Robot Learning from Human Demonstration with Remote Lead Through Teaching

Hsien-Chung Lin[1], Te Tang[1], Yongxiang Fan[1], Yu Zhao[1], Masayoshi Tomizuka[1], and Wenjie Chen[2]

***Abstract*— Industrial robots are playing increasingly important roles in factories. Many production applications require both position and force control; however, tuning the position-force controller is nontrivial. To simplify this process, the learning from demonstration (LfD) is proposed to transfer the human skills directly into robot applications. However, the current teaching methods, such as direct demonstration, lead through teaching, and teleoperation, all have their own drawbacks. Hence, Remote Lead Through Teaching (RLTT) is proposed to robot learn some tasks from human knowledge and skill. To implement the human skill model, the demonstration data is firstly synchronized by dynamic time warping (DTW), then decomposed into several actions by a support vector machine (SVM) based classifier. Lastly, the learning controller is trained by the Gaussian mixture regression (GMR). The experimental validation is realized on FANUC LR Mate 200*i*D/7L in a H7/h7 peg-hole insertion task and a surface grinding task.**

I. INTRODUCTION

Many industrial tasks performed by robots require position-force control. For example, the assembly and the surface polishing are realized by controlling not only the position of the end-effector but also the applied force to the workpiece. The position-force control can be categorized into two methods [1], the impedance control and the hybrid position/force control. The impedance control first proposed by Hogan [2] establishes a relationship between the velocity/position of the robot and the interaction force with the environment. The hybrid position/force control proposed by Raibert and Craig [3] separates the position and force control into two independent channels. Hence, the desired position or force for each can be individually specified. These controllers can achieve a good performance when the desired task and the environment are well-defined. However, tuning a set of control parameters is nontrivial. Also, even when the task changes slightly, tuning has to be repeated. Compared to the robot, human can learn from and adapt to various tasks with shorter time and fewer trials. Based on this observation, many researchers have investigated how to transfer the human knowledge and skill to the robot. To simplify the robot programming, the idea of Learning from Demonstration (LfD) is introduced by Schaal [4]. The main principle of robot LfD is that the users can teach the robots through demonstration instead of programming. Then, a question comes up: what is the interface for demonstration? The first intuitive answer is direct demonstration by human. Namely, operators use their own bodies to demonstrate the task, and the motion capture suits or markers record the demonstration. Calinon et al. [5] and Schaal [6] illustrate robot imitation learning from demonstration. This approach may be well suited to the humanoid or anthropomorphic robots. However, the different configurations between human and industrial robot make the mapping difficult. Furthermore, the wearable sensors usually only record the human motions, but many industrial applications require the force information as well.

This work was supported by FANUC Corporation, Japan

[1]The authors are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA. Email: {hclin, tetang, yongxiang_fan, yzhao tomizuka}@berkeley.edu

[2]The author is with FANUC Corporation, Yamanashi Prefecture 401-0597, Japan.

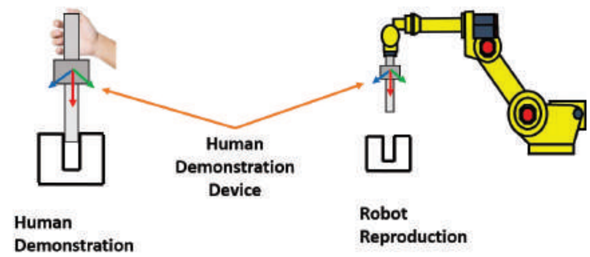


Fig. 1. The common tool frame of the human demonstration device is defined for both the human and robot workspace.

The lead through teaching [7], [8] is another common technique in LfD. The operator directly grasps the link or the handle mounted on the robot. The robot force controller allows the operator manually move the robot arm to pass though a desired path or a sequence of successive points so as to define the task. The lead through teaching provides a convenient and intuitive path planning approach, but it requires physical contact between the operator and the robot, which poses a potential danger to the operator. Also, the force measurement is the external force applied by human, the lead through teaching can not be used to teach the interactive force between the robot and the environment.

Teleoperation [9], [10] or virtual robot teaching [11] separates the workspace of human and robot. The operator manipulates the robot in a virtual reality environment by maneuvering a haptic interface. A sequence of robot commands are generated by recording the human motion/force on the haptic device. The remote operation ensures the operator's safety, but insufficient tactile feedback limits the applications. For instance, the teleoperation method may not be applicable to the complicated industrial tasks such as surface polishing.

