1

Introduction

Robot programming is the specification of the desired motions of the robot such that it may perform sequences of prestored motions or motions computed as functions of sensory input (Lozano-Pérez, 1983).

In today's competitive global economy, shortened life cycles and diversification of the products have pushed the manufacturing industry to adopt more flexible approaches. In the meanwhile, advances in automated flexible manufacturing have made robotic technology an intriguing prospect for small- and medium-sized enterprises (SMEs). However, the complexity of robot programming remains one of the major barriers in adopting robotic technology for SMEs. Moreover, due to the strong competition in the global robot market, historically each of the main robot manufacturers has developed their own proprietary robot software, which further aggravates the matter. As a result, the cost of robotic tasks integration could be many folds of the cost of robot purchase. On the other hand, the applications of robots have gone well beyond the manufacturing to the domains such as household services, where a robot programmer's intervention would be scarce or even impossible. Interaction with robots is increasingly becoming a part of humans' daily activities. Therefore, there is an urgent need for new programming paradigms enabling novice users to program and interact with robots. Among the variety of robot programming approaches, *programming by demonstration* (PbD) holds a great potential to overcome complexities of many programming methods.

This introductory chapter reviews programming approaches and illustrates the position of PbD in the spectrum of robot programming techniques. The PbD architecture is explained next. The chapter continues with applications of PbD and concludes with an outline of the open research problems in PbD.

Robot Learning by Visual Observation, First Edition. Aleksandar Vakanski and Farrokh Janabi-Sharifi.

1.1 Robot Programming Methods

A categorization of the robot programming modes based on the taxonomy reported by Biggs and MacDonald (2003) is illustrated in Figure 1.1. The conventional methods for robot programming are classified into manual and automatic, both of which rely heavily on expensive programming expertise for encoding desired robot motions into executable programs.

The *manual programming systems* involve text-based programming and graphical interfaces. In text-based programming, a user develops a program code using either a controller-specific programming language or extensions of a high-level multipurpose language, for example, C++ or Java (Kanayama and Wu, 2000; Hopler and Otter, 2001; Thamma *et al.*, 2004). In both cases, developing the program code is time-consuming and tedious. It requires a robot programming expert and an equipped programming facility, and the outcomes rely on programmer's abilities to successfully encode the required robot performance. Moreover, since robot manufacturers have developed proprietary programming languages, in industrial environments with robots from different manufacturers, programming robots would be even more expensive. The graphical programming systems employ graphs, flowcharts, or diagrams as a medium for creating a program code (Dai and Kampker, 2000; Bischoff *et al.*, 2002). In these systems, low-level robot actions are represented by blocks or icons in a graphical interface. The user creates programs by composing sequences of elementary operations through combination of the graphical units. A subclass of the graphical programming systems is the robotic simulators, which create a virtual model of the robot and the working environment, whereby the virtual robot is employed for emulating the motions of the actual robot (Rooks, 1997). Since the actual robot

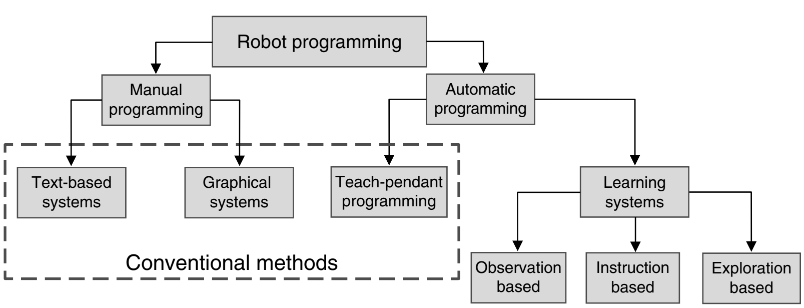

Figure 1.1 Classification of robot programming methods. (Data from Biggs and MacDonald (2003).)

is not utilized during the program development phase, this programming method is referred to as off-line programming (OLP).

The conventional *automatic programming systems* employ a teach-pendant or a panel for guiding the robot links through a set of states to achieve desired goals. The robot's joint positions recorded during the teaching phase are used to create a program code for task execution. Although programming by teach-pendants or panel decreases the level of required expertise, when compared to the text-based programming systems, it still requires trained operators with high technical skills. Other important limitations of the guided programming systems include the difficulties in programming tasks with high accuracy requirements, absence of means for tasks generalizations or for transfer of the generated programs to different robots, etc.

