An improved approach of task-parameterized learning from demonstrations for cobots in dynamic manufacturing

El Zaatari, S, Wang, Y, Hu, Y & Li, W

'An improved approach of task-parameterized learning from demonstrations for cobots in dynamic manufacturing', Journal of Intelligent Manufacturing, vol. (In-press), pp. (In-press).
https://doi.org/10.1007/s10845-021-01743-w

DOI 10.1007/s10845-021-01743-w
ISSN 0956-5515
ESSN 1572-8145

Publisher: Springer

The final publication is available at Springer via http://dx.doi.org/10.1007/s10845-021-01743-w

An Improved Approach of Task-Parameterized Learning from Demonstrations for Cobots in Dynamic Manufacturing

Shirine El Zaatari [1], Yuqi Wang [2], Yudie Hu [2] and Weidong Li [1, 2,*]

[1] Faculty of Engineering, Environment and Computing, Coventry University, Coventry, UK

[2] School of Logistics Engineering, Wuhan University of Technology, Wuhan, China

***** Correspondence: weidong.li@coventry.ac.uk

Abstract

Task-Parameterized Learning from Demonstrations (TP-LfD) is an intelligent intuitive approach to support collaborative robots (cobots) for various industrial applications. Using TP-LfD, human's demonstrated paths can be learnt by a cobot for reproducing new paths for the cobot to move along in dynamic situations intelligently. One of the challenges to applying TP-LfD in industrial scenarios is how to identify and optimize critical task parameters of TP-LfD, i.e., frames in demonstrations. To overcome the challenge and enhance the performance of TP-LfD in complex manufacturing applications, in this paper, an improved TP-LfD approach is presented. In the approach, frames in demonstrations are autonomously chosen from a pool of generic visual features. To strengthen computational convergence, a statistical algorithm and a reinforcement learning algorithm are designed to eliminate redundant frames and irrelevant frames respectively. Meanwhile, a B-Spline cut-in algorithm is integrated in the improved TP-LfD approach to enhance the path reproducing process in dynamic manufacturing situations. Case studies were conducted to validate the improved TP-LfD approach and to showcase the advantage of the approach. Owing to the robust and generic capabilities, the improved TP-LfD approach enables teaching a cobot to behavior in a more intuitive and intelligent means to support dynamic manufacturing applications.

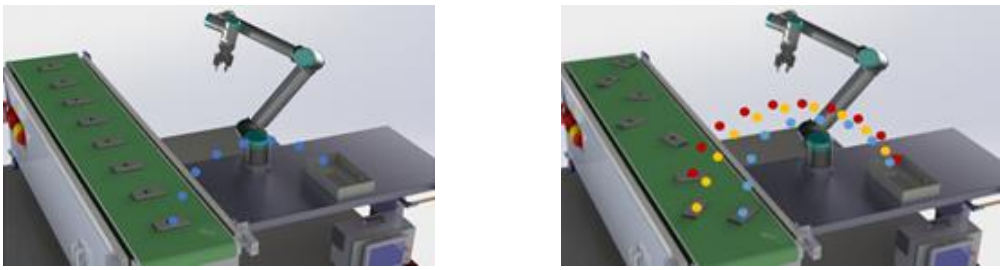
Keywords: Learning from demonstration; reinforcement learning; collaborative robots

1. Introduction

Modern manufacturing is characterized by shifting from mass production to mass customization, increased product personalization and shorter product lifecycles [1]. To meet the new industrial needs, it is required to develop more versatile robotic technologies. Industrial robots, which primarily execute pre-programmed tasks for mass production, lack adaptability to changes in customized manufacturing and dynamic uncertainties caused by counterpart human operators in the manufacturing processes [2]. In contrast, collaborative robots (cobots)

can effectively support human-robotic cooperation (HRC) for various dynamic manufacturing applications. Different from an industrial robot, a cobot is embedded with an intuitive mechanism that can enable the cobot to learn from its human companion intelligently. In the meantime, the mechanism can also support the cobot to adapt to its working environment to carry out a dynamic task flexibly [3]. Thus, cobots have been increasingly adopted in engineering and manufacturing companies, especially Small and Medium-sized Enterprises (SMEs) [4], to cater for fast-paced changes in their mass customization processes.

The following example illustrates the above concepts. In Figure 1 (a), a robot picks an object from one position on a manufacturing transmission belt and places it into a packaging box. The robot is pre-programmed to follow a fixed set of way points. In other words, the path is repeatable and does not cater for changes in robotic setting. However, in dynamic manufacturing settings, positions of objects for picking might vary, and dynamic adjustments on the pick-and-place path of the robot should be made adaptively. As shown in Figure 1 (b), the object for picking is on changing positions and orientations. Instead of having pre-programmed way points, a cobot with the capability to learn and interpolate from human's demonstrations, would be more flexible to the varying positions, thus implementing the pick-and-place task in a more flexible means.



(a) (b)

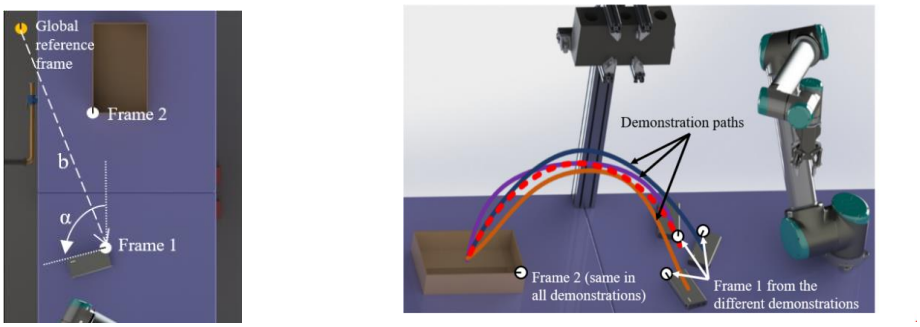
Figure 1. (a) Pre-programmed robotic operations in a pre-determined manufacturing setting. (b) Robotic motions that cater for un-predictable changes in a dynamic manufacturing setting.

Task-Parametrized Learning from Demonstrations (TP-LfD) is an effective intuitive approach for cobots to implementing intelligent flexible behaviors, such as apple picking [5], tennis playing [6] and industrial assembly [7]. For the above pick-and-place task, using TP-LfD, some human's demonstrations are taken to train a cobot to act under different task settings. Then, TP-LfD is used to generalize over the demonstrations and reproduce a path of the cobot to address a dynamic setting. More details of the process are depicted below [8]:

- A human operator demonstrates a path that a cobot should follow for picking an object from a start point and drop the object into an end-point. This is executed by dragging the end effector of the robot or inputting path points using a user interface. The positions of the start and/or end points might be varied between the demonstrations (Figure 2 (b));

- Using TP-LfD, the demonstration paths of the cobot are encoded with respect to task parameters. Task parameters usually refer to multiple frames of reference (frames in the rest of the paper) associated with objects that represent their positions in the demonstrations. The frames and demonstration paths will be further utilized for reproducing a new path for a dynamic manufacturing situation. Figure 2 (a) is an example of a task scenario on which the demonstrations are recorded. The object for picking in the demonstration path is associated with a frame (e.g., Frame 1), which is defined by a location vector and a rotation matrix. As the object varies positions, the frame does as well (e.g., Frame 2). One of the requirements for TP-LfD is to detect the frames of objects.
- Based on frames, a mathematical model, such as Task Parametrized Gaussian Mixture Model (TP-GMM), which is suitable to model complex systems [9], is introduced to encode the demonstrated paths. Then, based on TP-GMM and a regression mathematical model, such as Task Parameterized Gaussian Mixture Regression (TP-GMR), a new path can be reproduced. During the above encoding modeling and reproducing processes, the human operator is not required to possess programming skills and robotic knowledge in operating the cobot to accomplish dynamic tasks.

In TP-LfD, frames are critical inputs to ensure the quality of the reproduced path. Thus, essential tasks in TP-LfD include: 1) how to detect and localize frames in an application scene; 2) how to choose optimal frames to better support TP-LfD.



(a) (b)

Figure 2. (a) An example of a demonstration scenario showing two frames: Frame 1 on the object of a variable location and Frame 2 on the box in which the object should be placed in. The frames are defined by position and orientation with respect to a global reference frame. (b) A set of demonstration paths are recorded by an operator to teach a cobot to pick an object of a variable position and place it into a box.

Traditionally, a human operator is responsible for determining the frames of an object in an application scene and a computer vision algorithm is designed to detect the frames automatically. However, this process requires human's programming expertise to adapt TP-LfD to changes in the application/objects, which hinders the intuitiveness and flexibility of the algorithm. Thus, the gap between TP-LfD and dynamically customized manufacturing deployments is still significant [4]. To address the challenge, in this paper, an improved TP-

LfD approach is presented. The innovative characteristics of the improved TP-LfD approach include:

- Generic visual features of objects are identified intelligently to support frame extraction, thereby making the improved TP-LfD approach widely applicable without requiring further manual programming adjustment on the cobot;
- Optimal frames are determined from the set of the generic visual features using a statistical algorithm and a reinforcement learning algorithm. The algorithms can eliminate redundant frames and irrelevant frames to improve the quality of the reproduced path;
- Meanwhile, a B-Spline cut-in algorithm is designed and integrated with the reproduced process to further enhance the adaptability of the path for dynamic and customized manufacturing situations.

