [12], probabilistic dynamic movement primitives [13], and parameterized probabilistic principal component analysis [14] have been proposed for modeling human behavior in pHRI tasks [15], [16]. While relatively simple to implement and able to model stochasticity, these methods often do not capture all of the complex relationships between kinematic/kinetic data and future states in contact-rich tasks. Other data-driven techniques such as Dynamic Time Warping (DTW), Gaussian Process Regression (GPR), and Random Forests (RF) have also been used for intention recognition or motion prediction in pHRI [17], [18]. These can work effectively in simple tasks but may fail to extract deeper latent features and spatiotemporal dependencies in complex ones. Deep Learning (DL) has shown promising results in time series prediction and sequence modeling due to its ability to capture deeper representations and learn nonlinear relationships. It has been used for intent recognition, motion

[1] P. P. Niaz and C. Basdogan (*corresponding author*) are with the Robotics and Mechatronics Laboratory (RML) and the KUIS AI Center, Koc University, Sariyer, Istanbul 34450, Turkey. {pniaz20, cbasdogan}@ku.edu.tr

[2] E. Erzin is with the Multimedia, Vision, and Graphics Laboratory and the KUIS AI Center, Koc University, Sariyer, Istanbul 34450, Turkey. eerzin@ku.edu.tr

prediction, and human behavior modeling in pHRI [19]–[23].

In this study, a human and a robot, equipped with an admittance controller, were envisioned executing a collaborative drilling task, which involves contact with a stiff or rigid workpiece (Fig. 1). The task was divided into separate subtasks, each with its own set of desired/required control parameters, thereby needing an adaptive system for better task efficiency. To this end, a **subtask detector** recognizes the current subtask and adapts the admittance controller at subtask transitions as proposed in our earlier study [24]. Realistically, however, a subtask detector only ever reacts to changes in system behavior at subtask transitions with inevitable delay, *after* they occur. Hence, in a task like drilling, *Contact* begins with low damping values remaining from *Driving*. The stiff contact may instantly lead to instability as in oscillation or bounce-back *before* the subtask detector recognizes *Contact* and adapts the controller. This renders the subtask detector less effective in preventing instability at *Contact*. A higher damping value in *Driving* would mitigate the problem but instead fail to minimize human effort, nullifying the advantage of adaptive control. The goal is to choose low damping for *Driving* and high damping for *Contact* while maintaining stability regardless of material stiffness and contact rigidness. An additional **motion estimator** was thus proposed in this study, so we could not only adjust the controller flexibly during *Driving* but also anticipate *Contact* and adapt the controller earlier in time. As a result, as shown in the experimental results, human effort would be minimized during *Driving*, and stability would be maintained throughout *Contact*. We conducted collaborative drilling experiments in the virtual and physical worlds to validate the efficacy of the proposed system.

The contributions of this work are as follows:

- A two-layered system consisting of a subtask detector and a motion estimator was proposed for adapting the damping coefficient of an admittance controller in contact-rich pHRI tasks to minimize effort and maximize stability. The experimental results show that the proposed two-layer approach utilizing DL models leads to lower human effort and further reduces the instabilities at contact compared to the subtask detector alone [24].
- We conducted spring compression experiments in virtual environments (VEs) with 3-DoF admittance controllers with multiple human subjects under various conditions to train different ML models. Since the control loop ran at 500 Hz, our pHRI system could render contact forces and display real-time haptic feedback to users during the simulations. This feature allowed for the execution of spring compression experiments in VEs, imitating the actual drilling task, thereby eliminating the necessity for extensive physical trials, which is not very practical for tasks such as drilling. After the ML models were rigorously trained and cross-validated through simulations in virtual environments, they were implemented in the physical world, where their effectiveness was tested in real-life scenarios through a series of actual drilling experiments.

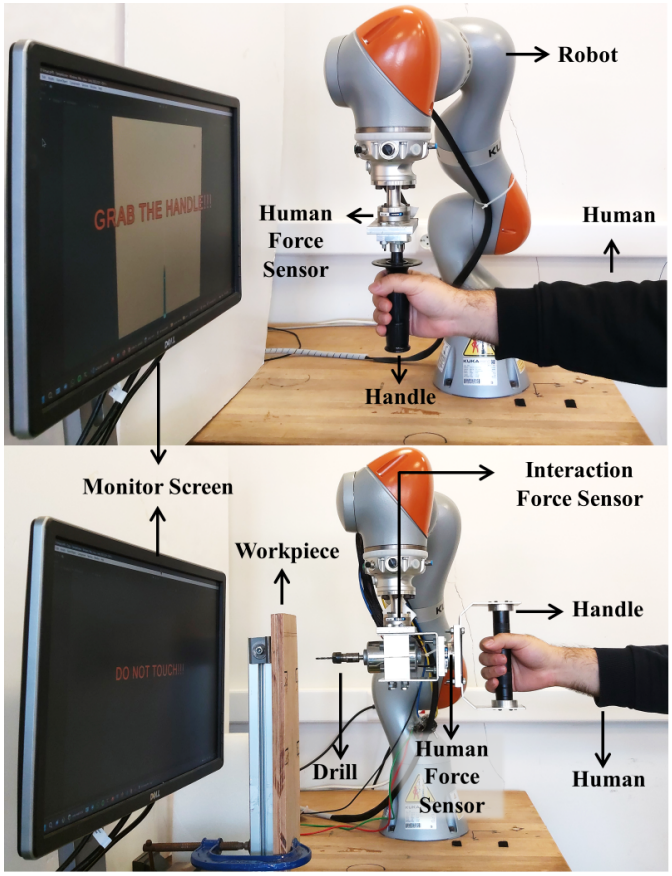

Fig. 1. Hardware setup in the virtual spring compression task (top) and the real drilling task (bottom).

