Current Robotics Reports
https://doi.org/10.1007/s43154-020-00023-4

Advanced Robot Programming: a Review

Zhongxiang Zhou[1] · Rong Xiong[1] · Yue Wang[1] · Jiafan Zhang[2]

Abstract

Purpose of Review The review presents an overview of advanced robot programming approaches which aims to ease robot programming and speed up the deployment of industrial robots, and then some considerations are shared with respect to requirements in new trends of manufacturing.

Recent Findings The new trend of customization along with Industry 4.0 is appearing which is a challenge for robotized systems. The bottleneck is mainly the efficiency of deployment of industrial robots because traditional programming methods are not intuitive and always time-consuming. Advanced robot programming techniques are expected to ease robot programming and make it accessible for non-experts.

Summary A review on advanced robot programming is here presented, firstly introducing the background of this research, followed by reviewing literatures in four categories: programming by demonstration for low-level motion, programming by demonstration for high-level task, speech recognition-based and augmented reality-based programming approaches, and finishing on discussing future works.

Keywords Advanced robot programming · Programming by demonstration · Speech recognition · Augmented reality

Introduction

With the deployment of industrial robots in manufacturing, productivity and quality have been boosting for decades. The critical role of the robot mainly owes to its capacity for repeating a wide variety of tasks with high speed and accuracy in long term; in terms of cost, the deployment of the robot takes days to months of programming by robotics engineers. On the other hand, the new trend of customization faced by the manufacturing enterprises changes this situation and brings new characteristics, production in small volume but large variants, and short cycle. This irreversible momentum urges the robot to be deployed from task to task efficiently.

However, traditional robot programming approaches, namely, lead-through (also called kinesthetic teaching), drive-through, and off-line programming, are time-consuming, unintuitive, and high skill demanded. Tedious programming has become the crucial bottleneck during robot deployment, which also has high prerequisites for most practitioners in manufacturing and makes industrial robots hard to be widely used in small and middle enterprises (SMEs).

In order to resolve the above bottleneck, several advanced robot programming techniques have been developed by researchers to ease robot programming and speed up the deployment of industrial robots. With the development of artificial intelligence, related technologies have also been applied to make industrial robots more intelligent and make robot programming accessible to non-experts. The main existing techniques of advanced robot programming can be viewed as different variants of programming by demonstration (PbD) pipeline; meanwhile, there are other methods based on speech recognition and augmented reality. The PbD approach is unique as

This article is part of the Topical Collection on *Robotics in Manufacturing*

✉ Rong Xiong
rxiong@zju.edu.cn

Zhongxiang Zhou
zhongxiangzju@gmail.com

Yue Wang
ywang24@zju.edu.cn

Jiafan Zhang
jiafan.zhang@cn.abb.com

[1] State Key Laboratory of Industrial Control Technology and Institute of Cyber-Systems and Control of Zhejiang Province, Zhejiang University, Hangzhou 310027, China

[2] Research Center China, Robotics and Discrete Automation ABB, Shanghai 201319, China

Published online: 17 August 2020

the robot system learns, improves, or duplicates actions based on human demonstrations. It provides the users with an intuitive and fast way of programming by allowing the task to be performed naturally and leaving the robot system to observe, follow, and learn from sensory information. Furthermore, approaches based on programming by demonstration can be classified to two categories that are PbD for low-level motion and PbD for high-level task. Speech recognition-based robot programming methods provide the translation between human voice and commands to be executed on an industrial robot. And approaches using augmented reality techniques aim to provide information closely related to robot tasks by placing virtual objects in the real world and allowing the human to interact with them using specific devices.

The rest of this paper reviews advanced robot programming techniques by classifying them into four categories as follows: PbD for low-level motion, PbD for high-level task, speech recognition based-, and augmented reality-based approaches, and finished by a discussion about future works.

PbD for Low-Level Motion

Programming by demonstration for low-level motion refers to learning robot trajectories or force controllers from human demonstrations specifically which includes two fundamental stages, namely, “data acquisition” and “data modeling.” In the data acquisition stage, methods are proposed to capture human motion data during the process of demonstration. In the data modeling stage, the aim is to extract critical information from captured data and provide the robot with the ability to adapt the learned skill to different situations. Researches focusing on programming by demonstration in trajectory level have a long history, and related theories and methods are gradually improved.

