

Optimised Learning from Demonstrations for Collaborative Robots

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

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:

Wang, Y, Hu, Y, El Zaatari, S, Li, W & Zhou, Y 2021, 'Optimised Learning from Demonstrations for Collaborative Robots', *Robotics and Computer-Integrated Manufacturing*, vol. 71, 102169.

<https://dx.doi.org/10.1016/j.rcim.2021.102169>

DOI 10.1016/j.rcim.2021.102169

ISSN 0736-5845

Publisher: Elsevier

© 2021, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

Optimised Learning from Demonstrations for Collaborative Robots

Y.Q. Wang¹, Y.D. Hu¹, S. El Zaatari², W.D. Li^{1,2*}, Y. Zhou^{1*}

¹ School of Logistics Engineering, Wuhan University of Technology, Wuhan, China

² Faculty of Engineering, Environment and Computing, Coventry University, Coventry, UK

*Correspondence author: weidong.li@coventry.ac.uk, zhoyong@whut.edu.cn

Abstract

The approach of Learning from Demonstrations (LfD) can support human operators especially those without much programming experience to control a collaborative robot (cobot) in an intuitive and convenient means. Gaussian Mixture Model and Gaussian Mixture Regression (GMM and GMR) are useful tools for implementing such a LfD approach. However, well-performed GMM/GMR require a series of demonstrations without trembling and jerky features, which are challenging to achieve in actual environments. To address this issue, this paper presents a novel optimised approach to improve Gaussian clusters then further GMM/GMR so that LfD enabled cobots can carry out a variety of complex manufacturing tasks effectively. This research has three distinguishing innovative characteristics: 1) a Gaussian noise strategy is designed to scatter demonstrations with trembling and jerky features to better support the optimisation of GMM/GMR; 2) a Simulated Annealing-Reinforcement Learning (SA-RL) based optimisation algorithm is developed to refine the number of Gaussian clusters in eliminating potential under-/over-fitting issues on GMM/GMR; 3) a B-spline based cut-in algorithm is integrated with GMR to improve the adaptability of reproduced solutions for dynamic manufacturing tasks. To verify the approach, cases studies of pick-and-place tasks with different complexities were conducted. Experimental results and comparative analyses showed that this developed approach exhibited good performances in terms of computational efficiency, solution quality and adaptability.

Keywords: Learning from Demonstrations, Gaussian Mixture Model, Collaborative Robots

List of Abbreviations and Symbols:

BIC	Bayesian Information Criterion
CEO	Cross Entropy Optimisation
DMP	Dynamic Moving Primitives
DTW	Dynamic Time Warping
EM	Expectation-Maximisation
$\ Eudis\ $	Euclidean Distance
GMM	Gaussian Mixture Model
GMR	Gaussian Mixture Regression
HMM	Hidden Markov Model
IBE	Integrated BIC and $\ Eudis\ $

LfD	Learning from Demonstrations
LCSS	Longest Common Subsequence
RMS	Root Mean Square
SA-RL	Simulated Annealing-Reinforcement Learning
TP-GMM	Task-Parameterised Gaussian Mixture Model
$\boldsymbol{\mu}_i$	The mean of the i -th Gaussian cluster
$\boldsymbol{\sigma}_i$	The covariance of the i -th Gaussian cluster
\mathbf{a}_{i1}	Eigenvector/direction of the major axis of the i -th Gaussian cluster
\mathbf{a}_{i2}	Eigenvector/direction of the minor axis of the i -th Gaussian cluster
$\sqrt{\lambda_{i1}}$	Length of the major axis of the i -th Gaussian cluster
$\sqrt{\lambda_{i2}}$	Length of the minor axis of the i -th Gaussian cluster
K	The total number of Gaussian clusters
D	The dimension of the coordinate system
$N_i(\mathbf{p}_j \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$	The probability density function of the i -th Gaussian
\mathbf{p}_j	The position of a constructive point from demonstrations
t	Time-step
x_j, y_j, z_j	The real coordinate of demonstration
ω_i	The weight of the i -th Gaussian
$P_{GMM}(\mathbf{p}_j)$	The probability density function of GMM
$\alpha_i(t_j)$	The mixture weights of GMR
LL	Log of the likelihood function of demonstrations
γ_{ji}	The posterior probability of each point \mathbf{p}_j for the i -th Gaussian
$\boldsymbol{\mu}_{noise}$	Mean of Gaussian noises
$\boldsymbol{\sigma}_{noise}$	Variance of Gaussian noises
r	A random value, which is considered as a noise
x'_j, y'_j, z'_j	Gaussian noise-enhanced points of demonstrations
$\ln L(\hat{\theta} B)$	The Logarithm of the likelihood function of the points in Gaussian clusters
p	The number of free parameters in Gaussian clusters
m	The number of the constructive points of demonstrations
P_i	Control points in B-spline
x_j^{GMR}	The regression data – x dimension
$E(x_j t_j)$	Expectation
$B_{i,k}(u)$	The base functions of k -times B-spline

1. Introduction

Industrial robots, which can execute pre-programmed tasks for mass production, lack versatility and adaptability to changes in dynamic and customised manufacturing environments [1,2]. In contrast, a collaborative robot (cobot) with intuitive programming capabilities can work with humans to conduct Human-Robot Collaboration (HRC) in a flexible and safe means [3-6]. In HRC, intuitive programming can facilitate human operators without much robotic programming experience to control cobots more efficiently and cost-effectively [7]. Learning from Demonstrations (LfD) has become an increasingly popular approach for implementing such intuitive programming in cobots [8]. LfD is inspired by imitating human behaviours to acquire new skills. That is, LfD enables learning from a set of human demonstrations on a cobot acting in a few varied settings, generalising them and reproducing solutions for new settings [9-10].

A pick-and-place example illustrated in Fig. 1 is used to explain the concepts of LfD. For this task, a cobot needs to pick an object and place it into a packaging box along a trajectory path. LfD can tackle the problem according to the following steps:

- A human operator demonstrates the operations of dragging the end effector of a cobot for picking the object from several start points and dropping it into the desired packaging box. As shown in Fig. 1, three trajectories (represented in yellow, purple and blue curves) of the cobot are considered as demonstration paths created by the human operator;
- A mathematical model, such as Gaussian Mixture Model (GMM), is used to encode the demonstrated paths. Then, a regression model, such as Gaussian Mixture Regression (GMR), is employed to specify a new reproduced path (illustrated in dash line in Fig. 1). The reproduced path is to interpolate (“mimic” or “learn from”) the geometric features of the demonstration paths. During the above process, the human operator is not required to possess programming skills and robotic knowledge in operating the cobot. Based on the demonstrations and encoding/decoding processes specified in GMM/GMR, the cobot can be adaptively controlled for dynamic settings.

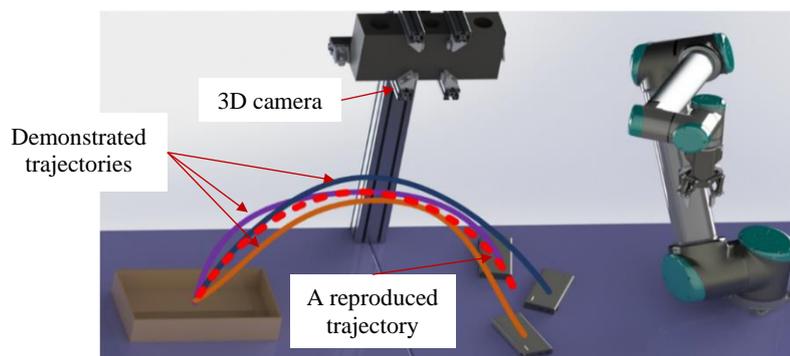


Fig. 1: A pick-and-place task with demonstrations and a reproduced trajectory for a new setting.

GMM/GMR enabled LfD approaches to control cobots intuitively have been actively researched in recent years [11-13]. There are two reasons of using GMM/GMR modelling for LfD: 1) real-world

modelling is close to the Gaussian probabilistic distribution; 2) GMM/GMR are effective in supporting LfD to achieve a good fit into demonstration data without prior knowledge on models' parameters. Gaussian clusters are building blocks for GMM/GMR, and their number and parameters are critical in generating a reproduced path. A good-quality reproduced path is a balance between the interpolation of the geometric features of demonstration paths and the smoothness of the reproduced path to avoid learning trembling and jerky features from the demonstration paths. If the number of Gaussian clusters is not optimised, over-fitting (learn too much from the demonstrations) or under-fitting (learn too little from the demonstrations) in GMM/GMR may occur. For instance, in the pick-and-place example shown in Fig. 1, constructive points of demonstration paths could be grouped based on different numbers of Gaussian clusters (i.e., 1, 3, 8 in Fig. 2). A reproduced path generated based on corresponding Gaussian clusters is “under-fitting”, “appropriate” or “over-fitting” respectively. In Fig. 2(a), although the reproduced path can reach the target, the deviation between the reproduced path and the demonstrations is too big due to a small number of Gaussian clusters used for GMM/GMR. S-shape features embedded in the demonstrations, which ensure the paths to be collision free from an intermediate obstacle, could be lost in the reproduced path. In Fig. 2(c), the path can complete the task and avoid collisions with the obstacle, but there are various sharp turnings in the reproduced path due to the excessive number of Gaussian clusters. This will generate unnecessary time loss and jerk during the robotic movement. Therefore, both Fig. 2(a) and Fig. 2(c) are not good enough compared with Fig. 2(b).

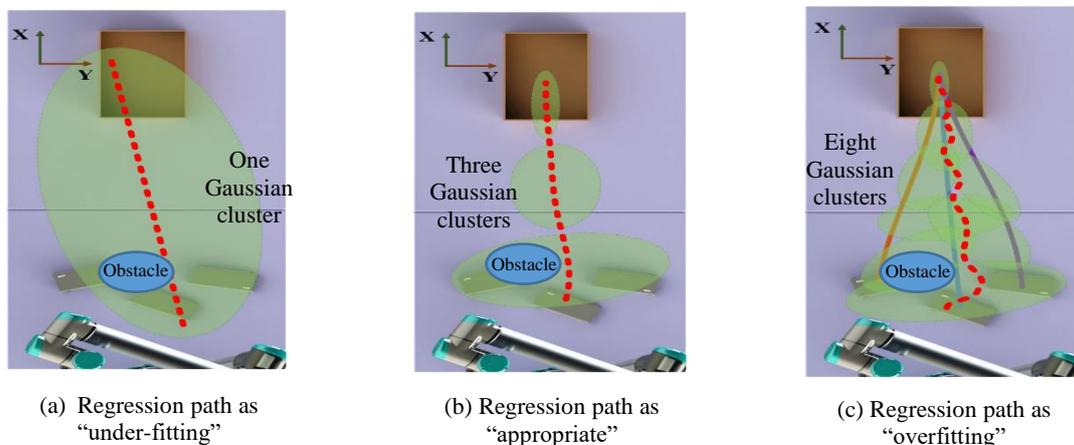


Fig. 2: Reproduced paths using different numbers of Gaussian clusters.

To address the above issue, in this paper, a novel optimisation approach for GMM/GMR based LfD is developed. Distinguishing innovations of the research include the following aspects:

- Design of a Gaussian noise strategy to enrich demonstrations: Trembling and abrupt changes in demonstrations caused by human operators result in generating jerky reproduced paths. With Gaussian noises added to scatter the demonstrations, trembling features in the demonstrations can be rectified during optimisation processes;
- Simulated Annealing-Reinforcement Learning (SA-RL) optimisation on Gaussian clusters: Using conventional optimisation criteria, such as Bayesian Information Criterion (BIC), could lead to a

computational failure in convergence and potential under-/over-fitting issues in GMM/GMR. In this research, a SA-RL based optimisation algorithm, supported by a new optimisation criterion embedded with the K-means and Expectation-Maximisation (EM) algorithms, is designed to avoid the under-/over-fitting issues;

- Adaptive reproduced path generation: For GMR, it could be not adaptive for dynamic manufacturing tasks if the start or end point of a new task is significantly deviated from those in demonstrations. A B-spline based cut-in algorithm is integrated with GMR to enhance the adaptability of the reproduced path to better address the situations.