In Section II, our hardware setup, adaptive control system, and evaluation metrics are explained. Section III explains the spring compression experiments performed in VEs for training the ML models offline, while Section IV reports the results of online deployment and testing of the models in the same environment. Section V reports real drilling experiments' subtask detection and motion estimation performance while deploying the ML models online. Section VI concludes the study.

II. Approach

A. Hardware Setup

In this study, two types of experiments were performed. Spring compression experiments were performed in a virtual environment (VE) for offline training and online testing of ML models. Drilling experiments were performed in the physical world for final deployment and assessment. Fig. 1 shows the hardware components of our experimental setup for both environments.

The virtual spring compression setup included a cobot (LBR iiwa 7, KUKA Inc.), a handle for the human to grab, and a monitor for visual feedback. A force sensor (Mini 45, ATI Inc.) connected the handle to the robot's end-effector to measure human forces. The real drilling setup included a DC motor,

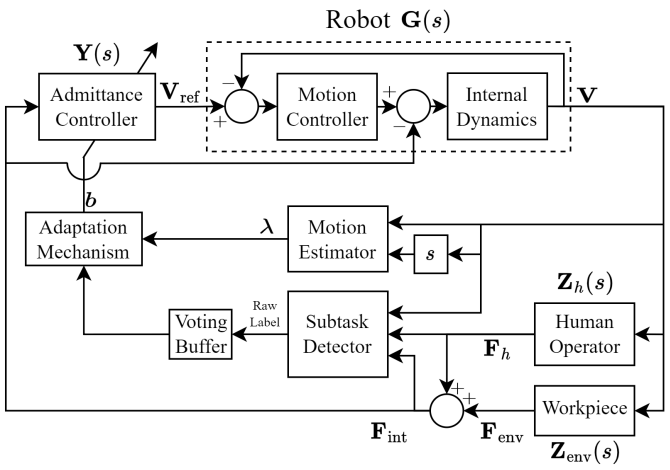

Fig. 2. The closed-loop control architecture used in our study. All variables are 3-dimensional vectors, corresponding to 3 translational degrees of freedom in the Cartesian space.

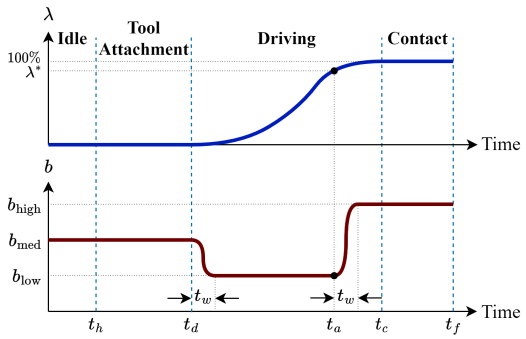

Fig. 3. Trajectory progress λ and admittance damping b vs. time t across all subtasks. Damping is adapted at t_d when *Driving* begins and at t_a when the estimated progress reaches the adaptation threshold, i.e., $\lambda = \lambda^*$, just before *Contact* occurs at t_c. The trial ends when the drilling/spring compression reaches its desired depth of 4 [mm] at t_f.

a drill bit, and another force sensor (Mini 45, ATI Inc.) to measure the interaction forces applied to the end-effector.

B. Closed Loop Control Architecture

Our control loop for pHRI, updated at 500 Hz, is shown in Fig. 2. Accordingly, $\mathbf{F}_h$ and $\mathbf{F}_{\text{env}}$ are human and environment forces, respectively, and $\mathbf{F}_{\text{int}} = \mathbf{F}_h + \mathbf{F}_{\text{env}}$ is the interaction force. $\mathbf{V}_{\text{ref}}$ is the reference velocity, and $\mathbf{V}$ is the real velocity. Additionally, $\mathbf{Z}_h(s)$ and $\mathbf{Z}_{\text{env}}(s)$ are the unknown human and environment (workpiece) mechanical impedances, respectively. The admittance controller allows three translational degrees of freedom, for each of which the admittance controller has the following transfer function:

$$Y(s) = \frac{V_{\text{ref}}(s)}{F_{\text{int}}(s)} = \frac{1}{m\,s + b} \tag{1}$$

where m [kg] and b [Ns/m] are the admittance mass and damping parameters, respectively, chosen to be equal among all 3 DOFs to maintain consistency when movement directions, target orientations, and material properties changed.

C. Adaptation Mechanism

a) ***Subtask Detector****:* We generically divided contact-rich pHRI tasks, such as drilling, polishing, cutting, etc., into four subtasks:

1) *Idle*: The robot is stationary, and there is no input.
2) *Tool-Attachment*: The human holds the handle to attach the drill bit, polishing pad, saw band, etc.
3) *Driving*: The human guides the robot in free space towards the workpiece.
4) *Contact*: The human and the robot perform drilling, polishing, cutting, etc., on the workpiece.

