

Review

# Robot Learning from Demonstration in Robotic Assembly: A Survey

Zuyuan Zhu \*  and Huosheng Hu 

School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; hhu@essex.ac.uk

\* Correspondence: z.zhu@essex.ac.uk; Tel.: +44-1206-874-092

Received: 11 February 2018; Accepted: 12 April 2018; Published: 16 April 2018



**Abstract:** Learning from demonstration (LfD) has been used to help robots to implement manipulation tasks autonomously, in particular, to learn manipulation behaviors from observing the motion executed by human demonstrators. This paper reviews recent research and development in the field of LfD. The main focus is placed on how to demonstrate the example behaviors to the robot in assembly operations, and how to extract the manipulation features for robot learning and generating imitative behaviors. Diverse metrics are analyzed to evaluate the performance of robot imitation learning. Specifically, the application of LfD in robotic assembly is a focal point in this paper.

**Keywords:** learning from demonstration; imitation learning; robotic assembly; machine learning

## 1. Introduction

### 1.1. Robotic Assembly

The industrial robots that are currently deployed in assembly lines are position-controlled and programmed to follow desired trajectories for conducting assembly tasks [1,2]. These position-controlled robots can handle known objects within the well-structured assembly lines very well, achieving highly precise control in position and velocity. However, they cannot deal with any unexpected changes in assembly operations, and need tedious reprogramming to adapt to new assembly tasks.

For instance, Knepper et al. investigated a multi-robot coordinated assembly system for furniture assembly [3]. The geometry of individual parts was listed in a table so that a group of robots can conduct parts delivery or parts assembly collaboratively. For the modeling and recognition of the furniture parts, the object's representation was predefined in CAD files so that the correct assembly sequence can be deduced from geometric data. Suárez-Ruiz and Pham proposed a taxonomy of the manipulation primitives for bi-manual pin insertion, which was only one of the key steps in the autonomous assembly of an IKEA chair [4].

In general, a typical robot-assembly operation involves operating with two or more objects/parts. Each part is a subset of the assembly. The aim of assembly is to compute an ordering of operations that brings individual parts together so that a new product appears. Examples of assembly tasks can be summarized below.

- Peg-in-hole, that is, a robotic gripper grabs the peg and inserts it in a hole. Peg-in-hole is the most essential and representative assembly task that has been widely researched [5–14].
- Slide-in-the-groove, that is, a robot inserts a bolt fitting inside a groove and slides the bolt to the desired position where the bolt is to be fixed [15].
- Bolt screwing, that is, a robot screws a self-tapping bolt into a material of unknown properties, which requires driving a self-tapping screw into an unstructured environment [15–17].

- Chair assembly, that is, a robot integrates chair parts together with a fastener [18,19].
- Pick-and-place, that is, a robot picks up an object as the base and places it down on a fixture [17,20,21].
- Pipe connection, that is, a robot picks and places two union nuts on a tube [17].

As typical robot-assembly operations need to contact the workpieces to be assembled, it is crucial to estimate the accompanying force–torque profiles besides position and orientation trajectories. To learn the execution of assembly operations, a robot needs to estimate the pose of the workpieces first, and an assembly sequence is then generated by learning from human demonstration. For some particular objects appearing in the assembly workspace, some specialized grippers should be designed to grab these parts with various shapes and acquire force–torque data. In particular, during a screwing task, the material to be screwed is unstructured, which makes the control of rotating angles more complicated.

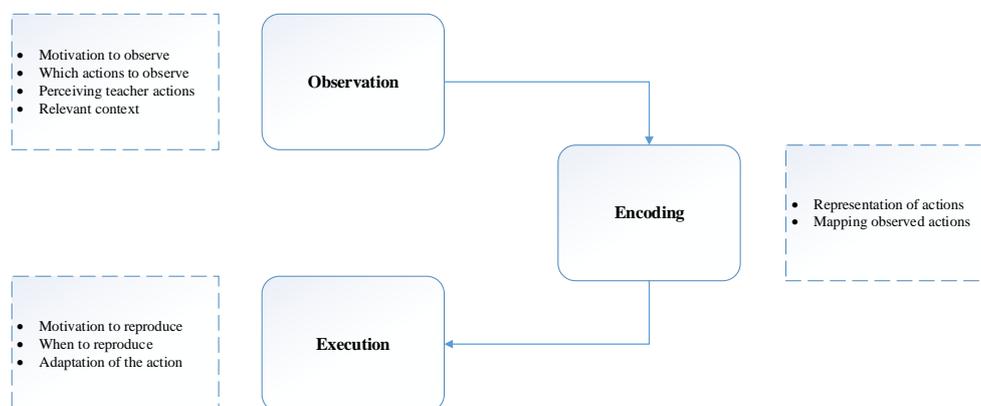
Considering these challenges, robotic assembly remains one of the most challenging problems in the field of robotics research, especially in unstructured environments. In contrast, humans have the excellent skills to perform assembly tasks that require compliance and force control. This motivates us to review the current research of learning from demonstration (LfD) in robotics assembly and its potential future directions.

### 1.2. Learning from Demonstration

Traditional robots require users to have programming skills, which makes the robots beyond the reach of the general public. Nowadays, robotics researchers have worked on a new generation of robots that could learn from demonstration and have no need of programming. In other words, these new robots could perceive human movements using their sensors and reproduce the same actions that humans do. They can be used by the general public who have no programming skills at all.

The term learning from demonstration (LfD), or learning by imitation, was analyzed in depth by Bakker and Kuniyoshi, who defined what imitation is and what robot imitation should be. From a psychological point of view, Thorndike defined imitation as learning to do an act being witnessed [22]. Based on this, Bakker indicated that imitation takes place when an agent learns a behavior from observing the execution of that behavior by a teacher [23]. This was the starting point for establishing the features of robot imitation: (i) adaptation; (ii) efficient communication between the teacher and the learner; (iii) compatibility with other learning algorithms; and (iv) efficient learning in a society of agents [24].

In addition, three processes in robot imitation have been identified, namely sensing, understanding and doing. In other words, they can be redefined as: observe an action, represent the action and reproduce the action. Figure 1 shows these three main issues and all the associated current challenges in robot imitation.



**Figure 1.** The three main phases in imitation learning according to [23].

Mataric et al. defined imitation learning from a biological perspective, that is, a behavior-based control [25]. They indicated that key challenges are how to interpret and understand observed behaviors and how to integrate the perception and motion-control systems to reconstruct what is observed. In other words, there are two basic tasks in imitation learning: (i) to recognize the human behavior from visual input; (ii) to find methods for structuring the motor-control system for general movements and imitation learning capabilities.

Current approaches to represent a skill can be broadly divided into two trends: (i) trajectories encoding—a low-level representation of the skill, taking the form of a nonlinear mapping between sensory and motor information; (ii) symbolic encoding—a high-level representation of the skill that decomposes the skill in a sequence of action–perception units [26]. In general, to achieve robot learning from demonstration, we need to address three challenges: the correspondence problem, generalization, and robustness against perturbation [27]. Firstly, the correspondence problem means how to map links and joints from a human to a robot. Secondly, learning by demonstration is feasible only if a demonstrated movement can be generalized, such as different goal positions. Finally, we need robustness against perturbation: exactly replaying an observed movement is unrealistic in a dynamic environment, in which obstacles may appear suddenly.

Most assembly tasks can be represented as a sequence of individual movements with specific goals, which can be modeled as dynamic movement primitives (DMPs, explained in Section 4.2), where DMPs are the fundamental blocks of the LfD architecture. In addition, LfD has been suggested recently as an effective means to speed up the programming of learning processes from the low-level control to the high-level assembly planning [28]. Therefore, LfD is a preferable approach for robotic assembly.

Recently, LfD has been applied to robotic assembly [29–31]. Takamatsu et al. introduced LfD to robotic assembly and proposed a method for recognizing assembly tasks by human demonstration [32]. They defined sufficient subskills and critical transition-assembly tasks, and implemented a peg-insertion task on a dual-arm robot with a real-time stereo-vision system. The assembly task is completed with two rigid polyhedral objects recognized by a conventional 6-DOF (degree of freedom) object-tracking system. The assembly tasks are encapsulated into chains of two-object relationships, such as maintaining, detaching and constraining. In order to make the process of assembly task smooth, critical transitions are also defined.

The human–robot cooperation in assembly tasks reduces the complexity of impedance control. Rozo et al. worked one step forward to achieve a multimodal LfD framework, in which a robot extracted the impedance-based behavior of the teacher recording both force patterns and visual information in a collaborative table-assembly task [33]. It should be noted that the experiments did not take into account the independence and autonomy of the robot. For the modeling of assembly task, Dantam et al. transferred the human demonstrations into a sequence of semantically relevant object-connection movements [34]. Then, the sequence of movements is further abstracted as motion grammar, which represents the demonstrated task. It should be noted that the assembly task is implemented in simulation.

### 1.3. Outline

Different from the previous survey of learning from demonstration [35], this paper mainly focuses on the applications of LfD techniques in robotic assembly. The rest of the paper is organized as follows. Section 2 outlines the major research problems in robotic assembly, which are classified into four categories. The key issue of how to demonstrate the assembly task to a robot is explained in Section 3, and how to abstract the features of the assembly task is illustrated in Section 4. Then, we examine the question of how to evaluate the performance of the imitator in Section 5. Finally, a brief conclusion and discussion on the open research areas in LfD and robotic assembly are presented in Section 6.

## 2. Research Problems in Robotic Assembly

Robotic assembly needs a high degree of repeatability, flexibility, and reliability to improve the automation performance in assembly lines. Therefore, many specific research problems have to be resolved in order to achieve automated robotic assembly in unstructured environments. The robot software should be able to convert the sequences of assembly tasks into individual movements, estimate the pose of assembly parts, and calculate the required forces and torques. As these are many challenges in robotic assembly, this section will be focused on four categories which are closely related to LfD: pose estimation, force estimation, assembly sequences, and assembly with screwing.

### 2.1. Pose Estimation

In assembly lines, it is often indispensable that the position and orientation of workpieces are predetermined with a high accuracy. The vision-based pose estimation is a low-cost solution to determine the position and orientation of assembly parts based on point cloud data [5]. The texture projector could also be used to acquire high-density point clouds and help the stereo matching process. Abu-Dakka et al. used a 3D camera Kinect for capturing 3D scene data, and the pose of the known objects could be estimated based on the point cloud data [11].

Before pose estimation, an object should be recognized by using local features, as this is an effective way for matching [36]. Choi et al. developed a family of pose-estimation algorithms that use boundary points with directions and boundary line segments along with the oriented surface points to provide high accuracy for a wide class of industrial parts [37]. However, in the cluttered environments where the target objects are placed with self-occlusions and sensor noise, assembly robots require robust vision to reliably recognize and locate the objects. Zeng et al. used a fully convolutional neural network to segment and label multiple views of a scene, and then fit predefined 3D object models to the segmentations to obtain the 6D object pose rather than 3D location [38].

However, the vision-based pose estimation has limitations due to the limited resolution of the vision system. In addition, in the peg-in-hole task, the peg would usually occlude the hole when the robot approaches the hole. Therefore, vision-based pose estimation is not suitable for the high-accuracy assembly tasks in which two parts occlude each other. If the camera is mounted on the robotic arm, the occlude problem can be eliminated, but additional sensory data is needed to estimate the camera pose [39].