The rest of the paper is organised as follows. Section 2 reviews the related work. The overall framework of the approach is outlined in Section 3. In Section 4, the technical details of the approach are depicted. Case studies and experimental results are explained in detail in Section 5. Finally, Section 6 concludes the research and envisages the future work.

2. Literature Survey

2.1 LfD and GMM/GMR

LfD is a widely used programming method for different robotic applications. Calinon et al. classified a LfD process roughly into observational learning and kinesthetic teaching [14]. Observational learning refers to visual observations by a human operator and reproduction of specific actions. Kinesthetic teaching refers to moving a robot manually by haptic interaction, and the robot learns from the demonstrations. Argall et al. summarised such the process of LfD as the similar steps but in different terminologies, i.e., gathering examples and deriving a policy from the examples [15]. A good LfD approach is that it can easily capture demonstration data and encode states and behaviours, and then reproduce a solution for a new environment [16]. LfD has been used for various applications and HRC is one of the effective applications of using LfD [14].

GMM/GMR are one of major intelligent technologies to implement LfD. Primary reasons of using GMM/GMR include being easy learning and simplicity of serialising learned behaviours as well as the capability to model internal correlations and constraints within the tasks. Demonstration trajectories can be encoded using GMM and regression trajectories can be generated using GMR for dynamic scenes based on the encoded knowledge. Duque et al. developed GMM/GMR based LfD to learn an assembly task [17]. GMM/GMR were used to encode multiple human demonstrations and a generalised assembly path was obtained by reproduction. However, in the research, the regression path owns several redundant turns, so that it is not effective to drive robots/cobots smoothly. Kyrarini et al. developed GMM/GMR for demonstration learning and reproducing, and Kinect was used to identify objects dynamically [18]. In the work, the smoothness of the regression path was not addressed effectively. Ogenyi et al. combined observational learning and kinaesthetic teaching [19]. In the research, a robot learns by ‘watching’ the movement of the arm of a human operator. The acquired dataset is

complemented using kinaesthetic teaching for further moving. GMM/GMR were used to process the joint angle data of the robot to obtain a regression path. However, the complexity of this system is high, which may bring about the problem of robustness. However, according to the experimental results in the paper, the regression path is relatively simple. A robotic assistance-as-needed framework was proposed for children with cerebral palsy to perform a 2D position-following task [20]. In the research, GMM/GMR were used to build a bridge between children with cerebral palsy and therapist. That is, the proposed system helps correct daily behaviours of children, which puts stricter requirements on the quality of the demonstration path, the smoothness and accuracy of the regression path. Considering the robustness of visual teaching, the research adopts human demonstration (kinaesthetic teaching) to obtain the data.

However, the reproduced path generated by GMM/GMR is inadaptable to a new task if there is a significant change either in the start or end point of the task. Dynamic Moving Primitives (DMP) can be used to enhance GMM/GMR in generating adaptive paths and so it has been used in LfD. Ti et al. developed an approach of combining DMP and GMM/GMR to smoothen a human-like regression path to a new goal [21]. In the research, GMM/GMR was applied to learn from multiple demonstration paths while DMP was used to generalise paths to new environments. However, compared with GMM/GMR or DMP, GMM/GMR combined with DMP increases the uncertainty and complexity of entire models by introducing more parameters. Calinon et al. proposed TP-GMM, i.e., Task-Parameterised GMM, to learn from demonstrations by training GMM by considering different frames [11]. TP-GMM can be adaptive by changing the parameters of frames. However, TP-GMM needs an extra frame recognition algorithm, and it is sensitive to directions leading to poor robustness.

In addition to the widely used GMM/GMR and DMP in LfD, there are some other methods. Rozo et al. used Hidden Markov Model (HMM) to encode a robot pouring task and a ball-in-box task based on demonstrations and GMR was adopted to generate a regression path [22]. Pignat et al. used Hidden semi-Markov Model to encode dressing and shoeing assistance to better support elderly and disable people [23]. However, HMM, which is as an extended model of GMM, is also more complicated, and GMM/GMR are of better operability for applications. Zhang et al. [24] proposed an Adaptive Curve Gaussian Mixture Model (AdC-GMM). In the research, curve Gaussian clusters was designed to model data, and a Cross Entropy Optimisation (CEO) algorithm instead of the traditional EM algorithm [25] was used to calculate the parameters of Gaussian clusters. This method improves the fitting performance of the model, and the reproduction path is also smoother. It has a clear effect on strong non-linear systems, but it makes nearly no difference for a relatively gentle demonstration path.

In summary, in order to improve LfD performance, some trade-offs should be made. That is, a high-precision task generally requires more advanced algorithms or sensors to satisfy users' needs but the system could be more complex. For the above reviewed research, there are two common issues:

- One issue is in demonstrations, i.e., the number of demonstration paths is limited and there could be some inevitable human trembling errors during the generation of the demonstrations. Effective optimisation algorithms are imperative to address this issue;
- Another issue is the inadaptability of LfD-based regression paths suitable for dynamic scenarios. As discussed in [26-27], B-spline [28] can generate different styles of smooth curves by adjusting the control points and weights. Compared with DMP and GMM/GMR, B-spline is more convenient and robust. Motivated by the idea, design of a B-spline cut-in algorithm based on GMM/GMR will be explored in this paper.

2.2 LfD criteria for model parameters selection

To encode demonstrations, criteria for model parameter selection are critical for learning results. BIC, which was proposed by Schwarz et al. [29], is one of the most popular criteria due to its simplicity and effectiveness. In [30-32], the effectiveness of BIC in solving GMM parameters was revealed. BIC can choose a model that maximises the posterior probability of the model. That is, it selects the model that best matches the data distribution. Various researchers have conducted research based on BIC. Mehrjou et al. proposed an improved BIC [33]. In clustering, the performance of the proposed method is better when there is a number of data points or the components of the dataset overlap. However, in LfD applications, as a regression path is used to drive the cobot, the quality of the regression path can directly affect the effect of task execution. BIC and its improved strategy are able to balance the complexity of the model and the degree of fitting. Therefore, BIC can confirm the GMM parameters. As mentioned above, the quality of the regression path is also crucial for driving cobots. Unfortunately, BIC can only guarantee that the selected parameter is optimised for GMM, but the evaluation of the regression path cannot be made using BIC. Therefore, some researchers determined the optimal number range of Gaussian clusters by manual selection and observing the regression path under the different number of Gaussian clusters [15]. In [16], the number of Gaussian clusters was determined by comparing the Root Mean Square (RMS) errors of the regression path and demonstration paths. However, the search range is small, and global or local optimal values were not be identified.

To obtain a proper regression path, it is necessary to calculate the relationship between the regression path and demonstration paths. A distance can be used as one of the criteria to measure the similarity of the paths. Because a regression path is composed of timeseries, three main methods could be employed, i.e., Euclidean Distance ($\|Eudis\|$), Longest Common Subsequence (LCSS), and Dynamic Time Warping (DTW) [34-35]. Considering $\|Eudis\|$ has the lowest computational complexity, if a regression path and demonstration paths share the same timeseries, $\|Eudis\|$ can be used to minimise the use of computing resources. Consequently, it is sensible to consider combining BIC and path similarity to optimise GMM/GMR.

3. Overview of GMM/GMR Optimisation

3.1 Concepts of Gaussian, GMM and GMR

Fig. 3 is used to illustrate the concept of Gaussian clusters. Demonstration paths are decomposed into a series of constructive points. These points are then clustered to form a number of Gaussian clusters (shown in Fig. 3(a)). Each Gaussian (denoted as the i -th Gaussian here) consists of the position mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\sigma}_i$. $\boldsymbol{\mu}_i$ is a $1 \times D$ vector and $\boldsymbol{\sigma}_i$ is a $D \times D$ matrix, where D is the dimension of the coordinate system ($D = 2$ or 3). For a 2-D problem, $\boldsymbol{\sigma}_i$ can be further decomposed to obtain eigenvalues λ_{i1} and λ_{i2} , and orthogonal eigenvectors \mathbf{a}_{i1} and \mathbf{a}_{i2} (shown in Fig. 3(b)). The eigenvectors \mathbf{a}_{i1} and \mathbf{a}_{i2} represent the directions of the major axis and minor axis of the i -th Gaussian, respectively. $\sqrt{\lambda_{i1}}$ and $\sqrt{\lambda_{i2}}$ are the lengths of the major axis and minor axis of the i -th Gaussian, respectively.

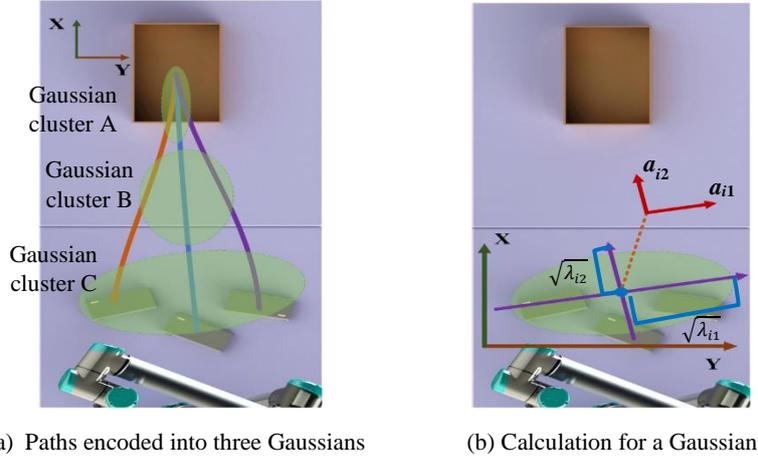


Fig. 3: Gaussian clusters encoding demonstration paths.

The probability density function of the i -th Gaussian $N_i(\cdot)$ is calculated below:

$$N_i(\mathbf{p}_j | \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\sigma}_i|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{p}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\sigma}_i^{-1} (\mathbf{p}_j - \boldsymbol{\mu}_i)\right) \quad (1)$$

where \mathbf{p}_j represents the position of a constructive point from demonstrations ($j=1, \dots, m$; m is the total number of points in all the demonstrations).

GMM is the summed up of weighted Gaussian clusters, and the probability density function of GMM, i.e., $P_{GMM}(\cdot)$, can be calculated as follows:

$$P_{GMM}(\mathbf{p}_j) = \sum_{i=1}^K \omega_i \cdot N_i(\mathbf{p}_j | \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i), \quad \text{s.t. } \sum_{i=1}^K \omega_i = 1 \quad (0 \leq \omega_i \leq 1) \quad (2)$$

Where K is the total number of Gaussian clusters, ω_i is the weight of the i -th Gaussian contributing to the construction process of GMM.

After obtaining GMM, a regression path can be reproduced based on GMR. To implement GMR, a series of time-step $t = (t_1, \dots, t_j, \dots, t_m)$ are added to divide each demonstration path evenly, and the points of each path can be re-written as $[(t_1, \mathbf{p}_1), \dots, (t_j, \mathbf{p}_j), \dots, (t_m, \mathbf{p}_m)]$. Thus, each path has the same number of points for better alignment between demonstrations.