In this paper, the performance of the improved TP-LfD approach was validated via some simulation and practical manufacturing case studies, showing the potential of the approach in supporting practical industrial applications.

In the rest of the paper, the related works are presented in Section 2. The methodology of this research is detailed in Section 3. In Section 4, case studies with different complexities are depicted. The research is concluded, and future research is outlined briefly in Section 5.

2. Related Works

2.1 Task Parameter Detection

One of the essential steps in TP-LfD is to detect its task parameters, which usually refer to the positions of objects relevant to task paths. Table 1 shows a list of methods developed previously for detecting task parameters to support TP-LfD for industrial applications. The table highlights the major characteristics and advantages of each method (also including the method to be presented in this paper).

Table 1. Methods of detecting the positions of objects and their characteristics.

Ref.	Detection Method	Non-invasive hardware	Capable to detect complex objects	Easily extended to other objects
[10]	Motion capturing	-	x	x
[11] [12]	Sticker markers	-	x	x
[13] [7]	Image processing, color/shape segmentation	x	-	-
[14] [15]	Contour matching	x	-	x
[16]	Cloud point matching	x	x	-
[17] [18] [19]	SVM/ANN classification	x	x	-

[20] [21]	Deep learning algorithm	x	x	-
Ours	Improved TP-LfD	x	x	x

In those research works, the methods of motion capturing [10] and sticker markers [11] [12] are reliable to support complex scenarios. However, the works require placing hardware, i.e., a motion capturing bulb or sticker, on parts. This way is still not desirable or even possible in practical manufacturing applications. Some methods were designed based on image processing techniques, such as color segmentation, shape detection [7] [13] or contour matching [14] [15]. However, the methods need to be customized and adjusted depending on each individual case. The customization process will be a strain on programmers for re-coding, and it is not suitable for customized manufacturing applications. Moreover, it is difficult to use the methods to detect objects with complex shapes or multiple colors, which, however, could be a normal case in manufacturing scenarios. Some methods were implemented based on machine learning techniques, such as support vector machine (SVM)-based classifiers [18] [19] and deep learning networks [20] [21]. The methods can be reliably used to detect and classify objects for a practical scene. However, the methods rely on a large amount of training data to function accurately, which introduces expensive processes of data collection and algorithm training for a new task. Furthermore, the method presented in [16] used a scanner to generate cloud points of an object and trained the algorithm of detecting the positions of the object from the cloud data. However, a complex system consisting of expensive hardware setup as well as a rotating table and a 3D camera/scanner is required. In this paper, the aim is to develop a more generic, flexible and intelligent method to detect task parameters to meet the changing needs of customized manufacturing. Meanwhile, the design of the method is also aimed to be based on a minimal and versatile hardware setup (e.g., only a 2D camera) to suffice the need for detecting objects of random shape/form. Some examples are illustrated in Figure 3.

(a) (b) (c)

Figure 3. Examples of object detection techniques: (a) using sticker markers [12], (b) using mask R-CNN [20], (c) using contour matching [15].

The methods for object/frame detection can also be used for vision-based motion planning, which is another way of programming cobots for variables tasks. Unlike TP-LfD, motion planning requests a programmer to provide a rule-based program detailing the relation between the path of a robot and the position of an object. However, in tasks where some path deviations

exist, TP-LfD is more appealing owing to its intuitive programming capability. That is, the relation between robot's path and object's position is captured intelligently by the TP-LfD algorithm and no programming expertise is necessary.

2.2 Task Parameter Optimization

Task parameters are usually chosen and designed manually. This could lead to sub-optimality, such that the user could chose a task-irrelevant frame or two frames that are redundant. Redundant frames are those frames that belong to the same rigid object, i.e., they have fixed relative positions with respect to each other. Due to the fixed relative positions between detected frames on a single object, training based on redundant frames is an unnecessarily computation-expensive process as the frames will generate the same TP-GMM. Moreover, irrelevant frames are those frames associated from environmental objects by a detection algorithm but irrelevant to the task. If redundant and/or irrelevant frames are considered in TP-LfD, the performance of the algorithm deteriorates and becomes falsely biased. Ideally, there should be one frame per task-relevant object.

To tackle the above problem, Alizadeh et al. defined an importance score for each frame to identify redundant frames [22] and eliminate irrelevant frames [23]. The importance score is the ratio of the covariance matrix of a frame over the total covariance of the path. It is a measure of how much a frame affects the path at a given path point. Frames with an equal importance score are deemed redundant, and only one of them is accounted for during path reproduction. However, in a real-life situation with slight frame position disturbances, even redundant frames will not have exactly equal importance scores. Therefore, an error threshold should be introduced to account for slight frame mis-localizations. Moreover, Alizadeh and Karimi's algorithm necessitates the training of TP-GMM to obtain the covariance matrices before grouping redundant frames [22]. However, in the cases with a large number of frames (e.g., more than ~50), TP-GMM would fail to converge. Therefore, it is important to note that it is paramount to design another algorithm to group redundant frames before training TP-GMM.

Alizadeh and Malekzadeh also employed the same importance score to identify irrelevant frames [23]. Frames with an importance score less than a user-specified threshold are deemed irrelevant and excluded from TP-LfD. However, choosing an appropriate threshold is tricky in the cases when some frames have subtle yet important effects on the path. Moreover, this method does not eliminate redundant frames in case some frames are missed by the algorithm presented in [23]. Therefore, a solution with a feedback system should be devised to identify optimal relevant frames for TP-LfD.

Other research works were conducted to improve the performance of TP-LfD using frame-based solutions. For example, Sena et al. used the same importance score to adjust the weights of TP-GMM for different frames during reproduction [24]. This approach provides more

accurate results at the start and end of the paths even when frames' positions are changed drastically compared to the demonstrations. To further improve the performance of this process, Huang et al. used a reinforcement learning algorithm to shift the position of frames until a task-specific cost function is minimized [25]. Moreover, the authors developed an automatic frame selection algorithm where which frames contribute most are identified to minimize the cost function. The approach eliminates the frames that have a low influence on the learning to speed up computation and improves performance. However, the limitation of the approach is that it does not tackle how to visually detect frames, but rather the frames are specified as fixed positions with respect to the cobot's end-effector. Thus, it is not functional in that the frames are intrinsically defined on non-static objects. Girgin et al. created an active learning approach to suggest new demonstrations that ensure improvement in the control policy [26]. This could include suggesting new positions for task parameters. However, this still does not tackle the problem of choosing where task parameters are on objects of interests or what the objects of interest are.

To overcome the above challenges, the research presented in this paper is aimed to develop a systematic approach with the following characteristics to facilitate dynamic manufacturing applications:

- The approach will be based on a simpler setup based on a 2D camera, which can be intuitively used by operators for a variety of industrial tasks/objects without making major algorithmic changes on individual cases. The approach should detect frames automatically to be used to train a cobot to perform tasks intelligently;
- An intelligent algorithm that identifies the optimal frames to boost the TP-LfD's performance should be developed. This will be conducted by grouping redundant frames and removing irrelevant frames using intelligent algorithms.

3. Methodology of the improved TP-LfD approach

3.1. Overall Methodology

The improved TP-LfD approach presented in this paper is comprised of two procedures: TP-GMM for encoding demonstrations (a training process), and TP-GMR for path reproduction (a reproduction process). The overview of the improved TP-LfD approach is shown in Figure 4.

In TP-GMM for encoding demonstrations, there are several critical steps: 1) recording demonstrations; 2) extracting visual features, i.e., frames; 3) grouping redundant frames; 4) calculating TP-GMM for each frame; 5) removing irrelevant frames. The improvements of the approach developed in this paper over conventional TP-GMM are in the above steps 2), 3) and 5) by developing new algorithms.

In the path reproduction process based on TP-GMR, the main steps are: 1) recording the new image of the new scene; 2) detecting relevant frames in the new image; 3) matching visual features between the new image and the images of the demonstrations to obtain relevant frames; 4) reproducing a path using TP-GMR; 5) fine-tuning the path to fit into a new dynamic environment using a B-Spline cut-in algorithm. The improvements of the approach developed in this paper over conventional TP-GMR are in the above steps 3) and 5) by developing new algorithms.

The above procedures are detailed in the following sub-sections.

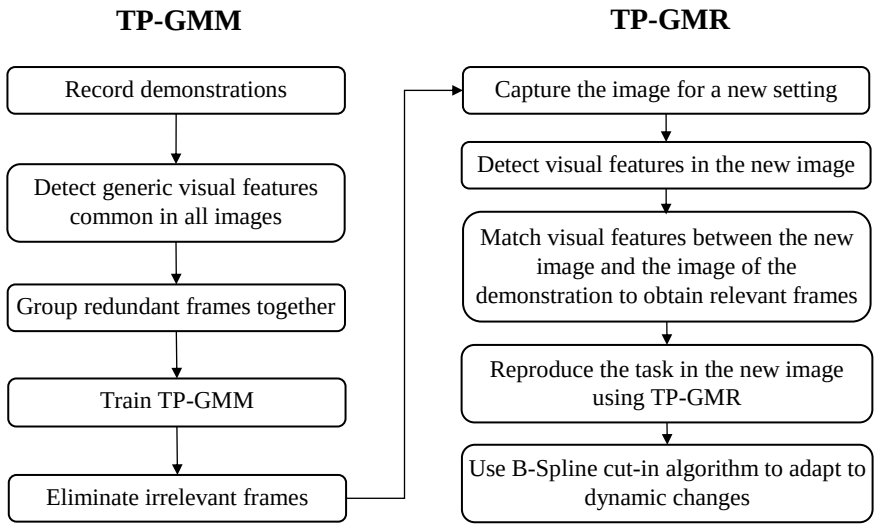


Figure 4. The overview process of the improved TP-LfD approach.