In this paper, a novel approach called remote lead through teaching (RLTT) is proposed to simplify the robot pro-

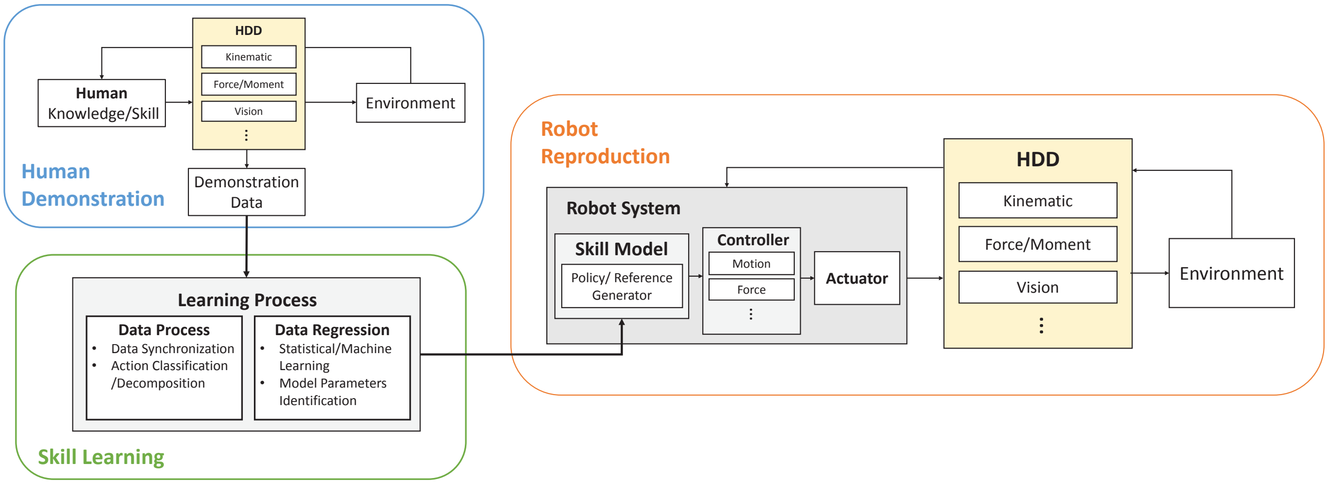


Fig. 2. The framework of remote lead through teaching, where it is decomposed into two phase, human demonstration and robot reproduction. In order to transfer human knowledge/skill to the robot, a skill learning process is established between the two phases.

gramming process. Under the framework of RLTT, human and robot share the common reference. Hence, the human demonstration data can be directly utilized by the robot. RLTT preserves the properties of the lead through teaching method. It also provides to the users a natural and safe demonstration approach.

This paper is organized as follows. The basic concepts and the framework of RLTT are outlined in Section II. The demonstration data processing and learning are introduced in Section III. Section IV presents two applications of RLTT teaching method. Finally, the conclusion is given in Section V.

II. Remote Lead Through Teaching

The basic idea of RLTT is illustrated in Fig. 1. A human demonstration device (HDD) is designed as a common tool for both human and robot. The tool frame in human demonstration phase is aligned to that in robot reproduction phase. HDD is a senor fusion system to record all the task required information during human demonstration. While the human is performing the task, the HDD records the demonstrator's motion and force. When the robot is assigned to reproduce the task, the HDD is mounted on the robot end-effector, and sends the measurement as feedback signal to the robot controller.

The framework of RLTT is shown in Fig. 2. As previously discussed, RLTT is decomposed into two phase, human demonstration and robot reproduction. In order to transfer human knowledge/skill to the robot, a skill learning process is established between the two phases.

A. Human Demonstration Phase

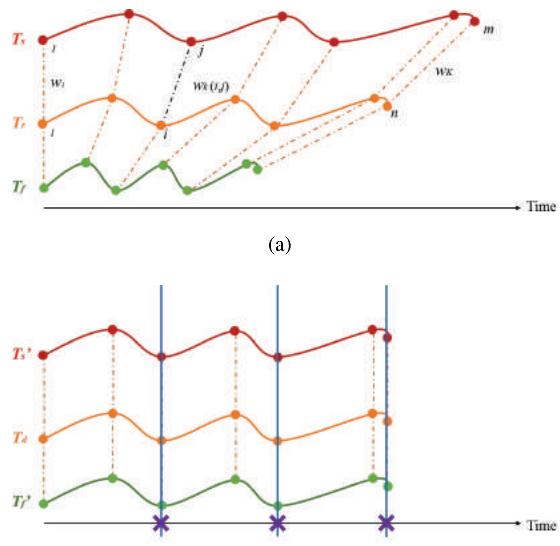
In human demonstration phase, the demonstrator uses HDD to naturally perform the task. At the same time, HDD records the information in the tool frame. As mentioned previously, HDD is a combination of different sensor devices, which may include the measurement of position, velocity, and force, etc. The design of HDD can be modified by the application specific requirement. The detail of the design of HDD and its data acquisition are discussed in [12]. There are several benefits of HDD in the human demonstration phase. Firstly, the human and robot workspaces are separated. Hence, the user's safety is guaranteed. Secondly, the HDD can be regarded as an add-on of the tool. It does not make significant changes of the user operating the task. Thus, the natural demonstration behavior can be preserved by RLTT.