To correct the pose of assembly parts, Xiao et al. devised a nominal assembly-motion sequence to collect data from exploratory complaint movements [40]. The data are then used to update the subsequent assembly sequence to correct errors in the nominal assembly operation. Nevertheless, the uncertainty in the pose of the manipulated object should be further addressed in the future research.

### 2.2. Force Estimation

In assembly tasks, force control could provide stable contact between the manipulator and the workpiece [5–7,30,41–45]. As human operators perform compliant motions during assembly, the robot should acquire the contact forces that occur in the process of assembly. During the task execution, the robot learns to replicate the learned forces and torques rather than positions and orientations from the trajectory. The force information may also be used to speed up the subsequent operations of assembly tasks [8,9,12–15,46].

The force applied to the workpiece is usually detected by using the force sensor on the robot end-effector, or using force sensors on each joint of the robot arm, for example, in the DLR (German Aerospace Centre) lightweight arm. The problem of using inside force sensors is that the measured forces must be compensated for disturbance forces (for example, gravity and friction) before use. The feedback force is then introduced to the control system that generates

a corresponding translational/rotational velocity command on the robot manipulator to push the manipulated workpiece.

To enable a robot to interact with the different stiffness, Peternel et al. used the impedance-control interface to teach it some assembly tasks [47]. The human teacher controlled the robot through haptic and impedance-control interfaces. The robot was taught to learn how to perform a slide-in-the-groove assembly task where the robot inserted one part fitted with a bolt into another part. However, the low movement variability did not necessarily correspond to the high impedance force in some assembly tasks such as slide-in-the-groove tasks.

The dedicated force–torque sensors can easily acquire the force, but may not be easily able to mount to the robot hand. Alternatively, Wahrburg et al. deployed motor signals and joint angles to reconstruct the external forces [42]. It should be noted that force/torque estimation is not a problem and has been successful in the traditional robotic assembly in structured environments. However, it is a problem for robotic assembly in unstructured environments. Force/torque estimation is not only about acquiring force/torque data but also about using these data for a robot to accommodate its interaction with the different stiffness.

### 2.3. Assembly Sequence

As an appropriate assembly sequence helps minimize the cost of assembly, the assembly sequences are predefined manually in the traditional robotic-assembly systems. However, the detailed assembly sequence defined in [44] significantly holds back the automation of next-generation assembly lines. To achieve an efficient assembly sequence for a task, an optimization algorithm is required to find the optimum plan. Bahubalendruni et al. found that assembly predicates (that is, some sets of constraints) have a significant influence on optimal assembly-sequence generation [48].

Wan and Harada presented an integrated assembly and motion-planning system to search the assembly sequence with the help of a horizontal surface as the supporting fixture [20]. Kramberger et al. proposed two novel algorithms that learned the precedence constraints and relative part-size constraints [8]. The first algorithm used precedence constraints to generate previously unseen assembly sequences and guaranteed the feasibility of the assembly sequences by learning from human demonstration. The second algorithm learned how to mate the parts by exploratory executions, that is, learning-by-exploration.

Learning assembly sequences from demonstration can be tailored for general assembly tasks [5,8,21,30,31,49]. Mollard et al. proposed a method to learn from the demonstration to automatically detect the constraints between pairs of objects, decompose subtasks of the demonstration, and learn hierarchical assembly tasks [19]. In addition, the learned sequences were further refined by alternating corrections and executions. It should be noticed that the challenge in defining assembly sequence is how to automatically extract the movement primitives and generate a feasible assembly sequence according to the manipulated workpieces.

### 2.4. Assembly with Screwing

Screwing is one of the most challenge subtasks of assembly, and requires robust force control so that the robot could screw a self-tapping bolt into a material of unknown properties. The self-tapping screw driving task consists of two main steps. The first step is to insert the screwdriver into the head of a bolt. The contact stiffness is kept at a constant value such that the screwdriver keeps touching the head of the bolt. The second step is to fit the screwdriver into the screw head and rotate it for a specific angle to actuate the bolt into the unstructured material.

To measure the specific force and angle needed for actuating the screwdriver, Peternel et al. used a human demonstrator to rotate the screwdriver first and captured the correspondence angle by a Haptic Master gimbal device. The information was then mapped to the end-effector rotation of the robot [15]. It should be noted that the torque used to compensate the rotational stiffness is unknown,

so the demonstrator manually commands high rotational stiffness through the stiffness control to make the robot exactly follow the demonstrated rotation.

There are significant uncertainties existing in the screwing task, such as the screwdriver may not catch the head of the bolt correctly. In fact, as the complexity of the task increases, it becomes increasingly common that tasks may be executed with errors. Instead of preventing errors from happening, Laursen et al. proposed a system to automatically handle certain categories of errors through automatic reverse execution to a safe status, from where forward execution can be resumed [16]. The adaptation of an assembly action is essential in the execution phase of LfD, as presented in Figure 1. Besides, the adaptation of uncertainties in robotic assembly still needs further investigation.

In summary, pose estimation, force estimation, assembly sequence, and assembly with screwing have been partially resolved in limited conditions, but are still far away from the industrial application. Most of the current assembly systems are tested in the relatively simple tasks, like peg-in-hole. In addition, a more robust and efficient control strategy is needed to deal with complicated assembly tasks in an unconstructed environment.

### 3. Demonstration Approach

Robot learning from demonstration requires the acquisition of example trajectories, which can be captured in various ways. Alternatively, a robot can be physically guided through the desired trajectory by its operator, and the learned trajectory is recorded proprioceptively for demonstration. This method requires that the robot is back-drivable [50,51] or can compensate for the influences of external forces [52–54]. In the following subsections, we discuss various works that utilize these demonstration techniques.

#### 3.1. Kinesthetic Demonstration

The advantage of kinesthetic guiding is that the movements are recorded directly on the learning robot and do not need to be first transferred from a system with different kinematics and dynamics. During the demonstration movement, the robot's hands are guided by a human demonstrator [55–57]. It should be noted that kinesthetic teaching might affect the acquired forces and torques, especially when joint force sensors are used to estimate forces and torques for controlling of assembly tasks on real robots. In addition, if the manipulated objects are large, far apart, or dangerous to deal with, kinesthetic guiding can be problematic.

Figure 2 shows that the robot was taught through kinesthetics in gravity-compensation mode, that is, by the demonstrator moving its arm through each step of the task. To achieve this, the robot motors were set in a passive mode so that each limb could be moved by the human demonstrator. The kinematics of each joint motion were recorded at a rate by proprioception during the demonstration. The robot was provided with motor encoders for every DOF. By moving its limb, the robot "sensed" its own motion by registering the joint-angle data provided by the motor encoders. The interaction with the robot was more playful than using a graphical simulation, which enabled the user to implicitly feel the robot's limitation in its real-world environment.

In [58], example tasks were provided to the robot via kinesthetic demonstration, in which the teacher physically moved the robot's arm in the zero-gravity mode to perform the task, and used the button on the cuff to set the closure of the grippers. By pushing the button on an arm, the recording began and the teacher started to move the same arm to perform manipulation. When the manipulation was done, the teacher pressed the button again to pause the recording. The teacher simply repeated the steps to continue the manipulation with another hand and the recording. The signals of arm activation and the grippers' state during the demonstration were recorded to segment the tool-use process into sequential manipulation primitives. Each primitive was characterized by using a starting pose, an ending pose of the actuated end-effector, and the sequence of the poses. The primitives are learned via the DMPs framework. These primitives and the sequencing of the primitives constitute the model for tool use.



**Figure 2.** Human demonstrator teaches the Kuka LWR arm to learn peg-in-hole operations by kinesthetic guiding [11].

### 3.2. Motion-Sensor Demonstration

The limb movements of a human demonstrator are very complex and difficult to capture. Computer vision could be used in capturing human-demonstrator motion with a low accuracy [59]. In contrast, the optical or magnetic marker-based tracking systems can achieve high accuracy and avoid visual overlapping of computer vision [60–63]. Therefore, marker-based tracking devices are deployed to track the manipulation movements of a human demonstrator for assembly tasks.

Skoglund et al. presented a method for imitation learning based on fuzzy modelling and a next-state-planner in the learning-from-demonstration framework [64]. A glove with LEDs at its back and a few tactile sensors on each fingertip was used in the impulse motion-capturing system. The LEDs were used to compute the orientation of the wrist, and the tactile sensors were to detect contact with objects. Alternatively, a motion sensor can be used for tracking instead of LEDs.

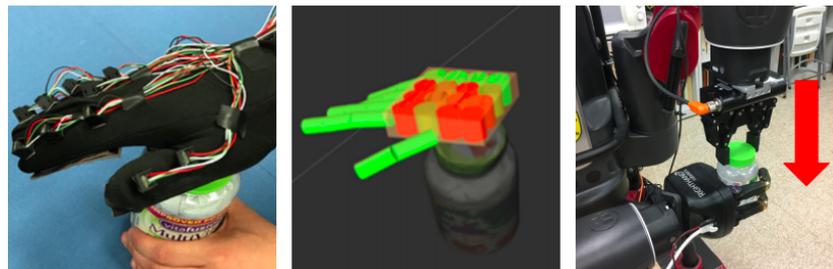
Color markers are a simple and effective motion tracking technique used in motion-sensor demonstration. Acosta-Calderon and Hu proposed a robot-imitation system, in which the robot imitator observed a human demonstrator performing arm movements [65]. The color markers on the human demonstrator were extracted and tracked by the color-tracking system. The obtained information was then used to solve the correspondence problem as described in the previous section. The reference point used to achieve this correspondence is the shoulder of the demonstrator, which corresponded to the base of the robotic arm.

To capture human movement with whole-body motion data, a total of 34 markers were used in a motion-capture setup [66]. During the data-collection process, a sequence of continuous movement data was obtained, for example, a variety of human walking motions, a squat motion, kicking motions and raising arm motions. Some of the motions are discrete, others are continuous. Therefore, the learning system should segment the motions automatically. The segmentation of full-body human-motion patterns was studied in [67]. The human also observed the motion sequence, manually segmented it and then labelled these motions. Note that the motions segmented by the human were set as ground truth, and no further criteria were used.

In Figure 3, a motion sensor mounted in the glove is used for tracking the 6D pose (the position and orientation) relative to the transmitter; the robot then receives a transformed pose on a 1:1 movement scale. The peg-in-hole experiments show that the data glove is inefficient compared with using an external device during the teleoperation process. In Figure 4, both hand-pose and contact forces are measured by a tactile glove. In robotic assembly, the force sensor is essential as the task requires accurate control of force. Therefore, the motion sensor and force sensor are usually combined together.



**Figure 3.** The motion sensor is intergrated in the glove at the back of the hand [68].



**Figure 4.** A tactile glove is used to reconstruct both forces and poses from human demonstrations, enabling the robot to directly observe forces used in demonstrations so that the robot can successfully perform a screwing task: opening a medicine bottle [69].

### 3.3. Teleoperated Demonstration

During the teleoperated demonstration, a human operator uses a control box or delivers hints to control a robot to execute assembly tasks, and the robot keeps recording data from its own sensors, see Figure 5. Similar to the kinesthetic demonstration, the movements performed by the robot are recorded directly on the robot, that is, the mapping is direct, and no corresponding issue exists. Learning from teleoperated demonstrations is an attractive approach to controlling complex robots.