3.2. Recording Demonstrations

Each demonstration entails the image of an application scene and a path that the cobot needs to follow to perform the task. In this paper, the images of the demonstrations are recorded in 2-dimension to simplify the computation complexity of the approach, i.e., the top view of objects located on a surface. The path recorded consists of five elements: the time step t, the x-y-z coordinates and the gripper state g. To record the path, a user saves a series of way points and the path is interpolated in between them. This is similar to kinaesthetic teaching. In this improved TP-LfD approach, to improve the computational performance, the number of demonstrations recorded is set to a default value of 5.

3.3. Extracting Visual Features

In an effort to simplify the process of frame detection for TP-LfD, generic visual features are used to identify the frames of reference (task parameters). It presents a generic solution that can work for a wide range of objects without tweaking. Moreover, only a minimal setup of a 2D camera is required. Two types of visual features are detected: Speeded Up Robust Features (SURF) and hand features.

SURF features: Interest points are detected such as corners, T-junctions and blobs. Such interest points are widely available in objects, which makes the improved TP-LfD approach applicable with a wide range of object shapes and textures.

Descriptor vectors are calculated to describe each interest point. The vectors are scale and rotation invariant, which allows them to be matched robustly from different images [27]. SURF features can be matched across the images of different demonstrations even when objects vary orientation on the table.

Moreover, SURF features provide a balance between detection time and number of features retrieved [28]. Therefore, enough features can be retrieved from task-relevant objects that serve as task parameters for TP-GMM. Also, when attempting to reproduce a path in a new scene, the features can be matched quickly between the new image and the demonstration images to provide close to real-time results.

Matching SURF features is done using the exhaustive nearest neighbor search algorithm [29]. Features that are matched across all the images of the demonstrations remain. If a feature is not found in at least one of the images, it is eliminated since TP-GMM can only be trained if a feature is found in all demonstration images.

Once a SURF feature is matched in the images of all demonstrations, it is saved as a task parameter to be used in training TP-GMM. Each task parameter i is characterized by a position vector $b_{i,m}=\{x, y\}_{i,m}$ and an orientation $\alpha_{i,m}$, where $m \in \mathbb{N}\{1,\ldots,M\}$, the total number of the demonstration images. From the orientation, a 2x2 rotation matrix $A_{i,m}$ is calculated. An example of detected visual features and identified frames based on the matched visual features is shown in Figure 5.

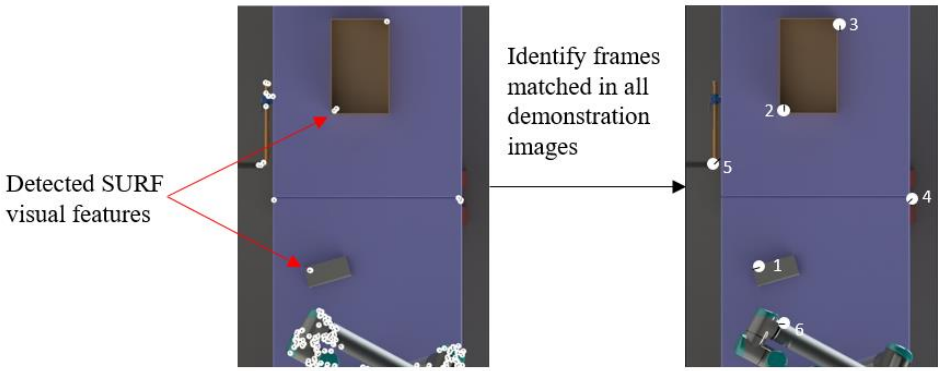


Figure 5. Examples of detected visual SURF features (left) and the common frames between all demonstration images (right).

Hand features: hand features are detected using a pre-trained YOLOv3 neural network [30]. A hand detection is a square bounding box on the 2D image. If multiple hands are detected in a demonstration image, there is no differentiating factor between them except a detection confidence score. Matching hand features across demonstration images is not possible.

Therefore, the hand of the highest confidence score from each demonstration image is considered and the rest of the hand features are ignored.

Moreover, the hand bounding box is not oriented. Therefore, the orientation of the hand is considered being equal in all the images and is set to zero degrees. This is an acceptable assumption since it is likely that the human operator stands in a consistent direction with respect to the cobot.

3.4. Grouping Redundant Frames

Redundant frames are those frames that belong to the same rigid object. Due to their fixed relative position with respect to each other, the obtained TP-GMM for each frame are identical. When reproducing a path using TP-GMM of all redundant frames, the path will be falsely biased towards the largest group of redundant frames. Therefore, redundant frames must be identified and grouped together as one object and only one frame should be used in the TP-GMM and TP-GMR algorithms. This one frame is randomly chosen from the object, and it is called the lead frame.

In this research, a statistic algorithm is devised to identify and group redundant frames based on their relative positions with respect to each other. The algorithm has the following steps:

1. The distance $\Delta b_{jk,m}$ and the relative orientation $\Delta\alpha_{jk,m}$ between two frames j and k in the m-th demonstration are calculated.

$$\Delta b_{jk} = \{\Delta b \; \forall \; 1 \leq m \leq M \; such \; that \; \Delta b_{jk,m} = \sqrt{(x_{j,m} - x_{k,m})^2 + (y_{j,m} - y_{k,m})^2} \quad (1)$$

$$\Delta\alpha_{jk} = \{\Delta\alpha_{jk,m} \; \forall \; 1 \leq m \leq M \; such \; that \; \Delta\alpha_{jk,m} = \; (\alpha_{j,m} - \alpha_{k,m}) \quad (2)$$

2. The standard deviation $\sigma(\Delta b_{jk})$ and the mean $\mu(\Delta b_{jk})$ of the distance Δb between the frames j and k across all the demonstrations are calculated.

$$\mu(\Delta b_{jk}) = \frac{1}{M}\sum_{m=1}^{M} \Delta b_{jk,m} \quad (3)$$

$$\sigma(\Delta b_{jk}) = \sqrt{\sum_{m=1}^{M} \left(\Delta b_{jk,m} - \mu(\Delta b_{jk})\right)^2 \Big/ M} \quad (4)$$

3. The standard deviation $\sigma(\Delta\alpha_{jk})$ and the mean $\mu(\Delta\alpha_{jk})$ of the relative orientation $\Delta\alpha$ between the frames j and k across all the demonstrations are calculated.

$$\mu(\alpha_{jk}) = \frac{1}{M}\sum_{m=1}^{M} \Delta\alpha_{jk,m} \quad (5)$$

$$\sigma(\Delta\alpha_{jk}) = \sqrt{\sum_{m=1}^{M} \left(\Delta\alpha_{jk,m} - \mu(\Delta\alpha_{jk})\right)^2 \Big/ M} \quad (6)$$

4. If the standard deviation divided by the mean, $\sigma(\Delta b_{jk})/\mu(\Delta b_{jk})$ and $\sigma(\Delta\alpha_{jk})/\mu(\Delta\alpha_{jk})$ are both below a certain threshold ε, the frames are deemed redundant.

$$if \; \frac{\sigma(\Delta b_{jk})}{\mu(\Delta b_{jk})} < \varepsilon \cap \frac{\sigma(\Delta\alpha_{jk})}{\mu(\Delta\alpha_{jk})} < \varepsilon \quad (7)$$

5. A *PxP* redundancy matrix *obj* is defined to represent the redundancy relationship between the frames, where *P* is the total number of the frames. If the frames *j* and *k* are redundant, obj_{jk} is set to 1. Then,

$$obj_{jk} = obj_{kj} = 1 \tag{8}$$

6. When the matrix is completed, frames that are redundant are grouped together into an object. From each object, the frame with the highest detection confidence value is set to be the *lead frame*. The *lead frame* will be used in training the TP-GMM.
7. If the total number of the obtained lead frames is more than the maximum number of the lead frames, *maxlead*, the threshold is increased by an increment and Steps 4 to 7 are repeated.
8. If the total number of the obtained lead frames is less than the *maxlead*, the process is completed.

The maximum number of lead frames allowed is set to be 25. The threshold ε is initialized as 0.05 and increased by increments of 0.01. Figure 6 illustrates the calculation process of the relative distances and angles between a pair of frames. The entire process of the algorithm is given below in Algorithm 1.