Firstly, we discuss two kinds of techniques used to record the demonstration trajectory of human as shown in Fig. 1. One is mapping-based method that records human motion data by cameras, inertial sensors, data gloves, and other sensors; then map these data from human to robot [1–3]. The advantages of this kind of method are two folds that one is intuitive for human demonstrators and the other is smooth trajectories can be captured. Because of the differences in configuration, size, and ability existing between human and robot, these advantages are valid only on the promise that a mapping function from human to robot can be built. The other is non-mapping-based methods that sensors on robot record its own motions as it is passively teleoperated by human or moved using kinesthetic teaching [4, 5]. This kind of method omits the correspondence problem between human and robot and is quite straightforward for robots, while the cost is that it is hard to acquire a smooth and accurate trajectory. And it is also not easy for demonstrators to control a multi-degree-of-freedom manipulator using non-mapping-based methods.

Given human demonstration data that have been acquired using one of the methods described above, we now discuss methods for generating robot movement using this data. The simplest way is to use this data directly, but it will fail if there exist differences between application and demonstration. Considering the generalization ability between different applications, it is necessary to learn a robot motion model on the base of demonstration data. The existing methods can be concluded into two broad categories based on what is learned, as stated in [6]. One category tries to learn a policy that is consistent with the demonstrated behaviors [7–12]. Here the policy is a mapping from states to actions, which can be executed by the robot to generate similar actions like demonstrations. The other category tries to learn a cost function that the demonstrator tries to optimize [13–18]. Here the cost function is a mapping from states to a scalar cost value. For the policy learning methods, the policy becomes invalid and needs to be relearned when the underlying state transition model of the robot system changes. For the cost learning method, the cost function together with the information of state transition model can be optimized to obtain a policy, which makes it consistently effective under changing domain dynamics.

Policy Learning

As we know, the traditional robot programming methods always rely on teach pendant to define trajectories or end-

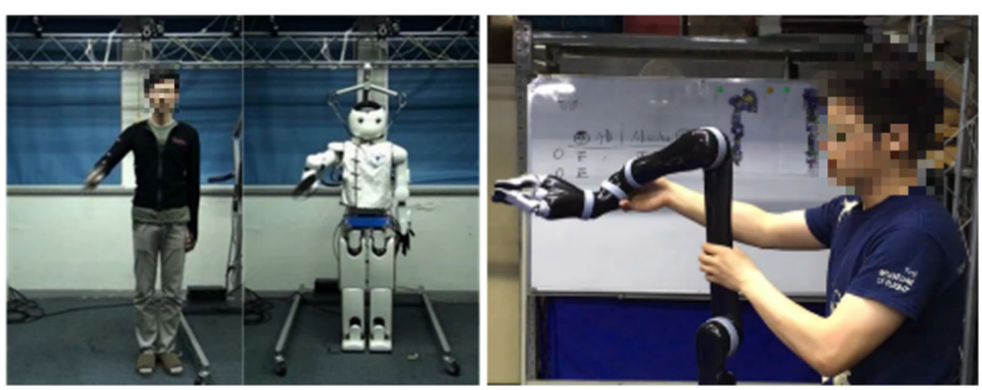


Fig. 1 Demonstrated trajectory acquisition methods. Left: mapping-based. Right: kinesthetic teaching

effector poses and then replay the same trajectories by robot. The shortcoming of such teaching reproduction way is obvious as it only follows the defined trajectories but cannot adapt to environmental changes. Policy learning is also known as behavioral cloning which can generate desired trajectories when the environment changes using a motion model built with critical information extracted from demonstrated data automatically. The most important phase of policy learning is data modeling.

In the results of early researches, critical points were extracted from trajectories demonstrated using teach pendant, and motion policies were modeled by a sequence of critical states and actions between two states [19, 20]. With the development of statistical learning, Hidden Markov Model (HMM) was applied in which the system was assumed to be a Markov process and the demonstrations were modeled by a series of transitions between discrete states [21, 22]. Because demonstrations were modeled as discrete states and transitions between them in these methods, it is hard to generate a smooth trajectory which makes it impossible to control robot joint motions directly. In practical applications, researchers generally use the average of multiple trajectories or interpolate between discrete states to obtain a continuous and smooth trajectory.