Teleoperation has the advantage of establishing an efficient communication and operation strategy between humans and robots. It has been applied in various applications, including remote control of a mobile robotic assistant [70–72], performing an assembly task [68,73,74], performing a spatial-positioning task [75], demonstrating grasp preshapes to the robot [76], transmitting both dynamic and communicative information on a collaborative task [77], and picking and moving tasks [78].

In assembly tasks, when a human demonstrator performs the assembly motions, the pose information is fed into a real-time tracking system so that the robot can copy the movements of the demonstrator [13]. The Robonaut, a space-capable humanoid robot from NASA, is controlled by a human through full-immersion teleoperation [79]. Its stereo camera and sound sensor transmit vision and auditory information to the teleoperator through a helmet worn on his or her head. Although the full-immersion teleoperation is a good strategy for the Robonaut, its dexterous control is very tedious and tiring.

During teleoperation, the human operator usually manipulates the robot through a controller, standing far away from the robot. In Figure 6, the human teaches the robot to perform the slide-in-the-groove task, as shown in the second column, and then the robot autonomously repeats the learned skill, as shown in the third column. For the bolt-screwing task, as shown in the right photo, DMPs are used to encode the trajectories after demonstrations. Sometimes the controller can be a part of the robot itself. Tanwani et al. applied teleoperation in the robot- learning process where the robot is required to do the tasks of opening/closing a valve and pick–place an object [80]. The human operator held the left arm of the robot and controlled its right arm to receive visual feedback from the camera mounted on the end of the right arm. In the teleoperation demonstration, the left arm of the robot plays the role of controller and the right arm plays as an effector.



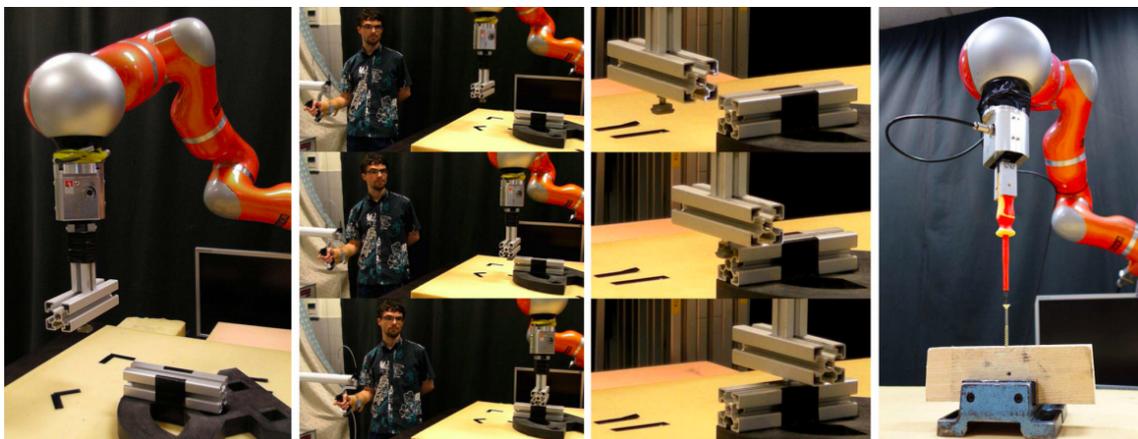
**Figure 5.** Human demonstrator performs the peg-in-hole task, in teleoperation mode, as the robot copies the movements of the human [13].

Delivering hints to robots is another way of teleoperation. Human operators deliver hints to the robot by repeating the desired tasks many times, or by pointing out the important elements of the skill. The hints can be addressed in various ways through the whole learning procedure. One of the hints is vocal deixis, that is, an instruction from a human operator.

Pardowitz et al. used vocal comments given by a demonstrator while demonstrating the task to speed up the learning of robots [81]. The vocal elucidations are integrated into the weight function, which determines the relevance of features consisted in manipulation segments.

Generally speaking, the acoustic of speech consists of three main bits information: the identification of the speaker, the linguistic content of the speech, and the way of speaking. Instead of focusing on the linguistic content of the vocal information, Breazeal et al. proposed an approach to teach the robot how to understand the speaker's affective communicative intent [82]. In the LfD paradigm, understanding of the intention of the human demonstrator through HRI (Human-Robot Interaction) is the key point for the robot to learn. Demonstrations are goal-directed, and the robot is expected to understand the human's intention and extract the goal of the demonstrated examples [83].

The vocal or visual pattern could be used in anthropomorphic robots [84]. The understanding of the human intention is transferred from the standpoint of movement matching [85,86] and joint motion replication [27,56,87–92]. Recent research works believe that robots will need to understand the human's intention as socially cognitive learners, even if the examples are not perfectly demonstrated [93–95]. However, to keep track of intentions to complete desired goals, the imitating robots need a learning method. The solution could be building a cognitive model of the human teacher [96,97], or using simulations of perspective [98].



**Figure 6.** Experimental setups for the slide-in-the-groove assembly task (left and middle photos) and bolt-screwing task (right photo) [15].

#### 4. Feature Extraction

When a dataset of demonstration trajectories (that is, state-action examples) has been collected by using the demonstration approaches mentioned above, we need to consider how to map these data into a mathematical model. There are thousands of position data points distributing along a demonstration trajectory, and no need to record every point, as the movement trajectories are hardly repeatable. In addition, direct replication of the demonstrated trajectories may lead to poor performance because of the limited accuracy of the vision system and uncertainties in the gripping pose. Therefore, learning a policy to extract and generalize the key features of the assembly movements is the fundamental part of LfD.

Hidden Markov models (HMMs) are a popular methodology to encode and generalize the demonstration examples [99–114]. HMM is a robust probabilistic method to encapsulate human motion, which contains spatial and temporal variables, through numerous demonstrations [115]. Initially, the training of HMMs is learned offline, and the sample data are manually classified into groups before learning. To make HMMs become online, Kulic et al. developed adaptive hidden Markov chains for incremental and autonomous learning of the motion patterns, which were extracted into a dynamic stochastic model.

The probabilistic approach can also be integrated with other methods to learn robust models of human motion through imitation. Calinon et al. combined HMM with Gaussian mixture regression (GMR) and dynamical systems to extract redundancies from a set of examples [112]. The original HMMs depend on a fixed number of hidden states, and model the observation as an independent state when segmenting continuous motion. To fix the two major drawbacks of HMMs, Niekum et al. proposed the beta process autoregressive hidden Markov model, in which the modes are easily shared among all models [114].

Based on Gaussian mixture models (GMMs), Chernova and Veloso proposed an interactive policy-learning strategy that reduces the size of training sets by allowing an agent to actively request and effectively represent the most relevant training data [116]. Calinon et al. encoded the motion examples by estimating the optimal GMM, and then the trajectories were generalized through GMR [56]. Tanwani and Calinon extended the semi-tied GMMs for robust learning and adaptation of robot-manipulation tasks by reusing the synergies in various parts of the task sharing similar coordination patterns [80].

Dynamic movement primitives (DMPs) represent a fundamentally different approach to motion representation based on nonlinear dynamic systems. DMP is robust to spatial perturbation and suitable for following a specific movement path. Calinon et al. [112] used DMPs to reproduce the smoothest movement, and the learning process was faster than HMM, TGMR (time-dependent Gaussian mixture regression), LWR (locally weighted regression), and LWPR (locally weighted projection regression). Li and Fritz [58] extended the original DMP formulation by adding a function to define the shape of the movement, which could adapt the movement better to a novel goal position through adjusting the corresponding goal parameter. Ude et al. [92] utilized the available training movements and the task goal to enable the generalization of DMPs to new situations, and able to produce a demonstrated periodic trajectory.

Figure 7 gives an overview of the input–output flow through the complete model, and shows how to extract the feature and how to carry on the whole imitation-learning process. Firstly, the human demonstrator performed demonstration motions  $X$ , then the motions were projected to a latent space using principal component analysis (PCA), in the *what-to-imitate* module. After being reduced of dimensionality, signals  $\zeta$  were temporally aligned through the dynamic time warping (DTW) method. The Gaussian mixture model (GMM) and Bernoulli mixture model (BMM) were then optimized to encode the motion as generalized signals  $\hat{\zeta}$  with the associated time-dependent covariance matrix  $\hat{\Sigma}_s$ .

In the *metric* module, a time-dependent similarity-measurement function was defined, considering the relative weight of each variable and dependencies through the variables included in the optional prior matrix  $P$ . After that, the trajectory of the imitating motion is computed in the *how-to-imitate*

module, aiming at optimizing the metric  $H$ . The trajectory was generated by using the robot's architecture Jacobian matrix  $\hat{J}$ , and the initial position of the object  $O_s$  within the workspace taken into consideration. Finally, the processed data  $\zeta'$  in the latent space is reconstructed to the original data space  $X'$  before the robot imitation.

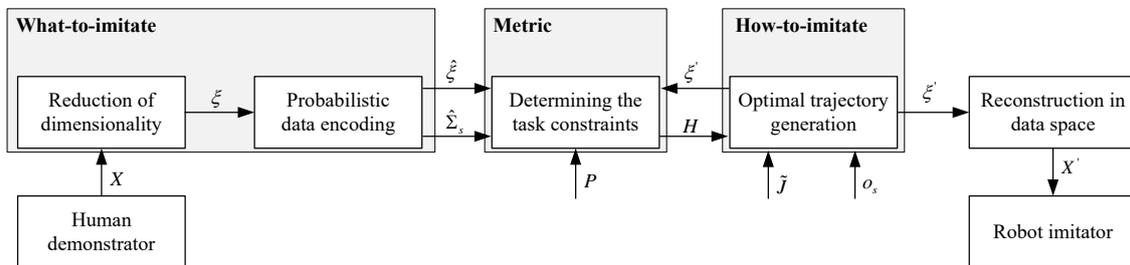


Figure 7. Information flow across the complete system [56].

Three main models used in the feature extraction of LfD will be analyzed in the following subsections, namely the hidden Markov model (HMM), dynamic movement primitives (DMPs), and Gaussian mixture model (GMM).

#### 4.1. Hidden Markov Models

The hidden Markov model (HMM, see Appendix A.1) is a statistical model used to describe a Markov process with unobserved states. An HMM can be presented as the simplest dynamic Bayesian network. The key problem in HMM is to resolve the hidden parameters that are used to do further analyzing (for example, pattern recognition) from the visible parameters. In the normal Markov model, the state is directly visible to the observer, therefore, the transformation probabilities between states are the whole parameters. While the states are not visible in the hidden Markov model, some variables that are influenced by the states are visible. Every state has a probability distribution on the token of the possible output, and therefore the sequence of the output token reveals the information of the state's sequence.

In general, HMM can be simply defined with three tuples as  $\lambda = (A, B, \pi)$ . HMM is the extension of standard Markov model in which the model is updated with observable state sets and the probabilities between the observable states and hidden states. In addition, the temporal variation of the latent representation of the robot's motion can be encoded in an HMM [55]. Trajectories demonstrated by human operators consist of a set of positions  $x$  and velocities  $\dot{x}$ . The joint distribution  $P(x, \dot{x})$  is represented as a continuous HMM of  $K$  states, which are encoded by Gaussian mixture regression (GMR). For each Gaussian distribution of HMM, the centre and covariance matrix are defined by the equation based on positions  $x$  and velocities  $\dot{x}$ .