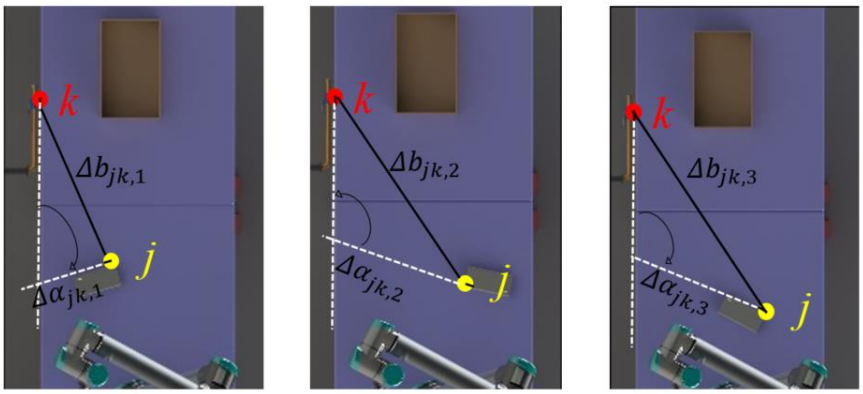


Figure 6. The relative distances and angles between a pair of frames.

Algorithm 1: The statistical algorithm for grouping redundant frames.

Inputs: frames detected from the demonstration images
Initialization:
Threshold ε = 0.05
The maximum number of lead frames *maxlead* = 25
Loop over each pair of frames *j* and *k:*
Step 1: Calculate the mean μ and the standard deviation σ of the relative distances Δb_{jk} and angles $\Delta\alpha_{jk}$;
Step 2: IF $\frac{\sigma(\Delta b_{jk})}{\mu(\Delta b_{jk})} < \varepsilon \cap \frac{\sigma(\Delta\alpha_{jk})}{\mu(\Delta\alpha_{jk})} < \varepsilon$, THEN
$obj_{jk} = obj_{kj} = 1$ (the frames *j* and *k* belong to the same object, i.e., redundant)
jump to Step 6
ELSE
Set $obj_{jk} = obj_{kj} = 0$ (the frames *j* and *k* do not belong to the same object)
END IF

END FOR

Step 3: Merge groups of the redundant frames to obtain clusters called *objects*; from each object, choose one *lead frame*;

Step 4: IF the number of lead frames < *maxlead* THEN

This algorithm completes

ELSE

$\varepsilon = \varepsilon + 0.01$, and go to Step 2.

END IF

Outputs: frames are grouped into objects, with a lead frame from each object

3.5. Training TP-GMM

Before the process of TP-GMM is introduced, some concepts of Gaussians from demonstrations are given. In Figure 7, the frames 2 from all the three demonstrations shown in Figure 5 are aligned as if the paths are observed from the frames 2. The paths are modeled as a group of Gaussians, A, B and C represented by ellipses in Figure 7. A Gaussian (denoted as the i-th Gaussian here) consists of the mean $\boldsymbol{\mu_i}$ and covariance Δ_i. $\boldsymbol{\mu_i}$ is a 1xD vector and Δ_i is a DxD covariance matrix (D is the dimension of the coordinate system, e.g., 2 or 3). TP-GMM approximates the data distribution of demonstrations based on a series of Gaussians.

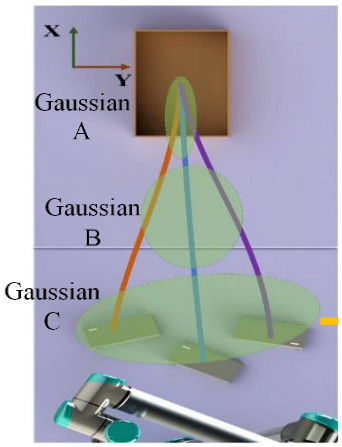


Figure 7. An example of Gaussians produced by TP-GMM to encode the demonstration paths.

The probability density function of a Gaussian is calculated according to the following formula:

$$N_i(\boldsymbol{x_j}|\boldsymbol{\mu_i}, \Delta_i) = \frac{1}{(2\pi)^{D/2}|\Delta_i|^{1/2}} exp(-\frac{1}{2}\left(\boldsymbol{x_j} - \boldsymbol{\mu_i}\right)^T {\Delta_i}^{-1}(\boldsymbol{x_j} - \boldsymbol{\mu_i})) \quad (9)$$

where N_i is the probability density function of the i-th Gaussian; $\boldsymbol{x_j}$ represents a point from demonstrations (the number of points in demonstration is j=1,.., M); D is the dimension of the point in the coordinate system; $\boldsymbol{\mu_i}$ is the mean of the i-th Gaussian; Δ_i is the covariance of the i-th Gaussian.

TP-GMM is summed up by the Gaussians with weights as follows:

$$p(\boldsymbol{x_j}) = \sum_{i=1}^{K}(\omega_i \cdot N_i(\boldsymbol{x_j}|\boldsymbol{\mu_i}, \Delta_i)), \text{ s.t. } \sum_{i=1}^{K}\omega_i = 1 \ (0<\omega_i<1) \quad (10)$$

where $p(\boldsymbol{x}_j)$ is the probability density function of TP-GMM; *K* is the total number of Gaussians, w_i is the weight of the i-th Gaussian contributing to the construction of TP-GMM.

Moreover, a differentiation was made between the two types of frames, oriented and orientation-less, as defined by the authors [33]. An adjusted ring Gaussian model is used to model demonstration paths with respect to orientation-less frames [33].

3.6. Identifying Irrelevant Frames

Irrelevant frames are those frames that are not related to the demonstration path's function. For example, when detecting SURF features in an image, some of the features might be extracted from the background, from noise or from useless clutters. The performance of TP-GMR is highly dependent on the set of frames given. If any of the frames are task-irrelevant, the performance of TP-GMR will deteriorate, and the reproduced path might be unsatisfactory.

Therefore, in this paper, a reinforcement learning algorithm is designed to enhance the reproduced path by optimizing the choice of frames used. That is, given the expected number of relevant frames, *nbRelev* is set to be 2, and the algorithm identifies a set of *nbRelev* frames that produce the best path reproduction. The reinforcement learning problem is formulated as follows:

State: The state of the environment is the reproduced path generated in each iteration of the algorithm;

Action: The action set is the various frame combinations that can be chosen to reproduce the path;

Model: The environmental model is the GMMs obtained from training the TP-GMM since they provide a mapping between the chosen frames and he reproduced path;

Reward: The reward *r* is a function that assesses how close the reproduced path is to the demonstration path (ground truth);

Policy: The policy π is a function that describes the probability of a frame being used in the next path reproduction, based on the previous performance of each frame.

The reinforcement learning algorithm is performed in the following steps:

1. Each lead frame *i* is given a relevancy probability, $probRelev_i$, that signifies how likely it is that a frame is relevant to the task, i.e., the probability that it should be used to reproduce the path. Initially, the *probRelev's* are assigned equal values summing up to 1. Therefore, given *P* is the total number of lead frames,
$$probRelev_i = {}^1/_P \tag{11}$$
2. The algorithm is trained over *iterTot* number of iterations. The value of *iterTot* is set to a default value of *10P*x*nbRelev*. In each iteration *iter*, a disturbance $\Delta probRelev_{iter}$ to the relevancy probability is introduced as part of the exploration tactic in the reinforcement learning algorithm. This results in temporary relevancy probabilities $\lambda probRelev$ that are

used as the policy π in iteration *iter*. The $\lambda probRelev$ for all lead frames sum up to 1. Given *rand*, a random number between 0 and 1,

$$\Delta probRelev_{iter} = (rand_i - probRelev) \times 0.05 \quad (12)$$

$$\lambda probRelev_{iter} = (probRelev + \Delta probRelev_{iter}) / \sum probRelev + \Delta probRelev_{iter} \quad (13)$$

3. In each iteration *iter*, a random demonstration *m* is chosen. The *nbRelev* lead frames to be used in path reproduction, i.e., action to be performed, is based on the values of $\lambda probRelev$, i.e. the policy. The *nbRelev* frames of maximum $\lambda probRelev$ are used to reproduce the path in the m-th demonstration. Given the lead frames with maximum *probRelev*, the state (i.e., the path) is reproduced using the model, i.e., the GMMs of these frames. For example, if the number of relevant frames is 2, then the 2 lead frames with the highest value of $\lambda probRelev$ at a certain iteration *iter* are used in TP-GMR.
4. Once a path is generated, a cost is calculated as the average of the Euclidean distance between the points of the reproduced path and the demonstration path for the m-th demonstration. Given that *T* is the total number of points in a path, (x_t, y_t) and ($X_{t,m}$, $Y_{t,m}$) are the *x*-*y* coordinates of the path point at time step *t* on the reproduced path and the path of the m-th demonstration, respectively,

$$cost_{iter} = \frac{1}{T} \sum_{t=1}^{T} \sqrt{(x_t - X_{t,m})^2 + (y_t - Y_{t,m})^2} \quad (14)$$

5. For every *iterPeriod* number of iterations, the costs are normalized between 0 and 1. The value of *iterPeriod* is set to a default of *P*. Then, the reward *r* is calculated using the normalized costs, such that,

$$r_{iter} = sigmoid(-5 \times cost_{iter}) \quad (15)$$

6. For every *iterPeriod* number of iterations, the value of *probRelev* is updated based on the rewards obtained from the different disturbances $\Delta probRelev$. The value of *probRelev* slowly converges to the lead frames obtaining to the highest reward value, i.e., the relevant lead frames.

$$probRelev = probRelev + \sum_{iter} r_{iter} \times \Delta probRelev_{iter} \quad (16)$$

Algorithm 2. A reinforcement learning algorithm for removing irrelevant frames.