B. Skill Learning Process

The purpose of skill learning process is to build the skill model from the demonstration data. The skill model is a policy or a reference generator for the robot system. When the robot is given a feedback signal from the HDD, the skill model generates a corresponding command to the robot control loops, where the policy is learned from the human demonstration data.

Before training the model from the demonstration data, some processes are required for improving the learning quality. For instance, the demonstrator has different motion speed in each demonstration. Although the demonstration behaviors are similar, the trajectories might look very different due to the mismatched timing. Besides, robot and human have their own expertise. It is not necessary to learn every single action from human demonstration. For example, the robot moves more precisely and faster than human, while human is more intelligent in assembly. Then, the robot does not need to imitate how human approaches the workpiece, but to learn how human assembles the parts together. Hence, the data processing involves two steps. The demonstration data is firstly synchronized, then decomposed into several action segments. The data in the target segments are further utilized for skill model learning.

Since it is difficult to directly derive the human skill model, the data must be first analyzed to describe the human behavior. The statistical learning and machine learning are the methods to identify the model by the training data. If the structure of model is known, the model parameter



(a)

(b)

Fig. 3. The illustration of data synchronization and data decomposition (a)Different trajectories synchronized by dynamic time warping (b) The synchronized data decomposed by the split points

identification technique is applicable to estimate the model parameters. The detail of the skill learning process is discussed in Section III.

C. Robot Reproduction Phase

In the robot reproduction phase, the HDD is mounted on the robot end-effector. In addition, the skill model established from the previous process is embedded into the robot control system. Because the HDD frame in the robot reproduction phase is aligned to that in the human demonstration phase, the sensory information in these two phases are shared in the same reference. Hence, the mechanism of skill model finds the closest human demonstrated policy to the current HDD measurements.

III. Skill Learning From Demonstration

This section introduces how to process and learn from the demonstration data, so as to build the human skill model. The three steps are 1) Data Synchronization, 2) Data Decomposition, and 3) Data Regression.

A. Data Synchronization

Berndt at el [13] proposed the dynamic time warping (DTW) method to deal with the speech recognition problem. The technique of DTW uses a dynamic programming approach to align the time series and a specific pattern so that the distance is minimized. Because each human demonstration is a time series with a specific pattern, DTW is applicable to synchronize multiple demonstrations.

Suppose there are three demonstration trajectories with different speed (see Fig. 3(a)). T_r is the trajectory with reference speed, while T_f and T_s are the fast and slow trajectories, respectively. In practice, the reference trajectory is determine by the user.

As shown in Fig. 3(a), the node i is the value of the trajectory at time step i. $w_k(i,j)$ is the warping distance between two trajectories. The goal of DTW algorithm is to find the optimal sequence of $w_k(i,j)$ such that the total warping distance is minimized.

$$\min_{i_k, j_k} \quad \sum_{k=1}^{K} \| w_k(i_k, j_k) \| \tag{1}$$

$$s.t \quad i_1 = 1, \quad j_1 = 1, \tag{2}$$

$$i_K = m, \quad j_K = n, \tag{3}$$

$$i_{k-1} \le i_k, j_{k-1} \le j_k \tag{4}$$

$$i_k - i_{k-1} \le 1, j_k - j_{k-1} \le 1 \tag{5}$$

$$|i_k - j_k| \le r \tag{6}$$

The constraints are designed to reduce the space of possible warping paths and to make the warping time index more reasonable. (2) and (3) are the initial and final condition of the time series, respectively. (4) implies that all the grid points are monotonically ordered with respect to time. The continuity of the time series is restricted by (5). The last constraint is to define the size of the warping window, $r \in \mathbb{Z}^+$, which makes indices searching more efficient.