The stated limitations of the conventional programming methods inspired the emergence of a separate class of automatic programming systems, referred to as *learning systems*. The underlying idea of robot learning systems originates from the way we humans acquire new skills and knowledge. Biggs and MacDonald (2003) classified these systems based on the corresponding forms of learning and solving problems in cognitive psychology: exploration, instruction, and observation. In *exploration-based systems*, a robot learns a task with gradually improving the performance by autonomous exploration. These systems are often based on reinforcement learning techniques, which optimize a function of the robot states and actions through assigning rewards for the undertaken actions (Rosenstein and Barto, 2004; Thomaz and Breazeal, 2006; Luger, 2008). *Instructive systems* utilize a sequence of high-level instructions by a human operator for executing preprogrammed robot actions. Gesture-based (Voyles and Khosla, 1999), language-based (Lauria *et al*., 2002), and multimodal communication (McGuire *et al*., 2002) approaches have been implemented for programming robots using libraries of primitive robot actions. *Observation-based systems* learn from observation of another agent while executing the task. The PbD paradigm is associated with the observation-based learning systems (Billard *et al*., 2008).

1.2 Programming by Demonstration

Robot PbD is an important topic in robotics with roots in the way human beings ultimately expect to interact with a robotic system. Robot PbD refers to automatic programming of robots by demonstrating sample tasks and can be viewed as an intuitive way of transferring skill and tasks knowledge to a robot. The term is often used interchangeably with learning by demonstration (LbD) and learning from demonstration (LfD) (Argall *et al.*, 2009; Konidaris *et al.*, 2012). PbD has evolved as an interdisciplinary field of robotics, human–robot interaction (HRI), sensor fusion, machine learning, machine vision, haptics, and motor control. A few surveys of robot PbD are available in the literature (e.g., Argall *et al.*, 2009). PbD can be perceived as a class of supervised learning problems because the robot learner is presented with a set of labeled training data, and it is required to infer an output function with the capability of generalizing the function to new contexts. In the taxonomy of programming approaches shown in Figure 1.1, PbD is a superior learning-based approach. Compared to the exploration-based learning systems (as an unsupervised learning problem), PbD systems reduce the search space for solutions to a particular task, by relying on the task demonstrations. The learning is also faster because the trial and errors associated with the reinforcement methods are eliminated.

In summary, the main purpose in PbD is to overcome the major obstacles for natural and intuitive way of programming robots, namely lack of programming skills and scarcity of task knowledge. In industrial settings, this translates to reduced time and cost of programming robots by eliminating the involvement of a robot programmer. In interactive robotic platforms, PbD systems can help to better understand the mechanisms of HRI, which is central to social robotics challenges. Moreover, PbD creates a collaborative environment in which humans and robots participate in a teaching/learning process. Hence, PbD can help in developing methods for robot control which integrate safe operation and awareness of the human presence in human–robot collaborative tasks.

1.3 Historical Overview of Robot PbD

Approaches for automatic programming of robots emerged in the 1980s. One of the earlier works was the research by Dufay and Latombe (1984) who implemented inductive learning for the robot assembly tasks of mating two parts. The assembly tasks in this work were described by the geometric models of the parts, and their initial and final relations. Synthesis of program codes in the robotic language was obtained from training and inductive (planning) phases for sets of demonstrated trajectories. In this pioneering work on learning from observation, the sequences of states and actions were represented by flowcharts, where the states described the relations between the mating parts and the sensory conditions.

Another early work on a similar topic is the assembly-plan-from-observation (APO) method (Ikeuchi and Suehiro, 1993). The authors presented a method for learning assembly operations of polyhedral objects. The APO paradigm comprises the following six main steps: temporal segmentation of the observed process into meaningful subtasks, scene objects recognition, recognition of performed assembly task, grasp recognition of the manipulated objects, recognition of the global path of manipulated objects for collision avoidance, and task instantiation for reproducing the observed actions. The contact relations among the manipulated objects and environmental objects were used as a basis for constraining the relative objects movements. Abstract task models were represented by sequences of elementary operations accompanied by sets of relevant parameters (i.e., initial configurations of objects, grasp points, and goal configurations).