The calculation process of GMR is based on x_j , y_j and z_j respectively ($(x_j, y_j, z_j) \in \mathbf{p}_j$). The basic steps are as follows (using the calculation process on x_j as an example) [36]:

Step 1: According to the conditional probability theorem, the joint probability $P(t_j, x_j)$ is computed below:

$$P(t_j, x_j) = \sum_{i=1}^K (\omega_i \cdot N_i(x_j|t_j; m_i(t_j), cov_i) \cdot N_i(t_j|\mu_{it}, \sigma_{it})), \text{ s.t. } \sum_{i=1}^K \omega_i = 1 \quad (0 \leq \omega_i \leq 1) \quad (3)$$

$$\boldsymbol{\mu}_i = \begin{bmatrix} \mu_{it} \\ \mu_{ix} \end{bmatrix} \quad \boldsymbol{\sigma}_i = \begin{bmatrix} \sigma_{itt} & \sigma_{itx} \\ \sigma_{ixt} & \sigma_{ixx} \end{bmatrix} \quad (4)$$

where μ_{it} represents the means of t_j for the i -th Gaussian; σ_{itt} , σ_{ixt} , σ_{itx} and σ_{ixx} represent the covariance of (t_j, t_j) , (x_j, t_j) , (t_j, x_j) and (x_j, x_j) for the i -th Gaussian respectively; $N_i(x_j|t_j; m_i(t_j), cov_i)$ is the conditional probability density function for x_j relative to t_j ; $m_i(t_j)$ and cov_i are the mean and covariance respectively, which can be computed below:

$$m_i(t_j) = \mu_{ix} + \sigma_{ixt} \cdot \sigma_{itt}^{-1} \cdot (t_j - \mu_{it}) \quad (5)$$

$$cov_i = \sigma_{ixx} - \sigma_{ixt} \cdot \sigma_{itt}^{-1} \cdot \sigma_{itx} \quad (6)$$

Step 2: Marginal probability $P(t_j)$ is computed below:

$$P(t_j) = \int P(t_j, x_j) dx = \sum_{i=1}^K \omega_i \cdot N_i(t_j|\mu_{it}, \sigma_{it}) \quad (7)$$

Step 3: The output data x_j' is computed using the following process:

The function of regression can be inferred from the joint probability and marginal probability. The conditional probability $P(x_j|t_j; m_i(t_j), cov_i)$ is shown in Equation (8):

$$\begin{aligned} P(x_j|t_j; m_i(t_j), cov_i) &= \frac{P(t_j, x_j)}{P(t_j)} = \frac{\sum_{i=1}^K (\omega_i \cdot N_i(x_j|t_j; m_i(t_j), cov_i) \cdot N_i(t_j|\mu_{it}, \sigma_{it}))}{\sum_{i=1}^K \omega_i \cdot N_i(t_j|\mu_{it}, \sigma_{it})} \\ &= \sum_{i=1}^K \alpha_i(t_j) \cdot N_i(x_j|t_j; (t_j), cov_i) \end{aligned} \quad (8)$$

where $\alpha_i(t_j)$ is the mixture weights of GMR, which can be defined below:

$$\alpha_i(t_j) = \frac{\omega_i \cdot N_i(t_j|\mu_{it}, \sigma_{it})}{\sum_{i=1}^K \omega_i \cdot N_i(t_j|\mu_{it}, \sigma_{it})} \quad (9)$$

The final regression data x_j^{GMR} from GMR can be computed in Equation (10):

$$x_j^{GMR} = E(x_j|t_j) = \sum_{i=1}^K \alpha_i(t_j) \cdot m_i(t_j) \quad (10)$$

3.2 Calculation the number of Gaussian clusters for GMM/GMR

The calculation of GMM consists of two primary steps: 1) constructive points are clustered to set up initial Gaussian clusters and the corresponding GMM; 2) the parameters of the initial Gaussian clusters, including ω_i , $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$, are continuously optimised to fine-tune the Gaussian clusters and furthermore GMM/GMR. The K-means algorithm is popularly used to develop Step 1). For Step 2), the EM algorithm is effective in improving these parameters iteratively.

The procedure of the K-means algorithm for the above Step 1) is below:

i) Selecting K cluster centre points in all the data of demonstrations randomly;

- ii) Calculating the distance between each point of demonstrations \mathbf{p}_j and each cluster centre point, and allocating data to clusters according to the shortest distances to the centres of the clusters;
- iii) Re-calculating the centre points of clusters according to the updated results;
- iv) Calculating the above ii) and iii) iteratively until the centre point of each cluster is unchanged;
- v) Considering each cluster as the base for Gaussian clusters, calculating $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ for each Gaussian, and deciding ω_i proportionally according to the number of points contained in each cluster;
- iv) Calculating GMM based on Gaussian clusters.

The EM algorithm for the above Step 2) will carry on the E-step and M-step iteratively until convergence. The E-step and M-step are depicted below:

E-step: the probability of each point \mathbf{p}_j from each Gaussian should be estimated based on the entire points and then the problem is transformed to solve the posterior probability. According to the Bayes' theorem, the posterior probability (γ_{ji}) of each point \mathbf{p}_j for the i -th Gaussian can be written below:

$$\gamma_{ji} = \frac{\omega_i \cdot N_i(\mathbf{p}_j | \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)}{\sum_{i=1}^K \omega_i \cdot N_i(\mathbf{p}_j | \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)} \quad (11)$$

where $N_i(\mathbf{p}_j | \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ is the probability density function of the i -th Gaussian; ω_i is the weight of the i -th Gaussian; K is the total number of Gaussian clusters.

M-step: the value of log of the likelihood function of demonstrations ($LL = \ln(\prod_{j=1}^m P_{GMM}(\mathbf{x}_j))$) is maximised to re-estimate the parameters (ω_i , $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$) of each Gaussian by γ_{ji} . The partial derivatives of $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ need to be calculated through log of the likelihood function (LL), i.e., $\frac{\partial(LL)}{\partial(\boldsymbol{\mu}_i)}$ and $\frac{\partial(LL)}{\partial(\boldsymbol{\sigma}_i)}$, while it is necessary to construct Lagrange multipliers and then partial derivatives to solve ω_i , i.e., $\frac{\partial(Lag(LL))}{\partial(\omega_i)}$.

The relevant details are shown below:

$$\omega_i = \frac{1}{m} \sum_{j=1}^m \gamma_{ji} \quad (12)$$

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^m \gamma_{ji} \cdot \mathbf{p}_j}{\sum_{j=1}^m \gamma_{ji}} \quad (13)$$

$$\boldsymbol{\sigma}_i = \frac{\sum_{j=1}^m \gamma_{ji} \cdot (\mathbf{p}_j - \boldsymbol{\mu}_i) \cdot (\mathbf{p}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^m \gamma_{ji}} \quad (14)$$

where m is the total number of constructive points from demonstrations.

3.3 Gaussian cluster optimisation

For the above EM algorithm, the number of Gaussian clusters is pre-determined and fixed. However, it could cause the generated GMM/GMR under-/over-fitting. If this happens, GMM/GMR are either incapable of keeping the features of demonstrations well, or cause a reproduced path much zigged. To address the challenge, a novel optimisation algorithm for GMM/GMR is developed. The algorithm includes the following innovative strategies: 1) Gaussian noises are added into demonstrations to enrich

and scatter their construction points for a better support on GMM/GMR generation; 2) based on the enriched demonstrations, an optimisation process enabled by a SA-RL algorithms, which uses a novel optimisation criterion and embeds the K-means and EM algorithms, to optimise the number and the parameters of Gaussian clusters leading to optimal GMM/GMR.

Meanwhile, for GMR, it could be still non-adaptive to dynamic situations, such that in the pick-and-place example the starting point of a task could be changed significantly different from those of demonstrations. Therefore, a new adaptive process is designed with the following steps: 1) a reproduced path based on GMR is generated; 2) to enhance the adaptability and the accuracy of the reproduced path fitting into the new situation, the path is amended using a B-spline based cut-in algorithm.

The framework of the above processes is shown in Fig. 4. The technical details are elaborated in the following Section 4.

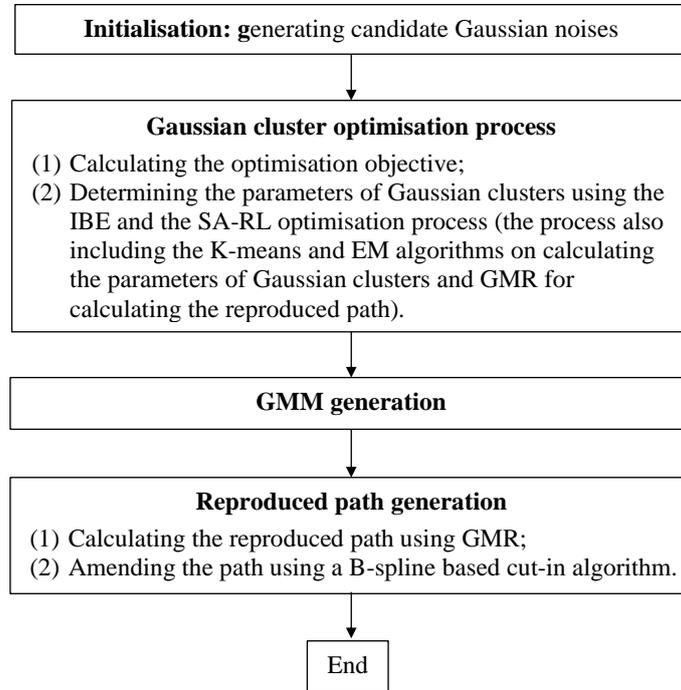


Fig. 4: The framework of the GMM/GMR optimisation processes.

4. GMM/GMR Optimisation

4.1 The Gaussian noise strategy

To better support the optimisation of Gaussian clusters and GMM/GMR, constructive points of demonstrations are enriched and scattered with Gaussian noises. The formulas are given below:

$$x'_j = x_j + r \quad (15)$$

$$y'_j = y_j + r \quad (16)$$

$$z'_j = z_j + r \quad (17)$$

$$r \sim N(\mu_{noise}, \sigma_{noise}) \quad (18)$$

$$N(\boldsymbol{\mu}_{noise}, \boldsymbol{\sigma}_{noise}) = \frac{1}{\sqrt{2\pi}\boldsymbol{\sigma}_{noise}} e^{-\frac{(r-\boldsymbol{\mu}_{noise})^2}{2\boldsymbol{\sigma}_{noise}^2}} \quad (19)$$

where x'_j, y'_j, z'_j are Gaussian noise-enhanced points of demonstrations; x_j, y_j, z_j are the original points of the demonstrations; $r \sim N(\boldsymbol{\mu}_{noise}, \Delta_{noise})$ means a random value r , which is considered as a noise, obeys the Gaussian distribution with a probability density function of $N(\boldsymbol{\mu}_{noise}, \boldsymbol{\sigma}_{noise})$; $\boldsymbol{\mu}_{noise}$ and $\boldsymbol{\sigma}_{noise}$ are the mean and variance of Gaussian noises respectively.

4.2 Optimisation criterion for Gaussian clusters

Bayesian Information Criterion (BIC) is an effective criterion to modelling a finite set of data (demonstrations in the context of this research). BIC can be used to evaluate the quality of a model from two aspects: 1) the fitting degree of the model, 2) the complexity of the model. BIC is defined below:

$$BIC = -2 \ln L(\hat{\theta}|B) + p \cdot \ln(m) \quad (20)$$

where $L(\cdot)$ is the likelihood function; $\ln L(\hat{\theta}|B)$ means the Logarithm of the likelihood function of the points in Gaussian clusters; p is the number of parameters in Gaussian clusters; m is the number of the constructive points of demonstrations.