Input: demonstration data (paths and images) + trained model of GMMs for each lead frame
Parameter Initialization:
nbRelev = *2 (*default, but adjustable according to task*)*
iterPeriod = *P*
iterTot = *Period* x 10 x *poolsize*
probRelev = 1x*P* vector of {1/*P}*
Loop over *iterTot* number of iterations:
Step 1: Reproduce:
Set *ΔprobRelev* to a random 1x*P* vector of values between 0 and 1-prob*Relev*
Let $\lambda probRelev_{iter}$ be prob*Relev* + *ΔprobRelev*
Sort $\lambda probRelev_{iter}$
m is a randomly chosen demonstration
filter_frames is the first *nbRelev* frame from Demonstration *m* of maximum $\lambda probRelev_{iter}$
Reproduce a path using TP-GMMs of *filter_frames* only
$cost_{iter}$ is the distance between the reproduced path and the demonstration path of *m*
Step 2: Update:
IF *iter* is a multiple of iter*Period* THEN
Normalize costs from all previous iterations between 0 and 1
FOR each *iter* in the past *iterPeriod* iterations
r_{iter} = sigmoid(-5 x $cost_{iter}$)
probRelev = *probRelev* + r_{iter} x $\lambda probRelev_{iter}$ /*iterPeriod*
END FOR
END IF
END FOR
Output: Relevant frames = the first *nbRelev* frames of maximum *probRelev*

3.7. Calculating TP-GMR for a Reproduced Path

Given a new setup with objects placed in previously unobserved positions, the aim is to reproduce the cobot's path based on what has been previously learnt from the demonstrations. Firstly, an image is recorded of the new scene and SURF features are extracted from the image. The features are matched with the relevant features from demonstration images. The aim to localize the relevant features in the new image in their new positions. These relevant features are then used in TP-GMR to reproduce the new path. The ring Gaussians of orientation-less frames are transformed into normal Gaussians prior to performing TP-GMR [33]. In the calculation process of TP-GMR, $(x_j, y_j)$is denoted as x_j. The basic steps of TP-GMR are as follows (using the calculation process on x_j as an example):

Step 1: Joint probability $p(t, x_j)$ – the calculation is as follows:

$$p(t, x_j) = \sum_{i=1}^{K} (\omega_i \cdot N_i(((x_j, t)|\boldsymbol{m_{it}}, \boldsymbol{cov_i}) \cdot N_i(t|\boldsymbol{\mu_{it}}, \Delta_{\boldsymbol{it}})), \text{ s.t. } \sum_{i=1}^{K} \omega_i = 1 \ (0<\omega_i<1) \quad (17)$$

where $\boldsymbol{\mu_i}$ and $\Delta_{\boldsymbol{i}}$ are represented in Equation (18); $\boldsymbol{\mu_{it}}$ represents the means of all the $\boldsymbol{x}$ for the i-th Gaussian; $\Delta_{\boldsymbol{itx}}$ represents covariance of *t* and all the $\boldsymbol{x}$ for the i-th Gaussian, and so on for each parameter.

$$\boldsymbol{\mu_i} = \begin{bmatrix} \boldsymbol{\mu_{it}} \\ \boldsymbol{\mu_{ix}} \end{bmatrix} \quad \Delta_i = \begin{bmatrix} \Delta_{itt} & \Delta_{itx} \\ \Delta_{ixt} & \Delta_{ixx} \end{bmatrix} \tag{18}$$

In Equation (17), $N_i((x_j, t)|m_{it}, cov_i)$ is the conditional probability density function for x_j relative to *t*; the mean $m_i(t)$ and covariance cov_i are shown in Equations (19) and (20):

$$\boldsymbol{m_{it}} = \boldsymbol{\mu_{ix}} + \Delta_{ixt} \cdot \Delta_{itt}^{-1} \cdot (t - \boldsymbol{\mu_{it}}) \tag{19}$$

$$\boldsymbol{cov_i} = \Delta_{ixx} - \Delta_{ixt} \cdot \boldsymbol{\Delta_{itt}^{-1}} \cdot \Delta_{itx} \tag{20}$$

Step 2: Marginal probability $p(t)$:

The marginal probability is shown in Equation (21):

$$p(t) = \int p(t, x_j) dx = \sum_{i=i}^{K} \omega_i \cdot N_i(t|\boldsymbol{\mu_{it}}, \Delta_{it}) \tag{21}$$

Step 3: The output data x_j':

To sum up, the function of regression can be inferred from the joint probability and marginal probability. The conditional probability $p(x_j|t)$ is shown in Equation (22):

$$p(x_j|t) = \frac{p(t, x_j)}{p(t)} = \frac{\sum_{i=1}^{K}(\omega_i \cdot N_i((x_j, t)|\boldsymbol{m_{it}}, \boldsymbol{cov_i}) \cdot N_i(x_j|\boldsymbol{\mu_{it}}, \Delta_{it}))}{\sum_{i=i}^{K} \omega_i \cdot N_i(t|\boldsymbol{\mu_{it}}, \Delta_{it})}$$
$$= \sum_{i=1}^{K} \alpha_i(t) \cdot N_i((x_j, t)|\boldsymbol{m_{it}}, \boldsymbol{cov_i}) \tag{22}$$

$\alpha_i(t)$ in Equation (22) is the mixture weights of TP-GMR:

$$\alpha_i(t) = \frac{\omega_i \cdot N_i(t|\boldsymbol{\mu_{it}}, \Delta_{it})}{\sum_{i=1}^{K} \omega_i \cdot N_i(t|\boldsymbol{\mu_{it}}, \Delta_{it})} \tag{23}$$

The final regression data x_j' from TP-GMR is represented in Equation (24):

$$x_j' = \sum_{i=1}^{K} \alpha_i(t) \cdot m_i(t) \tag{24}$$

3.8. Reproduced Path Adjustment for Dynamic Positions

To better address dynamic situations, in this research, a B-spline [31] based cut-in algorithm is integrated to be adaptive to changes on the generated regression path. The algorithm can smoothen the reproduced path while keeping the characteristics of the path for a dynamic setting. The process can be explained using the example shown in Figure 8. In Figure 8 (a), the curve *R* is the original regression path generated using TP-GMR, in which O_s is the start point, O_e is the end-point, and $O_1 - O_5$ are the intermediate points. Point *a* is a new start point in a dynamic situation, and O_e is still the end-point in the situation. Figure 8 (b) shows the adjusted reproducing path represented in the curve *NR* using the B-spline cut-in algorithm.

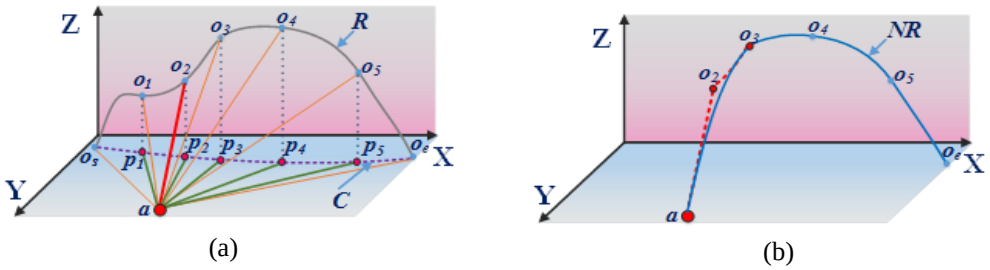


Figure 8. The B-spline based cut-in algorithm for path reproducing in a dynamic situation: (a) calculation of the cut-in point. (b) the improved reproduced path.

The procedures of the algorithm are below:

1. If Point *a* is simply inserted into Curve *R*, there could exist a redundant turning problem in the regression path. To avoid this problem, an improved procedure is designed here. As shown in the Figure 8 (a), based on Points *a*, O_s and O_e, an *x*-*y*-*z* coordinate system is built. The points on the demonstration path are projected on the *x*-*y* plane ($p_1 - p_5$). The distances between Points *a* and the projected points ($p_1 - p_5$) are calculated. The point in the shortest distance between its projected point on the *x*-*y* plane and Point *a* is chosen as the cut-in point, and the points before the cut-in point on the regression path are removed.
2. A new regression path, starting from Point *a*, should be integrated into the trimmed regression path starting from the cut-in point. To make the integrated path smooth, a B-spline curve is introduced. Point *a*, the cut-in point and the next point to the cut-in point on regression path are knot points for generating the B-spline curve. The formula is represented below.

$$P(u) = \sum_{i=0}^{n} P_i \cdot B_{i,k}(u) \tag{25}$$

where P_i is the set of control points (i.e., $P_i = (a, O_2, O_3)$ in this research); $B_{i,k}(u)$ means the base functions of *k-times* B-spline; in the research, k = 3 (called a cubic B-spline).