Munch *et al.* (1994) elaborated on the role of the teacher as a key element for successful task reproduction. The learning was accomplished through recognition of elementary operations for the observed tasks. The demonstrator supervised and guided the robot's knowledge acquisition by (i) taking into considerations the structure of robot's perceptibility sensors when providing examples, (ii) taking part in preprocessing and segmentation of the demonstrations, and (iii) evaluating the proposed task solution.

Ogawara *et al.* (2002a) proposed to generate a task model from observations of multiple demonstrations of the same task, by extracting particular relationships between the scene objects that are maintained throughout all demonstrations. Each demonstration was represented as a sequence of interactions among the user's hand, a grasped object and the environmental objects. The interactions that were observed in all demonstrations were called essential interactions, whereas the variable parts of the demonstrations called nonessential interactions were ignored in the task planning step. Generalization across multiple demonstrations was carried out by calculating the mean and variance of all trajectories for the essential interactions. A robot program was generated from the mean trajectory, and mapped to robot joints' motors using an inverse kinematics controller.

The advancements in the fields of machine learning and artificial intelligence in the past two decades produced an abundance of new methods and approaches. This trend was reflected by the implementation of approaches in robot PbD based on neural networks (Liu and Asada, 1993; Billard and Hayes, 1999), fuzzy logic (Dillmann *et al.*, 1995), statistical models (Yang *et al.*, 1994; Tso and Liu, 1997; Calinon, 2009), regression techniques (Atkeson *et al.*, 1997; Vijayakumar and Schaal, 2000), etc.

Significant body of work in PbD concentrated on utilizing virtual reality (VR) as an interaction medium to substitute the actual workspace (Takahashi and Sakai, 1991; Aleotti *et al*., 2004). The main advantages of performing demonstrations in VR include the availability of direct information for the positions and orientations of teacher's motions and the environmental objects during the demonstration phase, the accessible simulation of generated task solutions before the execution in the real world, reduced efforts and fatigue, increased safety of the user compared to physical demonstrations, etc. The concept of virtual fixtures, which refers to the use of virtual guidance and assistance to simplify and improve the tasks demonstration, was employed in Payandeh and Stanisic (2002) via provision of prior task information with an aim to restrict the demonstrated workspace and achieve more consistent performance. Aleotti *et al.* (2004) used visual and tactile virtual fixtures in PbD context, whereas adaptive virtual fixtures that correspond to different subtasks of a complex task were proposed by Aarno *et al*. (2005).

The recent progress in the field of HRI was also taken into consideration by several authors as a basis for improving the process of transfer of knowledge through demonstrations. Since building and developing social mechanisms between robots and humans in a PbD setting rely on successful training, Calinon and Billard (2007a) highlighted several interaction aspects that a demonstrator must reflect on before the demonstrations, such as what are the best ways to convey the knowledge considering the robot's abilities, which parts of the demonstration need special attention, etc. (Figure 1.2).

The latest research in the fields of neuroscience and bioinspiration also stimulated a new stream of research in PbD, and further enhanced its interdisciplinary character. Several researchers drew inspiration from the similar concepts in imitation learning among animals and children, where imitation is not only regarded as a product of social connection, but it also represents an important learning mechanism (Dautenhahn and Nehaniv, 2002).

Today, the PbD paradigm represents a multidisciplinary field which encompasses several research areas. From a general point of view, its goals are to enhance the process of transfer of knowledge to machines by providing motor skill examples through demonstrations.

1.4 PbD System Architecture

The principal steps in solving a typical PbD problem are depicted in Figure 1.3 (Billard *et al*., 2008). Note that some PbD systems include additional steps (e.g., dashed lines in Figure 1.3), such as evaluation of the

