Improving human robot collaboration through Force/Torque based learning for object manipulation

Al-Yacoub, A.; Zhao, Y. C.; Eaton, W.; Goh, Y. M.; Lohse, N.

DOI:
10.1016/j.rcim.2020.102111

Robotics and Computer–Integrated Manufacturing 69 (2021) 102111

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Improving human robot collaboration through Force/Torque based learning for object manipulation

A. Al-Yacoub [a,*], Y.C. Zhao [b], W. Eaton [a], Y.M. Goh [a], N. Lohse [a]

[a] Intelligent Automation Centre, Wolfson School, Loughborough University Epinal-way, Loughborough, LE11 3TU, UK
[b] Ewaybot, No. 18 Zhongguancun St, Building B, 19/F, Room 1930, Haidian District, Beijing 100080, PR China

ARTICLE INFO

Keywords:
Imitation learning
Human–Robot Collaboration
Random Forests regression
Gaussian mixture regression and ensemble-learning

ABSTRACT

Human–Robot Collaboration (HRC) is a term used to describe tasks in which robots and humans work together to achieve a goal. Unlike traditional industrial robots, collaborative robots need to be adaptive; able to alter their approach to better suit the situation and the needs of the human partner. As traditional programming techniques can struggle with the complexity required, an emerging approach is to learn a skill by observing human demonstration and imitating the motions; commonly known as Learning from Demonstration (LfD). In this work, we present a LfD methodology that combines an ensemble machine learning algorithm (i.e. Random Forest (RF)) with stochastic regression, using haptic information captured from human demonstration. The capabilities of the proposed method are evaluated using two collaborative tasks; co-manipulation of an object (where the human provides the guidance but the robot handles the objects weight) and collaborative assembly of simple interlocking parts. The proposed method is shown to be capable of imitation learning; interpreting human actions and producing equivalent robot motion across a diverse range of initial and final conditions. After verifying that ensemble machine learning can be utilised for real robotics problems, we propose a further extension utilising Weighted Random Forest (WRF) that attaches weights to each tree based on its performance. It is then shown that the WRF approach outperforms RF in HRC tasks.

1. Introduction

Over the last decade, there has been significant advancement in many areas of robotics. For collaborative robots (cobots) an essential area of improvement has been in safety, with commercial systems now in operation in many industrial roles. These typically work through the use of contact sensors (either tactile or force based), which detect resistance to motion and immediately stop the robot to prevent injury. Despite these improvements, there is still much to explore in the area. Although simply halting the robot upon detection of an external force is safe, it is unsuitable for any task involving direct Human–Robot Collaboration (HRC). For example, tasks such as holding components in place for a human to fasten together (i.e. a co-assembly task [1]) or moving heavy objects into a specific place under human guidance (i.e. a co-manipulation task [2,3], such as shown in Fig. 1), cannot be achieved if the robot must stop at the first external stimulus.

To overcome this limitation, the external forces provided by the human must be interpreted to produce the intended motion in a consistent and safe manner. Focusing on the intention to guide robots using physical contact with a human, the existing literature on human–robot collaboration can be roughly divided into three main approaches: **model-based** strategies, which depend on designing a control strategy based on an accurate mathematical model of the robots and the intended task, **human-based** strategies, which attempt to follow implicit human–human communication standards, and **learning-based** strategies, which use **Machine Learning (ML)** algorithms to attempt to generate models based on data alone. (Despite the use of this categorisation, there is overlap between the approaches and therefore the literature presented below is organised based on its major contributions.)

The remainder of this paper is organised as follows: this section covers the background literature related to HRC. Section 2 presents the problem statement and Section 3 describes the proposed approach. Section 5 describes the experimental setups and data collection. Then, the experimental results and validation examples are discussed in Section 6. Finally, the conclusions and future work are presented in Section 7.

1.1. Model-based strategies

Model-based control strategies were commonly used for earlier examples of HRC. Aside from creating the model, the challenge in

* Corresponding author.
E-mail address: a.al-yacoub@lboro.ac.uk (A. Al-Yacoub).

https://doi.org/10.1016/j.rcim.2020.102111
Received 29 August 2020; Received in revised form 1 December 2020; Accepted 6 December 2020
Available online 4 January 2021

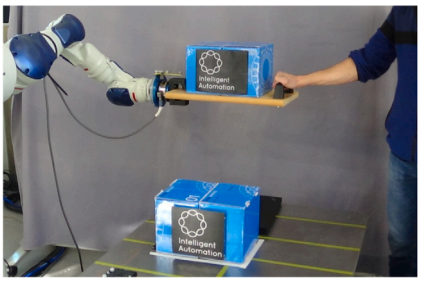

Fig. 1. Co-manipulation of an object is a common Human–Robot Collaborative task. In this scenario, the robot is intended to handle the weight of the object and follow the movements of the human detected through force sensors. The role of the human operator is to provide guidance, as human perception and understanding of the task should be far greater.

using this type of control is tuning the parameters to best fit the actual requirements. As any HRC task includes a human by definition, many solutions populate the model parameters based on tests with humans. For example, Ikeura and Inooka [4] proposed an adaptive impedance controller, where impedance parameters were experimentally determined by analysing two human operators performing a cooperative task. This created a model that could adaptively modify its parameters based on the change in rate of the measured forces, producing a similar force output as a human. Similarly, [5,6] and [7] proposed various elements of an admittance control strategy based on the apparent mechanical impedance of the manipulated object. By detecting the manually applied input force, the intended motion of the object could be inferred. This approach was successful in demonstrating a multi-robot system that could collaborate with a human through contact forces alone.

This was expanded on in [8], which introduced a two-level control scheme in which an admittance controller was driven by a higher level reflex control, triggered using a force-based threshold. As pure admittance controllers perceive any external force as an input command, the force threshold made the system more predictable as weak forces did not trigger motion. In addition, overcoming the initial resistance of the system was shown to better represent the human expectation of load when working with a collaborative partner. This was further expanded on in [9], where the control gain of the adaptive impedance controller was actively modified using gradient descent optimisation. The differentiation of the force information was also exploited to estimate the human intention in cooperative tasks and update the robot velocity control law accordingly. As an additional modification, motivated by biological studies, the stiffness parameter of the system was also actively changed to achieve a more human-like motion. These methods in combination were suggested as being highly effective in achieving compliant motion during collaboration.

Following this, other methods of adjusting model parameters have become more popular. Duchaine and Gosselin [10] and Ikeura et al. [11], proposed to adjust model damping parameters online optimally by minimising a cost function. Tsumugiwa et al. [12] proposed a real-time estimation approach, using the known stiffness of human arms to perform an online adjustment of the robot stiffness coefficients. To allow for a higher degree of compliance during HRC, Ficuciello et al. proposed a co-manipulation controller which combined impedance modulation (e.g. feedforward control) with the known kinematic redundancy of the robot [13].

1.2. Human-based strategies

Rather than considering HRC as a task, Klingspor et al. [14] recommended that human–robot physical interaction is a form of communication, and that greater co-operation could be achieved if the robot model was designed to follow implicit human–human communication standards. Accordingly, many studies have designed human–robot collaborative systems based on human–human collaboration. As outlined above, many model-based strategies already attempted to use human–human interaction in designing the system model. Rahman et al. [15] introduced a variable impedance control for HRC (specifically a push–pull task) in which the stiffness and damping parameters varied based on the analysis of human–human collaborative task (two humans jointly carrying an object). In a similar setup, Ikeura et al. [11] presented the use of stiffness parameters that minimise a cost function based on a human–human collaborative task.

While the aforementioned methods attempted to produce a system that could interact with humans in an adaptive way, in each case the actual model does not change, with only the parameters being tuned to accommodate the human input. As the model is fixed, control-based strategies are effectively 'task oriented', requiring that the controller be redesigned/reprogrammed should any aspect (object, task, robot or even user) change beyond what variance can be accommodated by altering parameters.

Recognising that conventional control strategies were difficult to apply, Hayashibara et al. [16] proposed that human co-operative behaviour could be modelled as a set of co-operative *rules*. Through experimentation, a set of rules was established for humans collaborating on a lifting task. These rules were then used to create a control system where the model changed based on the rules, which was further applied to a prototype collaborative robot. This robot demonstrated that a rule based approach was valid, albeit with poor low-frequency performance. This idea was expanded upon in [17], where the frequency of a repetitive collaborative task was identified in an effort to correct over time. It was shown that identifying repetition was sufficient for the controller to predict upcoming motion and act accordingly, improving the HRC effort.

More recently, Reed and Peshkin[18] agreed with the principal that a collaborative task can be considered a form of communication, and compared human performance when working alone to collaborating with a partner; specifically on a task with only haptic feedback. By adaptively regulating impedance of a robot, it was shown that human operators could not differentiate between a human partner and a robot partner, and in both cases completed the task more quickly than when working alone.

Perhaps the task which best demonstrates how collaboration can be viewed as a form of communication is 'handover', where actors co-ordinate to transfer control of an object between them. In addition to the physical transfer of an object, the understanding and recognition of when control has been passed are also required for the task to be successful. Strabala et al. [19] identified that humans are particularly effective at this, capable of achieving handover in differing conditions, even when it is unexpected. This suggests that humans share common in-built rules for this procedure, which can be adapted for human–robot interaction. By studying the physical and social-cognitive elements of handover (such as the signals and cues humans rely on) Strabala et al. [19] identified a *'what, when and where'* approach, in which human establish information to make an exchange. Human–robot handover based on this approach was then experimentally validated; holding out the robot arm (where), displaying the object for exchange (what) and waiting the same amount of time as a human once the other party had grasped the object before releasing control (when).

1.3. Learning-based strategies

Despite achieving better HRC, simply adjusting model parameters to better emulate humans can only bring so much improvement. As a collaborative task is essentially communication based, a traditional fixed control model is unlikely to be optimal. To properly capture the complexity of human interaction, recent literature predominantly uses

ML to generate a model which captures the entire scenario and allows for enhanced robot capabilities.

There is already a wide range of literature on ML for human–robot collaboration tasks. A learning-based approach for human–robot co-manipulation, in which a robot would be guided by human input force alone was presented in [20]. The controller combined Multi-Criteria Decision Making (MCDM) with synergy strategies (planning approaches), and was shown to be capable of correctly interpreting human intentions when evaluated for both 2DoF and 6DoF contexts.