In order to model continuous trajectories directly, various approaches were proposed, and one of the most representative methods is known as Dynamic Movement Primitives (DMP) which formulates the demonstrated trajectory as a dynamic system with a set of differential equations [2, 23]. Representing a movement with a differential equation has the advantage that a perturbance can be automatically corrected by the dynamics of the system. Furthermore, the equations were formulated in a way that adaptation to a new goal is achieved by simply changing a goal parameter. This characteristic allows generalization easily. There are two kinds of DMPs: discrete and rhythmic. For discrete movements, the base system is a point attractor, and for rhythmic movements, a limit cycle is used. Both point attractor and limit cycle attractors of almost arbitrary complexity can be achieved. DMP is one of the most used frameworks for trajectory learning from a single demonstration, since the demonstrations by different people for the same task cannot be the same; DMP is hard to take the uncertainties among multiple demonstrations into consideration. In addition, many hyperparameters contained in DMP need to be defined in advance, and the performance of DMP will decrease with improper hyperparameters.

Taking the uncertainties among multiple demonstrations into consideration, Gaussian mixture model (GMM)-based approaches were proposed in which different stages of demonstrations were modeled by multiple Gaussian distributions and uncertainties in the same stage were stated as the covariance of Gaussian distribution [24, 25] as shown in Fig. 2.

Cost Learning

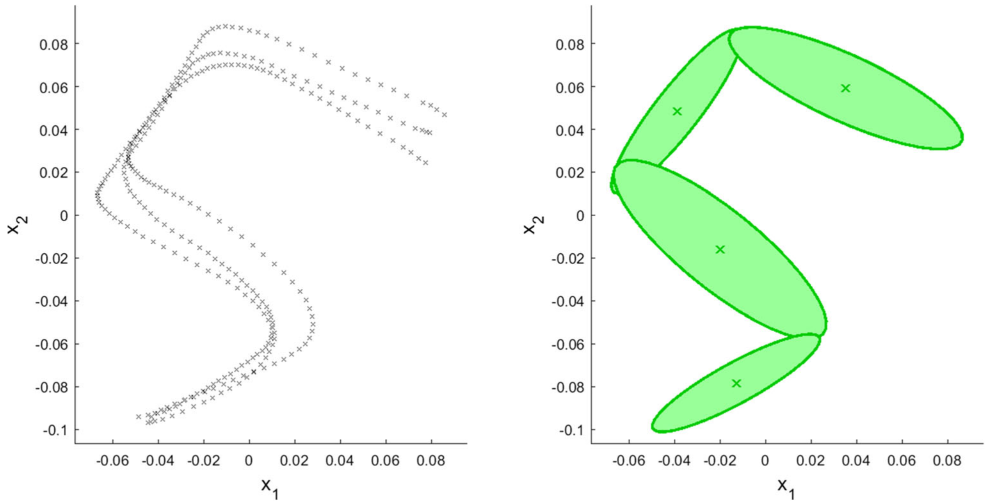
Cost learning also known as reward learning assumes that desired trajectory results from the optimization of a hidden function, known as a cost function or reward function. The goal of such methods is to extract the hidden function from the available demonstrations. Subsequently, the robot reproduces the learned motions by optimizing cost functions. Different from policy learning which needs to be relearned in the presence of heavy environmental changes, the learned cost functions can be optimized together with the information of a new environment to obtain a valid policy.

The problem of cost learning is commonly casted in the Markov decision process (MDP) setting and solved by the so-called inverse reinforcement learning (IRL) methods [13]. There are two challenges of cost learning: first, this problem is ill-posed as there might be multiple cost functions that optimally explain the available demonstrations; second, it is hard to use only one cost function to generate entire demonstrations since demonstrations may be acquired from different tasks or different stages of one task. For the first challenge, approaches were proposed to find a unique cost function by minimizing the differences between feature expectations of the resulting policy and feature expectations of demonstrations. And maximum-entropy-based and maximum-margin-based IRL were commonly used in these approaches [14, 15]. For the second challenge, rather than using one cost function for entire demonstrations, different cost functions are used to generate different demonstrations [26]. Researchers also proposed to apply multiple cost functions for one demonstration in the case that the demonstration was composed of multiple substages and every two successive substages were generated by different cost functions [6].