The smallest of a BIC value will support the better GMM/GMR. Nevertheless, based on the BIC criterion, unsmooth features in geometry from demonstrations, such as tremble and abrupt changes caused by human operators, could be kept in GMM/GMR and the corresponding reproduced path could become less smooth (over-fitting). Thus, Gaussian noises are measures introduced to scatter demonstrations so that the above unsmooth features could be alleviated during Gaussian clusters and GMM/GMR optimisation. However, through numerical trials shown in Fig. 5, it is observed that the BIC value will become significantly larger with an increase of $\boldsymbol{\sigma}_{noise}$. Meanwhile, with an increased $\boldsymbol{\sigma}_{noise}$, the number of Gaussian clusters (K) becomes smaller, which could lead to under-fitting. Based on the observation, the BIC criterion is not an ideal design, and a new criterion should be developed based on the following considerations:

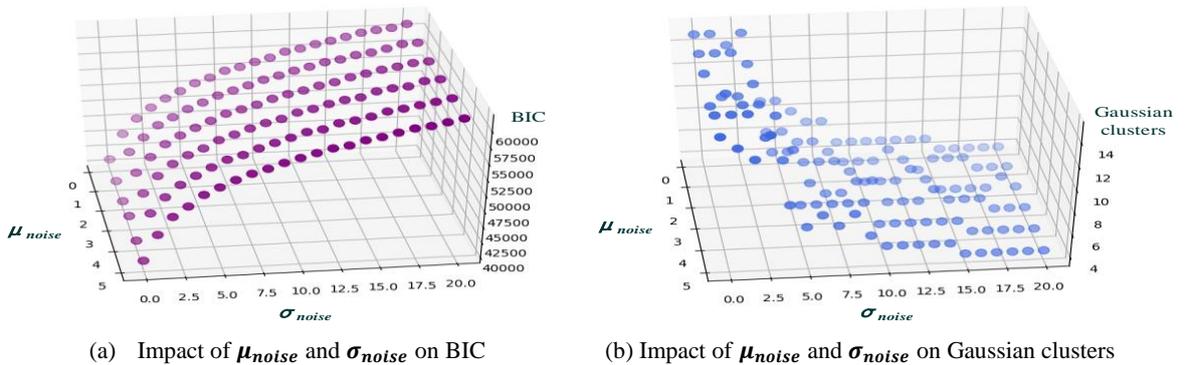


Fig. 5: Experimental results showing the impacts of $\boldsymbol{\mu}_{noise}, \boldsymbol{\sigma}_{noise}$ on K and BIC.

- It could achieve a good balance between under-fitting and over-fitting by adjusting the number of Gaussian clusters to an appropriate extend. Therefore, the Logarithm of the number of Gaussian clusters, i.e., $\ln K$ (K is the number of Gaussian clusters), can be designed as a penalty term to be

embedded into the criterion. That is, through introducing $\ln K \cdot BIC$, the numbers of Gaussian clusters corresponding to lower BIC values with little or no scatter can be excluded;

- It is also worth noticing that a regression path is not always the smoother the better. An excessively smooth regression path driven by $\ln K \cdot BIC$ could lose the desired features of demonstration paths. That is, it may generate a large deviation from leant demonstrations (under-fitting). Therefore, in the new criterion, it is necessary to integrate the assessment of similarity between demonstrations and their regression path in order to balance the under-/over-fitting aspects as well. Thus, a Euclidean Distance ($\|Eudis\|$) is introduced for the similarity assessment:

$$\|Eudis\| = \sum_{i=1}^n \sum_{j=1}^m \sqrt{(x'_j - x_{ij})^2 + (y'_j - y_{ij})^2 + (z'_j - z_{ij})^2} \quad (21)$$

where $Eudis_i$ is the distance between the regression path and all demonstrations; m is the total number of constructive points of all demonstrations; n is the total number of demonstrations; (x'_j, x_{ij}) , (y'_j, y_{ij}) and (z'_j, z_{ij}) are the corresponding constructive points in the reproduced path and the i -th demonstration;

- When the scattering level of Gaussian noises increases, the BIC value will change drastically (high variance) while the degree of change on $\|Eudis\|$ is lower (low variance). To incorporate the above elements and avoid excessively biased results, a scaling factor (sf) is added to scale the two elements. The new optimisation criterion (Integrated BIC- $\|Eudis\|$, in short IBE) can be written below:

$$IBE = \frac{\ln K \cdot BIC}{sf} + \|Eudis\| \quad (22)$$

$$sf = \frac{std(BIC)}{std(\|Eudis\|)} \quad (23)$$

where std is for the standardised deviation.

4.3 SA-RL optimisation algorithm

The optimised algorithm consists of the following three steps: 1) Input: the ranges of μ_{noise} , σ_{noise} and the number of Gaussian clusters are pre-specified; 2) Initialisation: a set of Gaussian clusters within the above ranges are generated; 3) Optimisation: the SA-RL optimisation process (including the K-means and EM algorithms for calculating BIC and $\|Eudis\|$) is conducted based on the IBE criterion to optimise the number of Gaussian clusters. In the algorithm, SA is designed for a global search, and RL is for fine-tuning a search area locally.

Compared with some metaheuristic optimisation algorithms that conduct computations on a large number of populations, such as PSO (Particle Swarm Optimisation), GA (Genetic Algorithm), ACO (Ant Colony Optimisation) and FOA (Fruit Fly Optimisation Algorithm), SA proved to be efficient in identifying an optimal or near-optimal solution [37]. Thus, SA is chosen here as a base to design this algorithm. RL, which is an increasingly popular technique to operate on complex problem spaces, is also able to identify an optimal or near-optimal solution [38]. In this research, RL is integrated to fine-tune the research within a small-scale scope near the minimum value obtained by SA under each

iterative temperature. By combining different strategies of SA and RL, the algorithm convergence could be accelerated and the success rate of obtaining the global minimum solution could be improved.

Fig. 6 is used to illustrate the process of the SA-RL optimisation. Some critical steps are below:

- The possible scopes of μ_{noise} , σ_{noise} and the corresponding Gaussian clusters are pre-specified according to some experiments. A set of μ_{noise} , σ_{noise} and the corresponding Gaussian clusters is called an index (a potential solution). They will be used to calculate BIC, $\|E_{udis}\|$ and IBE. In Fig. 6, the x axis represents indexes, and the y axis represents the IBEs of the indexes.
- A SA is executed within a search loop by lowering its working temperature continuously. Within the loop, under a temperature, a local search is executed by the SA for n times. The search is possible to fall into a local minimum point H after n rounds of computations. In Fig 6, the points in purple demonstrate the search process by the SA and an agent represents the current step of the SA (a solution, which includes an index and its IBE).
- Under the above temperature of the local search by the SA, a RL is then used to fine-tune the search by skipping from the local minimum H and identifying a smaller value nearby. For the RL, a local search range (indicated by brackets in Fig. 6) is specified around H . In the RL, a Q-table is used to define the best action for each state, and ϵ -greedy is used to define a probability to determine whether an agent will move to a next solution with a higher Q value (reward) or a solution with a smaller Q value (penalty). Based on the RL, the agent could reach a smaller value L closer to H .
- The above steps continue until the search loop of the SA is completed (reaching the last temperature).

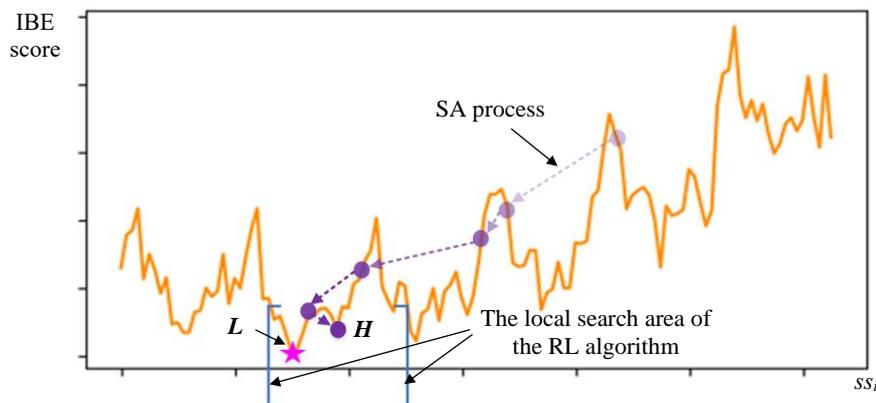


Fig. 6: Illustrations of the optimisation process of the SA-RL algorithm.

For the problem presented in this research, the local search process can be modelled as a Markov Decision Process (MDP). That is, the real-time position of a search agent is determined based on the current state. The future state only depends on the current state instead of historical states. It is similar to the Markov process so that the adoption of RL is appropriate. In comparison with swarm-based optimisation algorithms or the genetic algorithms, the Q-learning algorithm (a RL algorithm) uses a single agent for searching. Compared with other RL algorithms, the update method of the Q-learning algorithm is relatively simple and easy to implement. Meanwhile, the Q-learning algorithm is more efficient in local exploration as the search space is discrete and the amount of states is small. Therefore,

in this research, the Q-learning algorithm is devised to execute local search to fine-tune the global search conducted by the SA process.

In this research, the standard Q-learning algorithm is adopted and the reward function is set. According to the Q-learning algorithm, a reward is only obtained when the agent moves to the next state. To elaborate the Q-learning algorithm for this problem, Table 1 shows the definition of some key terms in the MDP, which includes the states, actions, transitions and rewards.

Table 1: Define the MDP in the RL process.

State — S	S represents a series of search parameters, which consist of μ_{noise} , σ_{noise} and Gaussian clusters. That is, $S = [SS_1:(\mu_{noise1}, \sigma_{noise1}$ and c_1 (Gaussian clusters)); ...; $SS_i:(\mu_{noise_i}, \sigma_{noise_i}$ and c_i); ...; $SS_n:(\mu_{noise_n}, \sigma_{noise_n}$ and $c_n)]$.
Action — A	A is a movement option of an agent. That is, $A = [a_1:$ “left”; $a_2:$ “right”; $a_3:$ “up”; $a_4:$ “down”].
Transition — T	T represents the probability of transition from the current state to the next state. In this research, to improve the computational efficiency, the probability is 100%.
Reward — R	R is assigned to a search parameter set in S . It is set as $\frac{1}{IBE_{SS_i}}$.

The local minimum H can be obtained when the SA process is completed. Subsequently, a local search process based on the Q-learning algorithm is carried out. In Fig. 6, the star point L means the last position of the search agent after the local search, and the bracket specifies the local search area of the Q-learning algorithm.

As shown in Fig. 6, the agent in the bracket (the local search area) only have two movement directions, i.e., “left” and “right”. To extend the exploration area of the agent and improve search efficiency, in this paper, the search space in the one-dimensional bracket is mapped into a two-dimensional checkerboard. For instance, as shown in Fig. 7, it is assumed that $S = [SS_1, SS_2, \dots, SS_{25}]$ in the bracket. The twenty-five sets are arranged into a 5*5 checkerboard.

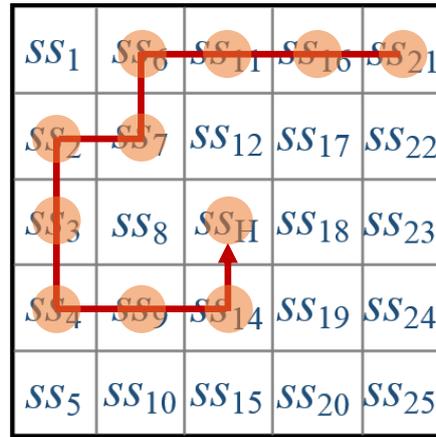


Fig. 7: The local exploration area represented by a checkerboard.

The search process of the Q-learning algorithm is also shown in Fig. 7. The red line represents the agent's movement path. According to Fig. 7, the agent ($ss_{current}$) is randomly initialised to start from ss_{21} ($ss_{current} = ss_{21}$). $Q(S, A)$ is a state-action value, which is the value of performing a specific A in the current agent. $Q(S, A)$ can update based on Equation (24). With the agent moves, the value of $Q(S, A)$ in the Q-table will be updated. The details are as follows:

- As aforementioned, the agent is randomly initialised as ss_{21} . The next A is selected according to ε -greedy (ε -greedy is a real number that is from zero to one);
- When the agent reaches ss_{16} , $R = \frac{1}{IBE_{ss_{16}}}$ is assigned to the agent;
- It is noted that when the agent reaches ss_{16} , it cannot move upward because of the boundary restriction as shown in Fig. 7. Thus, in ss_{16} , the agent only has three options of movement, that is, a_1 :“left”, a_2 :“right”, a_4 :“down”. The values of $Q(ss_{16}, a_1)$, $Q(ss_{16}, a_2)$, and $Q(ss_{16}, a_4)$ from the Q-table are calculated, and the maximum value is determined, i.e. $\max_{a_1, a_2, a_4} Q \begin{pmatrix} a_1 \\ ss_{16}, a_2 \\ a_4 \end{pmatrix}$;
- Set $\alpha \in (0,1)$ and $\gamma \in (0,1)$, where α is learning rate and γ is the attenuation value;
- $Q(ss_{21}, a_1)$ can be calculated as shown in Equations (24) and (25), and Equation (25) represents the agent position from ss_{21} to ss_{16} :

$$Q(ss_{21}, a_1) \leftarrow Q(ss_{21}, a_1) + \alpha \left[\frac{1}{IBE_{ss_{16}}} + \gamma \max_{a_1, a_2, a_4} Q \begin{pmatrix} a_1 \\ ss_{16}, a_2 \\ a_4 \end{pmatrix} - Q(ss_{21}, a_1) \right] \quad (24)$$

$$\text{The agent : } ss_{current} \leftarrow ss_{16} \quad (25)$$

- When s_H or the limit of iteration is reached, the above exploration process will stop.