In [21] a co-operative lifting task was demonstrated in which a human leader was assisted by a robotic follower, controlled by another human via teleoperation. The remote controls were recorded and Hidden Markov Models (HMM) were used to encapsulate the robot motion and the sensed forces. Gaussian Mixture Regression (GMR) was then used to generate the reference force required to reproduce collaborative robot behaviours. Not only was ML shown to be capable of producing a working control system, an additional benefit was the ability to implement capabilities not easily possible using other methods; with the robot becoming more proactive as it was further trained, going beyond simply following the human by incorporating predictive behaviour. Another interesting probabilistic approach was introduced by Duque et al. [22], in which a robot successfully carried out an assembly process using a combination of LfD, Gaussian Mixture Model (GMM), Task-Parameterised GMM for control, based on data acquired using machine vision.

Ude et al. [23] specifically sought to create a robot capable of responding to more extreme examples of a task, which were not previously encountered within the training set. Working with Dynamic Movement Primitives (DMPs) (a general approach for representing any recorded movement as a set of differential equations) the use of local weighting via Gaussian Process Regression (GPR) was suggested to best fit the human demonstration of a task, whilst also allowing the robot to respond to new situations. This highlighted the need for an accurate clustering approach that determines the desired behaviour in order to achieve a general solution.

Other solutions have suggested introducing frameworks to extend the capabilities of a system beyond a single task. Medina et al. [24] again uses ML to teach a robot collaborative actions. To allow these actions to be more complex, it proposes segmenting any complex task into simple elements, each of which can be accomplished more easily when attempted in isolation. To split the task, each element is semantically labelled through verbal communication with the human instructor. Should the robot not understand, it can request the name of the current task to help isolate the stage in which it should be operating, as well as further human demonstrations if needed. The demonstrated tasks were represented by a primitive graph and a primitive tree using HMMs that were incrementally updated during reproduction. The proposed framework aimed to improve the generalisation capabilities of the robot and was shown capable of completing more complex tasks when steps were attempted individually.

Sheng et al. [25] also presented an integrated human–robot collaborative framework. Rather than break each task into stages, the intention was for two control strategies to work concurrently; a reactive controller which attempts to steady the system, and a proactive controller which attempts to take proactive actions based on human motion prediction. This is demonstrated in a co-manipulation task in which a human works with a humanoid robot to lift a table. Both holding the table in a steady position (reactive) and manipulating the table (proactive) were trained through imitation learning. Confidence in the motion predictor is used to automatically switch between controller modes, and the results demonstrate that such an approach is successful for collaborative manipulation.

For HRC tasks with much greater variation, the changes in environment must also be modelled. Cui et al. [26] proposed changes in the environment (such as a larger manipulation object) not only affect how the robot must adapt to the task, but also alter how the human responds, which the robot must account for. The main idea in this work was to include variation in the handled object's size and include this variation in the final probabilistic model, in order to improve the generalisation of the model. A learning-from-demonstration framework was suggested, with training samples of both human and robot movement to finish a task under different environmental conditions. To encapsulate the additional environmental information, Environment-Adaptive Interaction Primitives (EAIP) were suggested, which are an extension to the well-known Interaction Primitives. This was shown to improve both collaboration and generalisation to new conditions.

As a final consideration, as this work intends to rely solely on haptic sensors, efforts made to enhance the precision of force detection are also worth considering. Typically Force/Torque (FT) sensors are quite sizeable and will reduce the already limited payload of smaller robots if required for every task. Attempting to mitigate the need for a dedicated FT sensor, Berger et al. [27] proposed that the inbuilt force sensors within a cobot could provide a similarly accurate measurement of force, if ML was used to map the readings of a dedicated FT sensor to internal force readings. After comparing various inputs and four different neural networks, the best result is obtained by using Transfer Entropy (TE) features and the recurrent neural network architecture. This was shown to have significantly improved the force estimation when compared to the robot native controller, producing similar accuracy to the purpose made FT sensor used for validation. With regards to the neural network architecture, Suleman et al. [28] have investigated the most suitable implementation for LfD applications, concluding that Elman networks and Nonlinear-Autoregressive-Exogenous-Model networks have the best performance, regardless of the high computational requirements.

1.4. LfD for HRC

This paper considers the HRC task of co-manipulation, in which a human and a robot transport an object from a starting point to a target point within a known workspace. The intended purpose of the collaboration is for the robot to support the weight of the transported load while simultaneously responding to human guidance. This interactivity will be achieved by using only haptic sensor data, through which the human must convey intentions to the robot. To achieve this, a control approach is required that can interpret the haptic information in the context of the required action.

As shown in the literature, safe haptic-based HRC could be implemented using classical or adaptive control strategies. However, this relies on having an accurate model of the human–robot interaction for each task. Creating such models requires a deep understanding of the system dynamics and requires assumptions to be made which may not hold true in practice. In addition to signal noise, hard coding HRC systems based on haptic information is also difficult due to signal ambiguity, whereby manual manipulation of the robot will produce unpredictable force components for the robot to interpret. As human inputs are imprecise, input signals are unlikely to be exclusive to any single intended action, making it difficult to determine the intention of the human. Therefore, a research gap exists in understanding haptic input signals when the robot is in direct contact with the surrounding environment [29].

The chosen method of bridging this gap is to employ Learning from Demonstration (LfD) [30]. LfD is an intuitive method of transferring human knowledge about a task using data directly from human demonstration. By creating a mapping between the human body and the equivalent joints of a robot, data captured on human motions can be fed into a ML algorithm to generate more human-like logic when performing a task [31]. This approach can be particularly useful for situations where conventional control strategies struggle. Teramae et al. [32] proposed using an LfD approach to work with non-linear systems in scenarios with minimal sensor feedback; specifically to teach an impact skill (i.e. hammering) to a pneumatically actuated 1DoF

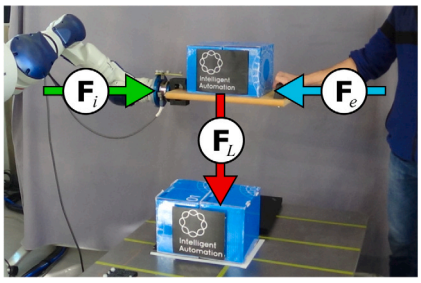

Fig. 2. Depiction of main forces within HRC task, consisting of the external human force (F_e), internal robot forces (F_i) and load force (F_L) of the manipulated object.

robot arm. The mapping between human and robot was established by directly mimicking the human arm motion via a human–robot interface, effectively creating a form of teleoperation. By having the human watch the robot arm while controlling the end effector, human-in-the-loop learning was achieved without the need for external sensors monitoring the robot. The results demonstrated that the hammering skill was learnt swiftly, validating the use of LfD for skill transfer.

For this work the intention is to use LfD for HRC, where the taught skills must not only mimic the human demonstration, but must also interpret information from co-operating humans. Although LfD has previously been used in HRC tasks, the majority of the research has focused on visual or aural data, with minimal work on haptic interaction. As the robot must interact with a human via physical connection (either directly or via a shared load), this requires more work on interpreting the complex force clues and signals that humans rely on.

In this paper we present an ensemble ML approach to learn a HRC task in a LfD framework. The foundation for this approach is Random Forest (RF), a popular ensemble learning method due to its simplicity, robustness under variance, and its capability to generalise well in novel scenarios. Furthermore, this paper also proposes to extend the RF approach by identifying better performing trees and weighting their influence within the final output.

2. Problem definition

To capture a skill using a learning-based approach, the input forces must be known in the context of the intended reaction. This requires that the co-manipulation task be demonstrated, with the FT sensor readings and motion trajectories recorded. Once collected, the data (**D**) can formally be expressed as shown in Eqs. (1) and (2):

$$\mathbf{D} = \left\{(\mathbf{f}_1, \mathbf{y}_1), (\mathbf{f}_2, \mathbf{y}_2), \ldots, (\mathbf{f}_N, \mathbf{y}_N)\right\} = \{(\mathbf{F}, \mathbf{Y})\} \tag{1}$$

$$\begin{aligned} \mathbf{f}_i &= \left[f_{xi}, f_{yi}, f_{zi}, t_{xi}, t_{yi}, t_{zi}\right] \\ \mathbf{y}_i &= \left[y_{xi}, y_{yi}, y_{zi}\right] \end{aligned} \tag{2}$$

where:

$\mathbf{f}_i$ = The $ith(X, Y, Z)$ components of the FT feature vector

$\mathbf{y}_i$ = The $ith(X, Y, Z)$ components of targeted trajectory

N = The number of data points

$\mathbf{F}$ = The input features, such that $\mathbf{F} \in \mathbb{R}^{6\times N}$

$\mathbf{Y}$ = The target trajectory, such that $\mathbf{Y} \in \mathbb{R}^{3\times N}$

Fig. 2 shows the forces for a co-manipulation task in which a human and robot share a load. The human haptic input is modelled as an external force ($\mathbf{F}_e$), with the robot internal forces and load force ($\mathbf{F}_i$ and $\mathbf{F}_L$) being compensated for using the method introduced in [33]. Hence, the HRC can be modelled as a spring–mass system, which can be formalised as shown in Eq. (3):

$$\Delta\ddot{\mathbf{Y}} = \boldsymbol{K}^p \Delta\mathbf{Y} - \boldsymbol{K}^v \Delta\dot{\mathbf{Y}} - \boldsymbol{F} \tag{3}$$

Where $\boldsymbol{K}^p$ is the compliance matrix and $\boldsymbol{K}^v$ is the damping factor. As jerky movements are not desirable (especially when a human is physically interacting with the robot) it was assumed that acceleration $\Delta\ddot{\mathbf{Y}}$ is equal to zero for each displacement $\Delta\mathbf{Y}$. Therefore, each sub-displacement, which is the displacement between two subsequent samples of synchronised sensory data, has a fixed velocity $\Delta\dot{\mathbf{Y}}$ based on the human generalised input force $\boldsymbol{F}$.