The influence of different Gaussian distributions is defined by the corresponding weight coefficients  $h_{i,t}$  by taking into consideration the spatial information and the sequential information in the HMM. In a basic HMM, sometimes the states may be unstable and have a poor solution. For this reason, Calinon et al. extended the basic control model by integrating an acceleration-based controller to keep the robot following the learned nonlinear dynamic movements [112]. Kuli'c et al. used HMM to extract the motion sequences tracked by reflective markers located on various joints of the human body [66]. HMM is chosen to model motion movements due to its ability to encapsulate both spatial and temporal variability. Furthermore, HMMs can be used to recognise labeled motions and generate new motions at the same time, because it is a generative model.

#### 4.2. Dynamic Movement Primitives

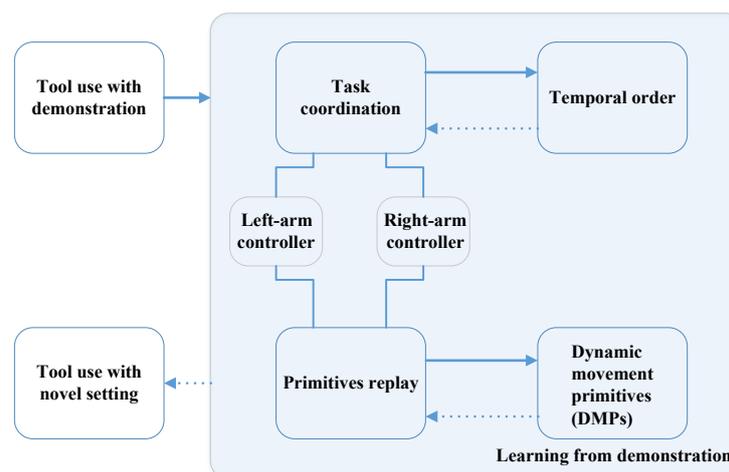
Dynamic movement primitives (DMPs, see Appendix A.2) was originally proposed by Ijspeert et al. [27], and further extended in [58]. The main question of DMPs is how to formulate

dynamic movement equations to flexibly represent the dynamic performance of robotic joint motors, without the need of manually tuning parameters and ensuring the system stability. The desired movement primitives are built on variables representing the kinematic state of the dynamic system, such as positions, velocities, and accelerations. DMP is a kind of high-dimensional control policy and has many favorable features. Firstly, the architecture can be applied to any general movements within the joint's limitation. Secondly, the dynamic model is time-invariant. Thirdly, the system is convergent at the end. Lastly, the temporal and spatial variables can be decoupled. Each DMP has localized generalization of the movement modeled.

DMP is a favorable way to deal with the complex assembly motion by decoupling the motion into a DMP sequence [6,8,31]. Nemeč et al. encoded the desired peg-insertion trajectories into DMPs, where each position/orientation dimension was encoded as an individual DMP [14]. In the self-tapping screw driving task, Peternel et al. collected the commanded position, rotation and stiffness variables, which were used to represent the phase-normalised trajectories [15].

Any demonstrated movements that are observed by the robot are encoded by a set of nonlinear differential equations to represent and reproduce these movements, that is, movement primitives. Based on this framework, a library of movement primitives is built by labeling each encoded movement primitive according to the task classification (for example, approaching, picking, placing, moving, and releasing). To address the correspondence problem, Pastor et al. [27] used resolved motion rate inverse kinematics to map the end-effector position and gripper orientation onto the corresponding motor angles. To pick up an object from a table, the sequence of movement primitives recalled from the library is approaching–picking–moving–placing.

To increase the end-effector's range, Li and Fritz proposed a hierarchical architecture to embed tool use from demonstrations by modelling different subtasks as individual DMPs [58]. In the learning-from-demonstration framework, the temporal order for dual-arm coordination is learned on a higher level, and primitives are learned by constructing DMPs from exemplars on the lower level. The pipeline is shown in Figure 8. The learning process is divided into temporal order at a higher level and DMPs at a lower level. The solid arrow means the process of modeling, while the dashed arrow is for the process of repeating the learned skill on the novel task.



**Figure 8.** The hierarchical architecture of teaching robots the use of human tools in a learning-from-demonstration framework [58].

Compared to an end-to-end trajectory model that queries the inverse kinematic solver for the start and end positions, applying DMPs to every movement primitive significantly improves the success rate for using tools [58]. Although the end-to-end model provides a motor-position solver, the physical constraint of the tool is neglected, leading to a failure manipulation.

### 4.3. Gaussian Mixture Models

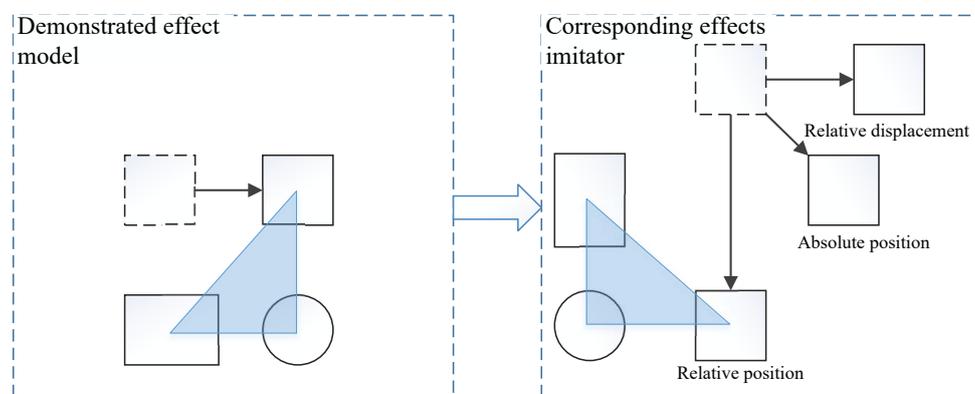
Gaussian mixture models (GMMs, see Appendix A.3) [117] are probabilistic models for clustering and density estimation. As mixture models, they do not need to know which classification a datapoint belongs to. Therefore, the model can learn the classification automatically. A Gaussian mixture model is constituted by two categories of parameters, the mixture component weights, and the component means and variances/covariances.

GMMs are powerful tools for robotic motion modeling, as the approaches are robust to noise. While in high-dimensional spaces and the sample data is noisy or the sample amount is not enough, Gaussian mixture components that have full covariance matrices are confronted with the problem of overfitting the sample datapoints. By decomposing the covariance into two parts—a common latent feature matrix and a component-specific diagonal matrix—the mixture components are then forced to align along a set of common coordination patterns. The semi-tied GMM yields favorable characteristics in the latent space that can be reused in other parts of the learning skill [80]. Combined with expectation maximization, GMM is outperforming many assembly modeling approaches, like gravitational search–fuzzy clustering algorithm (GS–FCA), stochastic gradient boosting (SGB), and classical fuzzy classifier [43].

In summary, HMMs are suitable for modeling motions with both spatial and temporal variability, while DMPs are time-invariant and can be applied to any general movement within the joint’s limitation. GMMs are robust to noise but may lead to overfitting when in high-dimensional spaces or when the sample data is not good enough. For a better performance, HMMs, DMPs, and GMMs are usually combined with other optimization algorithms as presented above.

## 5. Metric of Imitation Performance

Determining a metric of imitation performance is a key element for evaluating LfD. In the execution phase (see Figure 1), the metric is the motivation for the robot to reproduce. Once the metric is set, an optimal controller could be found by minimizing this metric (for example, by evaluating several reproductive attempts or by deriving the metric to find an optimum). The imitation metric plays the role of the cost function or the reward function for reproducing the skill [118]. In other words, the metric of imitation quantitatively translates human intentions during the demonstrations and evaluates the similarity of the robot-repeating performance. In the robotic assembly, the evaluation of the robot’s performance is intuitive, accomplishing the assembly sequence as demonstrated and assembling the individual parts together. However, a specific built-in metric is indispensable if we want to drive and optimize the learning process. Figure 9 shows an example of the use of an imitation metric, and how to use the metric to drive the robot’s reproduction.



**Figure 9.** An example of illustrating how to use three different displacements (relative displacement, absolute position, and relative position) to evaluate a reproduction attempt and find an optimal controller for the reproduction of a task [119].

LfD is a relatively young but rapidly growing research field in which a wide variety of challenges have been addressed. The most intuitive metric of evaluation is to minimize the difference between the observed motion repeated by the robot and the teaching motion demonstrated by the human teacher [120–122]. However, there exists little direct comparison between different feature-extracting models of LfD currently, since the evaluation is constrained to the specific learning task and robotic platform. LfD needs a set of unified evaluation metrics to compare the different imitation systems. The existing approaches mainly consider the variance and correlation information of joint angles, trajectory paths, and object–hand relation.

### 5.1. Weighted Similarity Measure

Taking only the position of the trajectory into consideration, a Euclidean distance measure can be defined as:

$$\sum_{i=1}^{D-1} \omega_i (\zeta_{s,i} - \hat{\zeta}_{s,i})^2 = (\zeta_s - \hat{\zeta}_s)^T W (\zeta_s - \hat{\zeta}_s), \quad (1)$$

where  $\zeta_s$  is the candidate position of the trajectory reproduced by the robot,  $\hat{\zeta}_s$  is the desired position of the optimum trajectory, and both position points have the same dimensionality ( $D - 1$ ). It should be noted that the optimum trajectory is not equal to the demonstrated trajectory, as the body schema of the human and robot are very different.  $W$  is the time-dependent matrix with dimensionality of  $(D - 1) \times (D - 1)$ . The matrix's diagonal variables  $\omega_i$  are the weights defining the influence of each corresponding point. Generally,  $W$  is a full covariance matrix, which represents the correlations across the different variables.

### 5.2. Generic Similarity Measure

Generic similarity measure  $H$  is a general formalism for evaluating the reproduction of a task, proposed in [123]. Compared to the weighted similarity measure of Euclidean distance defined in Equation (1), the similarity measure  $H$  takes into account more variables, such as the variations of constraints and the dependencies across the variables. It should be noted that the matrix is continuous, positive, and is estimable at any point along the trajectory. In the latent joint space, given the generalized joint angle trajectories  $\hat{\zeta}_s^\theta$ , the generalized hand paths  $\hat{\zeta}_s^x$ , and the generalized hands–object distance vectors  $\hat{\zeta}_s^y$ , which are obtained from the demonstrated examples, the generic similarity measure  $H$  is defined as:

$$H = (\zeta_s^\theta - \hat{\zeta}_s^\theta)^T W^\theta (\zeta_s^\theta - \hat{\zeta}_s^\theta) + (\zeta_s^x - \hat{\zeta}_s^x)^T W^x (\zeta_s^x - \hat{\zeta}_s^x) + (\zeta_s^y - \hat{\zeta}_s^y)^T W^y (\zeta_s^y - \hat{\zeta}_s^y), \quad (2)$$

where  $\{\zeta_s^\theta, \zeta_s^x, \zeta_s^y\}$  represent the candidate trajectories for repeating the movements.