The above steps of the SA-RL algorithm continue until the search loop of the SA process is completed (reaching the last temperature). Table 2 shows the pseudo-code for the SA-RL algorithm.

Table 2: A pseudo-code for the SA-RL algorithm.

Inputs:

Identify the feasible search ranges of μ_{noise} , σ_{noise} and the number of Gaussian clusters
 Calculate each index (μ_{noise} , σ_{noise} and the corresponding number of Gaussian clusters within their ranges)
 Calculate BIC, $||Eudis||$ and then sf for each index
 Store the above candidate solutions as a search space S
 Set a global minimum IBE as IBE_{best}

Initialisation:

Set SA contents: set the start temperature = T_{start} , end temperature = T_{end} , current temperature = T
 Set RL contents: set ε -greedy $\in (0,1)$, Q-table($Q(S, A)$), α , γ , S , A , T , R , and the limit step number = lsn

Optimisation:

<<SA process>>

Choose a solution randomly in S as a starting point; set it as the agent (its IBE is labelled IBE_{agent}) and

$$IBE_{best} \leftarrow IBE_{agent}$$

While $T \geq T_{end}$

 Loop under the current T for n times

 Choose a random neighbourhood of the agent in S as a new solution) and label its IBE as IBE_{new}

 If $IBE_{new} < IBE_{agent}$

$$IBE_{best} \leftarrow IBE_{agent}$$

```

Replace the agent using the new solution
Otherwise: If  $r < p$  ( $p = e^{-\frac{IBE_{new} - IBE_{agent}}{T}}$  and  $r \in [0,1]$ )
    Replace the agent using the new solution
End If
End Loop
<<RL process>>
Specify a local search range as exploration environment around the solution of  $IBE_{best}$ 
Adjusts the format of search area
Repeat:
    If  $x > \epsilon$ -greedy ( $x$  is randomly generated and  $x \in [0,1]$ ) or all  $Q(S, A) = 0$ 
        Choose a specific action  $a$  randomly from  $A$ , observe  $R$  and next  $S$  called  $s'$ 
        Calculate  $Q(s, a) = Q(s, a) + \alpha [R + \gamma \max_A Q(s', A) - Q(s, a)]$ 
         $s \leftarrow s'$ 
    Else
        Choose the action  $a$  according to  $\max Q(S, A)$ , observe  $R$  and next  $S$  called  $s'$ 
        Calculate  $Q(s, a) = Q(s, a) + \alpha [R + \gamma \max_A Q(s', A) - Q(s, a)]$ 
         $s \leftarrow s'$ 
    END if
Until  $l_{sn}$  or  $s_H$ 
 $IBE_{best} \leftarrow$  extract the minimum  $IBE_{agent}$  in RL process
Reduce  $T$  by  $T = \frac{T_{start}}{1 + current\ iteration}$ 
End While
Output:
The agent is the best solution

```

4.4 Adaptive reproduced path with GMR

To better address a dynamic situation, a B-spline based cut-in algorithm is integrated with GMR to be adaptive to changes. The algorithm can address the dynamic setting while keeping the characteristic of the regression path. The process can be explained using the example shown in Fig. 8. In Fig. 8(a) the curve R is the original regression path, 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 the new start point in a dynamic situation, and O_e is still the end point in the situation. Fig. 8(b) shows the reproduced path represented in the curve NR .

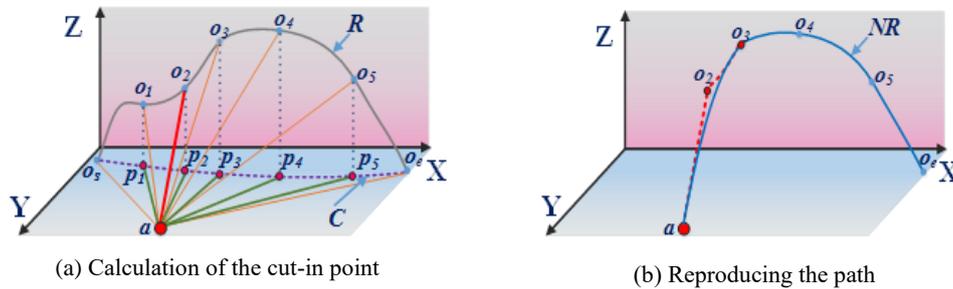


Fig. 8: B-spline based cut-in algorithm for path reproducing in a dynamic situation.

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 Fig. 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 with 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 reproduced path are removed.

2. A new path, starting from Point a , should be integrated into the trimmed reproduced 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 the reproduced path are knot points for generating the B-spline curve. The formula is represented below.

$$P(u) = \sum_{i=0}^n P_i \cdot B_{i,k}(u) \quad (26)$$

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

5. Case Studies and Discussions

An experiment platform shown in Fig. 9 was established to support the approach presented in this paper. The platform mainly includes two UR5 cobots (the right one was employed in the experiment), a two-finger adaptive gripper, a 3D camera, and a computer system.

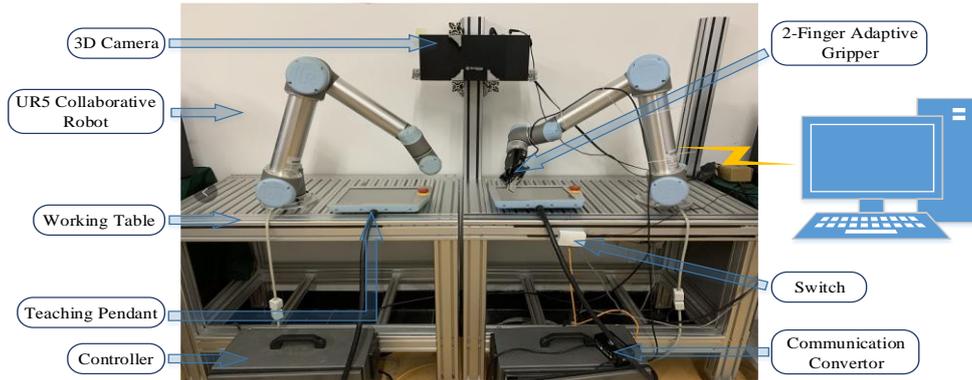
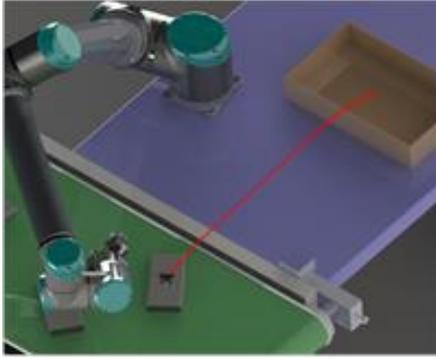


Fig. 9: The experimental platform.

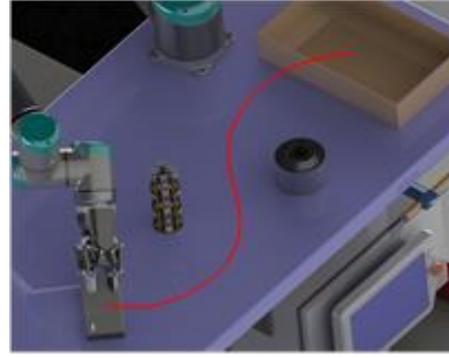
Two pick-and-place cases were used to validate the approach. They are:

- Case study 1: there are some objects (with slightly deviated positions) transmitted on a production conveyor, and a cobot with a gripper picks them up one-by-one and place them into a packaging box along a path;
- Case study 2: there are several obstacles along the paths of the pick-and-place operations, and the cobot should avoid collisions with the obstacles (e.g., humans) during the pick-and-place operations.

Fig. 10 shows the two case studies. Trajectories in red represent the paths required by the two tasks. It is obvious that the first task is barrier free along the path, while the second task needs to plan paths avoiding several obstacles so that the paths are in S-shape. Experimental results of the two cases are explained below.



(a) A pick-and-place task

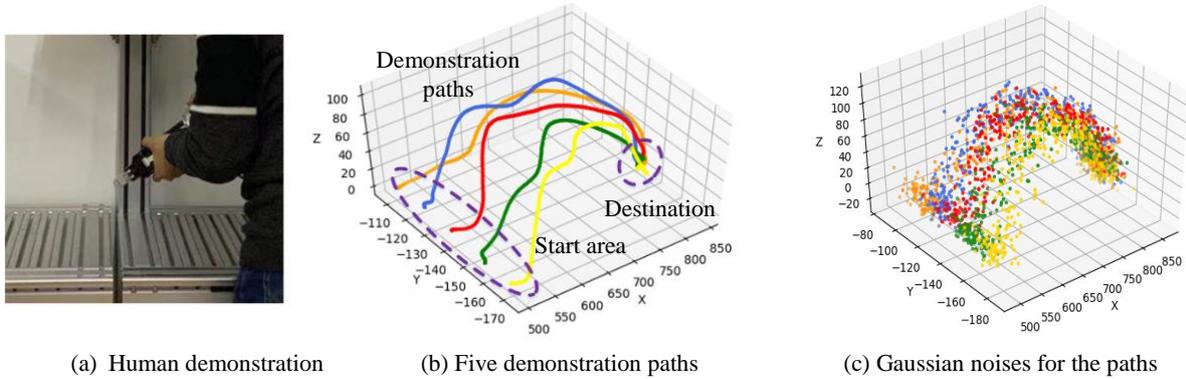


(b) A pick-and-place task with obstacle avoidance

Fig. 10: Two tasks of pick-and-place for approach validation.

5.1 Case study 1

Five demonstrations paths for the first pick-and-place task were generated (illustrated in Fig. 11).



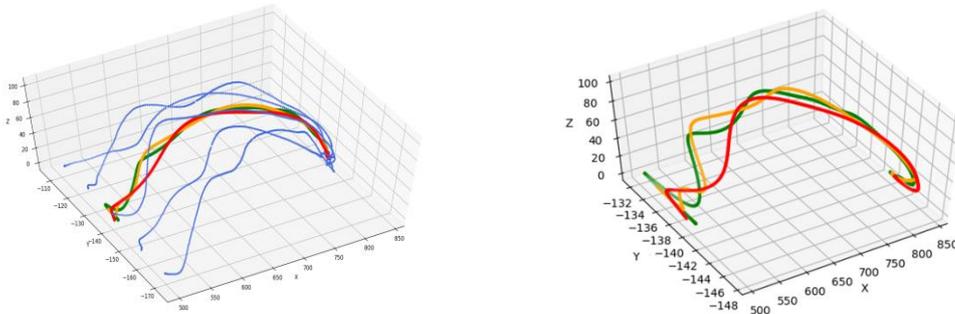
(a) Human demonstration

(b) Five demonstration paths

(c) Gaussian noises for the paths

Fig. 11: Demonstration paths and Gaussian noises for the pick-and-place task.

For the case study, the search ranges of μ_{noise} , σ_{noise} and the number of Gaussian clusters were pre-specified. After using the approach based on the BIC, $\|Eudis\|$ or IBE, the minimum number of Gaussian clusters identified as (15, 7, 5). It shows IBE can obtain the minimum number of Gaussian clusters, which means the calculation for Gaussian clusters and GMM/GMR will be more efficient in comparison with the calculation using BIC or $\|Eudis\|$. Three reproduced paths based on the demonstrations generated using BIC, $\|Eudis\|$ or IBE are shown in Fig 12. It can be observed that the reproduced paths optimised by using the three criteria all retain the characteristic of the demonstrations.