In our two-layer approach, the subtask detector shown in Fig. 2 was always active. It took sliding windows of magnitudes $||\mathbf{V}||$, $||\mathbf{F}_{\text{int}}||$ and $||\mathbf{F}_h||$ as the inputs to predict the current subtask. The "voting buffer" in Fig. 2 took the most frequent prediction in the last 100 steps of the control loop (i.e., 0.2 seconds).

b) ***Motion Estimator****:* The motion estimator shown in Fig. 2 was only active during *Driving*. It took sliding window sequences of magnitudes of velocity and acceleration signals and estimated the trajectory progress λ, which is the portion of the *Driving* trajectory length that has been traveled so far. It starts with 0.0% when *Driving* begins, monotonically increases regardless of the shape of the trajectory, and ends with 100% at *Contact* (see Fig. 3). It is calculated as follows:

$$\lambda = \frac{l_d}{L_d} = \frac{\int_{t_d}^{t} \|\mathbf{V}\|\,dt}{\int_{t_d}^{t_c} \|\mathbf{V}\|\,dt} \tag{2}$$

where t_d is when *Driving* begins, t_c is when *Contact* happens, l_d is the traveled length in *Driving*, and L_d is the total trajectory length in *Driving*.

The adaptation mechanism is summarized in Fig. 3. Accordingly, without touching admittance mass $m = 50$ [kg] in (1), admittance damping b was adapted for every degree of freedom. A low damping $b_{\text{low}} = 100$ [Ns/m] was chosen for *Driving* to minimize human effort, while a higher damping $b_{\text{high}} = 500$ [Ns/m] was chosen for *Contact* to maximize stability. These values were chosen empirically after preliminary trials, to facilitate smooth and stable interactions. To prevent jerky motion at the beginning of *Driving* and make the robot slightly more stable at the beginning, a medium value of admittance damping $b_{\text{med}} = 300$ [Ns/m] was chosen. As shown in Fig. 3, a progress value close to 100% called "Adaptation Threshold" was chosen (denoted as λ^*), so when the estimated progress reached it at time $t = t_a$, it was assumed that *Driving* is close to the end and *Contact* was imminent, so damping was adjusted for *Contact*. Adaptation occurred through a cubic polynomial lasting a short period of $t_w = 200$ [ms] to maintain a smooth damping profile.

D. Machine Learning Methods

Assuming unknown target position, orientation, etc., the goal in this study was to learn profiles of kinematic/kinetic

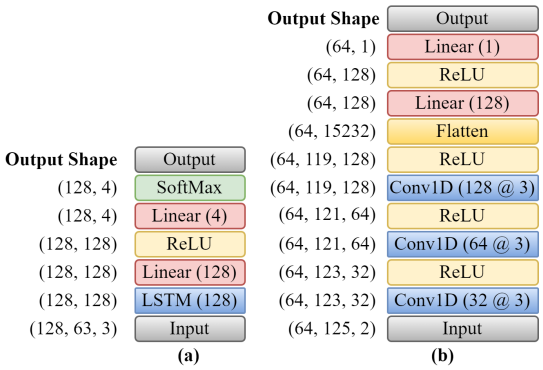

Fig. 4. Deep Learning architectures used in this study: (a) Subtask Detector LSTM, (b) Motion Estimator CNN. Numbers inside parentheses are hidden layer sizes for Linear layers, hidden sizes for LSTM layers, and the number of filters for Convolution layers, followed by filter sizes.

TABLE I
DL MODEL HYPERPARAMETERS

Parameter	Subtask Detection	Motion Estimation
Sequence Length [sec]	0.5	8.0
Downsampling Rate	4	32
Num. Timesteps in Input	63	125
Num. Input Features	3	2
Num. Output Features	4	1
Minibatch Size	128	64
Initial Learning Rate (LR)	1.0 e-4	1.0 e-4
LR Exponential Decay	0.95	0.95
Training Epochs	40	40
Validation Set (% Trainset)	5	5
Validation Patience (Epochs)	10	10
L1 Regularization	0.0	1.0 e-4
L2 Regularization	0.0	1.0 e-4

signals in different stages of a pHRI task, to be able to estimate motion and detect subtask in real time. To this end, even though multiple data-driven methods were explored, deep learning models performed best for subtask detection and motion estimation. After extensive hyperparameter optimization, a Long Short-Term Memory (LSTM) model was chosen for subtask detection, and a Convolutional Neural Network (CNN) model with 1D convolutions was chosen for motion estimation. Fig. 4 shows the model architectures, while Table I tabulates the hyperparameters chosen for these models. The data was sampled at 500 [Hz] and the sliding windows were pushed one step forward at each time step. The sequences extracted from the sliding windows were downsampled before being fed to the models to save time and memory during training and inference (See Table I).