Force Controller Learning

To further improve the robot ability of interacting with the environment, force controllers are necessary. Devices which can be used to record force data from human directly are not ready, and mapping-based data acquisition methods are impossible without available devices. Non-mapping-based methods such as teleoperation and kinesthetic teaching were used to record the force information of human demonstration as the base of modeling process. In terms of the outputs of learned model, we can summarize the existing methods into two categories. In the first category, the outputs of learned model are trajectories of positions and forces [27], while results of methods in the second category are impedance parameters [28, 29] which are inputs to a compliant controller. And a wide range of probabilistic modeling approaches such as GMM and Gaussian process regression (GPR) are applied to model the demonstrations [3].

Fig. 2 Modeling multiple demonstrations using GMM



PbD for High-Level Task

Programming by demonstration for low-level motion always focus on learning individual robot motions, while programming by demonstration for high-level task aims on learning complex tasks which are composed of a combination of individual actions such as setting a dinner table or assembly tasks in manufacturing. Programming by demonstration for high-level task can speed up the deployment of industrial robots greatly which is similar to the interaction between a "teacher" and a "student" in the factory. To get the information of tasks human demonstrated, non-mapping-based kinesthetic teaching was applied by many methods. As mentioned before, it is not easy for human to control a multi-degree-of-freedom manipulator using kinesthetic teaching especially for complex tasks. A more natural mapping-based approach was adopted in most researches in which the task can be demonstrated naturally and captured by external sensors such as cameras and motion capture systems, and then the semantics of human operations are extracted and transferred to robot motions. The main challenge of this approach is to extract the semantics of human operations accurately. Normally three kinds of techniques are proposed to tackle this challenge including object position-based semantic understanding, action- or gesture-based semantic understanding, and constraint-based semantic understanding.

Object Position-Based Semantic Understanding

The underlying idea of these approaches is that human operations during task demonstrations will change the positions of objects in the scene; thus, different operations can be identified using changes of objects positions before and after one operation.

Semantic scene graph was proposed in [30, 31]. It was extracted from image sequences and used to find the characteristic main graphs of the action sequence via an exact graph-matching technique, thus providing an event table of the action scene, which allows extracting object action relations and semantic understanding. An abstract action representation method was proposed in [32]. Given the tracked point clouds for all objects involved in the manipulation, a set of spatial relation predicates were evaluated for all object pairs at all video frames by object segmentation and tracking, and then action descriptors were built upon spatial Predicate Vector Sequences (PVS). As we can see, semantics of human actions were inferred from changes of object positions in the methods mentioned above; the limitation is that it cannot be applied to identify actions changing object positions insignificantly.

Action- or Gesture-Based Semantic Understanding

The underlying idea of these approaches is to convert the semantic understanding to human action recognition or gesture recognition since the semantics are corresponding to human actions directly.

For gesture recognition, some methods were achieved based on a single image [33, 34], and another methods were implemented using multiple historical images [35]. For action recognition, a template-based method for recognizing human actions was proposed in [36]. Since many actions are visually similar, a multimodal information-fused action recognition method was presented in [37]. With the development of deep learning, convolutional neural network-based action recognition methods were designed [38–41]. Although impressive results have been achieved in public datasets, the convolutional neural network-based methods rely on massive training data which may be impossible in industrial environment.

Constraint-Based Semantic Understanding

The underlying idea of these approaches is to extract the constraints or rules from multiple task demonstrations, and then apply these constraints to solve similar tasks.