(a) Demonstrations (in blue) and reproduced paths

(b) Reproduced paths generated using different criteria

Fig. 12: Demonstration paths and reproduced paths.

According to Fig. 13, GMM/GMR obtained by BIC and $\|Eudis\|$ were unable to eliminate trembling features of human demonstrations when calculating reproduced paths. It can clearly indicate that the path generated using BIC contains some unsmooth features leading to redundant twists and sharp turns in the paths, which will affect the movement of a cobot. For instance, a sharp turn will cause a sudden change of the speed and acceleration of the cobot generating vibration. The reproduced path obtained by $lnK \cdot BIC$ prefers a smaller number of Gaussian clusters while some features of the demonstrations are lost (thereby smoother). As analysed earlier, IBE is designed to balance the under-/over-fitting so that the reproduced path can preserve the characteristics of demonstration while keeping smooth. It means IBE outperforms other criteria in GMM/GMR optimisation.

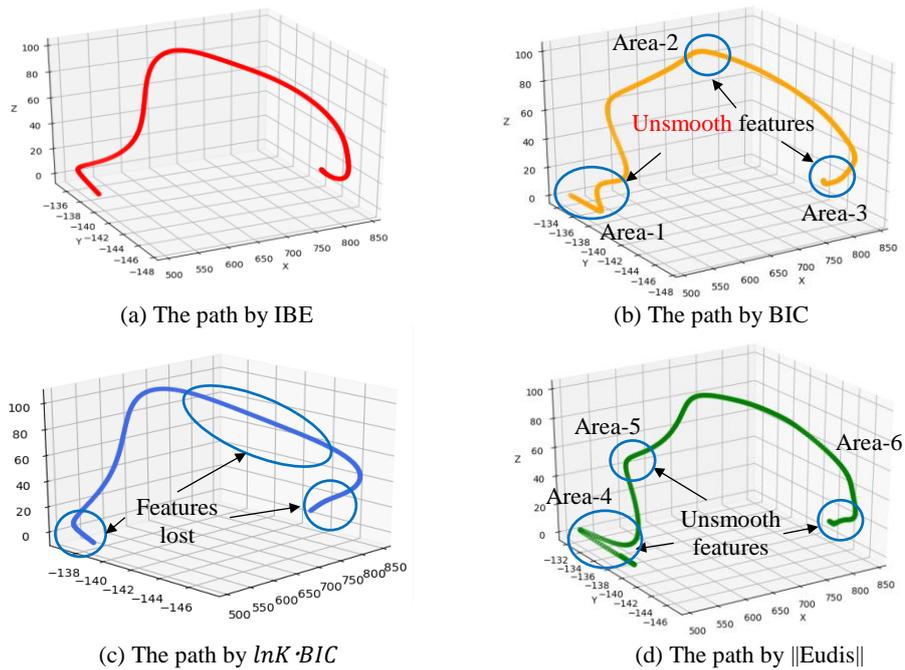


Fig. 13: Reproduced paths obtained by IBE, BIC, $lnK \cdot BIC$ and $\|Eudis\|$.

To qualitatively evaluate the results, curvatures are introduced. Fig. 14 illustrates the curvatures of some selected areas of the paths generated based on the criterion of IBE, BIC and $\|Eudis\|$ in Fig. 13.

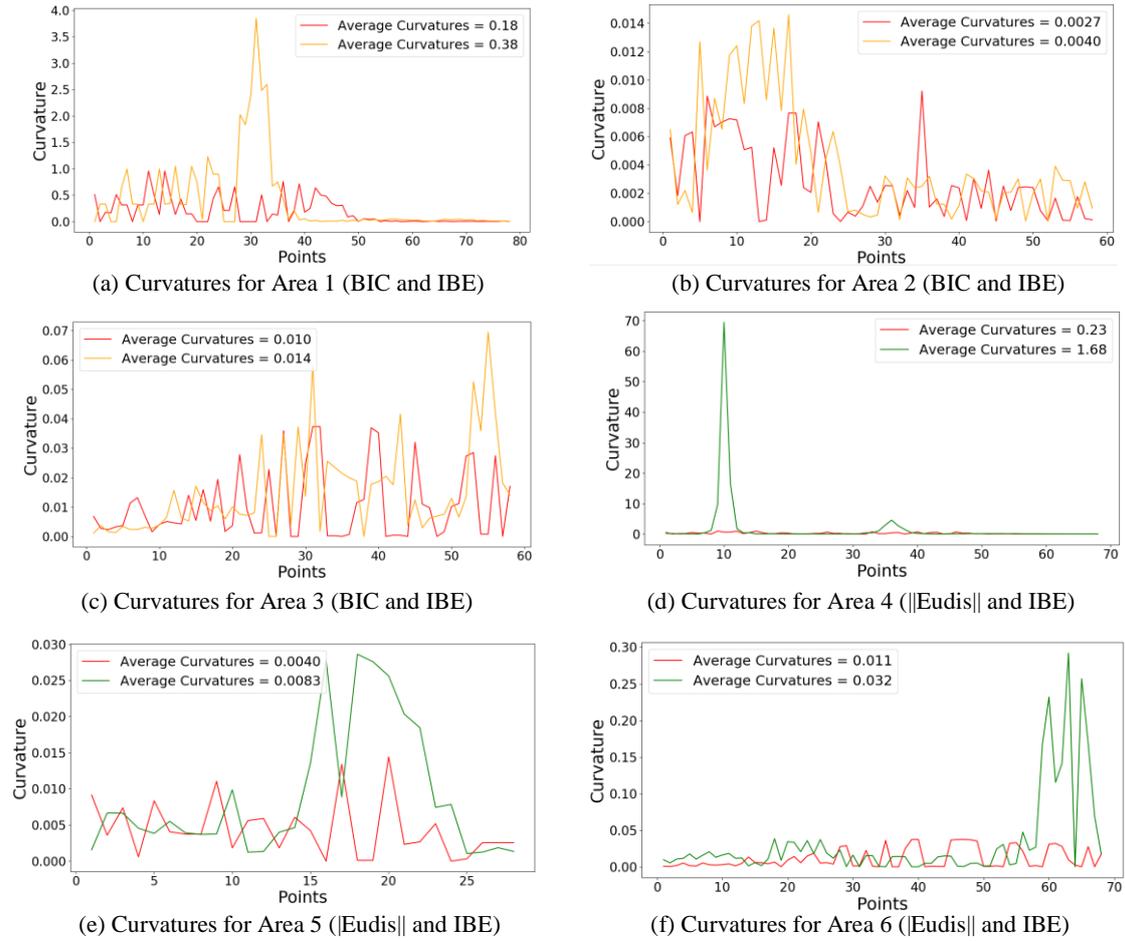


Fig. 14: Curvatures (curves projected on the x-y plane) of the examples in Fig. 13.

In the pick-and-place task, the reproduced path should be adaptive with a significant change of a start position in a dynamic manufacturing situation. The adaptive path calculated through the B-spline based cut-in algorithm is shown in Fig. 15. The start point is a . The purple curve passing through a and o_3 shows the adaptive cut-in path.

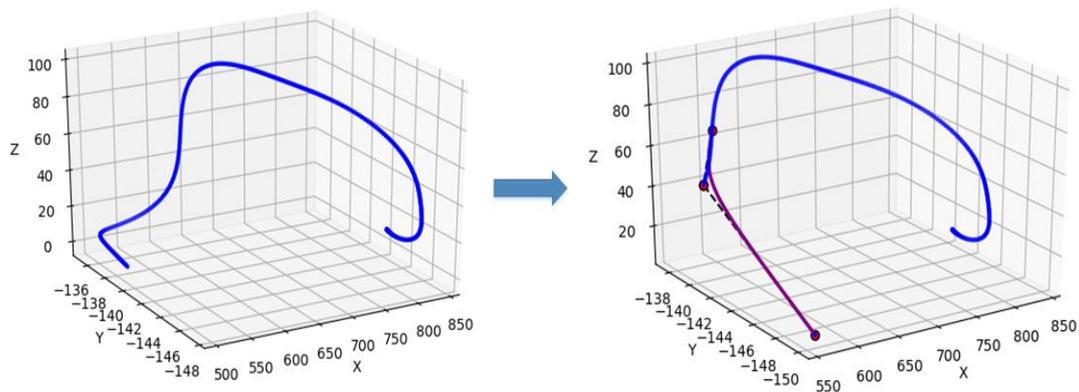


Fig. 15: A reproduced path adaptive to the start point using the B-spline based cut-in algorithm.

5.2 Case study 2

In Fig 16, three reproduced paths and demonstration paths for the second case study are shown. S-shape reproduced paths based demonstrated paths were established for task accomplishment and obstacle avoidance. The best results of S-shape demonstration paths using BIC, $\|Eudis\|$ and IBE in terms of the number of Gaussian clusters are (15, 13, 7). The same conclusion can be drawn as the case study 1. That is, it clearly shows that the number of Gaussian clusters obtained for the IBE criterion is the smallest and thereby more efficient in computing compared with BIC and $\|Eudis\|$.

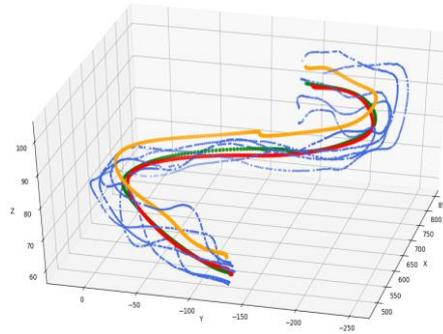


Fig. 16: Demonstrations paths and reproduced paths.

Fig. 17 shows reproduced paths optimised by using the IBE, BIC, $\|Eudis\|$ and $\ln K \cdot BIC$ respectively. It can be observed that the reproduced paths optimised by using the four criteria all retain the characteristic ‘S’ shape. Similarly as the first case study, it also clearly indicates that the paths by using BIC and $\|Eudis\|$ have some unsmooth features such as redundant twists and sharp turns in the paths (over-fitting) while the path obtained by $\ln K \cdot BIC$ lost a feature (under-fitting). The criterion of IBE balances the aspects and demonstrates a better result.

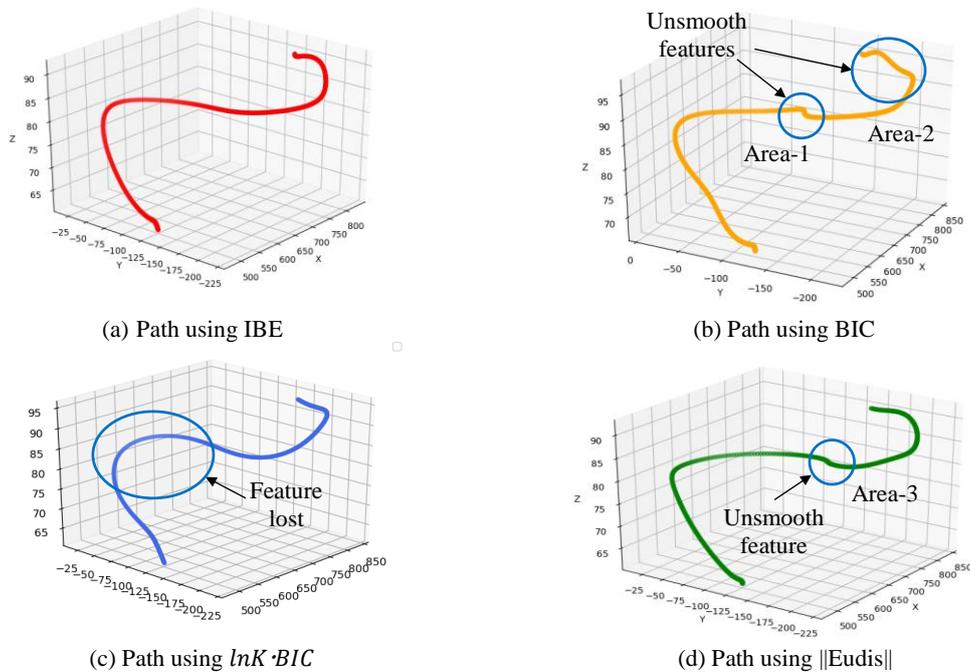


Fig. 17: Reproduced paths obtained by using different criteria.