This allows the problem to be defined as using the input force $\boldsymbol{F}$ to predict the sub-displacement velocity $\Delta\dot{\mathbf{Y}}$ (as shown in Eq. (4)). Within the context of ML, the $\mathbf{X}$ vector represents the system observation and $\Delta\dot{\mathbf{Y}}$ represents the expected output $\mathbf{Y}$. Hence, the formal problem description can be presented as shown in Eq. (5):

$$\boldsymbol{F} \rightarrow \Delta\dot{\mathbf{Y}} \tag{4}$$

$$\begin{aligned} \mathbf{X} : \{F_x, F_y, F_z, T_x, T_y, T_z\}_i \rightarrow \\ \Delta\dot{\mathbf{Y}} : \{\Delta\dot{\mathbf{y}}_x, \Delta\dot{\mathbf{y}}_y, \Delta\dot{\mathbf{y}}_z\}_i \end{aligned} \tag{5}$$

Where $\mathbf{X} \in \mathbb{R}^{6\times N}$ and $\Delta\dot{\mathbf{Y}} \in \mathbb{R}^{3\times N}$.

3. LfD framework

As aforementioned, LfD will be used to capture desired human capabilities and transfer these skills into an industrial robot. Fig. 3 shows the proposed LfD framework, consisting of both *offline* and *online* phases. The framework begins with the offline phase, where skills are captured from demonstration. Two different types of demonstrations

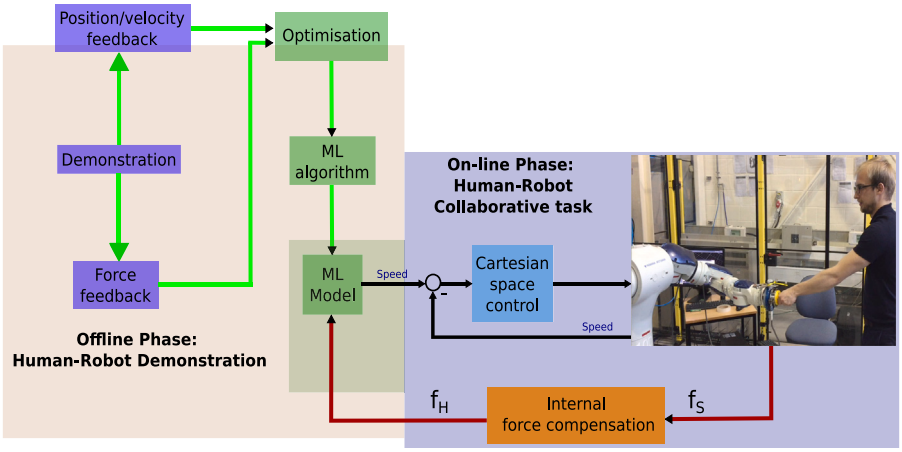

Fig. 3. Proposed LfD framework for HRC object manipulation. A velocity controller commands the robot in Cartesian space, with the reference velocity estimated using a ML model.

will be used (kinesthetic learning and human–human collaboration) with both approaches detailed in Section 5.2. In each case, synchronised data will be collected on the system input (i.e. FT) and resultant output (position/velocity feedback). The ML algorithm is then used to map the input/output relationship to generate the ML model, all within Cartesian space.

The online phase can be considered a closed-loop velocity controller. Beginning with input FT data from the sensors, an Internal Force Compensation (IFC) algorithm models the forces present within the robot. This is then subtracted from the raw FT input data, to better estimate the actual human input force F_e. The force data is then passed to the ML model which outputs the appropriate velocity command. As a final step, rather than directly feed this command to the robot, a PI controller is used to regulate the process. This helps to produce a smoother response and avoid jerky movements caused by sudden changes in input force. Furthermore, an advantage of this approach is that the PI gain values can be application specific; switching between tasks which require large motions and tasks which require delicate movements, all using the same underlying ML model.

4. Machine Learning (ML)

As the ML model sits at the centre of the proposed framework, the selection of the algorithm is critical to the systems success. ML algorithms are essentially a method of determining a function which maps input variables to an output. Learning a specific task requires optimising a cost function to identify the models' parameters that guarantee the best performance on training and validation datasets. As different algorithms are only suitable for certain types of data, the application scenario can often restrict which approaches can be used.

Returning to the literature, the majority attempt to capture human demonstration as a series of movements, recording the spatial data over time. This necessitates ML approaches that can handle sequential data. However, not all HRC applications are sequential and in this work the assumption is made that spatial data alone should be sufficient for haptic HRC. As spatial data can be matched directly to lower-level data (such as human force input) it may even generalise better than sequential processes in haptic based tasks. Assuming that only spatial and force information requires mapping, the problem definition (stated in Eqs. (4) and (5)) shows that the scenario is a regression problem. Typically, for regression problems the mapping relationship between input/output would be determined using either a statistical model (such as Linear Regression) or a probabilistic model (such as Gaussian Mixture Regression (GMR)). However, for LfD such methods can produce models which are over-fitted to the demonstration. For HRC, an approach is required that can generalise well for changing conditions, such as new users, objects and uncertainty.

The popular Random Forest (RF) method appears to be a strong candidate for addressing these problems. RF is an ensemble ML method, which averages the output from multiple predictors (i.e. decorrelated decision trees) simultaneously to produce an overall prediction output. Chandra et al. [34] state that multiple predictors have better formal and empirical performance in comparison with single predictor models (in terms of generalisation and robustness against bias) which should produce more responsive and safer robot behaviour. RF is also simple to implement, with its only main disadvantage being the increased computational complexity that comes with using multiple predictors at once.

Due to its capability to handle high-dimensional data and simplicity to implement, RF is highly popular. Despite these advantages, to the authors' knowledge the ensemble method has not previously been used to achieve LfD. Accordingly, this paper proposes using RF for the ML aspect of the framework. The capability of the proposed approach will be experimentally validated by performing a co-assembly task. The following section briefly summarises the RF approach within the context of this work (for more interested readers, the formal definition of RF is described in [35]).

4.1. Random forest regression

Random Forests are so named as they consist of a number of decision trees, with the overall prediction based on the average prediction of each individual tree. For standard decision trees, each node is split using the best split of all variables present at that node. However, this approach would result each tree being identical, and the overall forest having the same prediction as each tree. Therefore, the *'random'* element of RF is that each node is split based on the best variable amongst a randomly chosen subset at each node. This variability (combined with the subsequent averaging) reduces variance in the final prediction, as well as giving RF its ability to generalise better than other methods. To encapsulate the difference between trees, a 'characterisation' vector defines the following for each tree:

- Split variable subset for each node
- Cut-points (thresholds) for each node
- Terminal node size

For this work, the total variable set is shown in Eq. (5). The split variables are a randomly selected number of features (d) such that $d \leq M$, where M is the total number of captured features. The split-point among the d features then defines the nodes in each decision tree. As the terminal node size prevents subsequent splits when the remaining feature set becomes too small, it controls the 'height' of each tree. After growing the RF model on the training dataset, predictions can be made for each new observation, using Eq. (6):

$$h(\mathbf{o}, \psi) = \frac{1}{T} \sum_{t=1}^{T} \sum_{J}^{j=1} c_{j,t} I(\mathbf{o} \in \mathbf{R}_{j,t}, \psi_t) \tag{6}$$

where:

T = The total number of trees

t = An individual tree with distinct characterisation

ψ_t = The characterisation vector

$c_{j,t}$ = The average output in the given dataset

J = The number of regions (splits) in the tth tree

$\mathbf{R}_{j,t}$ = Region (split) $\mathbf{j}$ in the tth tree

$\mathbf{o}$ = The new observation vector

Finally, $\mathbf{I}()$ is a discrete identity function that returns 1 only if the new observation ($\mathbf{o}$) belongs to region $\mathbf{R}_{j,t}$. As the output of any RF is limited to a single scalar value, separate models need to be trained for each velocity component (i.e. outputs in X, Y, Z dimensions). Each model aims to map the generalised force vector into a velocity component in the Cartesian space $\mathbb{R}^6 \rightarrow \mathbb{R}^1$.

4.2. Weighted Random Forest (WRF) regression

In a conventional RF implementation, all trees in the forest contribute equally to the final prediction. However as the characterisation features (ψ_t) for each tree have been specified separately (each tree is effectively trained on a different bagged dataset) they vary in performance, with some trees consistently outperforming others. As such, identifying these trees and increasing their weighting has the potential to improve the overall performance.

In this paper, we present a new weighting mechanism which extends RF from a collaborative model into a competitor model, since the output of RF model is the average output of all trees within the RF model, while the WRF weight each tree based on trees performance on test dataset. In which, a stochastic weighting approach was employed to weight trees according to their Root Mean Square Error (RMSE), with trees that perform better on unseen test data receiving strong weights. The proposed WRF, unlike the work presented by Booth et al. [36], Zhang et al. [37] and Winham et al. [38], utilised RMSE

to calculate the weights at the tree levels after the training of the RF. On the one hand, Booth et al. [36] proposed to estimate the trees' weights during the training using an index that combines RMSE and the Mean Absolute Percentage Error (MAPE) for a regression problem. On the other hand, Zhang et al. [37] and Winham et al. [38] proposed weighting mechanisms for classification problem that focus of the data level(split mechanisms) then use classification accuracy to estimates some weights. The Weighted Random Forest (WRF) prediction can be calculated as shown in Eq. (7):

$$P^*(\mathbf{X}_i) = \sum_{j=1}^{M} w_j \cdot h_j(\mathbf{X}_i) + C_0 \tag{7}$$

where:

P^* = the overall prediction

$\mathbf{X}_i$ = the feature observation vector

w_j = the weight of tree j

h_j = the prediction of tree j

C_0 = a constant term for the forest

As Eq. (7) is a linear equation, the selection of weights can be seen as a linear regression problem, where the trees prediction vector $\mathbf{H}$ is the feature vector, and the goal is to find a weighting vector $\mathbf{W} = [w_0, \dots, w_j, \dots, w_T]$ which minimises a cost function $J(\mathbf{W})$. A common choice of cost function is the Mean Square Error (MSE), as shown in Eq. (8), where $\mathbf{Y}$ is the target output:

$$J(W) = \frac{1}{2}(\mathbf{Y} - \mathbf{W}\,\mathbf{H}(\mathbf{X}))^H(\mathbf{Y} - \mathbf{W}\,\mathbf{H}(\mathbf{X})) \tag{8}$$

Finding the optimal weights based on all points in the dataset can be computationally expensive, especially for online applications. To overcome this problem, weights can be updated incrementally, based on one observation point or a batch of observations at one time. The pseudo code in Algorithm 4.1 defines the implementation of finding the optimal weights by minimising a cost function, where $w_j \leftarrow w_j + min\left|J(w, h_j(x_i))\right|$. In this pseudo code, there are two loops:

- The outer loop which iterates between $0 - N$ representing the number of observations in the training dataset
- The inner loop which adjust trees' weights, so as to iterate over all the trees in the random forest.