E. Evaluation Metrics

Various evaluation metrics were used to assess the performance of the proposed system, calculated for every trial of the task. Subtask detection metrics included accuracy and weighted F_1 scores, while motion estimation metrics included Root Mean Squared Error (RMSE) and R^2 values. After deploying the trained models on the adaptive control system, the performance of the proposed system needed to be evaluated

TABLE II
DETAILS OF THE EXPERIMENTS PERFORMED IN THIS STUDY

Designation	A	B	C
Motivation	Collecting data and training models offline	Deploying trained models online and testing performance	Examining system performance in real tasks
Type	Virtual Spring Compression		Real Drilling
Subjects	5	3	3
Perpendicular Distances	$L_p \in \{12, 16, 20, 24\}$[cm]	$L_p = 18$ [cm]	$L_p = 18$ [cm]
Target Locations	4 corners	4 corners	4 corners
Target Sizes	IoD $\in \{3, 5\}$	IoD $= 4$	IoD $= 4$
Subtask Detection	Hard-coded	Trained Model	
Motion Estimation	N/A	Trained Model	

in terms of the final objectives we are trying to achieve in this study. These included the following:

- Average human force throughout *Driving*,
$$F_h^{\text{ave}} = \frac{1}{T_d} \int_{t_d}^{t_c} \|\mathbf{F}_h(t)\| \, dt, \ [\text{N}] \tag{3}$$
- Average velocity throughout *Driving*,
$$V_{\text{ave}} = \frac{1}{T_d} \int_{t_d}^{t_c} \|\mathbf{V}(t)\| \, dt, \ [\text{m/s}] \tag{4}$$
- Total human effort throughout *Driving*,
$$E_h^{\text{tot}} = \int_{t_d}^{t_c} \|\mathbf{F}_h(t)\| \, \|\mathbf{V}(t)\| \, dt, \ [\text{J}] \tag{5}$$
- Oscillation of the end-effector velocity at *Contact*, expressed as the maximum magnitude observed in its spectrogram, after detrending it with a high-order high-pass Butterworth filter with a cutoff frequency of 0.5 [Hz].

F. Experiments

A large amount of training data was required to train the DL models. Since extensive data collection with various distances and target locations/sizes would take a long time and consume significant material in a real drilling task, the experiments were conducted in virtual environments (VEs) using our collaborative robot and human subjects, where we simulated the workpiece as a linear spring with a coefficient of $K = 8000$ [N/m]. We first conducted a virtual spring compression experiment with no motion estimation. The admittance damping was adapted based on hard-coded rules (Experiment A). Then, using the collected data, different models for subtask detection and motion estimation were trained and tuned offline. Afterward, new experiments were performed to assess their online performance. This time, the models were deployed in the adaptive control system, and adaptation was based on these trained models, not hard-coded rules (Experiment B). Eventually, to examine the real-life performance of these trained models, drilling experiments were also performed in physical world (Experiment C) with adaptation based on the trained models. Table II summarizes the three experiments and their details.

III. EXPERIMENT A: OFFLINE TRAINING IN VES

In this experiment, the admittance damping was adapted based on subtask detection but not motion estimation. Since

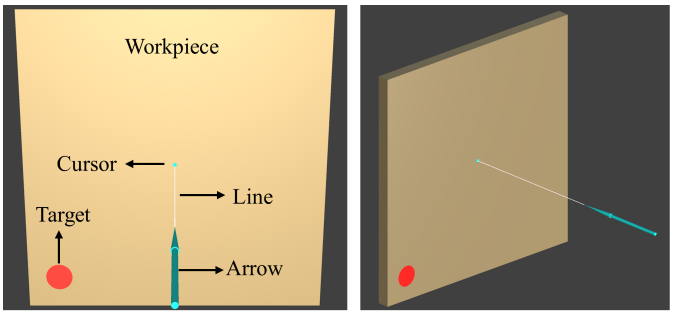

Fig. 5. Snapshot of the visual feedback displayed to the subjects for the spring compression experiments performed in VEs; (a) What the subjects see on the monitor screen; (b) Another view to aid the imagination.

all parameters of the experiment were fully known and under control, an ideal subtask detector was hard-coded into the system, which used predefined and controlled thresholds of human force for detecting *Idle*, *Tool-Attachment*, and *Driving*. The system detected *Contact* as soon as the tip of the virtual drill bit touched the virtual spring, which generated an artificial reaction force. A snapshot of the visual feedback in these experiments can be seen in Fig. 5.

A. Protocol

The following protocol was followed in this experiment:

The subject stood and waited inside the designated area (*Idle*) until the "GRAB THE HANDLE" command appeared on the screen. They then grabbed the handle of the robot and held it steady (*Tool-Attachment*). After 3 seconds, "GO" command appeared on the screen, at which point the subject started moving the robot towards the target. As the robot moved, a virtual arrow imitating the drill bit was displayed in real time on the computer screen. The subject was asked to move the arrow in free space (*Driving*) and penetrate the workpiece at the target (*Contact*), imitating the drilling process. The target was represented by a red circle on the workpiece, allowing the user to initiate penetration at any point within the circle. A progress bar appeared when the arrow touched the workpiece anywhere within the circle, showing the current compression depth and target depth (*Contact*). The subject kept penetrating the workpiece until the "RETRACT" command was shown, officially ending the trial. At this point, the subject retracted the arrow from the workpiece and stood clear for the robot to let it return to its home position and go to the subsequent trial.