A hybrid dynamic system was used to model spindle assembly task in [42], and events and states in the demonstrated sequences were represented by a directed graph, and then the optimal, task-level execution strategy was selected based on execution time and control effort. The underlying constraints were identified by the robot itself based on multiple human observations in [43]. The constraints were then considered in the planning phase, allowing the task to be executed without violating any of them. A syntactic approach aiming to captures important task structures in the form of probabilistic activity grammars from a reasonably small number of demonstrations was described in [44]; grammars can be recursively applied to help recognize unforeseen, more complicated tasks that share underlying structures after learning. And the proposed method was evaluated in Towers of Hanoi experiments.

To further verify the feasibility, several robot programming by demonstration systems for high-level tasks have been proposed in the scenarios of industrial assembly [45•, 46, 47].

In summary, the aims of PbD for low-level motion is learning trajectories or force controllers of single manipulation, and methods for trajectories learning are relatively mature and have been applied to several applications. In terms of force controller learning and PbD for high-level task, there still have issues to be solved in future researches such as non-rigid object manipulation.

Speech Recognition-Based Approaches

Since speech is one of the most common ways of human machine interaction; speech recognition-based advanced robot programming approaches are proposed for sending human commands to robots in the industrialized circumstance. In order to make it applicable in the communication with robots, the first thing to be done is to develop a system for automatic speech recognition (ASR) which is always available with the help of Microsoft Speech Engine.

Vocal commanding between a human and a robot, used in industrial applications, was explored in [48]. The speech recognition and text to speech application were developed capable of supporting a dialog between human and robot for pick-and-place and welding tasks. An interactive industrial robotic system for robot pick-and-place tasks, combining voice commands from humans, was presented in [49]. This study has also presented that vocal commands can be recognized successfully at the noise level of 89 dB in industrial environment. Similar system was also presented in [50] to manipulate the objects placed randomly on a table with industrial robot ABB IRB140. Although there are some works implementing robot programming by speech recognition, the tasks executed by robots were quite simple, and mapping between vocal command and robot motion is defined in advance. To program robot for complex tasks using speech recognition, simply recognizing vocal commands may be not enough, and enabling robots with the ability of understanding is important. And linking speech recognition with mixed reality [51] in robot programming is an interesting direction for future researchers. In this scenario, robot trajectories can be defined by operators using mixed reality techniques, and different choices in graphical user interface can be called using vocal commands.

Augmented Reality-Based Approaches

To alleviate the fact that reprogramming robotic systems requires expert knowledge, another advanced robot programming approach based on augmented reality techniques is proposed. Despite the development of more powerful hardware and software, the usage of augmented reality (AR) system is mainly limited to gaming applications. Merging AR with robotic systems brings new human-robot interaction (HRI) possibilities, and robot programming could be intuitive and flexible with the aid of these technologies.

An AR interface was proposed in [52] that uses a marker cube attached to a probe, which allows a user to guide a virtual robot by setting waypoints and orientations. The AR scene is visualized through a desktop monitor. As an improvement, an AR manufacturing paradigm was proposed in [53] in which a user can specify the fly of a robot trajectory through free space or in contact with a surface, visualize a preview of the robot movement, and monitor and modify robot variables during the simulation or execution mode. And a drag- and drop-like programming method for common pick and place tasks, using AR devices such as Microsoft HoloLens, was presented in [54]. AR was integrated with tactile feedback in [55]. It was proposed to help engineers for programming an industrial robot easily and naturally and provide assistance in real time. AR-based industrial robot programming framework is also described in [56]. It was found that the AR-based framework can significantly ease robot programming and motion planning and reduce the necessity for extensive training of the human workers, in other words, make robot programming accessible for non-experts.

A detailed review of AR research in robotics was given, and some future challenges were pointed out in [57•]. Even though advances in wearable devices enable integration of AR in different areas of robotics, there are still issues that need to be addressed. For instance, current wearable devices have a limited field-of-view, poor tracking stability, especially in the presence of occlusions, and crude user interfaces during interaction with the 3D contents of the augmented environment.

Despite improvements in robot programming brought by AR techniques, further researches are still needed for the usage of AR in robotic systems outside of the laboratories. For reliability and robustness of the real-world applications, the complexity of the visualization and registration methods should be reduced. Moreover, accurate and semi-automated calibration is needed in order to integrate AR in robotic systems.