As shown in Fig. 3(b), T'_f and T'_s are the trajectories aligned with the reference trajectory. With the synchronized trajectories, the pattern of the demonstration is more distinct to users. The data decomposition is thus applicable to the multiple demonstrations.

B. Data Decomposition

The data decomposition is to find the time steps that the demonstrator changes the action behavior. e.g. the purple points in Fig. 3(b). Since each action has its own pattern in motion/force trajectory, searching the action sequence can be formulated as a action classification problem. The support vector machine is a mature algorithm in classification problems [14], [15]. This paper introduce a SVM-based action classifier to decompose the demonstration data. A demonstration trajectory is given by

$$T = \begin{bmatrix} p_1 & \cdots & p_t & \cdots & p_n \\ f_1 & \cdots & f_t & \cdots & f_n \end{bmatrix} \in \mathbb{R}^{d \times n} \tag{7}$$

where p_t, and f_t are the position and the force measurement at t, d is the total dimension of the measurements, and n is the total time steps of the demonstration. The feature vector is written as

$$\Phi = \mathbf{vec}(T) \tag{8}$$

$$= \begin{bmatrix} p_1 \\ f_1 \\ \vdots \\ p_t \\ f_t \\ \vdots \\ p_n \\ f_n \end{bmatrix} \in \mathbb{R}^{nd} \tag{9}$$

Since the actions are the segments of the demonstration, each action can be represented similarly in the vector form.

To illustrate the action classification by SVM, a simple example is given. Suppose there are two sets of actions. The mechanism of SVM is to construct an optimal hyperplane in the middle of the two classes, so that the margin to the nearest positive or negative example is maximized. The decision function of the action classification is

$$f(\Phi) = \theta^\top \Phi + b \tag{10}$$

where $\theta \in \mathbb{R}^{nd}$ is the weighting parameter vector, and $b \in \mathbb{R}$ is the bias or offset scalar.

Although the actions are assumed to have different patterns, there are some cases that the different actions have a similar pattern, which makes it difficult to determine the decision boundary. Regarding the robustness issue, the soft margin SVM algorithm [16] is proposed by adding the slack variables $\xi_i \geq 0$. Hence, the training example Φ_i can satisfy the constraint even if it is on the wrong side of the decision boundary. The soft-margin SVM is given by

$$\min_{\theta,b,\xi} \quad \|\theta\| + C\sum_{i=1}^{N}\xi_i \tag{11}$$

$$s.t. \quad y_i(\theta^\top \Phi_i + b) \geq 1 - \xi_i, \qquad \forall i = 1, \cdots, N \tag{12}$$

$$\xi_i \geq 0 \tag{13}$$

where $C \geq 0$ is a penalized constant, and $y \in \{-1, 1\}$ is defined as the class label. The SVM classifier can be extended to multi-classes classification without losing the generality.

Hence, a SVM-based classifier can decompose the demonstration into several actions by classifying the motion/force trajectory based on trained class sets.

The demonstration is assumed to be a specific sequence, and the total number of actions is known. e.g. in a peg-hole insertion task, the demonstrator first approaches the hole, then rotates the peg to align with the hole, lastly inserts the peg into the hole. The total number of actions in the insertion task is three. The basic idea of the data decomposition is to find the split points, P_s, in the demonstration, where the split points are the timings that the demonstrator changes his/her action or move on to the next step in the task. To search the split points along the trajectory, the sliding window method is realized, which is a common technique in computer vision for finding the target objects in a picture[17].

The whole algorithm is presented in Algorithm 1. First, the $P_s \in \mathbb{R}^k$ is initialized as zeros, where k is the total number of actions in the task. In each iteration, a segment is firstly cropped by the sliding window, where the width of the window is designed by users. The time steps of cropped segments are aligned with the trained actions by DTW. Then, a classified label is assigned by SVM. The matched segments are labeled with a map score. By calculating the score map, the locations of split points are determined. Keep the iteration until the whole split points are found. With the set of split points, the demonstration can be decomposed into several actions, $\mathcal{A}(p, f)$.

Algorithm 1 Demonstration Decomposition

```
Input the demonstration trajectory
Initialize: P_s = 0, Iter = 1
if Iter < k then
    Crop the segment by the sliding window
    Align the segment dimension by DTW
    Classify by SVM
    if classified label = sequence(Iter) then
        Map Score = 1
    else  Map Score = 0
    end if
    Calculate the center of the local score map for the separate point P_s
    Update: Iter ← Iter +1, P_s ← P_s
else  return the decomposition with P_s
end if
```

C. Data Regression

To mathematically quantify the human skill is not a trivial work. There is no convincing deterministic model to describe the human decision during a single task [18]. Thus, the statistical learning model is utilized to represent the human skill. Suppose the human skill model is a black box, then the human perception Ψ and policy Π are the model input and output, respectively.