The outer loop convergence condition can be satisfied by either reaching the maximum number of iterations K_{max} or when the prediction error is less than a pre-defined tolerance value C_{th}. The main advantage of this method is that it can converge quickly, as well as allowing for a larger threshold to prevent over fitting.

Algorithm 4.1 Stochastic weighted RF

1. Find Weights $H(\mathbf{X})$,$\mathbf{Y}$,α,K_{max}, C_{th}
2. **Input:** $H(\mathbf{X})$: RF model prediction for the given features $\mathbf{X}$, $\mathbf{Y}$: target output,K_{max} is the maximum number of iteration and C_{th} is the cost threshold
3. $k = 0$ % where x is the number of iterations
4. While $k \leq K_{max}$ OR $J(\mathbf{W}) \leq C_{th}$ do :
5. $k = k + 1$
6. Calculate cost $J(\mathbf{W})$ using Equation (8)
7. For $i = 1$ `to` N do:
8. For $j = 1$ `to` T do :
9. Calculate $w_j \leftarrow w_j + \alpha(Y(x_i) - W\mathbf{H}(x_i))\,h_j(x_i)$
10. **Return**: W

It is noticeable from the pseudo code in Algorithm 4.1 that the first input is a prediction for the features of $\mathbf{X}$. For this reason WRF is only an extension to RF, as a full Random Forest model is a prerequisite. Furthermore, as the weight selection is performance-based, using the same dataset for both RF and WRF creation would likely bias the model [39]. To overcome this problem, it is recommended to use separate datasets for each phase.

Matching the RF input and output stated in Eq. (5) the WRF input feature vector is ($\mathbf{X}$) and the output considered as $\Delta\dot{\mathbf{Y}}$ which is the velocity in x, y and z directions. As with RF, the output is limited to a single scalar value. Hence, three different models need to be trained for each velocity component, with each model aiming to map the generalised force into a velocity component in the Cartesian space $\mathbb{R}^6 \rightarrow \mathbb{R}^1$.

In this section, a new WRF approach was proposed for LfD application. The proposed approach is designed for robots LfD to capture a co-manipulation task. Finally, the methods mentioned above were used to model the co-manipulation task and compared with the proposed WRF.

5. Experimental setup

The following section covers the experimental setup used for practical elements of the work; including the demonstration scenarios for training, the assessment scenarios used for model validation, and the data collection methods used throughout.

5.1. Data collection

As previously stated, this paper aims to use a LfD framework to capture human co-manipulation skills and replicate them using an industrial robot. In order to encapsulate the demonstration of the skill by a human, an experimental setup was required that could capture data for both the haptic input (using FT sensors) as well as the resultant trajectories of the object being manipulated. For both demonstration and validation, the setup centred around a PC workstation running Ubuntu 14.04. The Robot Operating System (ROS — specifically ROS-Industrial Indigo) was used as a framework for communication and processing. All other test equipment (robot controller, tracking system and the FT) were connected to the PC via Ethernet. The PC workstation was also responsible for data collection (for both training and testing) as well as controlling the robot manipulator during task execution.

A Schunk FTD-Delta SI-660-60 – a rigid 6-axis FT sensor[1] – was used to capture all haptic input data throughout. For human–human collaboration each side of the FT sensor was fitted with a handle, allowing a pair of humans to demonstrate the task, as shown in Fig. 4 In comparison, Fig. 5 shows the FT sensor mounted on an industrial robot arm for HRC validation. This particular setup also shows how the assembly parts are attached to the robot end effector, with the intention of validating the generalisation capability of the proposed approach in a collaborative assembly task. Fig. 6 better illustrates the assembly process, showing both the components and operator handle connected to the robot end effector.

The Motoman SDA10D dual-arm robot[2] was chosen as a suitable industrial robot collaborator. This was primarily due to its high Degrees of Freedom (DoF), which allows the robot to replicate human motion without being overly-restricted by joint limits. In addition, each arm of the robot is capable of lifting a 10 kg payload, which represents a mass heavy enough that robot assistance could be necessary.

During the task demonstration, an asymmetric marker was mounted on the FT sensor and the Cartesian position was captured using a VICON tracking system with 8 *Vantage* cameras.[3] Training the system using

[1] https://www.schunk.com/sg_en/gripping-systems/product/30876-ftd-delta-si-660-60/.

[2] https://www.motoman.com/en-us/products/robots/industrial/assembly-handling/sda-series/sda10d.

[3] https://www.vicon.com/products/camera-systems/vantage.

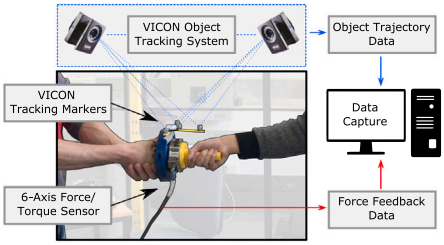

Fig. 4. Data capture setup for human–human collaboration.

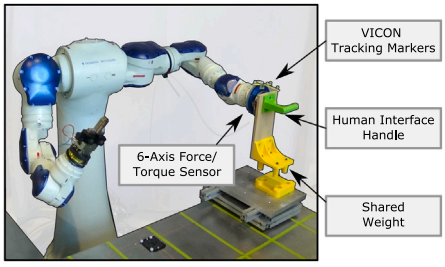

Fig. 5. Motoman SDA20D Dual-arm robot with 15 DoF. (7 DoF per arm and 1 additional DoF for the body rotation).

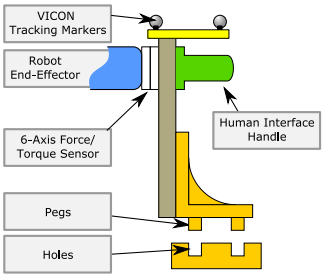

Fig. 6. Diagram of assembly components used in the HRC co-assembly task.

the external tracking system allowed the human–human trajectory to be accurately recorded. However, the collected position and orientation data of the tool were in the VICON coordinate system. As robot commands must be in the robot's coordinate system, transformation between the two systems was required. This was achieved using a second marker, mounted in a fixed position relative to the robot's base coordinate. The transformation between the VICON coordinate system was calculated and used to project all position/velocity data into the robot's coordinates, as shown in Fig. 7. The same tracking system was also mounted on the robot end effector to monitor position during validation of the HRC tasks.

5.2. Training scenarios

To learn co-manipulation behaviour, data was collected using two different demonstration methods:

- human–human
- Human–Robot (kinesthetic).

For both methods, five separate demonstrations were conducted in which the force, torque, position and velocity data were synchronised and recorded. Shown in Fig. 8, each demonstration simply consisted of performing 6 motions, in which the load moved along each of the cardinal directions of the workspace $(-x, +x, -y, +y, -z, +z)$ before returning to the origin. To prevent the follower anticipating the upcoming movement (and to ensure the movements were decoupled as

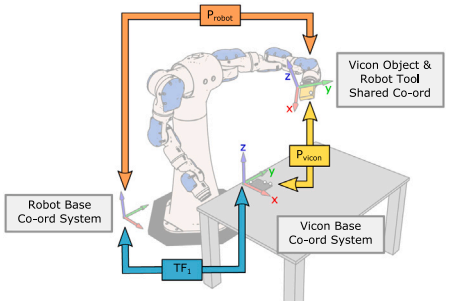

Fig. 7. Transformation between robot base and VICON base frames. TF_1 is the estimated homogeneous transformation matrix, P_{vicon} is the object position in the VICON origin frame and P_{robot} is the calculated object position in the robot base frame, which can be calculated as $P_{robot} = TF_1\, P_{vicon}$.

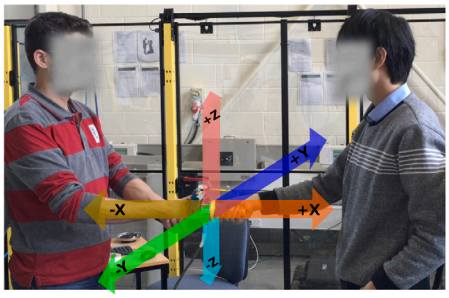

Fig. 8. human–human demonstration: two human operators holding a common object. The human operator on the right guides the demonstration by moving in each principal direction, in a random order.

much as possible), the order in which the motions was performed was randomised each time.

In the human–human demonstration, one operator knows the desired trajectory of the object (master), and the other operator does not (follower). The intention was to capture the behaviour of the follower, while they attempt to interpret the haptic clues and try to move accordingly. In the human–robot demonstration, the roles remained the same with the only difference being that the follower was acting through the robot via teleoperation based on the leader's verbal commands.

Due to the robot's rigidity, it is believed that the FT features collected using the robot will be more relevant for the learning process. This should result in a better co-manipulation task, in comparison with using the data from the human–human demonstration alone. This is mainly due to the robot's rigidity effectively acting as a low-pass filter, removing the high frequency position fluctuation typical in humans attempting to hold an object in a static position. However, by also capturing the human follower behaviour, the hypothesis is that the robot will imitate human-like compliant behaviour more closely, making both sources of data equally important.

Both RF and WRF models were fitted using the data from the demonstrations in an off-line phase, which mapped the FT information onto velocity in the Cartesian space. This can be represented as $F \rightarrow V$, where $F \in \mathbf{R}^6$ and $V \in \mathbf{R}^3$. Returning to Eq. (3), this mapping represents $\boldsymbol{K}^p$ and $\boldsymbol{K}^v$, which allows $\Delta\mathbf{Y}$ (the required input to the robot velocity controller) to be calculated.

Previously highlighted in Section 4.2, WRF is an extension to RF and therefore requires a full RF model as a prerequisite. As continuously using the same training data could bias the output, the WRF model should ideally be trained using a different dataset to that which was used for the RF model. In repeating each demonstration five times, the intention was to provide entirely separate datasets for this purpose. As shown in Fig. 9, the role of each dataset was as follows:

1. Train the decision trees for the original RF.

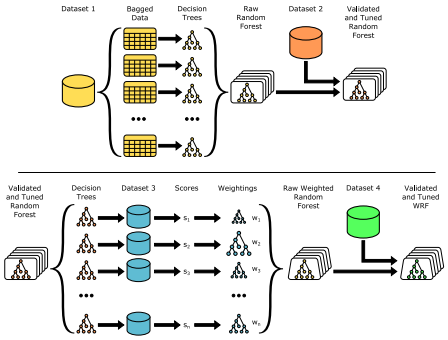

Fig. 9. Training process for both RF and WRF models, using Datasets 1–4.