Fig. 18 illustrates the curvatures for the selected areas of the paths generated based on the criterion of IBE, BIC and $\|Eudis\|$ in Fig. 17.

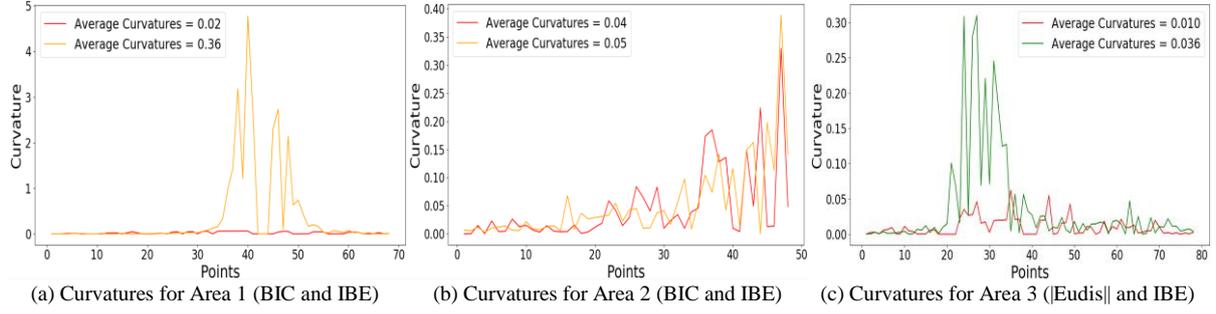


Fig. 18: Curvatures (curves projected on the x-z plane) of the examples in Fig. 17.

The SA-RL optimisation algorithm with IBE can effectively optimise the number and parameters of Gaussian clusters. In the actual task execution, the algorithm could obtain the global minimum solution within 10 steps. To show the effectiveness of the SA-RL optimisation, it was compared with a SA and a SA enhanced with a mutation operation. The probabilities of obtaining a global minimum solution for the algorithms are shown in Table 3. It clearly shows the advantage of the SA-RL optimisation algorithm in identifying a global minimum solution in terms of high success rate.

Table 3: The success rates of obtaining a global minimum solution using different algorithms.

Task	Algorithm	Success rate of obtaining minimum solutions
Case study 1	SA	65%
	SA-RL	93%
Case study 2	SA	68%
	SA-RL	90%

GMM is calculated by the K-means and EM algorithms. The quality of the reproduced path can directly reflect the algorithm quality of GMM, K-means and EM. Therefore, the evaluation of GMM can be based on the variance analysis of demonstration paths (as a training set) and the reproduced path (as a testing set). The details are shown in Table 4 to Table 7 (the research performed variance analysis to each point on the two case studies, i.e., five demonstration paths and S-shaped demonstration paths).

Table 4: Variances results with no Gaussian noises for Case study 1.

Paths	Element/criterion	x	y	z
Demonstration path	Average	16837.04	90.58	1309.93
	Maximum	18247.65	247.07	1347.57
Reproduced path and demonstration paths	Only BIC	6.19	5.30	6.62

Table 5: Variance results with Gaussian noises for Case study 1.

Paths	Element/criterion	x	y	z
Demonstration path	Average	16881.85	175.66	1378.54

	Maximum	18279.96	344.77	1439.43
Reproduced path and demonstration path	IBE	5.89	4.11	5.66

Table 6: Variance results with no Gaussian noises for Case study 2.

Paths	Element/criterion	x	y	z
Demonstration path	Average	14803.39	5065.06	116.55
	Maximum	17091.98	6002.62	165.5
Reproduced path and demonstration path	Only BIC	41.69	135.74	25.81

Table 7: Variance results with Gaussian noises for Case study 2.

Paths	Element/criterion	x	y	z
Demonstration path	Average	14836.19	5126.42	172.71
	Maximum	17177.39	6051.14	222.68
Reproduced path and demonstration path	IBE	11.61	131.14	1.45

In Tables 4-7, “Average” refers to the mean of the variance of each type of demonstration path (five demonstration paths /S-shaped demonstration paths), “maximum” is the extreme variance in each dimension (x, y or z) of the demonstration paths. It shows that the variance of the scattered data with Gaussian noises is slightly larger than that of the original demonstrations. The reason is that the data distributions of the original demonstrations are changed during the scattering process, which inevitably generates differences between the data thereby increasing the variance. At the same time, the variance (to each point) between the demonstration paths and its reproduced path is calculated based on BIC (no Gaussian noises) and IBE (Gaussian noises added). The variance for the reproduced path calculated by IBE is slightly smaller than that calculated by BIC. Compared with the appropriate state calculated by IBE, BIC contains some uncontrollable redundant turns, which further increases the difference between the reproduced path and the demonstration paths. Therefore, the overall variance calculated by IBE is smaller than that calculated by BIC.

5.3 Discussions

According to the above results, the appropriate parameters for Gaussian noise can eliminate errors from human demonstrations, e.g., smoothing reproduced paths. Randomly generated parameters of Gaussian noise may not obtain the best GMM. Therefore, BIC and $\|Eudis\|$ based IBE and the SA-RL optimisation process are designed to identify the best parameters. Among the processes, there are three topics that need further discussion.

- (1) The recommendation of the parameters of Gaussian noises (μ_{noise} , σ_{noise})

μ_{noise} is the mean, which controls the translation of the point cloud of Gaussian noises. σ_{noise} is the variance, which controls the degree of deviation of the point cloud around the mean. In other words, σ_{noise} decides the scattering degree of Gaussian noises added into demonstrations. A larger σ_{noise} makes the area of the point cloud bigger, and vice versa. While μ_{noise} is not for scattering the data, σ_{noise} plays a major role in forming the point cloud of Gaussian noises. This result shows that μ_{noise} has a smaller effect on the smoothness and correctness of reproduced paths than σ_{noise} . At the same time, a larger μ_{noise} will cause the overall displacement of demonstration paths, which are not preferred. It is worth noting that Gaussian noise is added into three dimensions respectively, and then integrate them into Gaussian noise points. Therefore, Gaussian noise can be independently controlled in task-relevant dimensions. That is, Gaussian noise can be added for a certain dimension or the entire data. At the same time, if there is a requirement for the variance of the data, the scope of σ_{noise} should be further discussed. Therefore, it suggests that when initialising the search range, the range of μ_{noise} should be limited to a small scope while the range of σ_{noise} can be designed to be relatively large.

In this research, Gaussian noises are added in the x, y, and z dimensions for generic conditions. Nevertheless, adding Gaussian noises could increase potential collision risks between an updated regression path and obstacles. For instance, as shown in Fig. 19, for a narrow passage problem, it would generate collisions if Gaussian noise is added along the axis (Y axis in this example) that is orthogonal to the passage direction.

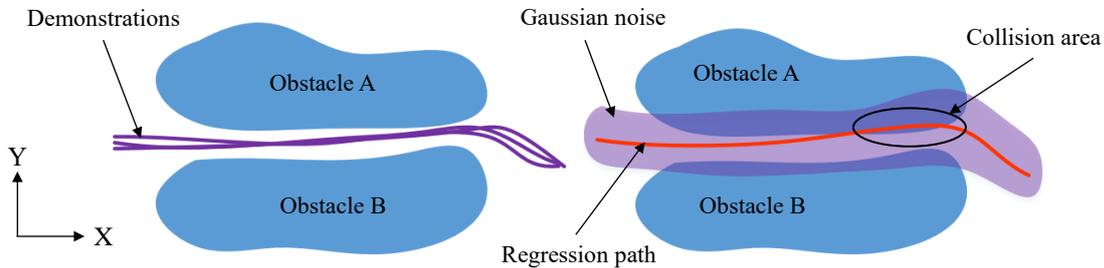


Fig. 19: A narrow passage problem where Gaussian noise-enhanced path could generate collisions.

In this research, for the strategy of adding Gaussian noise, collision avoidance from obstacles is not fully considered yet. Thus, the limitation of the approach is that it could be incompetent when addressing some complex practical situations such as the above narrow passage problem and other complex situations. A further work, which is undertaken by the authors and to be reported in near future, is that adding Gaussian noise is modelled as a constraint-based optimisation problem. In the model, the constraint is that the regression path generated from demonstrations with Gaussian noise should avoid obstacles. That is,

$$\text{Minimise: } IBE = \frac{\ln K \cdot BIC}{sf} + ||Eudis||$$

$$\text{Subject to: Distance (path, obstacles)} \geq \theta$$

where Distance (path, obstacles) refers to the shortest distance between the regression path and obstacles, and θ is a pre-set threshold that is the minimum value to allow the end-effect of a cobot to move along the regression path without collisions with the obstacles.

In the future work, the SA-RL optimisation algorithm presented in this research will be further enhanced by considering the constraint for obstacle avoidance during adding Gaussian noise.

(2) The sf term of IBE

sf is the scaling factor, and its main purpose is to scale the BIC value in the IBE criterion. Experiment results showed that the variance of BIC is larger than that of $\|Eudis\|$. Meanwhile, after adding the penalty term, the result of IBE is severely biased. It is important to choose a proper sf for good performance of the algorithm. Experimental results showed that the value of sf has good robustness within a certain range, the optimal value range of sf was determined to be around the standard deviation (BIC) divided by the standard deviation of $\|Eudis\|$, that is, $sf = \frac{Std(BIC)}{Std(Eudis)}$. The results for the two case studies are shown in Fig. 20 and Fig. 21, respectively.

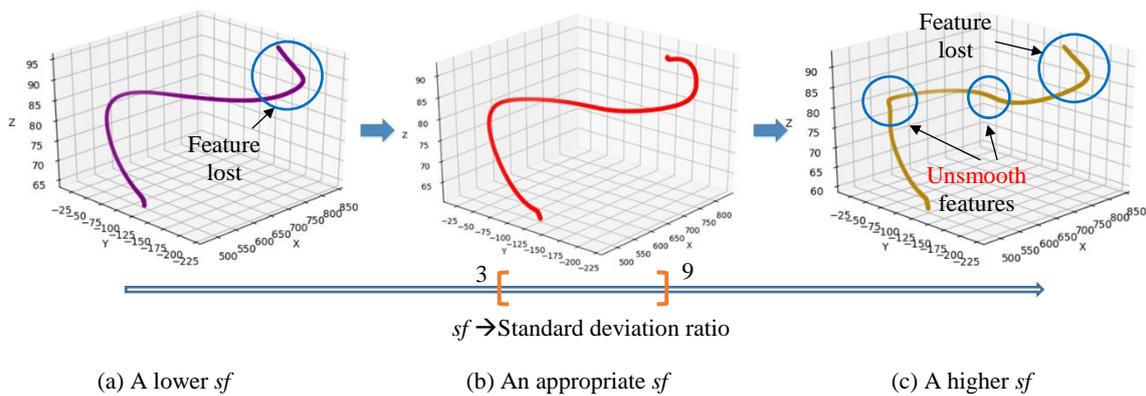


Fig. 20: sf for Case study 1.

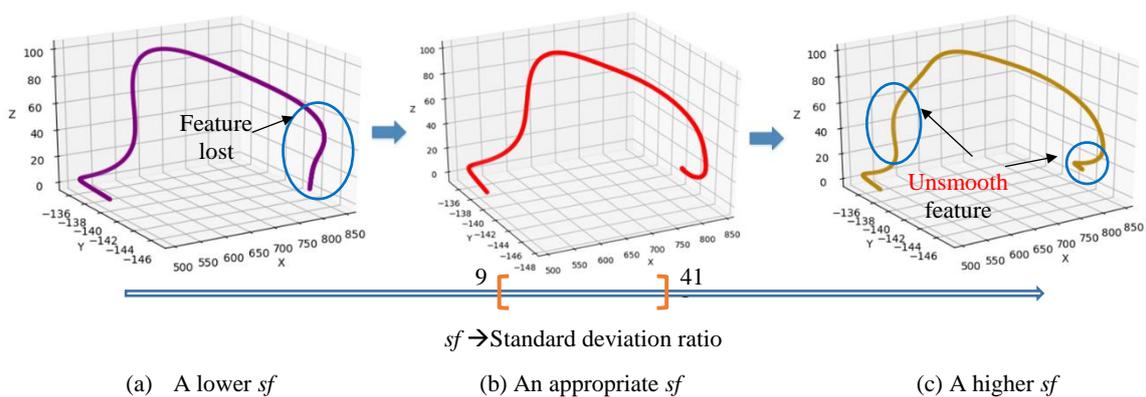


Fig. 21: sf for Case study 2.