Although the relationship between Ψ and Π are ambiguous, the pair of (Ψ, Π) of the human skill is obtained by observing the target action $\mathcal{A}(p, f)$. For instance, when demonstrator performs the insertion task, he/she usually senses the contact force first, then adjusts the motion. In surface grinding case, the demonstrator would think a desired shape first, then apply force on the workpiece.

In this paper, the skill model is estimated by a mixture of Gaussian models [19]. The joint probability of a human perception/policy is estimated by N Gaussian components

$$\Pr(\Psi, \Pi) = \sum_{i=1}^{N} \alpha^i \mathcal{N}(\mu^i, \Sigma^i) \tag{14}$$

where μ^i and Σ^i are the mean and covariance matrix, and α^i is the weighting factor of the i-th Gaussian component.

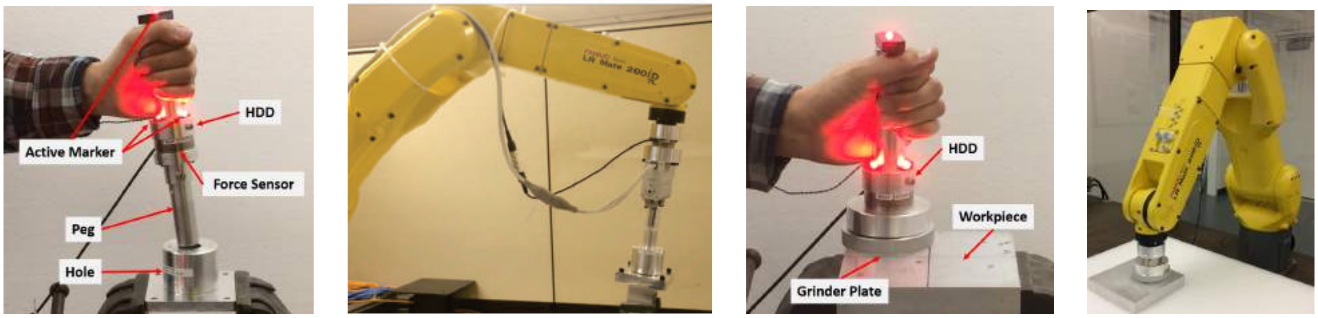
The purpose of skill model is to generate a policy/reference when given a perception. The conditional probability is given by

$$\Pr(\Pi|\Psi) = \mathcal{N}(\mu_{\Pi|\Psi}, \Sigma_{\Pi|\Psi}) \tag{15}$$

and is also a Gaussian due to the properties of Gaussian [19]. The GMR algorithm [20] determines the optimal policy Π^* by maximizing the likelihood of $\Pr(\Pi|\Psi)$

$$\Pi^* = \arg\max_{\Psi} \Pr(\Pi|\Psi) \tag{16}$$

$$= \sum_{i=1}^{N} \frac{\alpha^i \mathcal{N}(\mu^i_{\Pi|\Psi}, \Sigma^i_{\Pi|\Psi})}{\sum_{j=1}^{N} \alpha^j \mathcal{N}(\mu^j_{\Pi|\Psi}, \Sigma^j_{\Pi|\Psi})} \mu^i_{\Pi|\Psi} \tag{17}$$



(a) (b) (c) (d)

Fig. 4. Remote lead through teaching in two industrial applications. (a) The human demonstrator performs the peg-hole insertion task. (b) The robot reproduces the peg-hole insertion experiment. (c) The human demonstrator performs the grinding task. (d) The robot reproduces the grinding experiment.

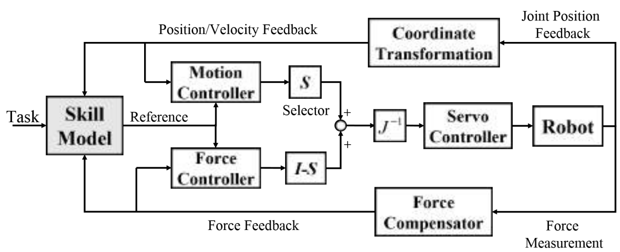


Fig. 5. Skill model in the position-motion hybrid control scheme

where α^i is the weighting factor of the i-th Gaussian component. To implement the GMR, the Gaussian parameters $(\mu^i_{\Pi|\Psi}, \Sigma^i_{\Pi|\Psi}, \alpha^i)$ are estimated by EM algorithm from the demonstration data [21]. Because the EM algorithm is usually sensitive to initialization, K-means clustering [22] is applied to the dataset for a good initial condition.