### 5.3. Combination of Metrics

Calinon et al. used five metrics to evaluate a reproduction attempt  $x' \in \mathbb{R}^{(D \times T)}$  reproduced from the demonstrated example set  $x \in \mathbb{R}^{(D \times M \times T)}$  [112]. The last two metrics consider the computation time of the learning and retrieval process, which partially depends on the performance of the central processing unit, so here, only the other three metrics will be introduced.

$\mathcal{M}_1$ : The metric evaluates the spatial and temporal information of the reproduced motion, where a root-mean-square (RMS) error is calculated based on the position difference:

$$\mathcal{M}_1 = \frac{1}{MT} \sum_{m=1}^M \sum_{t=1}^T \|x'_t - x_{m,t}\|, \quad (3)$$

where  $M$  is the number of demonstrations and  $T$  is the moment along the demonstrating processes.

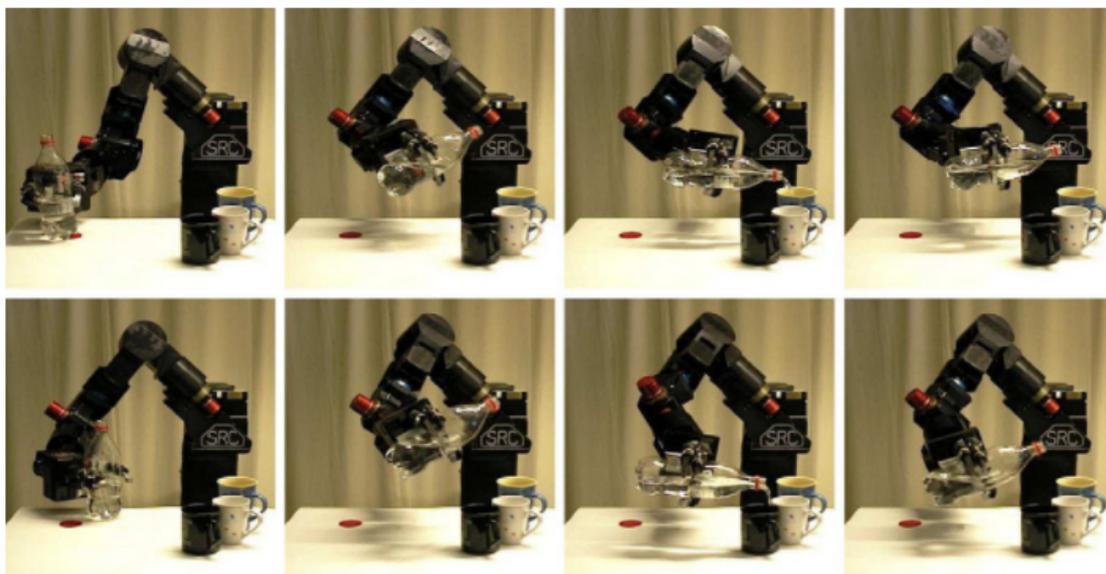
$\mathcal{M}_2$ : In this metric, the imitated motion is first temporally aligned with the demonstrations through dynamic time warping (DTW) [115], and then an RMS based on the position difference similar to  $\mathcal{M}_1$  is calculated. However, not like  $\mathcal{M}_1$ , spatial information has more priorities here, that is, the metric compares the whole path instead of the exact trajectory along time.

$\mathcal{M}_3$ : This metric considers the smoothness of the imitated motion by calculating the derivation of the acceleration extracted from the motion:

$$\mathcal{M}_3 = \frac{1}{T} \sum_{t=1}^T \|\ddot{x}'_t\|. \quad (4)$$

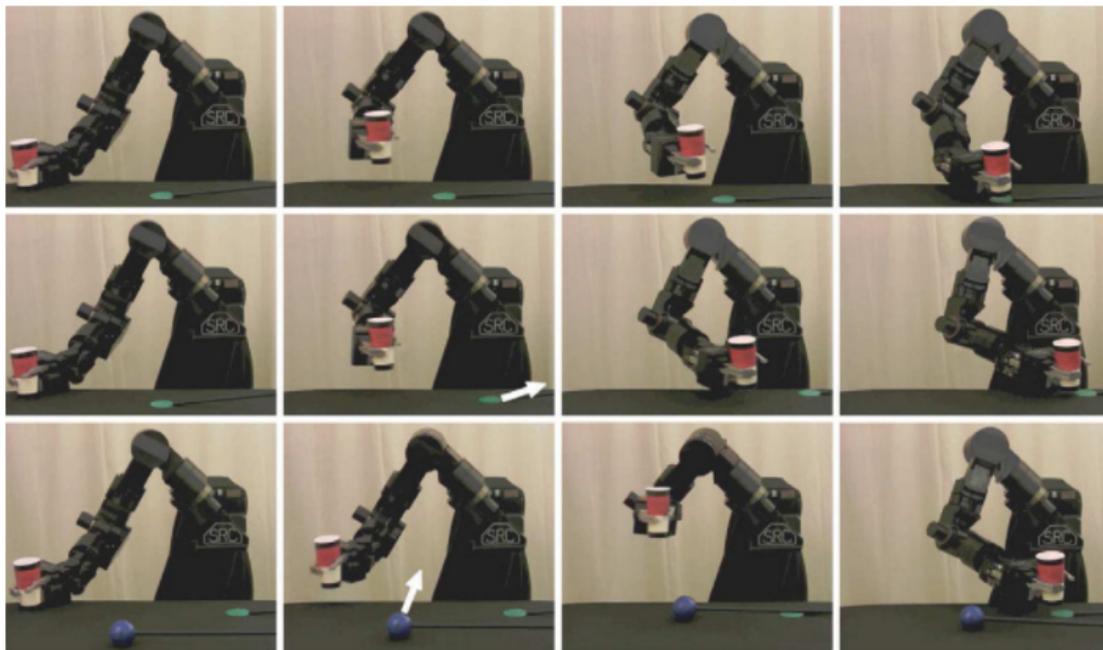
The smoothness is very useful, especially for evaluating the transition between different movement primitives.

Calinon et al. also used the above three metrics to evaluate the stability of the imitation system by superposing random force along the motion. The combination of the metrics provides a comprehensive evaluation of the imitation learning system. Most of the current imitative systems are evaluated through completing particular tasks as demonstrated by the teachers. Pastor et al. demonstrated the utility of DMPs in a robot demonstration of water-serving [27], as shown in Figure 10. Firstly, a human operator demonstrates the pouring task, including grasping, pouring, retreating bottle and releasing movements. Secondly, the robot extracts the movement primitives from the observed demonstrations and adds the primitives to the motion library. Thirdly, the experimental environment (water and cups) is prepared. Fourthly, the sequence of movement primitives is manually determined. Fifthly, appropriate goals are assigned to each DMP. Finally, the robot reproduces the demonstrated motion with the determined sequence of primitives and learns to apply the skill to new goal positions by adjusting the goal of the pouring movement.



**Figure 10.** Example of the movement reproduction and generalization in a new environment with the Sarcos Slave Arm. The top row shows that the robot executes a pouring task by pouring water into the inner cup. The bottom row is the reproduction of a pouring task with a new goal, the outer cup [27].

To demonstrate the framework's ability to adapt online to new goals, as well as to avoid obstacles, Pastor et al. extended the experimental setup with a stereo camera system. The robot needed to adapt movements to goals that changed their position during the robot's movement, as shown in Figure 11.



**Figure 11.** Example of placing a red cup on a green coaster. The top row shows that the robot places the cup in a fixed position. The middle row shows that the robot places the cup on a goal position which changes location during placing. The bottom row shows that the robot places the cup with the same goal as the middle row, while accounting for the interference of a blue ball [27].

## 6. Conclusions and Discussion

This article presents a comprehensive survey of learning-from-demonstration (LfD) approaches, with a focus on their applications in robotic assembly. The demonstration approaches are divided into three categories, in which the characters of each approach have been analyzed, and the theory behind the feature extraction is reviewed. The extraction is then segmented into three categories according to how the demonstration is modeled in the robot's program. Within these models, dynamic movement primitive (DMP) is highlighted for its favorable features of formalizing nonlinear dynamic movements in robotic assembly. Next, the metric of imitation performance is used as a cost function for reproducing the learning skill. The application of LfD in robotic assembly is clearly analyzed, in particular, how LfD facilitates the accomplishment of assembly tasks.

LfD has the unique advantage of enabling the general public to use the robot without the need of learning programming skills. Additionally, kinesthetic demonstration of LfD solves the correspondence problem between human demonstrators and robots. Consequently, LfD is an effective learning algorithm for robots, and has been used in many robotic-learning systems. There are several promising areas of LfD, ranging from insufficient data to incremental learning and effective demonstration, to be further investigated in the future. More specifically:

- Learning from insufficient data. LfD aims at providing non-experts an easy way to teach robots practical skills, although usually, the quantity of demonstrations is not numerous. However, the demonstration in robotic assembly may contain noise. Due to the lack of some movement features and the intuitive nature of interacting with human demonstrators, it becomes hard for the non-expert users to use LfD. Requiring non-experts to demonstrate one movement in a repetitive way is not a good solution. Future research work on how to generalize through a limited number of feature samples is needed.
- Incremental learning. The robot can learn a skill from a demonstrator, or learn several skills from different demonstrations. The study on incremental learning is still very limited during the past

research. The skills that the robot has learned are parallel, not progressive or incremental. DMPs are fundamental learning blocks that can be used to learn more advanced skills, while these skills cannot be used to learn more complicated skills. Incremental learning features should be given more attention for robotic assembly in the future research.

- **Effective demonstration.** When the demonstrator executes any assembly actions, the robot tries to extract the features from the demonstration. In most cases, the learning process is unidirectional, lacking timely revision, leading to less effective learning. The most popular approaches adopted in LfD systems are reward functions. However, the reward functions only give the evaluation of the given state, and no desirable information on which demonstration can be selected as the best example. One promising solution is that the human demonstrator provides timely feedback (for example, through a GUI [19]) on the robot's actions. More research on how to provide such effective feedback information is another aspect of future work.
- **Fine assembly.** Robotics assembly aims at enormously promoting industry productivity and helping workers on highly repeated tasks, especially in 'light' industries, such as the assembly of small parts. The robots have to be sophisticated enough to handle more complicated and more advanced tasks instead of being limited to the individual subskills of assembly, such as inserting, rotating, screwing and so on. Future research work on how to combine the subskills into smooth assembly skills is desired.
- **Improved evaluation.** A standardized set of evaluation metrics is a fundamentally important research area for future work. Furthermore, improved evaluation metrics help the learning process of imitation by providing a more accurate and effective goal in LfD. The formalization of evaluation criteria would also facilitate the research and development of the extended general-purpose learning systems in robotic assembly.

**Acknowledgments:** Zuyuan Zhu is financially supported by China Scholarship Council (CSC) and Essex University Scholarship for his PhD study.