B. Conditions and Subjects

Experimental conditions, summarized in Table II, included the following variables:

- Perpendicular distance: Normal distance between the home position and the workpiece
- Target location: Target was on one of the corners of a 15×15 [cm] square

TABLE III
TARGET DIAMETERS W [CM] AS A FUNCTION OF INDEX-OF-DIFFICULTY (IOD) AND PERPENDICULAR DISTANCE L_p

	IoD	3	5
L_p [cm]	12	1.71	0.39
	16	2.29	0.52
	20	2.86	0.65
	24	3.43	0.77

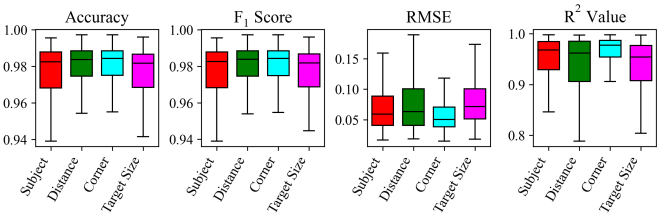

Fig. 6. Subtask detection (left pair) and motion estimation (right pair) test-set performance under various k-fold CV. Box plots show means and quartiles.

- Target size: Circle diameter; small and large

The above conditions were chosen considering that in real pHRI scenarios, target position and size are unknown and can change in every drilling session. To ensure consistency in the durations of *Driving* and human behaviors, and to facilitate meaningful comparisons of target sizes, we drew inspiration from the concept of "Index of Difficulty" (IoD, expressed in bits) as defined for the Fitts' reaching task ([25]). As per Shanon's formula [25], for a given IoD, the target diameter can be defined as follows:

$$W = \frac{L_p}{2^{\mathrm{IoD}} - 1} \tag{6}$$

where W [m] is the target diameter, and L_p [m] is the perpendicular distance from the home position to the workpiece. In our experiments, IoD values of 3 and 5 were chosen for large and small targets, respectively. Table III shows the actual target diameters for each L_p and IoD.

5 subjects (1 female and 4 males; average age: 26.25 ± 4.2 SD) participated in the experiment. There were 32 experimental conditions (4 distances × 4 corners × 2 IoD), and all subjects repeated each condition 5 times. Hence, the number of trials for each subject was 160, while the total was 800.

C. Cross Validation

Structured k-fold cross-validation (CV) was employed during the offline training of the models to evaluate their performance in unforeseen circumstances. In each fold, trials belonging to a single subject, perpendicular distance, target location (corner), or target size (IoD) were kept for testing, while the rest were used for training. This sort of CV showed the expected performance of the models when encountering new subjects or conditions unseen in the training set, which is important in real pHRI scenarios.

D. Results

Fig. 6 shows the statistics of the test-set performances of the deep learning models under various CV categories after

tuning and HPO. As can be seen, model performances are near-perfect, meaning the chosen models and architectures with the chosen input features can detect the subtask or estimate the trajectory progress when tested on new subjects and conditions. Multiple ML models and data-driven methods were tried, though only the best models are reported here [1].

IV. Experiment B: Online Testing in VEs

A. Protocol, Conditions and Subjects

These experiments were conducted within an identical virtual environment (VE), utilizing the same scenario and protocol as Experiment A. However, adaptation was based on the trained subtask detector and motion estimator deployed in the system. Also, numerical values for some conditions differed from the training experiments (Experiment A) to introduce new conditions unseen during training (Table II). Accordingly, while keeping the four corners of the workpiece, only one perpendicular distance (18 cm) and one target size with IoD = 4 were chosen, both unseen by the trained model. In these experiments, three admittance controllers were used: (**C1**) A fixed controller with a constant damping $b = 500$ [Ns/m], (**C2**) An adaptive controller with the same parameters used in Experiment A, which used learning-based subtask detection alone for damping adaptation, without any motion estimation, and (**C3**) An adaptive controller similar to the previous one, but also incorporating the motion estimator during *Driving*, to adapt the damping before *Contact* occurs.

3 subjects (1 female and 2 males; average age: 28.67 ± 4.92 SD) participated in the experiment. There were 12 experimental conditions (3 controllers × 1 perpendicular distance × 4 target locations × 1 IoD), and all subjects repeated each experimental condition twice. Hence, the number of trials for each subject was 24, while the total was 72.

B. Results

The performance metrics reported here are extracted from the system's real-time data, collected at 500 Hz. However, the DL models deployed in the system had much lower update rates, inevitably decreasing the real-time subtask detection and motion estimation performances. This can be alleviated by utilizing GPU or model parallelization. The subtask detector achieved an accuracy of 84.46% and a weighted F_1 score of 0.8423, which is considered satisfactory in most classification tasks. Most misclassifications by the subtask detector stemmed from delays in detecting subtasks during subtask transitions. This delay is attributed partly to the time needed to accumulate sufficient data from a new subtask and partly to the model's low update rate in real time. The motion estimator was deployed on the system and activated promptly upon detecting the *Driving* subtask by the subtask detector. A conservative adaptation threshold of 75% was chosen to adapt the damping before *Contact* occurred. Early adaptation of damping was successfully achieved in all trials under controller C3. The motion estimator achieved an overall RMSE of 0.0890 and R^2 score of 0.9562, which are acceptable for our pHRI application. Despite encountering limitations during their online deployment in the real-time system, the subtask detector and motion estimator demonstrated satisfactory performance, allowing for the appropriate adaptation of the admittance controller. Fig. 7 shows the confusion matrix for the subtask detector, as well as the overall regression performance of the motion estimator. Comparable model performances between Exp. B and Exp. A (Section III-D) despite unforeseen conditions showcase the generalization capability of the models.