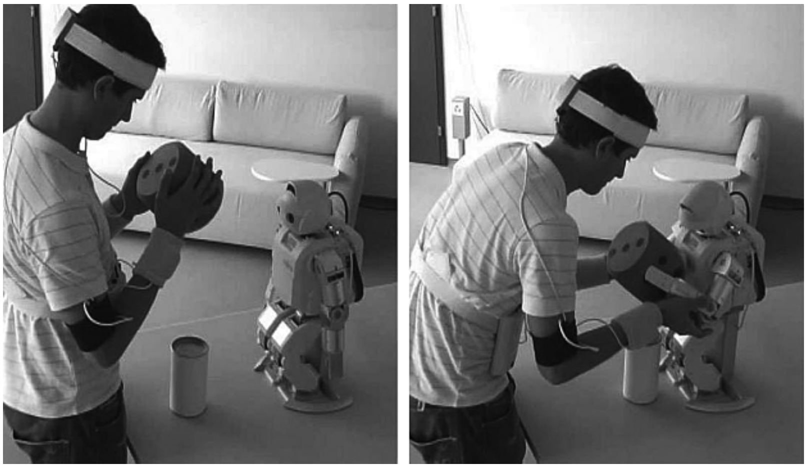

Figure 1.2 The user demonstrates the task in front of a robot learner, and is afterward actively involved in the learning process by moving the robot's arms during the task reproduction attempts to refine the learned skills (Calinon and Billard (2007a). Reproduced with permission of John Benjamins Publishing Company, Amsterdam/ Philadelphia, https://www.benjamins.com/#catalog/journals/is.8.3/main)

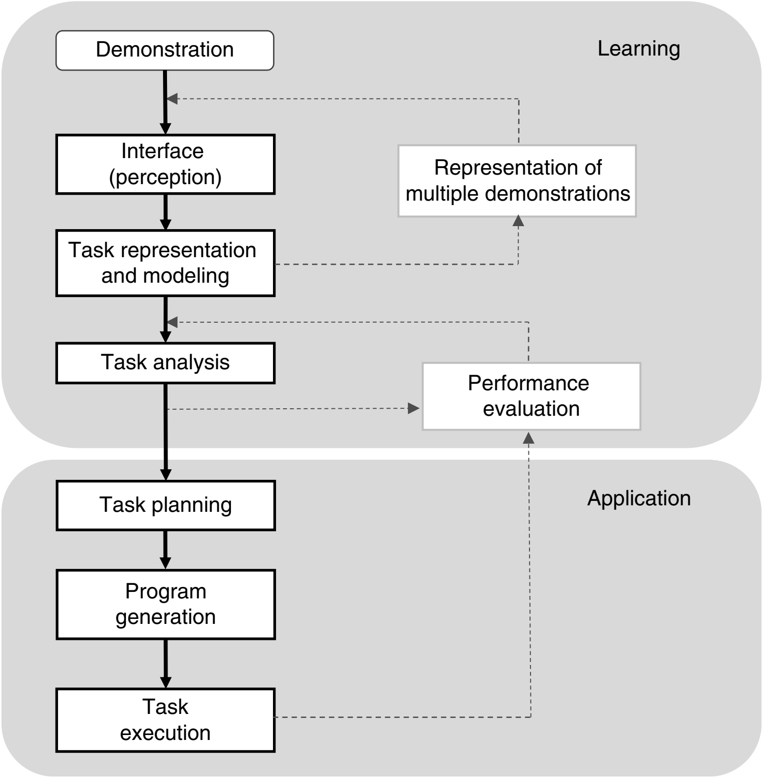

Figure 1.3 Block diagram of the information flow in a general robot PbD system. (Billard *et al.* (2008).)

reproduced task by the end-user and/or provision of additional information for improved robot performance (Chernova and Veloso, 2008a) and simulation of the planned solution before the deployment to robot executive code (Aleotti *et al.*, 2004).

1.4.1 Learning Interfaces

During the observation or perception phase of a PbD process, the teacher demonstrates the task, while the learner observes the teacher's actions and the environment. The learner must have abilities to record the movements of the teacher and the changes in the environment. In other words, the learner agent must possess attributes of perceptibility of actions and states in the world.

The presentation of demonstrations and quality of learning also depends on the learning interface (Figure 1.3). As the PbD advances, it will replace the traditional ways of guiding robots by more user-friendly interfaces, such as sensor-based techniques. The focus of this book will be on sensor-based learning methods. A brief review of all techniques is provided here.