4. Results

4.1. Simulation Results

The improved TP-LfD approach presented in this paper was used to program the cobot for four tasks as case studies for research validation based on the CoppeliaSim EDU software. The case studies are shown in Figure 9. The functions of the tasks are as follows:

- Task 1 – Apply Glue on Book Spine: This is a one-object operation in which the cobot needs to trace a line of glue on the spine of a book cover. The book varies position and orientation. This could be done in a human-robot interaction task where the human picks the hardcover afterwards and places it accurately on the book.
- Task 2 - Pick-and-place: This is a two-object operation in which the cobot picks up a cube and places it in a box. Both objects in this task change in positions and orientations. This task represents of a manufacturing task, i.e., such as assembly.
- Task 3 – Sort Circuit Boards: Given two circuit boards, the cobot picks the smaller one and places it in the green of the boxes. This is analogous to conditional manufacturing tasks such as sorting.
- Task 4 – Handover: In this task, the robot picks up an object and hands it to the human hand. The object and the hand vary positions. The hand is detected using YOLO (Redmon and Farhadi, 2018) which is integrated in improved TP-LfD approach.

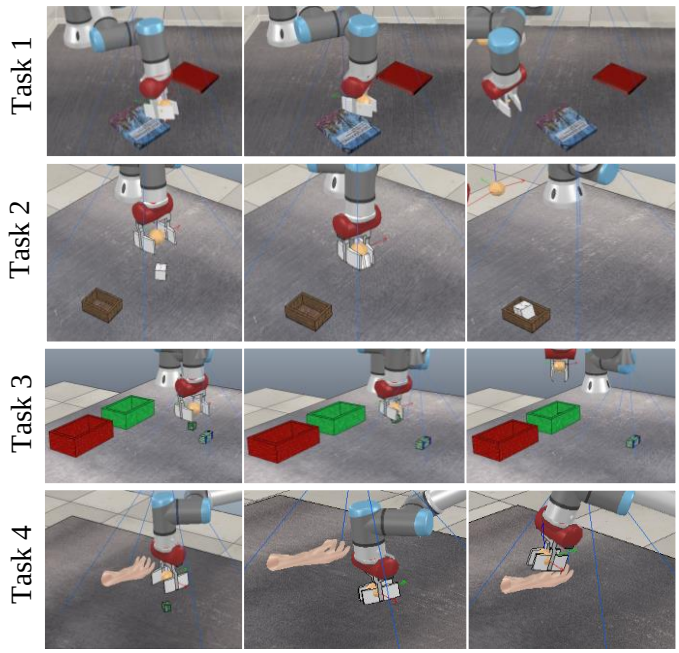


Figure 9. Four scenarios to validate the improved TP-LfD approach.

The improved TP-LfD approach was trained using five demonstrations. The approach identified the relevant frames that belong to the task-relevant object(s). The path is generated using TP-GMR based on TP-GMM of the relevant frames. Figure 10 shows the results in the validation for each task:

- The first row shows the results of grouping redundant frames together. In each scenario, feature of the same color belong to the same object, i.e. are redundant.
- The second row shows the demonstration path recorded by the user (green), the lead frames that are used in training TP-GMM, and the reproduced path (white) using these lead frames.
- The third row shows the results of the reinforcement learning algorithm eliminating the irrelevant frames. The identified relevant frames are shown as well as the reproduced path (white) using them. It can be observed that the path reproduced using relevant frames is significantly closer to the demonstration path than the path reproduced using all lead frames.

Table 2 presents a quantitative comparison between the paths. The comparison metrics are: d_{all}, the average of the Euclidean distance between the all points on the reproduced path and the demonstration path; d_{start}, the Euclidean distance between the start point of the reproduced and the demonstration paths; d_{end}, the Euclidean distance between the end point of the reproduced and the demonstration paths. The results show that overall, the path reproduced using relevant frames is closer to the demonstration path, and provides the higher accuracy for the path's target start and end positions in most cases.

Table 2. Comparison between reproduced paths using all frames vs. relevant frames only.

Path	Metric	Task 1	Task 2	Task 3	Task 4
Reproduced path using all frames	d_{all}	0.0858	0.064	0.04	0.0126
	d_{start}	0.1434	0.0071	0.022	0.00126
	d_{end}	0.0842	0.2443	0.038	0.1139
Reproduced path using relevant frames only (ours)	d_{all}	0.0032	0.0126	0.0102	0.0107
	d_{start}	0.0037	0.0043	0.0042	0.0033
	d_{end}	0.0035	0.0409	0.0123	0.022

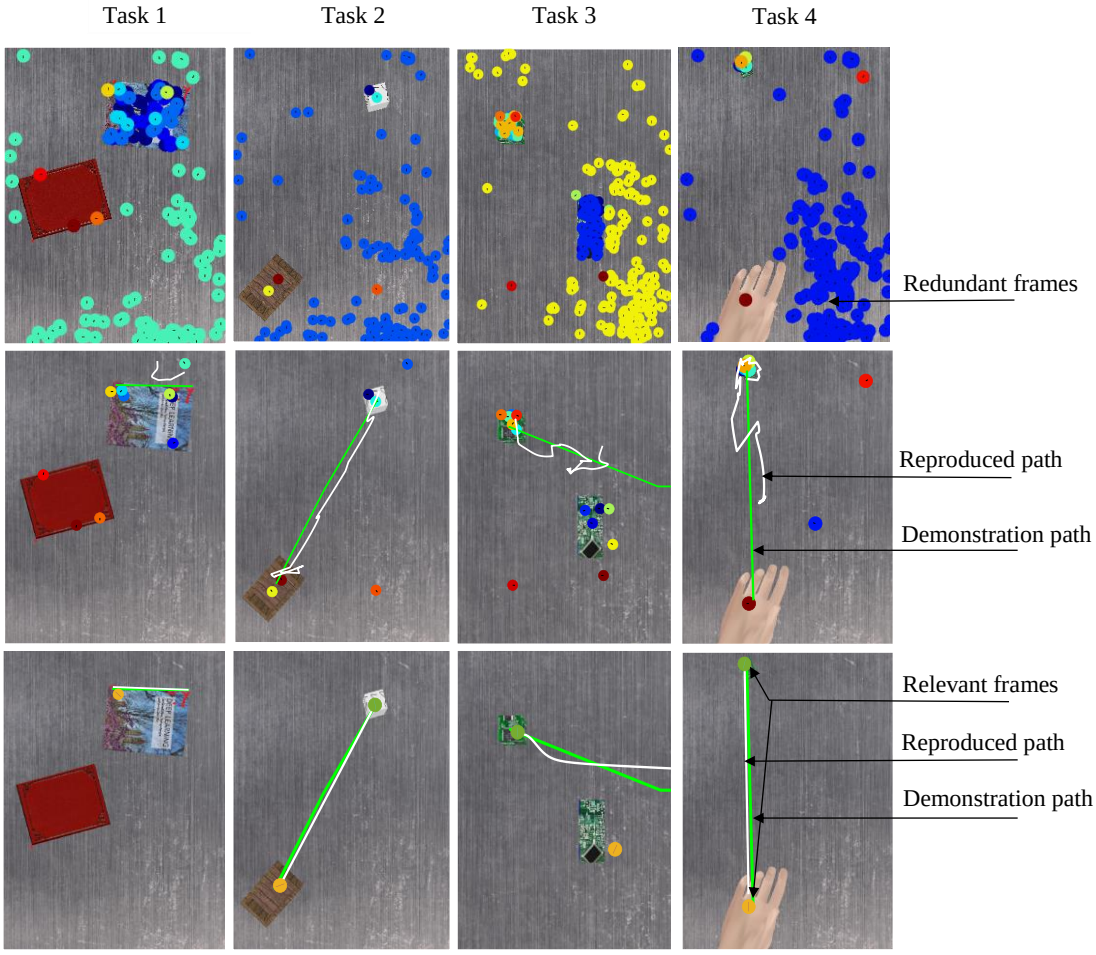


Figure 10. Training results on the validation image.

Moreover, the trained models are tested in new scenarios. The reproduced path is generated in 3D with gripper action. A recording of the cobot's reproduced path can be watched in this video: https://youtu.be/Y3m2M64OazI .

4.2. Experimental Results

In this sub-section, the improved TP-LfD approach was tested on a battery assembly process. The main aim of this sub-section is to highlight the adaptability of the approach to different tasks using different parts and textures. The following tasks are learnt by the approach:

1. Task 1 - Place Battery Holder on Plate: The robot needs to pick up a car battery holder and place it on a cooling plate. The battery holder varies position on a table, while the plate is stationary. The placing position on the plate is fixed but with loose tolerance.
2. Task 2 - Place Battery in Holder: This is a peg-in-hole application. The robot picks up a battery in upright position and place it in a slot in a car-battery holder. The tolerance of the placing position here is tighter than in Task 1.
3. Task 3 - Trace Battery Contact Point: After filling the battery holder with batteries and placing the holder's lid, this part of the task is to jet an insulating paste around the contact point.

In Figures 11, 12 and 13, the demonstration images show identified frames (colored numbered dots) obtained after the first step of filtering through redundant frames. After training TP-GMM and identifying the relevant frames (denoted as the white dots) using the reinforcement algorithm, TP-GMR is used to reproduce the path (denoted in white). Upon being given an image of a new setup, the relevant frames are localized (donated as yellow dots) and the path is reproduced using TP-GMR.