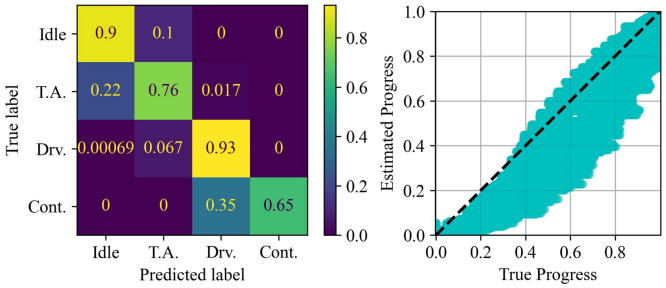

Fig. 7. The confusion matrix for subtask classification (left) and motion estimation performance (right) in Experiment B (online testing).

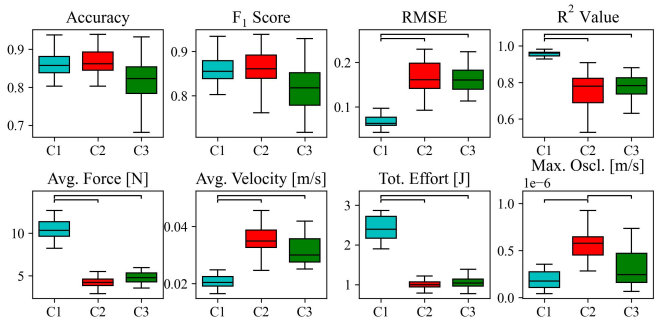

Fig. 8. Performance metrics observed in Experiment B (online testing). On the top row, the left pair of plots denotes subtask classification metrics, while the right pair denotes motion estimation metrics. The bottom row shows task performance metrics. Box plots show medians and quartiles. Horizontal brackets show statistically significant pairwise comparisons with $\alpha = 0.001$ after performing a Bonferoni-corrected Wilcoxon signed rank test.

The comparative statistics of all performance metrics introduced in Section II-E are reported in Fig. 8. Accordingly, even though the damping profiles and interaction dynamics differed among the three controllers, there were no statistically significant differences between subtask detection performances. This proves that a well-trained learning-based subtask detector will be sufficiently robust to changes in damping, adaptation mechanism, and kinematic/kinetic variables.

The right pair of plots on the top row of Fig. 8 reports motion estimation metrics. As can be seen, some differences exist between C1 (fixed controller) and other adaptive controllers C2 and C3, mainly because C1 incorporates high damping values

[1]Further details of data visualization and comparative performances of different ML models and data-driven methods for both subtask detection and motion estimation can be found at the following link, as well as a short demonstrative video:
https://github.com/pniaz20/progest

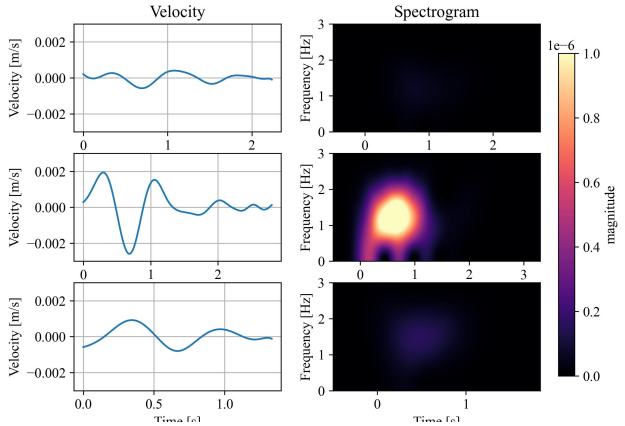

Fig. 9. Sample trial from Experiment B (online testing) showing oscillations of the end-effector velocity during the *Contact*. The left column shows the end-effector velocity as a function of time, and the right column shows its spectrogram with $\Delta f = 0.05$[Hz]. Rows 1 to 3 correspond with C1 to C3, respectively.

during the *Driving*, resulting in slower movement, giving more time for the DL model to predict the progress.

The bottom row in Fig. 8 shows adaptive control performance in Experiment B. Accordingly, as was expected, average human force and total human effort were significantly higher in C1 with fixed damping values, while average velocity was significantly lower in C1. This is because of the low damping value during *Driving* in C2 and C3. The lower right plot in this figure shows the peak oscillation amplitude of the end-effector velocity. Accordingly, in C1 and C3, because the *Contact* (when the robot touches the workpiece) began with high damping, there were significantly smaller oscillations than those observed in C2, in which the inevitable delay in subtask detection resulted in low damping at *Contact*. As can be seen from Fig. 8, when using C3, not only was the human effort and force minimized, but also, instabilities were avoided at the beginning of *Contact*, thereby maximizing stability and safety. Small differences in some metrics observed in Fig. 8 between C2 and C3 occurred because in C3 the damping increased to a high value during *Driving* and some time before *Contact*. This short period of time spent at the end of *Driving* with high damping lead to a slight increase in force and effort in *Driving*, though statistically insignificant.