The interfaces used in PbD can be categorized as follows:

- Kinesthetic guidance
- Direct control (through a control panel)
- Teleoperation
- Sensor based (e.g., vision, haptics, force, magnetic, and inertia)
- Virtual reality/augmented reality (VR–AR) environment

In the kinesthetic approach, the robot links are moved manually as shown in Figure 1.4, while in direct control technique the robot links are guided using a provided interface such as control panel. The joint angle or end-point trajectories are recorded and used in the next steps. Alternatively, robots can be controlled using teleoperation, as shown in Figures 1.5 and 1.6. Comparison of nonobservational methods shows that kinesthetic guidance outperforms the other two options in terms of efficiency and effectiveness. However, kinesthetic guidance has usability issues (Fischer *et al.*, 2016). Sensor-based approaches provide ergonomic convenience and ease of task demonstrations. When different sensor-based methods are compared, vision-based observation has the advantage of conveying natural task demonstrations in an unobtrusive way because no sensors are required to be attached to the demonstrator's body or demonstrated objects. Such attachments often lead to increased consciousness on the demonstrator side degrading his/her performance and efficacy.

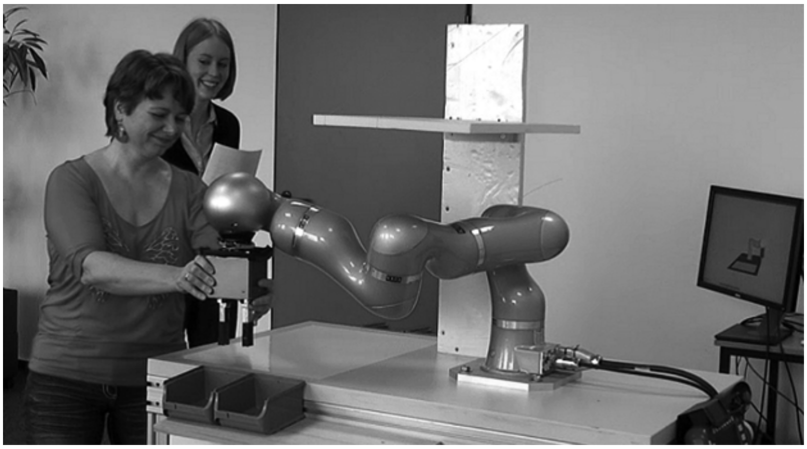

Figure 1.4 Kinesthetic teaching of feasible postures in a confined workspace. During kinesthetic teaching the human operator physically grabs the robot and executes the task. (Seidel *et al*. (2014). Reproduced with permission of IEEE.)

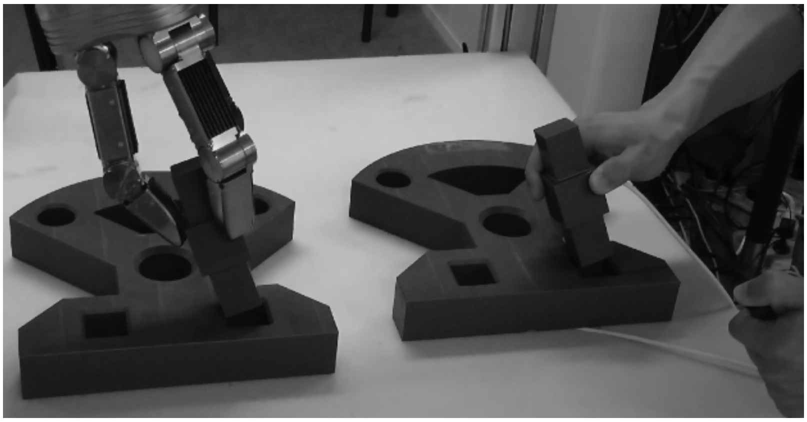

Figure 1.5 The PbD setup for teaching peg-in-hole assembly tasks includes a teleoperated robot gripper and the objects manipulated by a human expert. Tracking is done using magnetic sensors. (Yang *et al*. (2014). Reproduced with permission of IEEE.)