**Author Contributions:** All authors have made substantial contributions to this survey. Huosheng Hu supervised the research and provided suggestions on structuring the survey. Zuyuan Zhu analyzed and organised the methods presented in this paper and wrote the survey. All authors discussed and commented on the manuscript at all stages.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

### Appendix A.1. HMM

HMM can be described by five elements, namely two state sets and three probability matrices:

- **Hidden state  $S$**   
The states (for example,  $S_1, S_2, S_3, \dots$ ) are the actual hidden states in HMM which satisfy the Markov characteristics and cannot be directly observed.
- **Observable state  $O$**   
The observable state  $O$  is associated with the hidden state and can be directly observed. (For example,  $O_1, O_2, O_3$  and so on, the number of observable states is not necessarily the same as hidden states.)
- **Initial state probability matrix  $\Pi$**   
 $\Pi$  is the probability matrix of hidden state at the initial moment  $t = 1$ . (For example, given  $t = 1$ ,  $P(S_1) = p_1, P(S_2) = p_2, P(S_3) = p_3$ , then the initial state probability matrix  $\Pi = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}$ .)
- **Hidden state transition probability matrix  $A$**   
Matrix  $A$  defines the transition probabilities between different states of HMM, where  $A_{ij} = P(S_j|S_i), 1 \leq i, j \leq N$ , which means given the time  $t$  and state  $S_i$ , the state is  $S_j$  with probability  $P$  at time  $t + 1$ .

- Observable state transition probability matrix  $B$

Assume that  $N$  is the number of hidden states, and  $M$  is the number of observable states, then:  $B_{ij} = P(O_i|S_j), 1 \leq i \leq M, 1 \leq j \leq N$ , which means that given the time  $t$  and hidden state  $S_j$ , the observed state is  $O_i$  with probability  $P$ .

### Appendix A.2. DMP

In a basic point movement system, the discrete DMP motion can be represented by the following formulations:

$$\tau \dot{v} = K(g - x) - Dv - K(g - x_0)s + Kf(s), \tag{A1}$$

$$\tau \dot{x} = v, \tag{A2}$$

where  $g$  is the desired goal position;  $x_0$  is the initial start position;  $x$  and  $v$  are position and velocity of the DMP motion;  $t$  is the temporal scaling factor which determines the duration of the movement;  $K$  is the spring constant; and  $D$  is a damping term. The nonlinear function  $f$ , which changes the rather trivial exponential and monotonic convergence of the position  $x$  towards goal position  $g$ , is defined as

$$f(s) = \frac{\sum_i \psi_i(s) \theta_i s}{\sum_i \psi_i(s)}, \tag{A3}$$

where  $\psi_i(s) = \exp(-h_i(s - c_i)^2)$  are Gaussian basis functions characterized by a centre  $c_i$  and bandwidth  $h_i$ ;  $\theta_i$  is the adjustable parameter which differs one DMP from another.  $s$  is a 'phase' value which monotonically decreases from the start '1' to the end '0' of the movement and is generated by a canonical system which is among the most basic dynamic systems used to formulate a point attractor:

$$\tau \dot{s} = -\alpha s, \tag{A4}$$

where  $\alpha$  is a known time constant;  $t$  is the temporal scaling factor as above.

To encapsulate a primitive movement into a DMP, the kinematic variables such as position, velocity, and acceleration are computed for each time-step using the recorded movement trajectory. Then, the corresponding coefficients are deduced according to the above Equations (A1)–(A4), and represent different DMPs. The discrete DMP then generates various motions with specific sequence of primitives.

### Appendix A.3. GMM

For a dataset of  $N$  datapoints  $\{\xi_j\}_{j=1}^N$  with  $\xi_j \in \mathbb{R}^D$ , which can be either joint angles, hand paths, or hand-object distance vectors, a Gaussian mixture model with a weighted sum of  $M$  components  $\{C_i\}_{i=1}^M$  is defined by a probability density function:

$$p(\xi_j|\lambda) = \sum_{i=1}^M \omega_i g(\xi_j|\mu_i, \Sigma_i), \tag{A5}$$

where  $\xi_j$  is a  $D$ -dimensional data vector,  $\omega_i, i = 1, \dots, M$ , are the mixture component weights of component  $C_i$ , with the constraint that  $\sum_{i=1}^M \omega_i = 1$ , and  $g(\xi_j|\mu_i, \Sigma_i), i = 1, \dots, M, j = 1, \dots, N$  are the component Gaussian densities. For each component  $C_i$ , the Gaussian density is defined as:

$$g(\xi_j|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\xi_j - \mu_i)' \Sigma_i^{-1} (\xi_j - \mu_i)\right\}, \tag{A6}$$

where  $\mu_i$  are the component means,  $\Sigma_i$  are the covariance matrices.  $\lambda = \{\omega_i, \mu_i, \Sigma_i\}_{i=1}^M$  are the set of parameters to be estimated in the density function; these parameters define a Gaussian mixture model.  $\lambda$  is usually estimated by maximum likelihood estimation using the standard expectation maximization (EM) algorithm [124].

## References

1. Krüger, J.; Lien, T.K.; Verl, A. Cooperation of human and machines in assembly lines. *CIRP Ann.-Manuf. Technol.* **2009**, *58*, 628–646.
2. Hu, S.; Ko, J.; Weyand, L.; ElMaraghy, H.; Lien, T.; Koren, Y.; Bley, H.; Chryssolouris, G.; Nasr, N.; Shpitalni, M. Assembly system design and operations for product variety. *CIRP Ann.* **2011**, *60*, 715–733, doi:10.1016/j.cirp.2011.05.004.
3. Knepper, R.A.; Layton, T.; Romanishin, J.; Rus, D. IkeaBot: An autonomous multi-robot coordinated furniture assembly system. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 855–862. doi:10.1109/icra.2013.6630673.
4. Suárez-Ruiz, F.; Pham, Q.C. A framework for fine robotic assembly. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 421–426, doi:10.1109/icra.2016.7487162.
5. Yang, Y.; Lin, L.; Song, Y.; Nemeč, B.; Ude, A.; Buch, A.G.; Krüger, N.; Savarimuthu, T.R. Fast programming of Peg-in-hole Actions by human demonstration. In Proceedings of the 2014 International Conference on Mechatronics and Control (ICMC), Jinzhou, China, 3–5 July 2014; pp. 990–995, doi:10.1109/icmc.2014.7231702.
6. Kramberger, A.; Gams, A.; Nemeč, B.; Chrysostomou, D.; Madsen, O.; Ude, A. Generalization of orientation trajectories and force–torque profiles for robotic assembly. *Robot. Auton. Syst.* **2017**, *98*, 333–346. doi:10.1016/j.robot.2017.09.019.
7. Kim, Y.L.; Song, H.C.; Song, J.B. Hole detection algorithm for chamferless square peg-in-hole based on shape recognition using F/T sensor. *Int. J. Precis. Eng. Manuf.* **2014**, *15*, 425–432, doi:10.1007/s12541-014-0353-6.
8. Kramberger, A.; Piltaver, R.; Nemeč, B.; Gams, M.; Ude, A. Learning of assembly constraints by demonstration and active exploration. *Ind. Robot Int. J.* **2016**, *43*, 524–534. doi:10.1108/ir-02-2016-0058.
9. Nottensteiner, K.; Sagardia, M.; Stemmer, A.; Borst, C. Narrow passage sampling in the observation of robotic assembly tasks. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 130–137, doi:10.1109/icra.2016.7487125.
10. Jain, R.K.; Majumder, S.; Dutta, A. SCARA based peg-in-hole assembly using compliant IPMC micro gripper. *Robot. Auton. Syst.* **2013**, *61*, 297–311, doi:10.1016/j.robot.2012.12.001.
11. Abu-Dakka, F.J.; Nemeč, B.; Kramberger, A.; Buch, A.G.; Krüger, N.; Ude, A. Solving peg-in-hole tasks by human demonstration and exception strategies. *Ind. Robot Int. J.* **2014**, *41*, 575–584, doi:10.1108/ir-07-2014-0363.
12. Tang, T.; Lin, H.C.; Zhao, Y.; Fan, Y.; Chen, W.; Tomizuka, M. Teach industrial robots peg-hole-insertion by human demonstration. In Proceedings of the 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, 12–15 July 2016; pp. 488–494, doi:10.1109/aim.2016.7576815.
13. Savarimuthu, T.R.; Buch, A.G.; Schlette, C.; Wantia, N.; Rossmann, J.; Martinez, D.; Alenya, G.; Torras, C.; Ude, A.; Nemeč, B. Teaching a Robot the Semantics of Assembly Tasks. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, 1–23, doi:10.1109/tsmc.2016.2635479.
14. Nemeč, B.; Abu-Dakka, F.J.; Ridge, B.; Ude, A.; Jorgensen, J.A.; Savarimuthu, T.R.; Jouffroy, J.; Petersen, H.G.; Krüger, N. Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile. In Proceedings of the 16th International Conference on Advanced Robotics (ICAR), Montevideo, Uruguay, 25–29 November 2013; pp. 1–7, doi:10.1109/icar.2013.6766568.
15. Peternel, L.; Petrič, T.; Babič, J. Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation. *Autonom. Robots* **2018**, *42*, 1–17.
16. Laursen, J.S.; Schultz, U.P.; Ellekilde, L.P. Automatic error recovery in robot-assembly operations using reverse execution. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1785–1792, doi:10.1109/iros.2015.7353609.
17. Laursen, J.S.; Ellekilde, L.P.; Schultz, U.P. Modelling reversible execution of robotic assembly. *Robotica* **2018**, 1–30, doi:10.1017/s0263574717000613.
18. Dogar, M.; Spielberg, A.; Baker, S.; Rus, D. Multi-robot grasp planning for sequential assembly operations. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 193–200, doi:10.1109/icra.2015.7138999.