Fig. 9 shows the oscillations of the end-effector velocity for a sample trial of Experiment B. As can be seen, in C1 and C3 where *Contact* began with high damping, oscillations are minimal. In contrast, in C2 (middle row), they are distinctly larger because damping is not adapted appropriately during the initial moments of *Contact*. In real-life scenarios with rigid materials, these instabilities occurring in C2 can easily damage the sensors in the pHRI system and the workpiece. This demonstrates the benefit of the motion estimator in contact-rich pHRI tasks.

V. Experiment C: Verification in Physical World

These experiments evaluated whether the proposed models, originally trained in VEs, could be seamlessly transferred to real-world settings without compromising performance. To that end, model performances should not drop dramatically in these experiments compared to Experiment B. These experiments involved collaborative drilling of four corners of a 7×7 [cm] square on a flat 1-cm-thick plywood plate, positioned vertically, using a powered drill attached to the robot's end-effector.

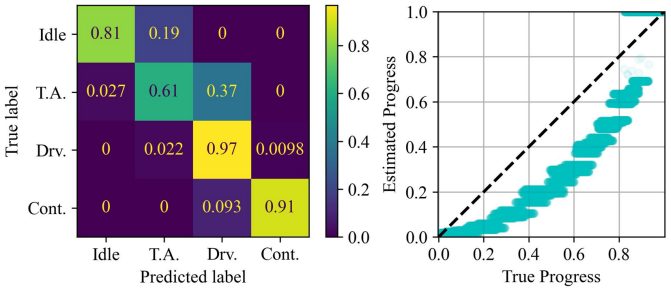

Fig. 10. The confusion matrix for subtask classification (left) and motion estimation performance (right) in Experiment C (real drilling in the physical world).

A. Protocol, Conditions, and Subjects

The drilling experiments in the physical world followed a protocol and conditions similar to Experiment B, except that the screen was solely utilized for providing instructions and visual feedback. Furthermore, only the C3 configuration (adaptive control with subtask detection and motion estimation) was employed (see Table II). The same subjects as Experiment B were included. Model performances were evaluated after the experiments.

B. Results

In the drilling experiments performed in the physical world, the system achieved a subtask classification accuracy of 83.61%, and a weighted F_1 Score of 0.8361, along with a motion estimation RMSE of 0.0948 and an R^2 Score of 0.96. These values, similar to those reported in Experiment B, are often considered adequate in most classification and regression tasks. Despite the possible differences in human behavior and/or trajectories between the virtual world and the real world, the performances of the models are sufficiently similar to those observed in Experiment B and adequate for successful adaptation in *Driving* and just before *Contact*. The performances achieved by the subtask detector and motion estimator are depicted in Fig. 10.

VI. Discussion and Conclusion

This study emphasizes that many contact-rich pHRI tasks in small-batch manufacturing, such as drilling, grinding, polishing, cutting, etc., can be deconstructed into subtasks, e.g., *Idle*, *Tool-Attachment*, *Driving*, and *Contact*. We define human intention as the specific subtask currently in progress and how much of the *Driving* subtask is progressed, which defines human motion intention more precisely and helps with handling possible *Contact* instabilities. We argue that if a robot can

recognize the subtask and estimate motion progress effectively, it can adapt its controller accordingly to improve both task efficiency and contact safety. To this end, we implemented subtask classification utilizing kinematic and kinetic signals. Following subtask detection, the admittance damping was adjusted to comply with human intention. Starting from an intermediate value to eliminate any jerky movements at the *Tool-Attachment*, a low admittance damping was used during the *Driving* to improve transparency (minimal resistance to human movement), while a high value was used during the *Contact* for stable interactions. One of the primary challenges we faced was the timely detection of *Contact* before it arose to mitigate potential stability issues, including oscillation or sudden bounce-back. To address this challenge, we introduced a motion estimator that allowed us to anticipate and proactively adapt the controller before the *Contact* subtask was initiated.

We conducted 3D experiments in virtual and physical worlds to validate our approach, focusing on collaborative drilling. The virtual environment enabled us to conduct simulated drilling experiments under different experimental conditions, reducing the number of trials for training the ML model in the physical world. Our experimental results showed that while subtask detection leads to significantly lower human effort and faster task execution, motion estimation allows the anticipation of *Contact* and early adaptation of the controller, leading to significantly smaller oscillations and better contact stability.

ACKNOWLEDGMENT

P.P.N. acknowledges the research fellowship provided by the KUIS AI Center.

REFERENCES

[1] Y. Aydin, O. Tokatli, V. Patoglu, and C. Basdogan, "Stable Physical Human-Robot Interaction Using Fractional Order Admittance Control," *IEEE Transactions on Haptics*, vol. 11, no. 3, pp. 464–475, 2018.

[2] Y. Aydin, D. Sirintuna, and C. Basdogan, "Towards collaborative drilling with a cobot using admittance controller," *Transactions of the Institute of Measurement and Control*, vol. 43, no. 8, pp. 1760–1773, 5 2021.