(3) Comparison of optimisation criteria

On one hand, if only BIC is considered as the optimisation criterion, the result will be biased towards zero variance. Since demonstration paths are in low entropy without Gaussian noises, it has a lower

BIC value and the number of Gaussian clusters would be greater. This can result in more zigs in reproduced path. Such a path is not satisfactory. On the other hand, if $\|Eudis\|$ is used for consideration only, it will choose a reproduced path that is the most similar to demonstration paths, i.e., a path with the minimum $\|Eudis\|$ value.

Smoothness is an important factor to ensure the quality of a reproduced path. There could be some trembles in demonstration paths by human operators, leading to an unsmooth reproduced path. Thus, only using Eudis is not sufficient to ensure the smoothness of a reproduced path. Thus, $\|Eudis\|$ is not conducive to eliminating human errors in demonstrations as well. Thus, Gaussian noises are designed addressing not only the over-/under-fitting issue but also the potential un-smoothness issue by adding “deviating” constructive points to repair trembles. IBE, which incorporates BIC, Gaussian noises and a scaling factor, is designed as an optimisation criterion to support the SA-RL optimisation algorithm to select the most appropriate reproduced path. That is, by using IBE, the above aspects are comprehensively assessed and balanced so that it can conclude that IBE is the best choice. In the case studies, curvatures are calculated to evaluate the smoothness and compare the results obtained using BIC, Eudis and IBE. Experimental results prove that IBE outperformed other criteria.

Meanwhile the smoothness of an adaptive reproduced path is further ensured owing to the introduction of NURBS and its C_0 - C_2 continuity.

(4) Analysis of the appropriate number of demonstrations

In this research, the appropriate number of demonstrations was trialled using the case studies. It was concluded that too many demonstration paths are not recommended. There are three reasons: 1) in the same demonstration area, if there are many demonstration paths, i.e., massive data, it will consume more computing resources; 2) the role of Gaussian noises is to eliminate demonstration errors and expand the demonstration area, so that excessive demonstration paths are not necessary; 3) setting up a large number of demonstrations is also time-consuming. Therefore, for manufacturing applications in relative complexities, 4-7 demonstration paths should be sufficient and therefore recommended.

In the research, $\|Eudis\|$ is employed to measure the similarity between a demonstrated path and a reproduced path. This requires the reproduced path to have the same number of time-step points in corresponding to those of the demonstration path one-by-one. In applications, GMR is adjustable to adding or removing some intermediate points to meet the requirement of each specific task.

6. Conclusions

GMM/GMR based LfD approaches are useful technologies to support humans in operating cobots intuitively. However, the effectiveness of the approaches could be crippled by potential issues such as under-/over-fitting in reproduced solutions. To tackle the issues, this paper presents a novel optimised approach to improve GMM/GMR so that LfD enabled cobots can carry out a variety of complex manufacturing tasks in a robust and adaptive means. In the research, innovative strategies are developed, analysed and validated. Firstly, a Gaussian noise strategy is introduced to enhance demonstrations by

scattering data clouds. Based on it, a SA-RL based optimisation algorithm with an effective optimisation criterion (i.e., IBE) is then developed to eliminate under-/over-fitting issues in GMM/GMR. That is, GMM/GMR can achieve refinement in terms of important feature preservation and smoothness in reproduced paths. Finally, a B-spline based cut-in algorithm is integrated with GMR to improve the adaptability of reproduced paths for dynamic manufacturing tasks. Case studies were conducted and experimental analyses highlighted that this approach was boosted by the strategies and algorithms. Computational efficiency, solution quality and adaptability were significantly improved, and the potential applicability of the approach to practical manufacturing scenarios were clearly demonstrated.

In near future work, research will be conducted in the following aspects:

- For GMM/GMR, it is important to develop a sensible strategy to handle complex obstacles in practical situations, such as a movable human operator during operations;
- Different distance criteria such as the Mahalanobis distance will be carefully evaluated and compared in future research;
- Our future research will also consider the constraints of orientation, singularity and accessibility as well as robotic kinematics and dynamics in the optimisation;
- It is essential to consider validating the research in more complex industrial applications, including assembly and disassembly for complex products.

Acknowledgement:

This research was partially sponsored by the National Natural Science Foundation of China (Project No. 51975444), and partially funded by the UK industrial and research partners (the Unipart Powertrain Application Ltd. (UK) and the Institute of Digital Engineering (UK)). The authors would thank reviewers sincerely for their critiques and valuable suggestions to help us improve the manuscript in the rigour, clarity and scientific aspects.

References:

- [1] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, J. Pérez-Oria, Working together: a review on safe human-robot collaboration in industrial environments, *IEEE Access*. 99 (2017) 1-1.
- [2] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, O. Madsen, Robot skills for manufacturing: From concept to industrial deployment, *Robot. Comput. Integr. Manuf.* 37 (2016) 282–291.
- [3] Z. Gao, T. Wanyama, I. Singh, A. Gadhri, R. Schmidt, From Industry 4.0 to Robotics 4.0 - A conceptual framework for collaborative and intelligent robotic systems, *Procedia Manufacturing*. 46 (2020) 591-599.
- [4] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, P. Fraisse, Collaborative manufacturing with physical human–robot interaction, *Robot. Comput. Integr. Manuf.* 40 (2016) 1-13.

- [5] A. Mohammed, B. Schmidt, L. H. Wang, Active collision avoidance for human–robot collaboration driven by vision sensors, *Int. J. Comput. Integr. Manuf.* 30(9) (2016) 970–980.
- [6] B. Chandrasekaran and J. M. Conrad, Human-robot collaboration: A survey, *Proceedings of the SoutheastCon IEEE 2015.* (2015).
- [7] S. Blankemeyer, R. Wiemann, L. Posniak, C. Pregizer, A. Raatz, Intuitive robot programming using augmented reality, *Procedia CIRP.* 76 (2018) 155-160.
- [8] 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.
- [9] J. Lee. A survey of robot learning from demonstrations for human-robot collaboration, *arXiv:1710.08789.* (2017).
- [10] P. Tsarouchi, S. Makris, G. Chryssolouris, Human–robot interaction review and challenges on task planning and programming, *Int. J. Comput. Integr. Manuf.* 29(8) (2016) 916-931.
- [11] S. Calinon, A tutorial on task-parameterized movement learning and retrieval, *Intell. Serv. Robot.* 9(1) (2016) 1-29.
- [12] J. Qu, F. Zhang, Y. Wang, Y. Fu, Human-like coordination motion learning for a redundant dual-arm robot, *Robot. Comput. Integr. Manuf.* 57 (2019) 379-390.
- [13] H. Lin, Design of an intelligent robotic precise assembly system for rapid teaching and admittance control, *Robot. Comput. Integr. Manuf.* 64 (2020) 101946.
- [14] S. Calinon, Learning from demonstration (programming by demonstration). *Encyclopaedia of Robotics* (Springer). (2018).
- [15] B. D. Argall, S. Chernova, M. Veloso, B. Browningm, A survey of robot learning from demonstration, *Robot. Auton. Syst.* 57(5) (2009) 469-483.
- [16] S. E. Zaatari, M. Marei, W. D. Li, Z. Usman, Cobot programming for collaborative industrial tasks: an overview, *Robot. Auton. Syst.* 116 (2019) 162-180.
- [17] 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.
- [18] M. Kyrarini, M. A. Haseeb, D. Ristić-Durrant, A. Gräser, Robot learning of industrial assembly task via human demonstrations, *Auton. Robot.* 43 (2019) 239–257.
- [19] U. E. Ogenyi, G. Zhang, C. Yang, Z. Ju, H. Liu, An intuitive robot learning from human demonstration, *Proceedings of the 11th International Conference Intelligent Robotics and Applications (ICIRA 2018).* (2018) 10984.
- [20] M. Najafi, M. Sharifi, K. Adams, M. Tavakoli, Robotic assistance for children with cerebral palsy based on learning from tele-cooperative demonstration, *Int. J. Intell. Robot Appl.* 1 (2017) 43-54.
- [21] B. Ti, Y. Gao, Q. Li, J. Zhao, Dynamic movement primitives for movement generation using GMM-GMR analytical method, *Proceedings of the IEEE 2nd International Conference on Information and Computer Technologies (ICICT).* (2019).

- [22] L. Rozo, P. Jiménez, C. Torras, A robot learning from demonstration of force-based manipulation tasks, *Intell. Serv. Robot.* 6 (2013) 33-51.
- [23] E. Pignat, S. Calinon, Learning adaptive dressing assistance from human demonstration, *Robot. Auton. Syst.* 93 (2017) 61-75.
- [24] H. Zhang, Y. Leng, Motor skills learning and generalization with adapted curvilinear Gaussian mixture model, *J. Intell. Robot. Syst.* 96 (2019) 457-475.
- [25] A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Statistical Society Ser. B.* 39(1) (1977) 1-38.
- [26] T. Lai, S. Xiao, H. Aoyama, and C. Wong, Path Planning and Obstacle Avoidance Approaches for Robot Arm, *Proceedings of the SICE Annual Conference.* (2017).
- [27] B. Luigi, M. Lorenzo, M. Claudio, Improving the accuracy of industrial robots via iterative reference trajectory modification, *IEEE Trans. Control Syst. Technol.* 28(3) (2020) 831-843.
- [28] F. Xie, L. Chen, Z. Li, K. Tang, Path smoothing and feed rate planning for robotic curved layer additive manufacturing, *Robot. Comput. Integr. Manuf.* 65 (2020) 101967.
- [29] G. E. Schwarz, Estimating the dimension of a model. *The Annals of Statistics*, 6(2) (1978).
- [30] A. Hewitt, C. Yang, Y. Li, R. Cui, DMP and GMR based teaching by demonstration for a KUKA LBR robot, *Proceedings of the 2017 IEEE International Conference on Automation & Computing.* (2017).
- [31] L. Rozo, S. Calinon, D. G. Caldwell, P. Jiménez, C. Torras, Learning physical collaborative robot behaviors from human demonstrations, *IEEE Trans. Robot.* 32(3) (2016) 513-527.
- [32] C. Yang, C. Chen, W. He, R. Cui, Z. Li, Robot learning system based on adaptive neural control and dynamic movement primitives, *IEEE Trans. Neural Netw. Learn. Syst.* 30(3) (2019) 777-787.
- [33] A. Mehrjou, R. Hosseini, B. N. Araabi, Improved Bayesian information criterion for mixture model selection, *Pattern Recognit. Lett.* 69(1) (2016) 22-27.
- [34] Y. Wang, P. Lei, H. Zhou, X. Wang, M. Ma, X. Chen, Using DTW to measure trajectory distance in grid space, *Proceedings of the 2014 IEEE International Conference on Information Science & Technology.* (2014).
- [35] L. Chen, M. T. Özsu, V. Oria, Robust and fast similarity search for moving object trajectories, *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data.* (2005).
- [36] S. Calinon, *Mixture Models for the Analysis, Edition, and Synthesis of Continuous Time Series* (Springer). (2020) 33-57.
- [37] W. D. Li, C. A. McMahon, A simulated annealing – based optimization approach for integrated process planning and scheduling, *Int. J. Comput. Integr. Manuf.*, 20(1) (2007), 80-95.
- [38] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement learning: A survey, *Art. Intell. Res.*, 4 (1996) 237-285.