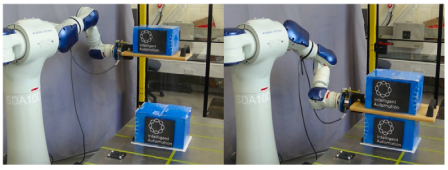

Fig. 10. The start point and goal position for the co-manipulation task.

2. Validate the RF and tune parameters.
3. Generate scores and weighting for each of the original decision trees in order to produce the WRF model.
4. Validate the WRF and tune parameters.
5. Test and compare the performance of the two approaches.

Once fitted, both RF and WRF models were compared and evaluated. As a final point, the weightings of each tree within WRF could alternatively be calculated online, making incremental changes based directly on the human–robot interaction. However, this has not been explored in this paper and remains a topic for future exploration.

5.3. Validation scenarios

To assess whether the skill has been captured successfully, both the RF and WRF approaches were validated experimentally. Despite both models only being trained on data from co-manipulation demonstrations, the assessment consisted of both co-manipulation and co-assembly tasks. This was intended to assess the hypothesis that the proposed approach can generalise well when compared to other ML approaches. The validation scenarios were:

1. A co-manipulation task that involves moving a heavy object between two points in the robot workspace; intended to validate the reproduction of the demonstrated skill.
2. A co-assembly task that involves attaching a movable component to a static component; intended to test the generalisation capabilities of the proposed approach.

In the co-manipulation task, a human operator was asked to work collaboratively with the robot to move an object between two known points in the workspace. The operator's goal was to guide the robot such that the object (a box) would come to be placed upon another box fixed to a workbench, as shown in Fig. 10. The role of the robot was to share the weight of the object, while following a trajectory based on the operators' input force. For validation purposes, the starting point was a fixed location known to the robot controller, allowing the same position at the beginning of each trial.

For the co-assembly task, the human operator was again asked to guide the robot from a random starting point in the robot workspace to a fixed target location, with the robot responding in a similar way.

Fig. 11. Components used for co-assembly task: (*A*) A 3D printed plate with four cylindrical holes of 28 mm diameter. (*B*) A 3D printed plate with four pegs each of 25 mm diameter.

However, this task included the additional physical constraint of aligning two interlocking components. Fig. 11 shows the two components, consisting of a rectangular plate with four pegs (attached to the robot) and a reciprocal rectangular plate with four holes. The target state for this task was more specific, requiring the full insertion of all four pegs into their respective holes.

As previously stated, a PI controller was used to refine the output from the RF and WRF models and regulate the velocity command sent to the robot. For the co-manipulation task, the same values were used throughout. However, for the co-assembly task, the gain values were actively switched between values optimised for *gross* movements and values for *fine* control. Switching between the gains occurred automatically, based on a threshold distance of 20 mm in the XY 2D plane (i.e. vertical separation was not considered). The PI gain values were empirically determined and are shown in Table 1, with far smaller values used for the fine motion control. The co-manipulation task used the *gross* gains throughout.

5.4. Assessment criteria

Three different factors were used to assess the success of each approach:

- The time required to complete the task,
- The interaction force required to guide the robot, and
- The similarity in the generated trajectory to the demonstration.

The overall smoothness and responsiveness to human input are important for collaborative tasks. These are most easily assessed by the execution time and interaction forces that occur. The execution time (i.e. the total duration of the HRC task) indicates controller response, with shorter times being preferable. The interaction force (i.e. the force needed for the human to guide the robot $\mathbf{F}$, as shown in Eq. (3)), implies how smooth the robot behaviour is under the fitted model, with less force showing less work for the operator.

In addition, a strong indicator of successful skill capture is how similar the output is compared to the original demonstration. In this case, the trajectory generated by the robot should be compared to the trajectory of a manual demonstration. As the previously defined training scenarios consist of collaborative efforts designed to encompass a wide range of input/output conditions, they do not provide an optimal example of any one motion. Therefore, a human-operator was asked to manipulate a load in a series of straight lines (independently along each of the X, Y and Z axes) to produce a reference for comparison. The predicted trajectory can then be compared against the demonstration trajectories using the Normalised Root Mean Square Error (NRMSE).

Table 1
The PI controller parameters used for gross and fine motion control in the co-assembly task.

	P	I
Gross motion	41.0	0.5
Fine motion	2.5	0.0

Table 2
human–human demonstrations.

Demo	Number of samples	Max force N	Max speed m s^{-1}
1	10,967	81.39	0.134
2	11,707	70.25	0.126
3	10,894	72.18	0.139
4	14,604	76.39	0.06
5	7494	74.96	0.066

6. Results and discussion

The following section presents the results of this work, beginning with an analytical assessment of the RF and WRF models. This includes a review of the data captured for demonstration, as well as a comparison between the model-predicted results and actual data from an unseen demonstration dataset (as explained in Section 4.2. Following this, the capabilities of the models are experimentally validated through two representative HRC tasks, as defined in Section 5.3). Finally, the straight line performance of the models is investigated.

Repeated here for clarity, a total of 12 models have been created for this work, which fulfil each configuration of the following attributes:

- **Approach:** Random Forest (RF) and Weighted Random Forest (WRF)
- **Dataset:** human–human and Human–Robot demonstrations
- **Axis:** 3 separate models for robot control in each cardinal axis of Cartesian space (i.e. X, Y and Z)

6.1. Analytical assessment

Tables 2 and 3 show the respective statistical properties of demonstrations for human–human and human–robot collaboration. In each table, the first four demonstrations are used for training, with the fifth used for validation as an unseen dataset. In Table 2, it can be seen that the resulting force (F_e) between two human participants varies between 70.25 N and 81.39 N, whereas Table 3 shows that the interaction force between the human and the robot is almost $\sim$ 300 times greater. This difference can be explained as a result of the more compliant human body, with the robot being much more rigid and slower to respond. In addition, despite the maximum human–human speed varying between 0.06 m s^{-1} and 0.134 m s^{-1}, the maximum human–robot speed is far more consistent due to an imposed safety limit of 0.06 m s^{-1}.

From these results, it can be concluded that despite attempting to replicate the capabilities shown in the human–human demonstration, the physical constraints of the robot will likely limit how closely it can replicate the human motion. As such, model predictions based on the human–robot demonstrations are likely to better reflect the actual robot motions.

To test this and assess if the models produce suitable outputs, the final demonstration in both datasets was kept 'unseen' (i.e. not used for training), allowing the captured force data to be used as an input for each of the 12 models. The models are used to produce 'predicted' position and velocity values ($\hat{Y}$) which can be compared to the *true* values captured during the test (Y). As the motions in each dataset were performed with human operators, the number of sample points and range of motions differ between demonstrations. To make comparison

Table 3
Human–Robot demonstrations.

Demo	Number of samples	Max force N	Max speed m s^{-1}
1	3770	211.00	0.060
2	5640	275.77	0.061
3	4596	235.81	0.061
4	4272	294.97	0.061
5	4780	273.67	0.061

easier, the Normalised Root Mean Square Error (NRMSE) has been chosen as the metric to compare the results. This is defined in Eq. (9):

$$NRMSE = \frac{\sqrt{MSE}}{Y_{max} - Y_{min}} \tag{9}$$

where:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{10}$$

Figs. 12 and 13 show the velocity prediction of each model compared to the actual data over time. Beginning with the human–human data in Fig. 12, it is noticeable that the RF prediction not only has greater error overall, but also features substantial instances of rapid acceleration. WRF prediction has a smaller error and tends to accelerate more gradually. For the human–robot data in Fig. 13, it is immediately evident that both models perform better using this dataset. These results confirm that models trained on the human–robot data perform better than those trained on the human–human data. As previously stated, this is likely due to the human–robot model being a better fit for the experimental setup.

Fig. 14 shows the NRMSE results for each fitted model, based on their respective datasets. As expected, it can be seen that the WRF consistently outperforms the RF in all cases. For the human–human results, the WRF is noticeably better, significantly outperforming RF especially in the X-axis. By comparison, for the human–robot data the RF and WRF results are far closer. Although the WRF prediction also seems to be smoother and closer to the real data in Figs. 12 and 13, the similarity between the two would seem to indicate the WRF is not significantly better than RF in this case.

The reason for this similarity between RF and WRF can be explained by returning to the structure of the Random Forest approach. Variation in RF performance is introduced during training by the use of randomised subsets of the overall dataset (i.e. *bagging*). This forms the basis for tree weighting in WRF, as variables with better correlation between the input and output will tend to outperform other trees. However, for the human–robot demonstration, the rigidity of the robot will filter out much of the output variation, and allow the human to apply a more consistent input force. This increase in consistency results in similar bagged data for many of the trees. With many trees in the Random Forest being created with near identical data, applying weighting based on performance will not greatly improve the final result.