Conclusions and Future Works

In summary, researches on advanced robot programming focus on developing the easy-to-use and affordable, especially for SMEs, robot programming tools. The role of humans in advanced robot programming is expected to be an assistant, without requiring expertise in robotics. Four categories of approaches are reviewed and impressive progresses have been achieved both in programming for low-level motions and high-level tasks.

Based on recent research results and demands of Industry 4.0, we think that the following issues are still challenging for existing methods and need to be tackled in future researches.

- Accurate pose estimation of parts in demonstration. Since assembly task is a crucial process in manufacturing as it takes 40–60% of total production time with 20–30% of overall production cost [58], applying robot programming by demonstration techniques in assembly tasks is quite promising direction in the future. One of the most challenging issues is to locate parts manipulated by human accurately owing to the high-precision characteristics of assembly task.
- Non-rigid object manipulation. Rigid objects are mainly considered in existing robot manipulation tasks; however, plenty of non-rigid object manipulation tasks are still finished by humans in factories, such as laying wire harness along a guiding groove. If these works can be done with robots, efficiency of production can be improved further. However, modeling and controlling of non-rigid object manipulation tasks are still open problems.
- Multimodal information-fused robot feedback control. Feedback control is the foundation of robot interaction with environment; visual information is widely used in existing methods, while force and tactile information are hardly used. Further study on multimodal feedback control methods using vision, force, and tactile information can effectively improve anti-disturbance ability of robot systems.
- Real-time collision avoidance. Human-robot collaboration may be a common scene in future factories, and ensuring the safety of human is the most important thing in this situation. Real-time collision avoidance and path generation is a way to provide the required safety, but the efficiency of obstacles modeling and robot path generation is needed to be improved.
- Grasping planning. Many great improvements have been achieved for robust grasping planning, such as Dex-Net [59••]. However, existing methods mainly focus on planning for a parallel robot gripper with fixed stroke; grasping planning for grippers with different strokes and more fingers is worth to be studied in the future.

Funding Information This work was supported in part by the National Nature Science Foundation of China (Grant No. U1609210) and in part by Science and Technology Project of Zhejiang Province (Grant No. 2019C01043).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Human and Animal Rights and Informed Consent This article does not contain any studies with human or animal subjects performed by any of the authors.

References

Papers of particular interest, published recently, have been highlighted as:
• Of importance
•• Of major importance

1 Pardowitz M, Knoop S, Dillmann R, et al. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007, 37(2): 322–332.
2 Hoffmann H, Pastor P, Park DH, et al. Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance//2009 IEEE international conference on robotics and automation. IEEE. 2009:2587–92.
3 Silvério J, Huang Y, Rozo L, et al. Probabilistic learning of torque controllers from kinematic and force constraints//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2018:1–8.
4 Peters R A, Campbell C L, Bluethmann W J, et al. Robonaut task learning through teleoperation//2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422). IEEE, 2003, 2: 2806–2811.
5 Inamura T, Kojo N, Inaba M. Situation recognition and behavior induction based on geometric symbol representation of multimodal sensorimotor patterns//2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 2006:5147–52.
6 Hu J, Xiong R. Trajectory generation with multi-stage cost functions learned from demonstrations. Robot Auton Syst. 2019;117:57–67.
7 Billard A, Calinon S, Dillmann R, et al. Survey: robot programming by demonstration. Handbook of robotics, 2008, 59(BOOK_CHAP).
8 Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S. Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput. 2013;25(2):328–73.
9 Paraschos A, Daniel C, Peters JR, et al. Probabilistic movement primitives. Adv Neural Inf Process Syst. 2013:2616–24.