[3] A. Kucukyilmaz, T. M. Sezgin, and C. Basdogan, "Intention recognition for dynamic role exchange in haptic collaboration," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 58–68, 2013.

[4] Y. Aydin, N. Arghavani, and C. Basdogan, "A new control architecture for physical Human-Robot Interaction based on haptic communication," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2014, pp. 122–123.

[5] D. Sirintuna, Y. Aydin, O. Caldiran, O. Tokatli, V. Patoglu, and C. Basdogan, "A Variable-Fractional Order Admittance Controller for pHRI," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2020, pp. 10 162–10 168.

[6] Y. M. Hamad, Y. Aydin, and C. Basdogan, "Adaptive Human Force Scaling via Admittance Control for Physical Human-Robot Interaction," *IEEE Transactions on Haptics*, 2021.

[7] A. Madani, P. P. Niaz, B. Guler, Y. Aydin, and C. Basdogan, "Robot-Assisted Drilling on Curved Surfaces with Haptic Guidance under Adaptive Admittance Control," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2022-October. Institute of Electrical and Electronics Engineers Inc., 2022, pp. 3723–3730.

[8] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter, and H. Van Brussel, "Human-inspired robot assistant for fast point-to-point movements," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 4 2007, pp. 3639–3644.

[9] J. Zhao, S. Gong, B. Xie, Y. Duan, and Z. Zhang, "Human arm motion prediction in human-robot interaction based on a modified minimum jerk model," *Advanced Robotics*, vol. 35, no. 3-4, pp. 205–218, 2 2021.

[10] Q. Wang, W. Jiao, R. Yu, M. T. Johnson, and Y. M. Zhang, "Virtual Reality Robot-Assisted Welding Based on Human Intention Recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 799–808, 4 2020.

[11] C.-H. Lin, K.-J. Wang, A. A. Tadesse, and B. H. Woldegiorgis, "Human-robot collaboration empowered by hidden semi-Markov model for operator behaviour prediction in a smart assembly system," *Journal of Manufacturing Systems*, vol. 62, pp. 317–333, 1 2022.

[12] M. Vochten, T. De Laet, and J. De Schutter, "Generalizing demonstrated motion trajectories using coordinate-free shape descriptors," *Robotics and Autonomous Systems*, vol. 122, p. 103291, 12 2019.

[13] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 3 2018.

[14] C. V. Perico, J. de Schutter, and E. Aertbelien, "Learning robust manipulation tasks involving contact using trajectory parameterized probabilistic principal component analysis," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2020-October. IEEE, 10 2020, pp. 8336–8343.

[15] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent Advances in Robot Learning from Demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, 5 2020.

[16] R. Burlizzi, M. Vochten, J. De Schutter, and E. Aertbeliën, "Extending extrapolation capabilities of probabilistic motion models learned from human demonstrations using shape-preserving virtual demonstrations," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2022-October. IEEE, 10 2022, pp. 10 772–10 779.

[17] Z. Al-Saadi, D. Sirintuna, A. Kucukyilmaz, and C. Basdogan, "A Novel Haptic Feature Set for the Classification of Interactive Motor Behaviors in Collaborative Object Transfer," *IEEE Transactions on Haptics*, vol. 14, no. 2, pp. 384–395, 4 2021.

[18] L. Wang, G. Wang, S. Jia, A. Turner, and S. Ratchev, "Imitation learning for coordinated human–robot collaboration based on hidden state-space models," *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102310, 8 2022.

[19] O. Mazhar, S. Ramdani, B. Navarro, R. Passama, and A. Cherubini, "Towards Real-Time Physical Human-Robot Interaction Using Skeleton Information and Hand Gestures," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10 2018, pp. 1–6.

[20] H. Maithani, J. A. C. Ramon, and Y. Mezouar, "Predicting Human Intent for Cooperative Physical Human-Robot Interaction Tasks," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, vol. 2019-July. IEEE, 7 2019, pp. 1523–1528.

[21] D. Sirintuna, I. Ozdamar, Y. Aydin, and C. Basdogan, "Detecting Human Motion Intention during pHRI Using Artificial Neural Networks Trained by EMG Signals," in *29th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2020*, 2020, pp. 1280–1287.

[22] H. Liu and L. Wang, "Collision-free human-robot collaboration based on context awareness," *Robotics and Computer-Integrated Manufacturing*, vol. 67, p. 101997, 2 2021.

[23] J. Atkins and H. Lee, "MIntNet: Rapid Motion Intention Forecasting of Coupled Human-Robot Systems With Simulation-to-Real Autoregressive Neural Networks," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6363–6370, 10 2023.

[24] B. Guler, P. P. Niaz, A. Madani, Y. Aydin, and C. Basdogan, "An adaptive admittance controller for collaborative drilling with a robot based on subtask classification via deep learning," *Mechatronics*, vol. 86, p. 102851, 10 2022.

[25] R. W. Soukoreff and I. S. MacKenzie, "Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI," *International Journal of Human-Computer Studies*, vol. 61, no. 6, pp. 751–789, 12 2004.