The skill model trained by (17) is embedded into the position-force hybrid control scheme, which is shown as a shaded block in Fig. 5. The skill model uses the current measurements to represent Ψ, and then generates a policy Π as a robot reference. In this way, the human skill model could be transferred for robot applications. Note that the choice of perception will vary by tasks. For example, in the peg-hole insertion application, the force measurement is used as a feedback to generate the corrective velocity for insertion. In the surface grinding application, the current pose is used to generate the desired force.

IV. Experiment

In order to validate the proposed RLTT framework, it is applied on two classical position-force industrial tasks. The first application scenario is the assembly task, which is represented by peg-hole insertion. The second one is the grinding scenario, which is represented by a simplified testbed. In this section, the experimental setup is firstly introduced. Secondly, the data synchronization and decomposition are illustrated in peg-hole insertion demonstration. Lastly, the policy learning by GMR and the robot execution in the two scenarios are provided.

A. Hardware

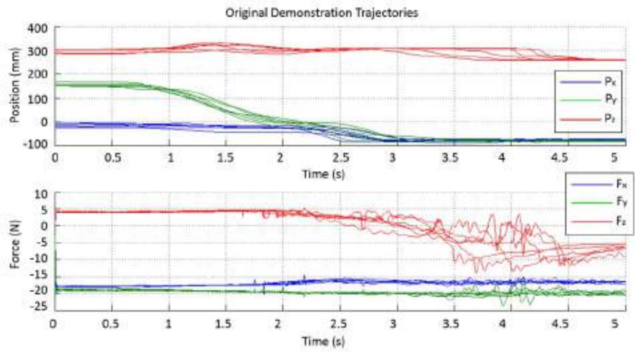
Fig. 4 shows the experimental setup for the remote lead through teaching. Fig. 4(a) and 4(b) are the photos of the peg-hole insertion in human demonstration and robot reproduction phases, respectively. Fig. 4(c) and 4(d) show both phases in the surface grinding. The prototype of the HDD device is shown in Fig. 4(a). The ATI mini 45 F/T transducer [24] is fixed in the center of the HDD. The PhaseSpace Impulse X2 [25] LED markers are placed on the top and body of the HDD. In the human demonstration phase, the demonstrator's motion is captured by tracker cameras. The robot in Fig. 4(b) and 4(d) is FANUC LR Mate 200*i*D/7L [23], where the force sensor is installed on the end-effector. However, markers are not necessary to place on the robot body because the motion of the end-effector can be obtained by calculating the forward kinematics of the current robot joint position.

B. Data Processing

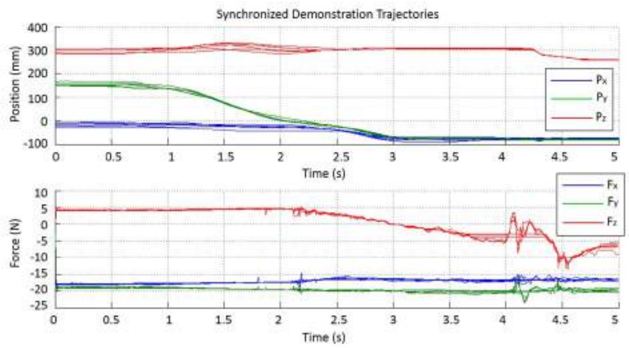
To illustrate the data processing, sixty demonstrations of peg-hole insertion are recorded. The original and synchronized demonstration motion/force trajectories of peg-hole insertion are shown in Fig. 6(a) and 6(b), respectively. In order to have a better illustration, there are only five demonstrations plotted in the figures. The original insertion timings are inconsistent due to the different demonstrator's motion speed. The data decomposition is realized by the SVM-based action classifier with the SVM toolbox package [26]. The training data is acquired by manually decomposing and labeling in the 20 demonstrations, and the rest of demonstrations are automatically decomposed by the SVM-based action classifier. The result is shown in Fig. 6(c), where the three actions decomposed from the peg-hole insertion demonstration are approaching, rotation, and insertion. In this paper, the insertion part is used to train the human skill model.