19. Mollard, Y.; Munzer, T.; Baisero, A.; Toussaint, M.; Lopes, M. Robot programming from demonstration, feedback and transfer. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1825–1831, doi:10.1109/iros.2015.7353615.
20. Wan, W.; Harada, K. Integrated assembly and motion planning using regrasp graphs. *Robot. Biomim.* **2016**, *3*, 18. doi:10.1186/s40638-016-0050-2.
21. Lambrecht, J.; Kleinsorge, M.; Rosenstrauch, M.; Krüger, J. Spatial programming for industrial robots through task demonstration. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 254, doi:10.5772/55640.
22. Thorndike, E.L. *Animal Intelligence. Experimental Studies*; Macmillan: London, UK, 1911; Volume 39, p. 357, doi:10.1097/00005053-191205000-00016.
23. Bakker, P.; Kuniyoshi, Y. Robot see, robot do: An overview of robot imitation. In Proceedings of the AISB96 Workshop on Learning in Robots and Animals, Kobe, Japan, 12–17 May 2009; pp. 3–11.
24. Rozo, L.; Jiménez, P.; Torras, C. A robot learning-from-demonstration framework to perform force-based manipulation tasks. *Intell. Serv. Robot.* **2013**, *6*, 33–51, doi:10.1007/s11370-012-0128-9.
25. Mataric, M.J. Getting humanoids to move and imitate. *IEEE Intell. Syst. Their Appl.* **2000**, *15*, 18–24, doi:10.1109/5254.867908.
26. Billard, A.; Calinon, S.; Dillmann, R.; Schaal, S. Robot Programming by Demonstration. In *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1371–1394, doi:10.1007/978-3-540-30301-5\_60.
27. Pastor, P.; Hoffmann, H.; Asfour, T.; Schaal, S. Learning and generalization of motor skills by learning from demonstration. In Proceedings of the ICRA'09. IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 763–768, doi:10.1109/robot.2009.5152385.
28. Krüger, N.; Ude, A.; Petersen, H.G.; Nemeč, B.; Ellekilde, L.P.; Savarimuthu, T.R.; Rytz, J.A.; Fischer, K.; Buch, A.G.; Kraft, D. Technologies for the fast set-up of automated assembly processes. *KI-Künstliche Intell.* **2014**, *28*, 305–313, doi:10.1007/s13218-014-0329-9.
29. Aleotti, J.; Caselli, S.; Reggiani, M. Toward programming of assembly tasks by demonstration in virtual environments. In Proceedings of the 12th IEEE International Workshop on Robot and Human Interactive Communication, Millbrae, CA, USA, 2 November 2003; pp. 309–314, doi:10.1109/roman.2003.1251863.
30. Suomalainen, M.; Kyrki, V. A geometric approach for learning compliant motions from demonstration. In Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots, Birmingham, UK, 15–17 November 2017; pp. 783–790. doi:10.1109/humanoids.2017.8246961.
31. Stenmark, M.; Topp, E.A. From demonstrations to skills for high-level programming of industrial robots. In *AAAI Fall Symposium Series: Artificial Intelligence for Human-Robot Interaction*; AAAI: Arlington, VA, USA, 2016; pp. 75–78.
32. Takamatsu, J.; Ogawara, K.; Kimura, H.; Ikeuchi, K. Recognizing assembly tasks through human demonstration. *Int. J. Robot. Res.* **2007**, *26*, 641–659.
33. Rozo, L.; Calinon, S.; Caldwell, D.; Jiménez, P.; Torras, C. Learning collaborative impedance-based robot behaviors. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, Washington, USA, 11–18 July 2013; Volume 1, p. 1.
34. Dantam, N.; Essa, I.; Stilman, M. Linguistic transfer of human assembly tasks to robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 237–242, doi:10.1109/iros.2012.6385749.
35. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A survey of robot learning from demonstration. *Robot. Auton. Syst.* **2009**, *57*, 469–483, doi:10.1016/j.robot.2008.10.024.
36. Liu, Q.; Li, R.; Hu, H.; Gu, D. Extracting semantic information from visual data: A survey. *Robotics* **2016**, *5*, 8, doi:10.3390/robotics5010008.
37. Choi, C.; Taguchi, Y.; Tuzel, O.; Liu, M.Y.; Ramalingam, S. Voting-based pose estimation for robotic assembly using a 3D sensor. 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 1724–1731, doi:10.1109/icra.2012.6225371.
38. Zeng, A.; Yu, K.T.; Song, S.; Suo, D.; Walker, E.; Rodriguez, A.; Xiao, J. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, Singapore, 29 May–3 June 2017; pp. 1386–1383, doi:10.1109/icra.2017.7989165.

39. Schmitt, R.; Cai, Y. Recognition of dynamic environments for robotic assembly on moving workpieces. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 1359–1369, doi:10.1007/s00170-013-5455-2.
40. Sarić, A.; Xiao, J.; Shi, J. Reducing uncertainty in robotic surface assembly tasks based on contact information. In Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), Evanston, IL, USA, 11–13 September 2014; pp. 94–100, doi:10.1109/arso.2014.7020987.
41. Likar, N.; Nemec, B.; Žlajpah, L.; Ando, S.; Ude, A. Adaptation of bimanual assembly tasks using iterative learning framework. In Proceedings of the IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, South Korea, 3–5 November 2015; pp. 771–776, doi:10.1109/humanoids.2015.7363457.
42. Wahrburg, A.; Zeiss, S.; Matthias, B.; Ding, H. Contact force estimation for robotic assembly using motor torques. In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), Taipei, Taiwan, 18–22 August 2014; pp. 1252–1257, doi:10.1109/coase.2014.6899487.
43. Jasim, I.F.; Plapper, P.W. Contact-state modeling of robotic assembly tasks using Gaussian mixture models. *Procedia CIRP* **2014**, *23*, 229–234, doi:10.1016/j.procir.2014.10.061.
44. Stolt, A.; Linderoth, M.; Robertsson, A.; Johansson, R. Force controlled robotic assembly without a force sensor. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 1538–1543. doi:10.1109/icra.2012.6224837.
45. Suomalainen, M.; Kyrki, V. Learning compliant assembly motions from demonstration. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, 9–14 October 2016; pp. 871–876. doi:10.1109/iros.2016.7759153.
46. Navarro-Gonzalez, J.; Lopez-Juarez, I.; Rios-Cabrera, R.; Ordaz-Hernández, K. On-line knowledge acquisition and enhancement in robotic assembly tasks. *Robot. Comput.-Integr. Manuf.* **2015**, *33*, 78–89, doi:10.1016/j.rcim.2014.08.013.
47. Peternel, L.; Petric, T.; Babic, J. Human-in-the-loop approach for teaching robot assembly tasks using impedance-control interface. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1497–1502, doi:10.1109/icra.2015.7139387.
48. Bahubalendruni, M.R.; Biswal, B.B.; Kumar, M.; Nayak, R. Influence of assembly predicate consideration on optimal assembly sequence generation. *Assem. Autom.* **2015**, *35*, 309–316, doi:10.1108/aa-03-2015-022.
49. Liang, Y.S.; Pellier, D.; Fiorino, H.; Pesty, S. Evaluation of a Robot Programming Framework for Non-Experts using Symbolic Planning Representations. In Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication, Lisbon, Portugal, 28 August 2017, doi:10.1109/roman.2017.8172444.
50. Hersch, M.; Guenter, F.; Calinon, S.; Billard, A. Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations. *IEEE Trans. Robot.* **2008**, *24*, 1463–1467, doi:10.1109/tro.2008.2006703.
51. Peters, J.; Schaal, S. Reinforcement learning of motor skills with policy gradients. *Neural Netw.* **2008**, *21*, 682–697, doi:10.1016/j.neunet.2008.02.003.
52. Mistry, M.; Mohajerian, P.; Schaal, S. An exoskeleton robot for human arm movement study. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 4071–4076, doi:10.1109/iros.2005.1545450.
53. Albu-Schäffer, A.; Ott, C.; Hirzinger, G. A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *Int. J. Robot. Res.* **2007**, *26*, 23–39, doi:10.1177/0278364907073776.
54. Hyon, S.H.; Hale, J.G.; Cheng, G. Full-Body Compliant Human–Humanoid Interaction: Balancing in the Presence of Unknown External Forces. *IEEE Trans. Robot.* **2007**, *23*, 884–898, doi:10.1109/tro.2007.904896.
55. Calinon, S.; Guenter, F.; Billard, A. On learning the statistical representation of a task and generalizing it to various contexts. In Proceedings of the International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; Volume 37, pp. 2978–2983, doi:10.1109/robot.2006.1642154.
56. Calinon, S.; Guenter, F.; Billard, A. On Learning, Representing, and Generalizing a Task in a Humanoid Robot. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2007**, *37*, 286–298, doi:10.1109/tsmcb.2006.886952.
57. Ye, G.; Alterovitz, R. Demonstration-Guided Motion Planning. *Springer Tracts Adv. Robot.* **2017**, 291–307, doi:10.1007/978-3-319-29363-9\_17.
58. Li, W.; Fritz, M. Teaching robots the use of human tools from demonstration with non-dexterous end-effectors. In Proceedings of the IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, South Korea, 3–5 November 2015; pp. 547–553, doi:10.1109/humanoids.2015.7363586.

59. Moeslund, T.B.; Hilton, A.; Krüger, V. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.* **2006**, *104*, 90–126, doi:10.1016/j.cviu.2006.08.002.
60. Riley, M.; Ude, A.; Atkeson, C.G. Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching. In Proceedings of the Workshop Interactive Robot and Entertainment, Pittsburgh: PA, USA, 30 April–1 May 2000; pp. 35–42.
61. Pollard, N.S.; Hodgins, J.K.; Riley, M.J.; Atkeson, C.G. Adapting human motion for the control of a humanoid robot. In Proceedings of the ICRA'02 IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 2, pp. 1390–1397, doi:10.1109/robot.2002.1014737.
62. Ude, A.; Atkeson, C.G.; Riley, M. Programming full-body movements for humanoid robots by observation. *Robot. Auton. Syst.* **2004**, *47*, 93–108, doi:10.1016/j.robot.2004.03.004.
63. Ruchanurucks, M.; Nakaoka, S.; Kudoh, S.; Ikeuchi, K. Humanoid robot motion generation with sequential physical constraints. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, Orlando, FL, USA, 2006, 15–19 May 2006; pp. 2649–2654, doi:10.1109/robot.2006.1642101.
64. Skoglund, A.; Iliev, B.; Palm, R. Programming-by-Demonstration of reaching motions—A next-state-planner approach. *Robot. Auton. Syst.* **2010**, *58*, 607–621.
65. Acosta-Calderon, C.A.; Hu, H. Robot imitation: Body schema and body percept. *Appl. Bionics Biomech.* **2005**, *2*, 131–148, doi:10.1533/abbi.2004.0043.
66. Kulić, D.; Ott, C.; Lee, D.; Ishikawa, J.; Nakamura, Y. Incremental learning of full body motion primitives and their sequencing through human motion observation. *Int. J. Robot. Res.* **2012**, *31*, 330–345.
67. Kulic, D.; Nakamura, Y. Scaffolding online segmentation of full body human-motion patterns. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Nice, France, 22–26 September 2008; pp. 2860–2866, doi:10.1109/iros.2008.4650619.
68. Kuklinski, K.; Fischer, K.; Marhenke, I.; Kirstein, F.; aus der Wieschen, M.V.; Solvason, D.; Kruger, N.; Savarimuthu, T.R. Teleoperation for learning by demonstration: Data glove versus object manipulation for intuitive robot control. In Proceedings of the 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), St. Petersburg, Russia, 6–8 October 2014; pp. 346–351. doi:10.1109/icumt.2014.7002126.
69. Edmonds, M.; Gao, F.; Xie, X.; Liu, H.; Qi, S.; Zhu, Y.; Rothrock, B.; Zhu, S.C. Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017, doi:10.1109/iros.2017.8206196.
70. Tzafestas, C.S. Teleplanning by human demonstration for VR-based teleoperation of a mobile robotic assistant. In Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication, Paris, France, 18–21 September 2001; pp. 462–467, doi:10.1109/roman.2001.981947.
71. Chong, N.Y.; Kotoku, T.; Ohba, K.; Komoriya, K.; Tanie, K. Exploring interactive simulator in collaborative multi-site teleoperation. In Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication, Paris, France, 18–21 September 2001; pp. 243–248, doi:10.1109/roman.2001.981909.
72. Makiishi, T.; Noborio, H. Sensor-based path-planning of multiple mobile robots to overcome large transmission delays in teleoperation. In Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan, 12–15 October 1999; Volume 4, pp. 656–661, doi:10.1109/icsmc.1999.812482.
73. Chen, J.; Zelinsky, A. Programing by Demonstration: Coping with Suboptimal Teaching Actions. *Int. J. Robot. Res.* **2003**, *22*, 299–319, doi:10.1177/0278364903022005002.
74. Bohren, J.; Papazov, C.; Burschka, D.; Krieger, K.; Parusel, S.; Haddadin, S.; Shepherdson, W.L.; Hager, G.D.; Whitcomb, L.L. A pilot study in vision-based augmented telemanipulation for remote assembly over high-latency networks. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3631–3638, doi:10.1109/ICRA.2013.6631087.
75. Argall, B.D.; Browning, B.; Veloso, M. Learning robot motion control with demonstration and advice-operators. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 399–404, doi:10.1109/iros.2008.4651020.
76. Sweeney, J.D.; Grupen, R. A model of shared grasp affordances from demonstration. In Proceedings of the 7th IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, PA, USA, 29 November–1 December 2007; pp. 27–35, doi:10.1109/ichr.2007.4813845.