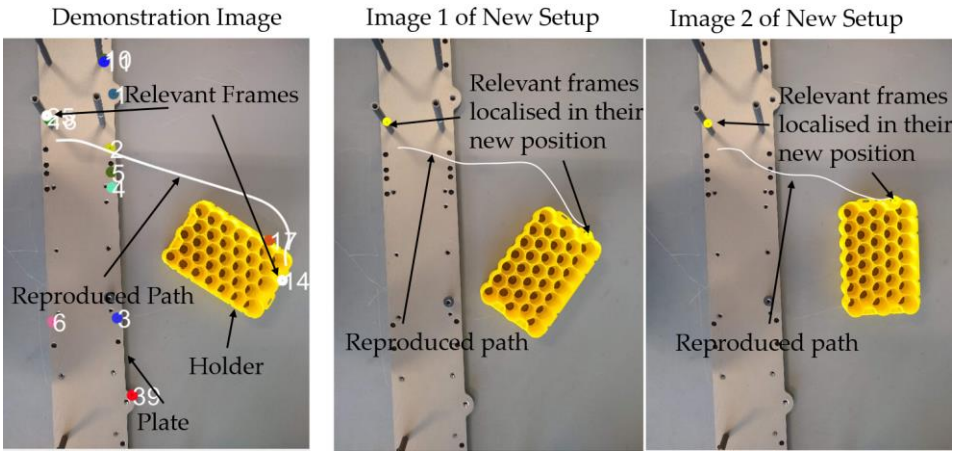


Figure 11. Results of applying the improved TP-LfD approach on Task 1 – Placing the battery holder on the cooling plate.

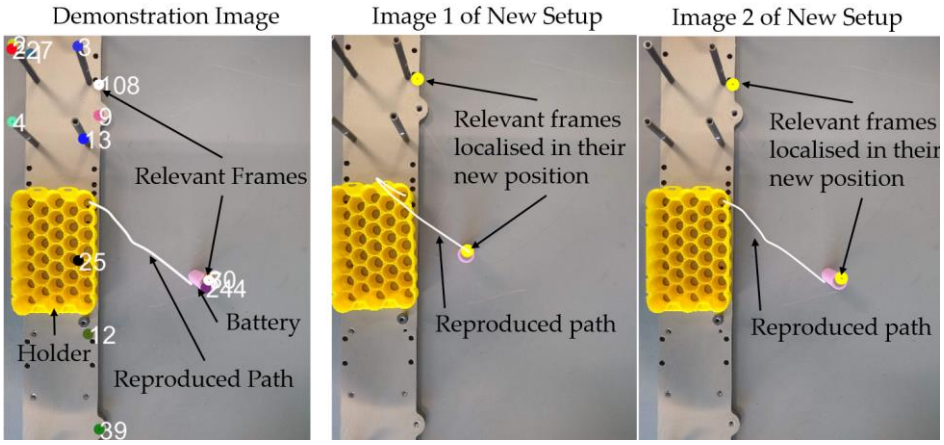


Figure 12. Results of applying the improved TP-LfD approach on Task 2 – Placing the battery inside the battery holder.

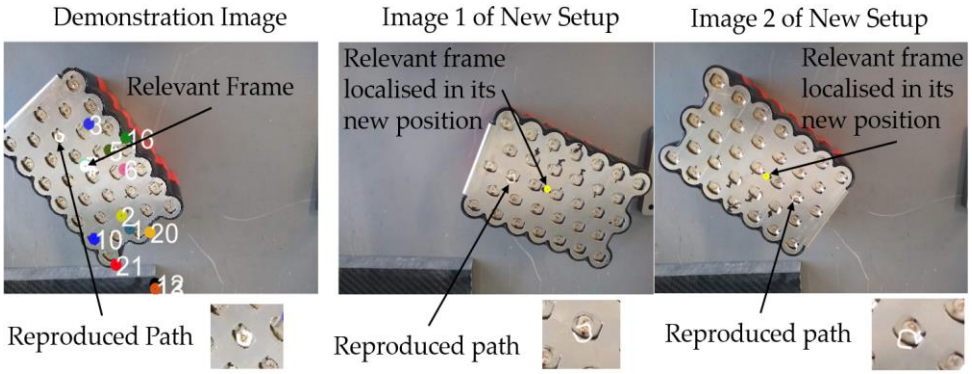


Figure 13. Results of the applying the improved TP-LfD on Task 3 – Jetting insulating pasting on contact point of battery with the holder's lid.

Results show the improved TP-LfD approach successfully learnt demonstrations involving a range of object material from plastic to metallic surfaces, and a range of object sizes from small batteries to large plates. Some of the observations include:

- Detected relevant frames might be mis-localized in objects having symmetrical features, such as the battery holder in Task 1. For example, in the demonstration image, the relevant feature is on the left half of the battery holder. However, in the new setup images, the relevant feature is detected to be to the right of the battery holder. The error originates from the SURF feature matching algorithm, which could be prone to error when handling symmetrical objects. For this particular task, this error does not present a major issue since the tolerance in moving the object is high. However, in future works, other 2D features will be investigated for detection/matching to overcome such an error. In Tasks 2 and 3, the error does not occur since the objects in the image are not symmetrical.
- In Tasks 2 and 3, the improved TP-LfD approach performed well, showing that the path successfully reaches the hole in the battery holder or successfully traces the battery's contact point, respectively. However, a task such as lining batteries in a battery holder is an arrayed task. The improved TP-LfD approach does not yet provide an intelligent solution for arrayed tasks, so each operation in the array of pick-and-place would have to be coded individually. However, since arrayed tasks often occur in repetitively patterned objects, the visual features are repeated along the object. Therefore, this can be exploited to enable the cobot to learn the arrayed action from a few sparse demonstrations. In future works, we aim to extend the improved TP-LfD approach to intelligently support more complex tasks, such as arrayed tasks as well as conditional tasks.

4.3. Implementation

In this sub-section, a full physical implementation on a pick-and-place task was demonstrated. The hardware set-up is shown in Figure 14. The experimental set-up includes a UR5 cobot, a 2-finger adaptive gripper, a switch, a 3-D camera and a computer. In the experiment, the images in the demonstrations are recorded in 2-D using the camera to simplify the computation process.

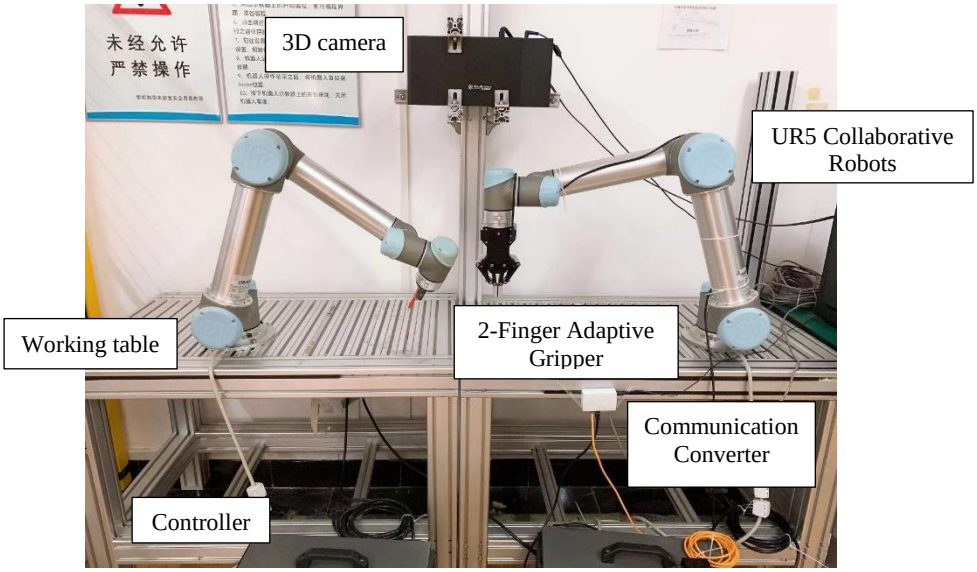


Figure 14. The experimental platform.

The aim of the pick-and-place task is to pick a bolt (target-A) and place it in a rectangular storage box (des-B). The position and direction of target-A can be varied while des-B is chosen to be fixed. Figure 15 shows the table-top images for a demonstration, the frames detected in the demonstration and the demonstration path. There are four demonstrations built up.

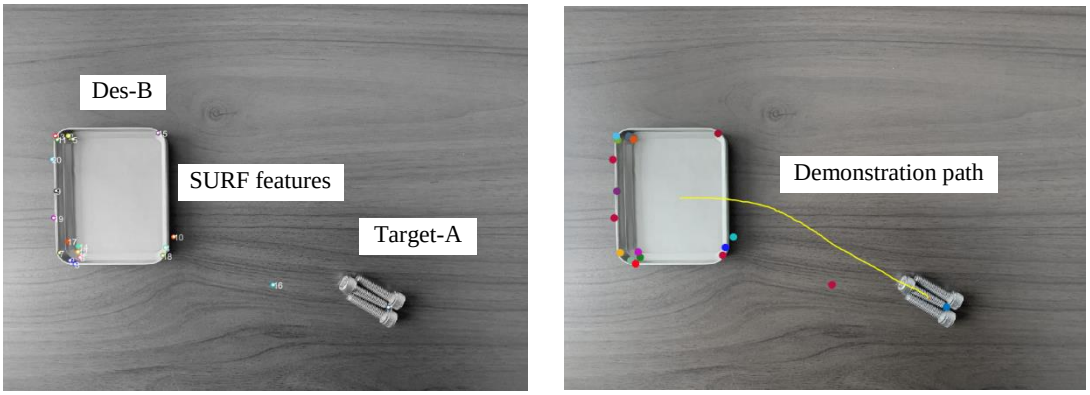


Figure 15: A demonstration, the identified features and the demonstration path.