6.2. Validation scenario results

As described at the beginning of this section, a total of 12 models were created for this work. For the analytical results above, all 12 models were of interest; allowing comparison between the model predicted and actual results in all 3 axes. By contrast, the experimental validation scenarios (as described in Section 5.3) involve unconstrained movement in all axes to achieve a goal, without any other data for direct comparison. As such, the control models per axis will not be considered independently, instead contrasting the effects of dataset and ML model. With only the modelling approach and dataset being evaluated, the total number of models being tested is effectively reduced to just four:

HHRF = Human–Human dataset - RF

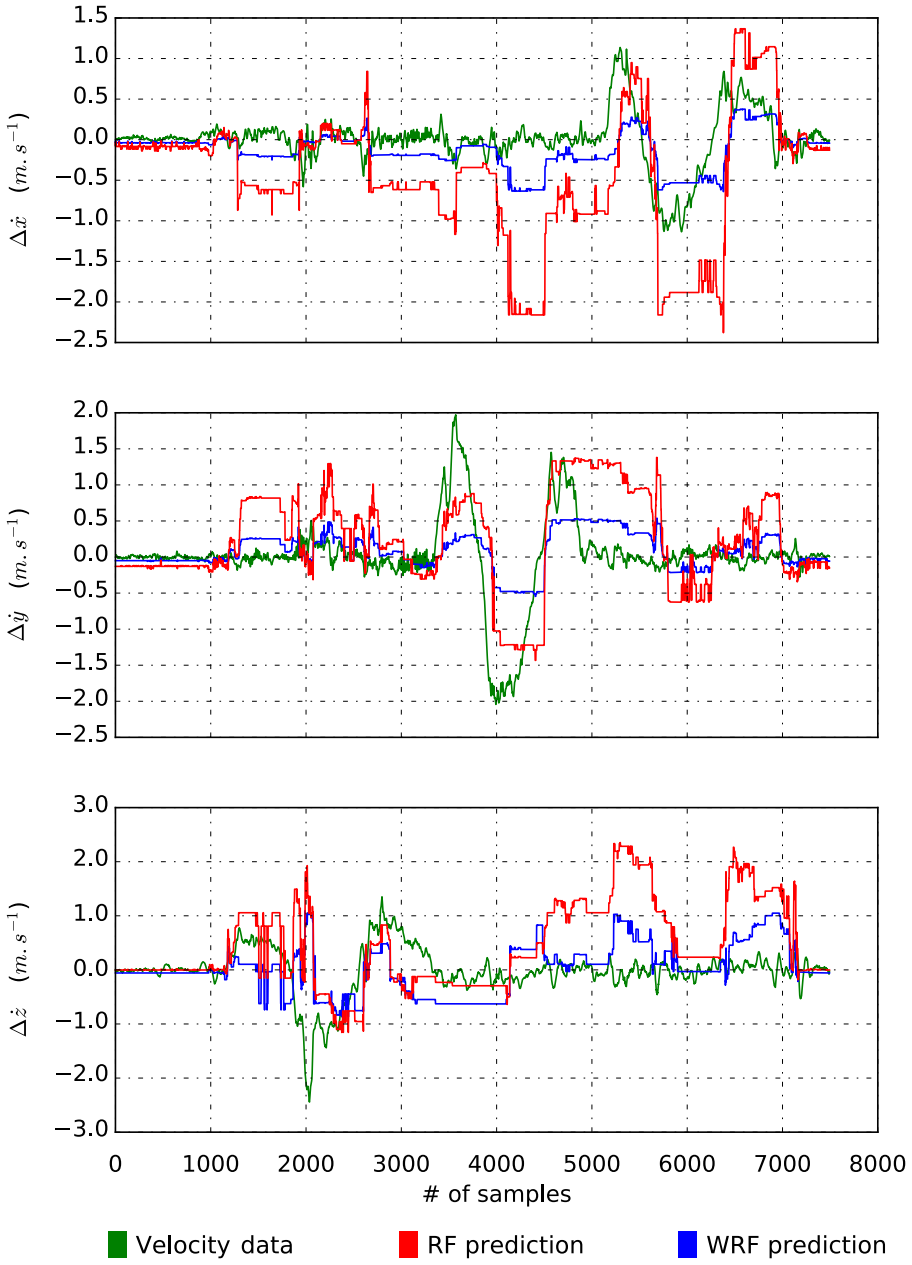

Fig. 12. Comparison of recorded and model-predicted velocity data for both RF and WRF methods on the human–human dataset.

HHWRF = Human–Human dataset - WRF

HRRF = Human–Robot dataset - RF

HRWRF = Human–Robot dataset - WRF

Each of these four models were applied to both the co-manipulation and co-assembly tasks, as previously described in Section 5.3. To increase the number of results, two human operators were asked to perform each task with each model twice (i.e. 4 models, 2 tasks, 2 operators and 2 attempts at each), for a total of 32 trials.

As described in Section 5.4, the performance of the fitted models in the validation scenarios were assessed on the time required to complete each task, as well as the interaction force between the human operator and the robot; with faster and less forceful results indicating better collaboration. For each trial, the force components (F_x, F_y and F_z) and torque components (T_x, T_y and T_z) were recorded directly from the FT sensor, while position and velocity data were captured using the VICON tracking system.

Fig. 15 depicts the average interaction forces and total execution times of each co-manipulation trial, as well as the average value for each ML model. Beginning with the force data, WRF approaches can be seen to have lower average interaction forces than the RF models for both datasets, with only a single example of WRF forces exceeding the minimum RF force. The lowest single-trial average interaction force was $\sim$ 14 N achieved using a HRWRF model, whereas the highest single-trial average interaction force was $\sim$ 55 N which occurred when using a HRRF model.

By contrast, it is more difficult to draw any conclusion about the performance of the models based on dataset, with both the best and worst performing models being trained using the human–robot demonstration data. Looking at the execution time required to complete each trial, the HRRF model had the shortest execution time of only 24 s whereas the HRWRF had the longest execution time, which lasted around 37 s. Despite this difference, when looking at the averages per model the execution time is very similar for all methods. From the co-manipulation trial alone, it can be said that the WRF approach does

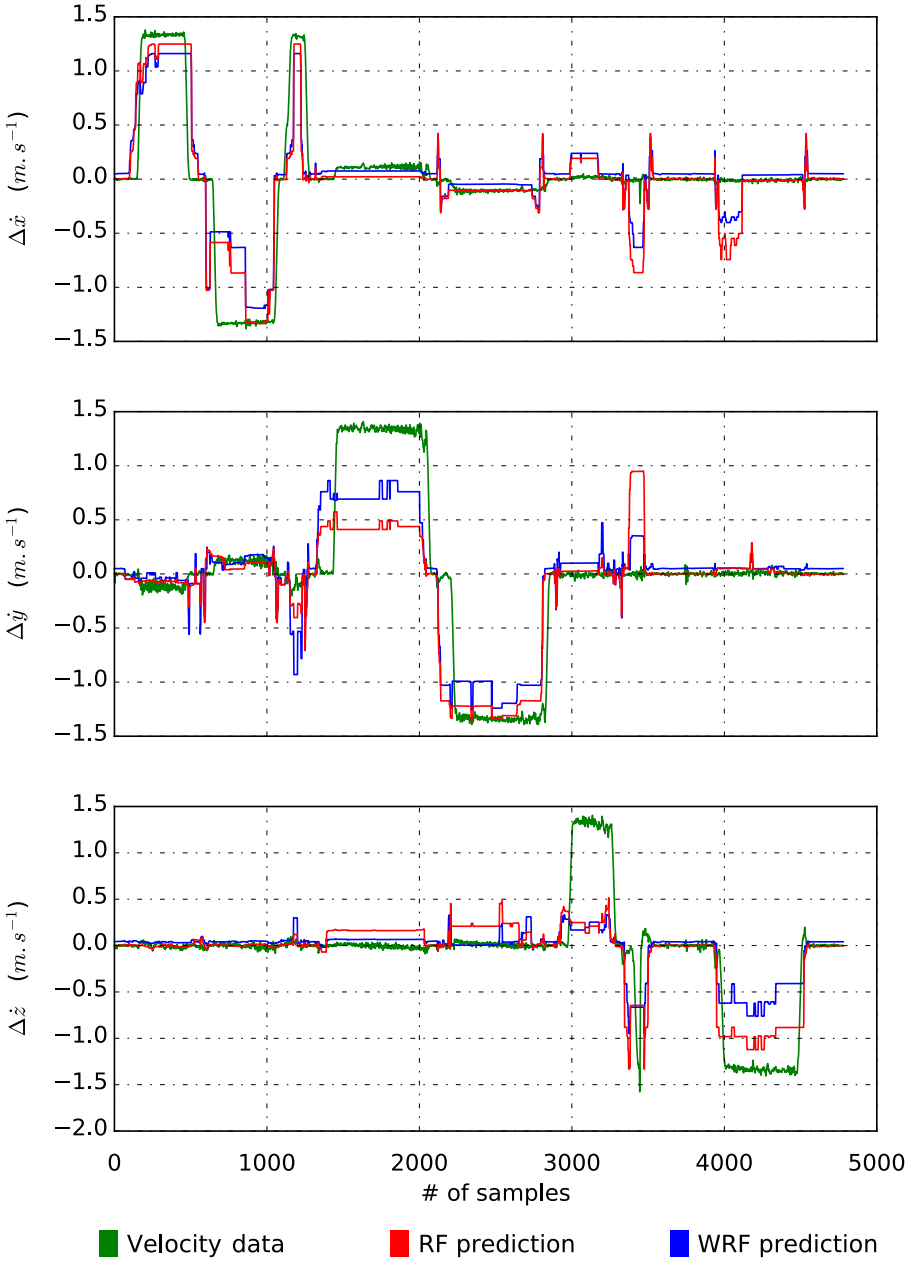

Fig. 13. Comparison of recorded and model-predicted velocity data for both RF and WRF methods on the human–robot dataset.

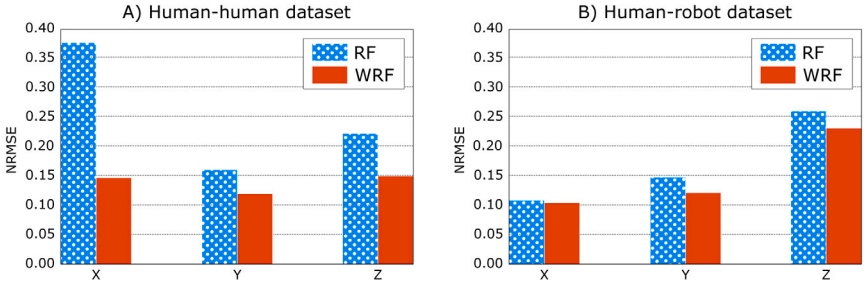

Fig. 14. NRMSE for RF and WRF models, for both human–human and human–robot datasets. The error is based on Cartesian offset in all 3 axes, with each bar (i.e. X, Y, Z) representing the control model specific to that axis.

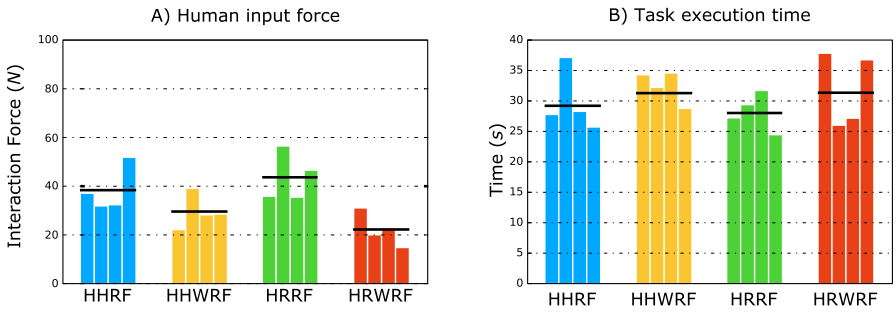

Fig. 15. Human–Robot interaction force and task execution time for all co-manipulation trials, including average results per ML model.