In order to avoid damaging the actual robot, recent approaches also rely on three-dimensional (3D) simulators before executing the program on a real robot. Examples of such platforms include V-REP (Freese *et al*., 2010) and Gazebo (Koenig and Howard, 2004), which are also open source. The main issue with the simulators is that even if accurate geometric models of

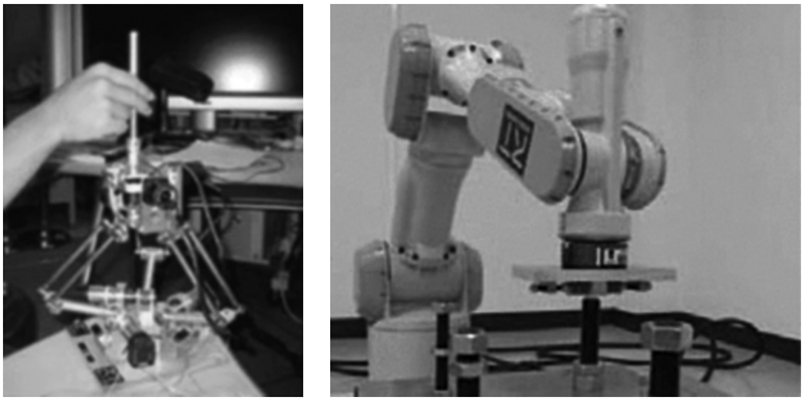

Figure 1.6 Teleoperation scheme for PbD—master arm (on the left) and slave arm (on the right) used for human demonstrations. (Shimizu *et al.* (2013). Reproduced with permission of IEEE.)

the objects become available, the dynamics and control effects are not considered in the simulations, and hence the actual behavior of the robots will deviate from those in the simulations (Angelidis and Vosniakos, 2014). AR, VR, and mixed reality (MR) methods have also been proposed (Fang *et al.*, 2012; Aleotti *et al.*, 2014) to eliminate the need for programming setup. VR–AR techniques bring the advantage of using setups and objects that might not be immediately available or affordable. They also present improved information content because they allow intermittent offline and online programming, enabling a user to modify digital model of the robots while enhancing cognition by adding extra models (AR) or through presenting parts of the real world (MR). AR–VR methods with embedded perceptual/cognitive aids (Figures 1.7 and 1.8) have been shown to outperform working with real robots in training of online programming of industrial robots (Nathanael *et al.*, 2016).

1.4.1.1 Sensor-Based Techniques

Tracking of teacher's hand is an essential observation goal for many tasks, because it is closely related to the grasping states, guiding of the tools, and/or manipulation of the objects of interest. The tracking is usually accomplished by data acquisition from sensing devices mounted directly on the teacher's body (Figure 1.9). In most of the research works on PbD, electromagnetic sensors and data gloves have been used for tracking the teacher's movements (Dillmann, 2004; Martinez and Kragic, 2008). Although these sensing systems are characterized with high measurement accuracy, their operation is sensitive to presence of ferrous parts in the working

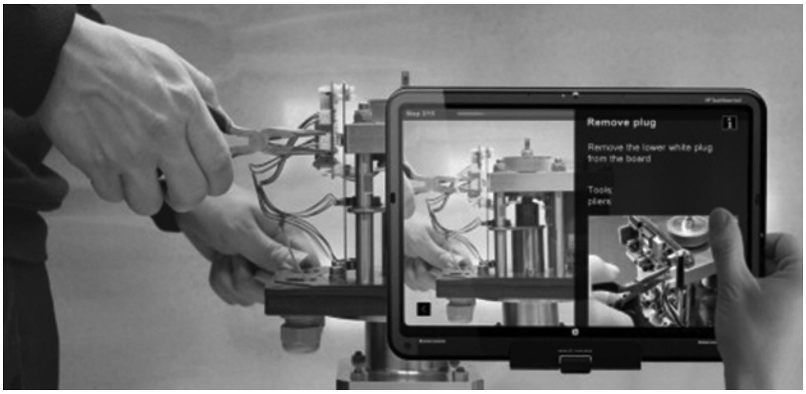

Figure 1.7 AR training of an assembly task using adaptive visual aids (AVAs). (Webel *et al*. (2013). Reproduced with permission of Elsevier.)