10 Akgun B, Thomaz A. Simultaneously learning actions and goals from demonstration. Auton Robot. 2016;40(2):211–27.
11 Hung PN, Yoshimi T. Programming everyday task using primitive skills and generative model of movement demonstrated by human// 2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS). IEEE. 2017:26–32.
12 Mao R, Yang Y, Fermüller C, et al. Learning hand movements from markerless demonstrations for humanoid tasks//2014 IEEE-RAS International Conference on Humanoid Robots. IEEE. 2014:938–43.
13 Ng AY, Russell SJ. Algorithms for inverse reinforcement learning. Icml. 2000;1:2.
14 Abbeel P, Ng A Y. Apprenticeship learning via inverse reinforcement learning//Proceedings of the twenty-first international conference on Machine learning 2004: 1.
15 Ziebart BD, Maas AL, Bagnell JA, et al. Maximum entropy inverse reinforcement learning. Aaai. 2008;8:1433–8.
16 Ramachandran D, Amir E. Bayesian Inverse Reinforcement Learning. IJCAI. 2007;7:2586–91.
17 Boularias A, Kober J, Peters J. Relative entropy inverse reinforcement learning. Proceedings of the Fourteenth International Conference on AISTATS, in PMLR. 2011;15:182–9.
18 Kalakrishnan M, Pastor P, Righetti L, et al. Learning objective functions for manipulation//2013 IEEE International Conference on Robotics and Automation. IEEE. 2013:1331–6.
19 Levas A, Selfridge M. A user-friendly high-level robot teaching system//Proceedings. 1984 IEEE International Conference on Robotics and Automation. IEEE, 1984, 1: 413–416.
20 Segre A, DeJong G. Explanation-based manipulator learning: acquisition of planning ability through observation Proceedings. 1985 IEEE International Conference on Robotics and Automation. IEEE, 1985, 2: 555–560.
21 Tso SK, Liu KP. Hidden Markov model for intelligent extraction of robot trajectory command from demonstrated trajectories// Proceedings of the IEEE International Conference on Industrial Technology (ICIT'96). IEEE. 1996:294–8.
22 Calinon S, Pistillo A, Caldwell DG. Encoding the time and space constraints of a task in explicit-duration hidden Markov model// 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 2011:3413–8.
23 Ijspeert AJ, Nakanishi J, Schaal S. Movement imitation with nonlinear dynamical systems in humanoid robots//Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292). IEEE. 2002;2:1398–403.
24 Calinon S, Guenter F, Billard A. On learning, representing, and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007, 37(2): 286–298.
25 Calinon S, Billard A. A probabilistic programming by demonstration framework handling constraints in joint space and task space// 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 2008:367–72.
26 Babes M, Marivate V, Subramanian K, et al. Apprenticeship learning about multiple intentions//Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011: 897–904.
27 Montebelli A, Steinmetz F, Kyrki V. On handing down our tools to robots: single-phase kinesthetic teaching for dynamic in-contact tasks//2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2015:5628–34.
28 Kormushev P, Calinon S, Caldwell DG. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. Adv Robot. 2011;25(5):581–603.
29 Kronander K, Billard A. Learning compliant manipulation through kinesthetic and tactile human-robot interaction. IEEE transactions on haptics. 2013;7(3):367–80.
30 Aksoy EE, Abramov A, Wörgötter F, et al. Categorizing object-action relations from semantic scene graphs//2010 IEEE International Conference on Robotics and Automation. IEEE. 2010:398–405.
31 Aksoy EE, Abramov A, Dörr J, Ning K, Dellen B, Wörgötter F. Learning the semantics of object–action relations by observation. Int J Robot Res. 2011;30(10):1229–49.
32 Zampogiannis K, Yang Y, Fermüller C, et al. Learning the spatial semantics of manipulation actions through preposition grounding// 2015 IEEE international conference on robotics and automation (ICRA). IEEE. 2015:1389–96.
33 Mitra S, Acharya T. Gesture recognition: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2007, 37(3): 311–324.
34 Chalechale A, Safaei F, Naghdy G, et al. Hand gesture selection and recognition for visual-based human-machine interface//2005 IEEE International Conference on Electro Information Technology. IEEE, 2005: 6 pp.-6.
35 Alahari K, Jawahar CV. Discriminative actions for recognising events//computer vision. Berlin, Heidelberg: Graphics and Image Processing. Springer; 2006. p. 552–63.
36 Rodriguez MD, Ahmed J, Shah M. Action mach a spatio-temporal maximum average correlation height filter for action recognition// 2008 IEEE conference on computer vision and pattern recognition. IEEE. 2008:1–8.
37 Wang Y, Jiao Y, Xiong R, Yu H, Zhang J, Liu Y. MASD: a multimodal assembly skill decoding system for robot programming by demonstration. IEEE Trans Autom Sci Eng. 2018;15(4):1722–34.
38 Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks. 2014 IEEE Conference on CVPR. Columbus, OH, 2014, pp. 1725–32.
39 Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. Adv Neural Inf Process Syst. 2014: 568–76.
40 Tran D, Bourdev L, Fergus R, et al. Learning spatiotemporal features with 3d convolutional networks. Proc IEEE Int Conf Comput Vision. 2015:4489–97.
41 Tran D, Wang H, Torresani L, et al. Video classification with channel-separated convolutional networks. Proc IEEE Int Conf Comp Vision. 2019:5552–61.
42 Chen JR. Constructing task-level assembly strategies in robot programming by demonstration. Int J Robot Res. 2005;24(12):1073–85.
43 Ekvall S, Kragic D. Robot learning from demonstration: a task-level planning approach. Int J Adv Robot Syst. 2008;5(3):33.
44 Lee K, Su Y, Kim TK, Demiris Y. A syntactic approach to robot imitation learning using probabilistic activity grammars. Robot Auton Syst. 2013;61(12):1323–34.
45• Wang Y, Xiong R, Yu H, et al. Perception of demonstration for automatic programing of robotic assembly: framework, algorithm, and validation. IEEE/ASME T Mech. 2018;23(3):1059–70 **An intelligent perception system was developed in this paper that can generate the task program for the robot to realize programming by demonstration in industrial assembly tasks**.
46. Savarimuthu TR, Buch AG, Schlette C, et al. Teaching a robot the semantics of assembly tasks. IEEE Trans Syst Man Cybern Syst. 2017;48(5):670–92.
47. Rozo L, Jiménez P, Torras C. A robot learning from demonstration framework to perform force-based manipulation tasks. Intell Serv Robot. 2013;6(1):33–51.
48. Norberto Pires J. Robot-by-voice: experiments on commanding an industrial robot using the human voice. Ind Robot. 2005;32(6):505–11.
49. van Delden S, Umrysh M, Rosario C, et al. Pick-and-place application development using voice and visual commands. Ind Robot: An Int J, 2012