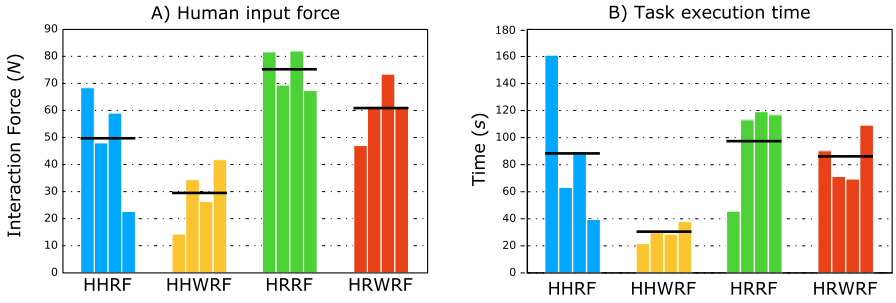

Fig. 16. Human–Robot interaction force and task execution time for all co-assembly trials, including average results per ML model.

show some advantage, but that the datasets have had minimal influence. As the co-manipulation scenario is closest to the demonstration datasets, this similarity may simply be due to all models performing well.

Unlike the co-manipulation data, the co-assembly data shown in Fig. 16 are more clear, with the WRF outperforming RF in nearly all trials. This is especially true for the models based on the human–human dataset, with WRF requiring 40% less input force, whilst also completing the task 64% quicker on average. Although less pronounced, the HRWRF was also 14% quicker and required 25% less force on average than the RF model trained on the human–robot dataset. This reinforces the conclusion that selectively weighting trees within the random forest can improve performance.

Fig. 16 also shows that the models trained on the human–human dataset outperform those trained on the human–robot data. On average, the models fitted using the human–human dataset require 42% less interaction force and complete the task 36% quicker when compared to the models based on the human–robot dataset. It is believed that this difference is caused by the rigidity of the robot, which requires a certain force threshold to be met before the robot will start moving. Not only does this require a greater level of input force, but will also increase the length of the trial as the robot will not move until the human operator overcomes this initial force requirement. As the ML models try to capture the compliance of the follower (human or robot), models fitted on the human–human demonstration will start moving at a much lower interaction force in comparison with models fitted on the human–robot demonstration, thus achieving better compliance and collaboration. As could be expected, the HHWRF results have both the lowest average interaction force and execution time, as well as the individual trial with the best results (at $\sim 15\ N$ and 29.57 s).

In summary:

- WRF models typically have lower interaction forces and shorter execution times representing more compliant behaviour.
  - Controlling the robot using WRF models during the assembly task requires less effort in comparison with RF models.
  - The WRF models have a better performance during fine motions compared with RF models. They result in lower interaction forces during gross motions, as shown in the co-manipulation example.
- The human–human dataset captured collaboration using far less input force than is required to overcome the robot rigidity, resulting in more compliant (human-like) behaviour for the co-assembly scenario.

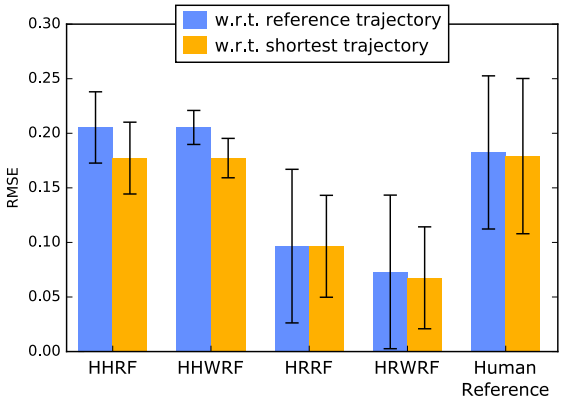

Fig. 17. Comparison of overall RMSE error for each model and dataset, as well as the human–human reference, with respect to the reference trajectory and the shortest possible trajectory between the start and finish points.

6.3. Straight line performance

As stated in Section 5.4, the final indicator of successful skill capture is if the trajectory generated by the robot is very similar to the trajectory of a manual demonstration. As the previously covered validation scenarios do not provide any clear example of a single repeatable motion, an additional task was undertaken in which the two human-operators were asked to collaboratively manipulate a load in a series of straight lines (independently along each of the X, Y and Z axes) to produce a reference for comparison. Following this, one of the human

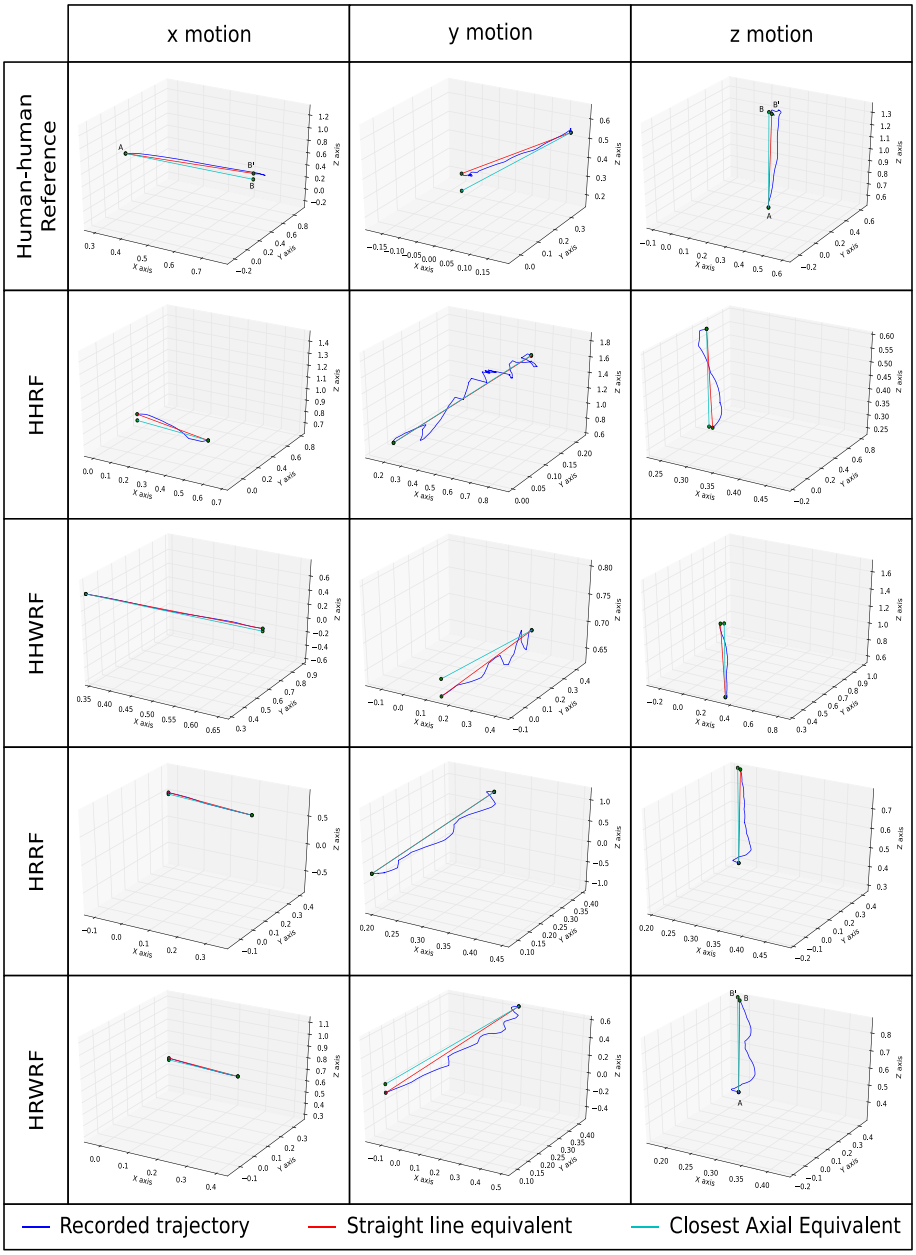

Fig. 18. Captured trajectory information for human–robot collaboration, performing a straight-line trajectory in X, Y and Z directions for each model. Each trajectory is presented with equivalent straight line trajectories, as well as a human–human collaboration reference.

operators was asked to perform the same task in collaboration with the robot, using each of the models for control.

Fig. 18 shows the trajectories captured during collaboration, as well as the human–human reference trajectories. To help compare the human–human and human–robot results, for each trial the optimal shortest line between the start and finish points for that test (shortest trajectory), and the closest axial straight line to the test (reference trajectory) are also shown. For all graphs within this figure the axes are presented to maximise visibility, and therefore scale is not consistent between axes or graphs.

As expected, the human trajectories captured for reference are not optimal, due to the limitations of human perception and muscle control. However, they do tend to be smooth and relatively straight and provide a suitable reference for visual comparison. Similarly, it can be seen that the motion along the X axis was similarly smooth for all control models, with only the HRRF model showing a minor deviation. In comparison, the Y and Z motions are less straight, featuring recurring oscillation. This is likely due to the robot not being responsive enough for the human to precisely stop the overshoot.

Fig. 17 shows the RMSE values for WRF and RF models on both datasets, with respect to the reference and shortest trajectories. In general, the trajectories reproduced using models fitted on the human–robot dataset have lower RMSE values in comparison with trajectories reproduced using models fitted on the human–human dataset, with the HRWRF model achieving the lowest error overall. This is likely due to the fact that systems trained on data which is more similar to the actual use-case will outperform those trained for general use. The models trained on the human–robot dataset also outperform two humans working in collaboration, likely due to the rigidity of the robot making the task easier to perform.

In comparison, the human–human reference and the robot trained on the human–human dataset did not perform as well. However, despite the increased error, the similarity between the human and the

robot suggests that the modelling approach has successful captured the skill.

7. Conclusions

This paper focused on methods of transferring human skills to an industrial robot, using a Learning from Demonstration (LfD) framework. Two demonstration datasets were created (human–human and human–robot) and skill capture was achieved using the Random Forest (RF) ensemble learning approach. An extension to RF was proposed and tested, in which trees in the RF are weighted based on their performance, to improve the generalisation capabilities of the method.