According to the improved TP-LfD approach, the paths are used to train TP-GMM. Moreover, two optimal task parameters, i.e., frames, are autonomously detected, one belonging to target-A and one to des-B (as shown in Figure 15). Given a new image of the setting, shown in Figure 16, the optimal frames are localised and the path is generated using TP-GMR.

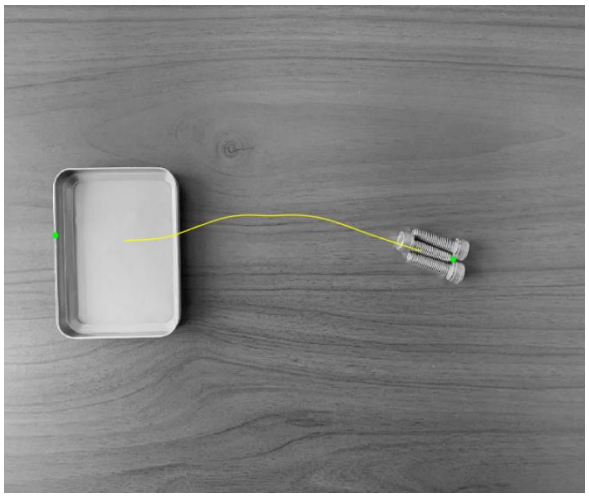


Figure 16: The test image and reproduction path.

5. Conclusions

In this paper, an improved TP-LfD approach is presented to support cobots to adapt their trajectory paths intelligently and adaptively for a variety of dynamic manufacturing applications. To enhance computational efficiency and performance robustness of the improved TP-LfD approach, frames identified in demonstrations needs to be optimized.

The optimization process includes: 1) a statistical algorithm is developed to group redundant frames, and 2) a reinforcement learning algorithm is devised to eliminate irrelevant frames. Meanwhile, a B-Spline cut-in algorithm is designed to enable reproduced paths to better address dynamic situations. Case studies with different complexities showed that the overall performance of the improved TP-LfD approach is significantly enhanced in terms of adaptability and robustness.

In future works, improvements of further eliminating irrelevant frames will be explored. That includes experimenting with more visual feature detection techniques besides SURF. It is targeted to extend the improved TP-LfD approach to more complex industrial tasks including arrayed operations. It is expected to conduct a user-study for operators. Based on the research, it is aimed to eventually leading to the deployment of the improved TP-LfD approach on factory floors.

Acknowledgement:

This research is funded by the Coventry University, the Unipart Powertrain Application Ltd. (U.K.), the Institute of Digital Engineering, U.K., and a research project sponsored by the National Natural Science Foundation of China (Project No. 51975444). We would also acknowledge reviewers for their valuable and constructive comments for us to improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References:

[1] D.-Y. Kim, J.-W. Park, S. Baek, K.-B. Park, H.-R. Kim, J.-I. Park, H.-S. Kim, B.-B. Kim, H.-Y. Oh, K. Namgung and W. Baek, "A modular factory testbed for the rapid reconfiguration of manufacturing systems," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 661-680, 2019.

[2] A. Realyvásquez-Vargas, K. C. Arredondo-Soto, J. L. García-Alcaraz, B. Y. Márquez-Lobato and J. Cruz-García, "Introduction and configuration of a collaborative robot in an assembly task as a means to decrease occupational risks and increase efficiency in a manufacturing company," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 315-328, 2019.

[3] L. Tamas and M. Murar, "Smart CPS: vertical integration overview and user story with a cobot," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 4-5, pp. 504-521, 2019.

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

[5] M. S. Malekzadeh, J. F. Queißer and J. J. Steil, "Multi-level control architecture for Bionic Handling Assistant robot augmented by learning from demonstration for apple-picking," *Advanced Robotics*, vol. 33, no. 9, pp. 469-485, 2019.

[6] O. Koc and J. Peters, "Learning to serve: an experimental study for a new learning from demonstrations framework," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1784-1791, 2019.

[7] D. A. Duque, F. A. Prieto and J. G. Hoyos, "Trajectory generation for robotic assembly operations using learning by demonstration," *Robotics and Computer Integrated Manufacturing*, vol. 57, pp. 292-302 , 2019.

[8] S. Calinon, "A tutorial on task-parametrized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1-29, 2016.

[9] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016, pp. 64.

[10] D. Vogt, S. Stepputtis, S. Grehl, B. Jung and H. B. Amor, "A system for learning continuous human-robot interactions from human-human demonstrations," in *the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[11] C. Paxton, A. Hundt, F. Jonathan, K. Guerin and G. D. Hager, "CoSTAR: Instructing collaborative robots with behavior trees and vision," in *the Proceedins of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[12] C. Perez-D'Arpino and J. A. Shah, "C-LEARN: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy," in *the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[13] A. M. Ghalamzan and M. Ragaglia, " Robot learning from demonstrations: Emulation learning in environments with moving obstacles," *Robotics and Autonomous Systems*, vol. 101, p. 45-56, 2018.

[14] S. Zheng, "Design of intelligent manufacturing product identification and detection system based on machine vision," in *the Proceedings of the International Conference on Cyber Security Intelligence and Analytics*, pp. 258-265, 2020.

[15] A. Rogowski and P. Skrobek, "Object identification for task-oriented communication with industrial robots," *Sensors*, vol. 20, no. 6, pp. 1773, 2020.

[16] Y. Gu, W. Sheng, C. Crick and Y. Ou, "Automated assembly skill acquisition and implementation through human demonstration," *Robotics and Autonomous Systems*, vol. 99, pp. 1-16, 2018.

[17] M. Chouai, M. Merah, J.-L. Sancho-Gomez and M. Mimi, "Supervised feature learning by adversarial autoencoder approach for object classification in dual X-ray image of luggage," *Journal of Intelligent Manufacturing*, vol. 31, no. 5, pp. 1101-1112, 2019.

[18] D. P. Penumuru, S. Muthuswamy and P. Karumbu, "Identification and classification of materials using machine vision and machine learning in the context of industry 4.0," *Journal of Intelligent Manufacturing*, vol. 31, no. 5, pp. 1229-1241, 2019.

[19] K. D. Joshi, V. Chauhan and B. Surgenor, "A flexible machine vision system for small part inspection based on a hybrid SVM/ANN approach," *Journal of Intelligent Manufacturing*, vol. 31, no. 1, pp. 103-125, 2018.

[20] Z. Jia, M. Lin, Z. Chen and S. Jian, "Vision-based robot manipulation learning via human demonstrations," arXiv:2003.00385, 2020.

[21] P. Shi, Q. Qi, Y. Qin, P. J. Scott and X. Jiang, "A novel learning-based feature recognition method using multiple sectional view representation," *Journal of Intelligent Manufacturing*, vol. 31, no. 5, pp. 1291-1309, 2020.

[22] T. Alizadeh and N. Karimi, "Exploiting the task space redundancy in robot programming by demonstration," in *the Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation*, Changchun, 2018.

[23] T. Alizadeh and M. Malekzadeh, "Identifying the relevant frames of reference in programming by demonstration using task-parameterized Gaussian mixture regression,"

in *the Proceedings of the 2016 IEEE/SICE International Symposium on System Integration*, Sapporo, 2016.

[24] A. Sena, B. Michael and M. Howard, "Improving Task-Parameterised Movement Learning Generalisation with Frame-Weighted Trajectory Generation," in *the Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[25] Y. Huang, J. Silverio, L. Rozo and D. G. Caldwell, "Generalized task-parameterized skill learning," in *the Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[26] H. Girgin, E. Pignat, N. Jaquier and S. Calinon, "Active improvement of control policies with Bayesian Gaussian mixture model," in *the Proceedings of the IEEE International Conference on Robotics and Intelligent Systems*, 2020.

[27] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," in *the Proceedings of the European Conference on Computer Vision*, 2006.

[28] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," in *the Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies - iCoMET 2018*, 2018.

[29] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *the Proceedings of the International Conference on Computer Vision Theory and Applications*, 2009.

[30] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv:1804.02767*, 2018.

[31] F. Xie, L. Chen, Z. Li and K. Tang, "Path smoothing and feed rate planning for robotic curved layer additive manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 65, pp. 101967, 2020.

[32] S. E. Zaatari and W. D. Li, "Visual features as task parameters in task parametrised learning from demonstration," in *the Proceedings of the UK-RAS19 Conference*, Loughborough, 2019.

[33] S. E. Zaatari, W. D. Li and Z. Usman, " Ring Gaussian mixture modelling and regression for collaborative robot", submitted to *Robotics and Autonomous Systems*, 2020.