50. Tasevski J, Nikolić M, Mišković D. Integration of an industrial robot with the systems for image and voice recognition. Serbian J Electr Eng. 2013;10(1):219–30.
51. Neves J, Serrario D, Pires JN. Application of mixed reality in robot manipulator programming. Ind Robot. 2018;45(6):784–93.
52. Fang HC, Ong SK, Nee AYC. Novel AR-based interface for human-robot interaction and visualization. Adv Manuf. 2014;2(4):275–88.
53. Quintero CP, Li S, Pan MKXJ, et al. Robot programming through augmented trajectories in augmented reality//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2018:1838–44.
54. Rudorfer M, Guhl J, Hoffmann P, et al. Holo Pick'n'Place//2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE. 2018;1:1219–22.
55. Chan W P, Quintero C P, Pan M K X J, et al. A multimodal system using augmented reality, gestures, and tactile feedback for robot trajectory programming and execution//ICRA workshop on robotics in virtual reality 2018. 2018.
56. Ong SK, Yew AWW, Thanigaivel NK, Nee AYC. Augmented reality-assisted robot programming system for industrial applications. Robot Comput Integr Manuf. 2020;61:101820.
57• Makhataeva Z, Varol HA. Augmented Reality for Robotics: A Review. Robotics. 2020;9:21 **A review of AR research in robotics was presented in this paper and some future challenges were pointed out for researchers.**
58. Shen F, Wu W, Yu D, et al. High-precision automated 3-D assembly with attitude adjustment performed by LMTI and vision-based control. IEEE/ASME T Mech. 2014;20(4):1777–89.
59•• Mahler J, Matl M, Satish V, et al. Learning ambidextrous robot grasping policies. Sci Robot. 2019;4(26):eaau4984 **A Dexterity Network was proposed in this paper for robot universal picking using two or more heterogeneous grippers such as vacuum gripper and parallel gripper.**