Both RF and WRF models were created based on each dataset, and the models were evaluated using three different methods; an analytical assessment based on prediction error using data from an unseen demonstration trial, an experimental assessment using validation scenarios, and a final practical assessment of the models ability to co-manipulate an object along a straight line. It was found that our proposed method of WRF improves the prediction by $\approx$ 20% in comparison with unchanged RF models. In addition, this performance increase is notably better than achieved in other literature (being $\approx$ 2% in [37], $\approx$ 2.5% in [38] and $\approx$ 10% in [36]).

The validation scenarios used in the experimental assessment had two goals; the co-manipulation task was intended to directly validate the fitted models' competence in reproducing the demonstrated skill, whereas the co-assembly task was intended to validate the generalisation capabilities of the proposed approach.

Collectively, the results show that the ensemble ML method can be used for *agile* Human–Robot Collaboration, with the proposed WRF extension generally outperforming the original RF implementation. As humans themselves are often suboptimal, models based on human–human datasets were often shown to be less accurate, with the human–robot trained models requiring lower interaction forces whilst producing smoother trajectories, more suitable for fine motion. However, the models created using the human–human dataset were shown to feature a faster response, allowing them to be more easily applied to scenarios outside of the training dataset.

7.1. Future work

Looking forward, we plan to extend this research to include online weight estimation for WRF. This would require RF models to be actively adjusted during the human–robot co-manipulation, allowing them to generalise better than fixed weight approaches. This should also allow the fast response of human–human collaboration to be transferred more easily, without the loss of accuracy. This could be achieved using offline batch learning to incrementally refine the data-driven models, after deployment to a robot.

Furthermore, there is also scope for improving collaboration by isolating human haptic inputs from other forces. Although the human operator is the only input during co-manipulation, during co-assembly the frictional interaction between components can affect the trajectory. For this work, the assembly parts were designed with a small clearance to weaken these disturbance forces during the co-assembly task, however this is impractical for real-world applications. Instead, this limitation should be addressed by distinguishing between haptic inputs from the operator and those generated by the process itself. Therefore, this is also an area to be explored, either through including an external source of information, or more intelligent interpretation of the force input.

CRediT authorship contribution statement

A. Al-Yacoub: Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Y.C. Zhao:** Methodology, Software, Validation. **W. Eaton:** Formal analysis, Writing - review & editing, Visualization. **Y.M. Goh:** Supervision, Writing - review & editing. **N. Lohse:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was funded by the EPSRC, UK as part of the "Digital Toolkit for optimisation of operators and technology in manufacturing partnerships" project (DigiTOP; EP/R032718/1).

References

[1] Y. Li, K.P. Tee, R. Yan, W.L. Chan, Y. Wu, D.K. Limbu, Adaptive optimal control for coordination in physical human-robot interaction, in: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, 2015, pp. 20–25.

[2] J. Aleotti, S. Caselli, Learning manipulation tasks from human demonstration and 3d shape segmentation, Adv. Robot. 26 (16) (2012) 1863–1884.

[3] J. Mainprice, D. Berenson, Human-robot collaborative manipulation planning using early prediction of human motion, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, 2013, pp. 299–306.

[4] R. Ikeura, H. Inooka, Variable impedance control of a robot for cooperation with a human, in: Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on, Vol. 3, IEEE, 1995, pp. 3097–3102.

[5] K. Kosuge, H. Yoshida, T. Fukuda, Dynamic control for robot-human collaboration, in: Robot and Human Communication, 1993. Proceedings., 2nd IEEE International Workshop on, IEEE, 1993, pp. 398–401.

[6] K. Kosuge, H. Yoshida, D. Taguchi, T. Fukuda, K. Hariki, K. Kanitani, M. Sakai, Robot-human collaboration for new robotic applications, in: Industrial Electronics, Control and Instrumentation, 1994. IECON'94., 20th International Conference on, Vol. 2, IEEE, 1994, pp. 713–718.

[7] K. Kosuge, N. Kazamura, Control of a robot handling an object in cooperation with a human, in: Robot and Human Communication, 1997. RO-MAN'97. Proceedings., 6th IEEE International Workshop on, IEEE, 1997, pp. 142–147.

[8] O.M. Al-Jarrah, Y.F. Zheng, Arm-manipulator coordination for load sharing using reflexive motion control, in: Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, Vol. 3, IEEE, 1997, pp. 2326–2331.

[9] O.M. Al-Jarrah, Y.F. Zheng, Intelligent compliant motion control, IEEE Trans. Syst. Man Cybern. B 28 (1) (1998) 116–122.

[10] V. Duchaine, C.M. Gosselin, General model of human-robot cooperation using a novel velocity based variable impedance control, in: EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint, IEEE, 2007, pp. 446–451.

[11] R. Ikeura, T. Moriguchi, K. Mizutani, Optimal variable impedance control for a robot and its application to lifting an object with a human, in: Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on, IEEE, 2002, pp. 500–505.

[12] T. Tsumugiwa, R. Yokogawa, K. Hara, Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task, in: Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, Vol. 1, IEEE, 2002, pp. 644–650.

[13] F. Ficuciello, L. Villani, B. Siciliano, Variable impedance control of redundant manipulators for intuitive human–robot physical interaction, IEEE Trans. Robot. 31 (4) (2015) 850–863.

[14] V. Klingspor, J. Demiris, M. Kaiser, Human-robot communication and machine learning, Appl. Artif. Intell. 11 (7) (1997) 719–746.

[15] M. Rahman, R. Ikeura, K. Mizutani, Investigating the impedance characteristic of human arm for development of robots to co-operate with human operators, in: Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on, Vol. 2, IEEE, 1999, pp. 676–681.

[16] Y. Hayashibara, T. Takubo, Y. Sonoda, H. Arai, K. Tanie, Assist system for carrying a long object with a human-analysis of a human cooperative behavior in the vertical direction, in: Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on, Vol. 2, IEEE, 1999, pp. 695–700.

[17] Y. Maeda, A. Takahashi, T. Hara, T. Arai, Human-robot cooperation with mechanical interaction based on rhythm entrainment-realization of cooperative rope turning, in: Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, Vol. 4, IEEE, 2001, pp. 3477–3482.

[18] K.B. Reed, M.A. Peshkin, Physical collaboration of human-human and human-robot teams, IEEE Trans. Haptics 1 (2) (2008) 108–120.

[19] K. Strabala, M.K. Lee, A. Dragan, J. Forlizzi, S.S. Srinivasa, M. Cakmak, V. Micelli, Toward seamless human-robot handovers, J. Hum.-Robot Interact. 2 (1) (2013) 112–132.

[20] M. Lawitzky, J.R. Medina, D. Lee, S. Hirche, Feedback motion planning and learning from demonstration in physical robotic assistance: differences and synergies, in: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE, 2012, pp. 3646–3652.
[21] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, A. Kheddar, Learning collaborative manipulation tasks by demonstration using a haptic interface, in: Advanced Robotics, 2009. ICAR 2009. International Conference on, IEEE, 2009, pp. 1–6.
[22] D.A. Duque, F.A. Prieto, J.G. Hoyos, Trajectory generation for robotic assembly operations using learning by demonstration, Robot. Comput.-Integr. Manuf. 57 (2019) 292–302.
[23] A. Ude, A. Gams, T. Asfour, J. Morimoto, Task-specific generalization of discrete and periodic dynamic movement primitives, IEEE Trans. Robot. 26 (5) (2010) 800–815.
[24] J.R. Medina, M. Lawitzky, A. Mörtl, D. Lee, S. Hirche, An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE, 2011, pp. 2416–2422.
[25] W. Sheng, A. Thobbi, Y. Gu, An integrated framework for human–robot collaborative manipulation, IEEE Trans. Cybern. 45 (10) (2015) 2030–2041.
[26] Y. Cui, J. Poon, T. Matsubara, J.V. Miro, K. Sugimoto, K. Yamazaki, Environment-adaptive interaction primitives for human-robot motor skill learning, in: Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on, IEEE, 2016, pp. 711–717.
[27] E. Berger, D. Vogt, S. Grehl, B. Jung, H.B. Amor, Estimating perturbations from experience using neural networks and information transfer, in: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, IEEE, 2016, pp. 176–181.
[28] M.U. Suleman, M.M. Awais, Learning from demonstration in robots: Experimental comparison of neural architectures, Robot. Comput.-Integr. Manuf. 27 (4) (2011) 794–801.
[29] X. Gao, J. Ling, X. Xiao, M. Li, Learning force-relevant skills from human demonstration, Complexity 2019 (2019).
[30] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Handbook of robotics chapter 59: Robot programming by demonstration, in: Handbook of Robotics, Springer, 2008.
[31] S. Calinon, Z. Li, T. Alizadeh, N.G. Tsagarakis, D.G. Caldwell, Statistical dynamical systems for skills acquisition in humanoids, in: Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on, IEEE, 2012, pp. 323–329.
[32] T. Teramae, K. Ishihara, J. Babič, J. Morimoto, E. Oztop, Human-in-the-loop control and task learning for pneumatically actuated muscle based robots, Front. Neurorobotics 12 (2018).
[33] A. Al-Yacoub, S. Sharifzadeh, N. Lohse, Z. Usman, Y.M. Goh, M.R. Jackson, Learning industrial robot force/torque compensation: A comparison of support vector and random forests regression, 2016.
[34] A. Chandra, X. Yao, Evolving hybrid ensembles of learning machines for better generalisation, Neurocomputing 69 (7–9) (2006) 686–700.
[35] A. Liaw, Wiener m, Classification and regression by random forest. R News, 2, 2002, pp. 18–22.
[36] A. Booth, E. Gerding, F. Mcgroarty, Automated trading with performance weighted random forests and seasonality, Expert Syst. Appl. 41 (8) (2014) 3651–3661.
[37] C. Zhang, Y. Li, Z. Yu, F. Tian, A weighted random forest approach to improve predictive performance for power system transient stability assessment, in: 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), IEEE, 2016, pp. 1259–1263.
[38] S.J. Winham, R.R. Freimuth, J.M. Biernacka, A weighted random forests approach to improve predictive performance, Stat. Anal. Data Min.: ASA Data Sci. J. 6 (6) (2013) 496–505.
[39] H.B. Li, W. Wang, H.W. Ding, J. Dong, Trees weighting random forest method for classifying high-dimensional noisy data, in: E-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on, IEEE, 2010, pp. 160–163.