C. Assembly Scenario

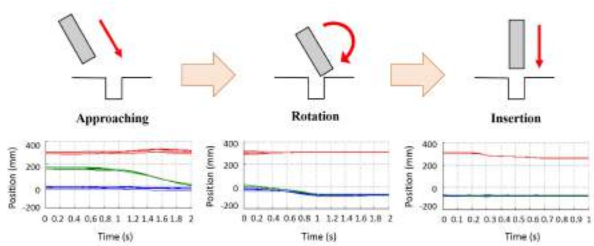
The assembly scenario is designed as an industrial standard H7/h7 peg-hole insertion task. The material and the dimension of the workpieces are listed in Table I, where the tolerance is below 0.030 mm. The action segments of



(a)



(b)



(c)

Fig. 6. The demonstration data processing. (a) The original demonstration data. (b) The synchronized demonstration data. (c) The demonstration decomposed into three actions: approaching, rotation, insertion.

the insertion from the data processing are used to train the skill model. The sensed wrench is regarded as the input of the model, and the corrective velocity is regarded as the output of the model. Since the dimensions of wrench is six, it is difficult to visualize the model. An insertion action segment is used as a query data, and the comparison of the skill model output and the original demonstration is shown in Fig. 7. The red-dash line is the corrective velocity from human demonstration, while the blue solid line is the velocity reference generated by the skill model. The robot has completed 50 trials in the experiment. The statistical result for the robot reproduction is shown in Table II. The robot achieves the 96% successful rate in the H7/h7 peg-hole testbed. The two failure cases is caused by the fact that the peg is tilted due to the frequent insertion.

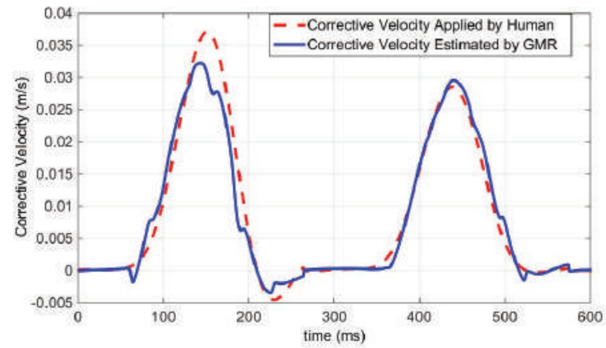


Fig. 7. The GMR model for the assembly scenario

TABLE I

THE SPECIFICATION OF PEG-HOLE INSERTION TESTBED

	Material	mm (SI)	inch (UI)	Tolerance
Hole	T6101	$\phi 25.400^{+0.003}_{-0.000}$	$\phi 1.000^{+0.0001}_{-0.0000}$	H7
Peg	T6101	$\phi 25.400^{+0.000}_{-0.030}$	$\phi 1.000^{+0.0012}_{-0.0000}$	h7

D. Grinding Scenario

In the grinding scenario, the demonstrator performs the grinding task as shown in Fig 4(c). For the safety consideration and the process simplification, an aluminum plate represents a grinder in this experiment. Also, the demonstrator only presses large forces at two specific regions of the workpiece, and moves on the surface with small force. The skill model is trained by the demonstration, where the demonstrator's pose is the input and the pressing force is the output of the model. The skill model of the demonstrator's force distribution on the workpiece is shown in Fig. 8(a). The orange dash line in Fig. 8(b) is the force reference that sliced in the diagonal direction from Fig. 8(a), and the blue solid line is the force that the robot applies on the workpiece along the path. The result indicates that the robot learns the human's intension in applying the different forces along the trajectory.

V. CONCLUSION

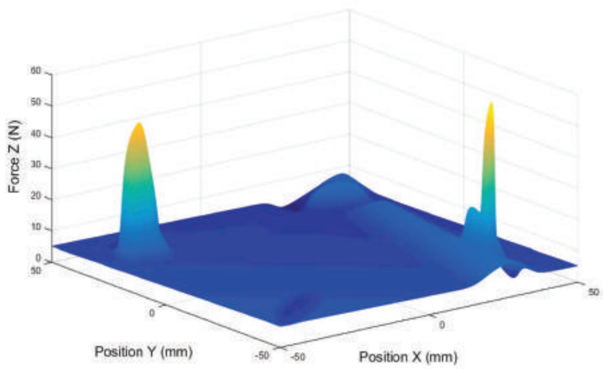
In this paper, a novel framework of the remote lead through teaching (RLTT) was introduced to simplify the robot programming problem. The idea of RLTT is to design a common reference between human demonstration and robot reproduction. Hence, the demonstration data from human can be directly utilized in the robot reproduction.