77. Calinon, S.; Evrard, P.; Gribovskaya, E.; Billard, A.; Kheddar, A. Learning collaborative manipulation tasks by demonstration using a haptic interface. In Proceedings of the International Conference on Advanced Robotics, ICAR, Munich, Germany, 22–26 June 2009; pp. 1–6.
78. Dong, S.; Williams, B. Motion learning in variable environments using probabilistic flow tubes. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 1976–1981, doi:10.1109/icra.2011.5980530.
79. Peters, R.A.; Campbell, C.L.; Bluethmann, W.J.; Huber, E. Robonaut task learning through teleoperation. In Proceedings of the ICRA'03. IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; Volume 2, pp. 2806–2811, doi:10.1109/robot.2003.1242017.
80. Tanwani, A.K.; Calinon, S. Learning Robot Manipulation Tasks With Task-Parameterized Semitied Hidden Semi-Markov Model. *IEEE Robot. Autom. Lett.* **2016**, *1*, 235–242, doi:10.1109/lra.2016.2517825.
81. Pardowitz, M.; Knoop, S.; Dillmann, R.; Zollner, R.D. Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2007**, *37*, 322–332, doi:10.1109/TSMCB.2006.886951.
82. Breazeal, C.; Aryananda, L. Recognition of affective communicative intent in robot-directed speech. *Autonom. Robots* **2002**, *12*, 83–104.
83. Calinon, S.; Guenter, F.; Billard, A. Goal-directed imitation in a humanoid robot. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, Barcelona, Spain, 18–22 April 2005; pp. 299–304, doi:10.1109/robot.2005.1570135.
84. Nicolescu, M.; Mataric, M.J. Task learning through imitation and human-robot interaction. In *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*; Cambridge University Press: Cambridge, UK, 2007; pp. 407–424.
85. Demiris, J.; Hayes, G. *Imitative Learning Mechanisms in Robots and Humans*; University of Edinburgh, Department of Artificial Intelligence: Edinburgh, UK, 1996.
86. Muhlig, M.; Gienger, M.; Hellbach, S.; Steil, J.J.; Goerick, C. Task-level imitation learning using variance-based movement optimization. In Proceedings of the ICRA'09. IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 1177–1184, doi:10.1109/robot.2009.5152439.
87. Asfour, T.; Azad, P.; Gyarfas, F.; Dillmann, R. Imitation learning of dual-arm manipulation tasks in humanoid robots. *Int. J. Humanoid Robot.* **2008**, *5*, 183–202. doi:10.1142/s0219843608001431.
88. Breazeal, C.; Scassellati, B. Robots that imitate humans. *Trends Cogn. Sci.* **2002**, *6*, 481–487, doi:10.1016/s1364-6613(02)02016-8.
89. Ijspeert, A.J.; Nakanishi, J.; Schaal, S. Movement imitation with nonlinear dynamical systems in humanoid robots. In Proceedings of the ICRA'02 IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 2, pp. 1398–1403, doi:10.1109/robot.2002.1014739.
90. Billard, A.; Mataric, M.J. Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. *Robot. Auton. Syst.* **2001**, *37*, 145–160, doi:10.1016/s0921-8890(01)00155-5.
91. Schaal, S.; Ijspeert, A.; Billard, A. Computational approaches to motor learning by imitation. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **2003**, *358*, 537–547, doi:10.1098/rstb.2002.1258.
92. Ude, A.; Gams, A.; Asfour, T.; Morimoto, J. Task-Specific Generalization of Discrete and Periodic Dynamic movement primitives. *IEEE Trans. Robot.* **2010**, *26*, 800–815, doi:10.1109/tro.2010.2065430.
93. Breazeal, C.; Berlin, M.; Brooks, A.; Gray, J.; Thomaz, A.L. Using perspective taking to learn from ambiguous demonstrations. *Robot. Auton. Syst.* **2006**, *54*, 385–393, doi:10.1016/j.robot.2006.02.004.
94. Demiris, Y. Prediction of intent in robotics and multi-agent systems. *Cognit. Process.* **2007**, *8*, 151–158, doi:10.1007/s10339-007-0168-9.
95. Breazeal, C.; Gray, J.; Berlin, M. An embodied cognition approach to mindreading skills for socially intelligent robots. *Int. J. Robot. Res.* **2009**, *28*, 656–680, doi:10.1177/0278364909102796.
96. Jansen, B.; Belpaeme, T. A computational model of intention reading in imitation. *Robot. Auton. Syst.* **2006**, *54*, 394–402, doi:10.1016/j.robot.2006.01.006.
97. Chella, A.; Dindo, H.; Infantino, I. A cognitive framework for imitation learning. *Robot. Auton. Syst.* **2006**, *54*, 403–408.
98. Trafton, J.G.; Cassimatis, N.L.; Bugajska, M.D.; Brock, D.P.; Mintz, F.E.; Schultz, A.C. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2005**, *35*, 460–470, doi:10.1109/tsmca.2005.850592.

99. Pook, P.K.; Ballard, D.H. Recognizing teleoperated manipulations. In Proceedings IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 2–6 May 1993; Volume 2, pp. 578–585, doi:10.1109/ROBOT.1993.291896.
100. Yang, J.; Xu, Y.; Chen, C.S. Hidden Markov model approach to skill learning and its application to telerobotics. *IEEE Trans. Robot. Autom.* **1994**, *10*, 621–631, doi:10.1109/70.326567.
101. Hovland, G.E.; Sikka, P.; McCarragher, B.J. Skill acquisition from human demonstration using a hidden markov model. In Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22–28 April 1996; Volume 3, pp. 2706–2711, doi:10.1109/robot.1996.506571.
102. Lee, C.; Yangsheng, X. Online, interactive learning of gestures for human/robot interfaces. In Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22–28 April 1996; Volume 4, pp. 2982–2987, doi:10.1109/ROBOT.1996.509165.
103. Tso, S.K.; Liu, K.P. Hidden Markov model for intelligent extraction of robot trajectory command from demonstrated trajectories. In Proceedings of The IEEE International Conference on Industrial Technology (ICIT '96), Shanghai, China, 2–6 December 1996; pp. 294–298, doi:10.1109/ICIT.1996.601593.
104. Yang, J.; Xu, Y.; Chen, C.S. Human action learning via hidden Markov model. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **1997**, *27*, 34–44, doi:10.1109/3468.553220.
105. Rybski, P.E.; Voyles, R.M. Interactive task training of a mobile robot through human gesture recognition. In Proceedings 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 1999; Volume 1, pp. 664–669, doi:10.1109/ROBOT.1999.770051.
106. Voyles, R.M.; Khosla, P.K. A multi-agent system for programming robots by human demonstration. *Integr. Comput. Aided Eng.* **2001**, *8*, 59–67.
107. Inamura, T.; Tushima, I.; Nakamura, Y. Acquiring motion elements for bidirectional computation of motion recognition and generation. In *Experimental Robotics VIII*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 372–381.
108. Dixon, K.R.; Khosla, P.K. Learning by observation with mobile robots: A computational approach. In Proceedings of the ICRA'04. 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; Volume 1, pp. 102–107, doi:10.1109/robot.2004.1307136.
109. Inamura, T.; Kojo, N.; Sonoda, T.; Sakamoto, K.; Okada, K. Intent imitation using wearable motion-capturing system with online teaching of task attention. In Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan, 5 December 2005; pp. 469–474. doi:10.1109/ICHR.2005.1573611.
110. Lee, D.; Nakamura, Y. Stochastic Model of Imitating a New Observed Motion Based on the Acquired Motion Primitives. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4994–5000, doi:10.1109/IROS.2006.282525.
111. Lee, D.; Nakamura, Y. Mimesis Scheme using a Monocular Vision System on a Humanoid Robot. In Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 2162–2168, doi:10.1109/ROBOT.2007.363641.
112. Calinon, S.; D'halluin, F.; Sauser, E.; Caldwell, D.; Billard, A. A probabilistic approach based on dynamical systems to learn and reproduce gestures by imitation. *IEEE Robot. Autom. Mag.* **2010**, *17*, 44–54.
113. Lee, K.; Su, Y.; Kim, T.K.; Demiris, Y. A syntactic approach to robot imitation learning using probabilistic activity grammars. *Robot. Auton. Syst.* **2013**, *61*, 1323–1334, doi:10.1016/j.robot.2013.08.003.
114. Niekum, S.; Chitta, S.; Barto, A.; Marthi, B.; Osentoski, S. Incremental Semantically Grounded Learning from Demonstration. *Robot. Sci. Syst.* **2013**, *9*, doi:10.15607/rss.2013.ix.048.
115. Kulić, D.; Takano, W.; Nakamura, Y. Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains. *Int. J. Robot. Res.* **2008**, *27*, 761–784, doi:10.1177/0278364908091153.
116. Chernova, S.; Veloso, M. Confidence-based policy learning from demonstration using gaussian mixture models. In Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, Honolulu, HI, USA, 14–18 May 2007; p. 233, doi:10.1145/1329125.1329407.
117. Cohn, D.A.; Ghahramani, Z.; Jordan, M.I. Active learning with statistical models. *Adv. Neural Inf. Process. Syst.* **1995**, 705–712, doi:10.21236/ada295617.
118. González-Fierro, M.; Balaguer, C.; Swann, N.; Nanayakkara, T. Full-Body Postural Control of a Humanoid Robot with Both Imitation Learning and Skill Innovation. *Int. J. Humanoid Robot.* **2014**, *11*, 1450012, doi:10.1142/s0219843614500121.

119. Alissandrakis, A.; Nehaniv, C.L.; Dautenhahn, K.; Saunders, J. Evaluation of robot imitation attempts: comparison of the system's and the human's perspectives. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, Salt Lake City, Utah, USA, 2–3 March 2006; pp. 134–141, doi:10.1145/1121241.1121265.
120. Kuniyoshi, Y.; Yorozu, Y.; Inaba, M.; Inoue, H. From visuo-motor self learning to early imitation—A neural architecture for humanoid learning. In Proceedings of the International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; Volume 3, pp. 3132–3139, doi:10.1109/robot.2003.1242072.
121. Billard, A.; Epars, Y.; Calinon, S.; Schaal, S.; Cheng, G. Discovering optimal imitation strategies. *Robot. Auton. Syst.* **2004**, *47*, 69–77, doi:10.1016/j.robot.2004.03.002.
122. Lopes, M.; Santos-Victor, J. A developmental roadmap for learning by imitation in robots. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2007**, *37*, 308–321, doi:10.1109/tsmcb.2006.886949.
123. Billard, A.G.; Calinon, S.; Guenter, F. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robot. Auton. Syst.* **2006**, *54*, 370–384, doi:10.1016/j.robot.2006.01.007.
124. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–38.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).