This paper also introduced the skill learning process by three steps. First, the dynamic time warping (DTW) synchronizes the demonstration data in the same time horizon. Second, a support vector machine (SVM) based action classifier is designed to decompose the demonstration data into several action segments. Lastly, the Gaussian mixture regression (GMR) is used to train the human skill model.

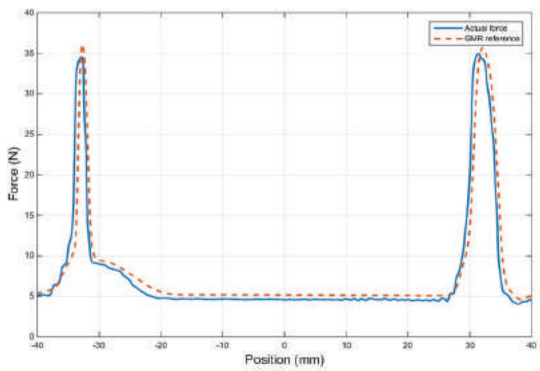
Two experimental verifications in the classical industrial application scenarios were given. In the assembly scenario, the robot has achieved 96% successful rate in H7/h7 peg-hole

TABLE II
THE ROBOT EXECUTION RESULT IN PEG-HOLE INSERTION

Total trials	Success	Fail	Success Rate(%)
50	48	2	96.00%



(a)



(b)

Fig. 8. The skill model for the grinding scenario. (a) The grinding force reference distribution over the workpiece trained by GMR. (b) The force reference of the skill model and the actual robot force along the desired path.

insertion, where the tolerance was below 0.030 mm. In the grinding scenario, the robot successfully imitated the human behavior to press large forces in the desired spots.

REFERENCES

[1] J. Pliego-Jiménez and M. A. Arteaga-Pérez, "Adaptive position/force control for robot manipulators in contact with a rigid surface with uncertain parameters," *European Journal of Control*, vol. 22, pp. 1–12, 2015.

[2] N. Hogan, "Impedance control: An approach to manipulation: Part iiimplementation," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 8–16, 1985.

[3] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[4] S. Schaal *et al.*, "Learning from demonstration," *Advances in neural information processing systems*, pp. 1040–1046, 1997.

[5] S. Calinon and A. Billard, "Learning of Gestures by Imitation in a Humanoid Robot," in *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge University Press, 2007, pp. 153–177, k. Dautenhahn and C.L. Nehaniv (Eds).

[6] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 358, no. 1431, pp. 537–547, 2003.

[7] T. L. Graf, "Lead-through robot programming system," Mar. 9 1999, uS Patent 5,880,956.

[8] S. Choi, W. Eakins, G. Rossano, and T. Fuhlbrigge, "Lead-through robot teaching," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–4.

[9] F. A. Candelas Herías, C. A. Jara Bravo, and F. Torres Medina, "Flexible virtual and remote laboratory for teaching robotics," 2006.

[10] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, "Learning collaborative manipulation tasks by demonstration using a haptic interface," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.

[11] H. Kawasaki, S. Nanmo, T. Mouri, and S. Ueki, "Virtual robot teaching for humanoid hand robot using muti-fingered haptic interface," in *Communications, Computing and Control Applications (CCCA), 2011 International Conference on*, March 2011, pp. 1–6.

[12] H.-C. Lin, T. Tang *et al.*, "Remote lead through teaching by human demonstration device," in *ASME 2015 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, oct 2015.

[13] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[14] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152. [Online]. Available: http://doi.acm.org/10.1145/130385.130401

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] Q. Wu and D.-X. Zhou, "Svm soft margin classifiers: linear programming versus quadratic programming," *Neural computation*, vol. 17, no. 5, pp. 1160–1187, 2005.

[17] S.-S. Zhuang and S.-H. Lai, "Face detection directly from h. 264 compressed video with convolutional neural network," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 2485–2488.

[18] G. E. Hovland, P. Sikka, and B. J. McCarragher, "Skill acquisition from human demonstration using a hidden markov model," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 3. IEEE, 1996, pp. 2706–2711.

[19] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[20] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[21] J. A. Bilmes *et al.*, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[22] F. Badano, A. Jutard, and M. Betemps, "Chamferless robotic assembly using random search," in *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*. IEEE, 1991, pp. 1598–1601.

[23] FANUC Corporation, "http://www.fanucrobotics.com."

[24] ATI Industrial Automation, "http://www.ati-ia.com."

[25] PhaseSpace, "http://http://www.phasespace.com/."

[26] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.