Figure 1.8 Mobile AR component including a haptic bracelet. (Webel *et al*. (2013). Reproduced with permission of Elsevier.)

environment. In addition, the measurement volume of the magnetic trackers is limited, as well as the sensor response becomes nonlinear toward the edges of the measurement volume. PbD interfaces with inertial sensors have also been used for capturing the demonstrator's movements (Calinon, 2009). This type of motion capture device employs gyroscopes for measurement of the rotational rates of the sensors. They are characterized by a large measurement volume, although on the account of reduced accuracy. The main disadvantages of inertial sensory systems

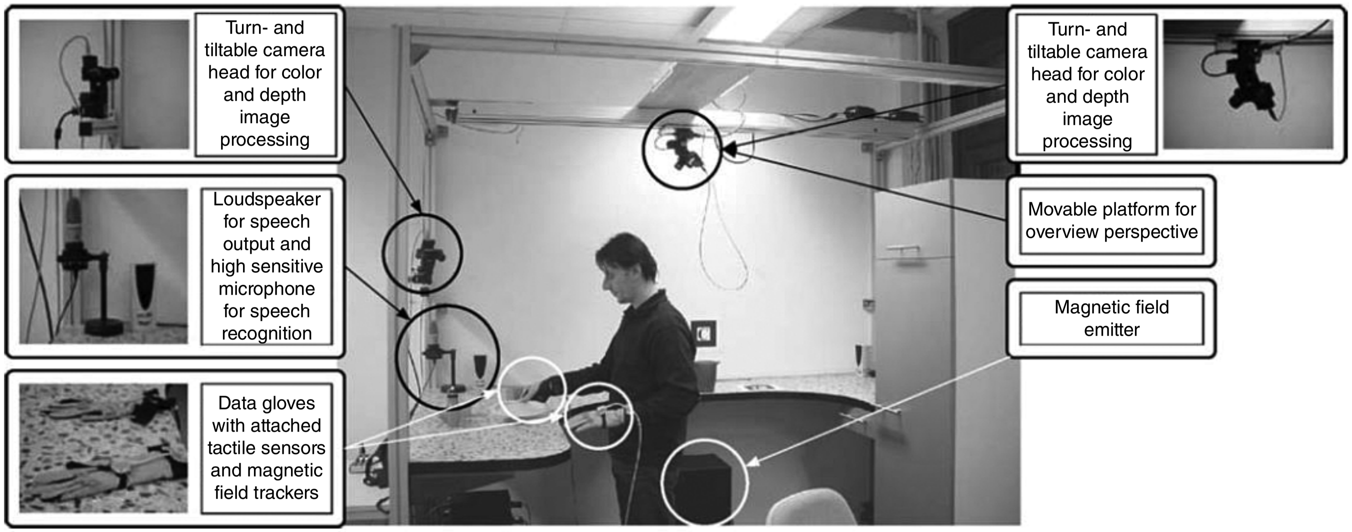

Figure 1.9 Sensory systems used for PbD task observation. (Dillmann (2004). Reproduced with permission of Elsevier.)

are the limited positional tracking accuracy, as well as the positional drift that increases over time. Some researchers employed optical marker-based motion capturing for observation of demonstrator's actions in PbD (Kruger *et al.*, 2010). The optical tracking systems are highly accurate, and do not suffer from the interference problems encountered with the magnetic systems. In addition, the measurement volume of the optical systems is large, and it is easily expanded by adding multiple sensors in the working space.

Different from the approaches that involve perception of demonstrations with sensing devices mounted directly on teacher's body, the perception sensors can also be placed externally with respect to the teacher (Argall *et al.*, 2009), as shown in Figure 1.9. This modality for recording of the teacher's motions usually employs vision sensors, for example, a single vision camera (Ogino *et al.*, 2006; Kjellstrom *et al.*, 2008), stereo cameras (Asada *et al.*, 2000), or multiple cameras (Ekvall *et al.*, 2006). The external form of perception is more challenging, due to the difficulties associated with object recognition in cluttered and dynamic environments, determining depths of the scene objects from projections onto the image space, occlusion problems, sensitivity to lighting conditions, etc. On the other hand, this type of perception enables manipulation of scene objects in a natural way without the motion intrusions caused by sensors' wires, and it represents an important step toward the expansion of robotic applications in the service industries.

Fusion of measurements from multiple sensory systems in order to extract the maximum possible information from the demonstrations was proposed by Ehrenmann *et al.* (2001). This work presented an approach for fusing force sensors, a data glove, and an active vision system in a PbD setting, as shown in Figure 1.9. Subsequently, the next generation of intelligent robots must be furnished with efficient techniques for data fusion from multiple sensors, in order to achieve reliable perception of the environment.

1.4.2 Task Representation and Modeling

The learning process in PbD typically relies on the similarities of demonstrated tasks which can be represented in either the *symbolic level* (symbolic encoding) or *trajectory level* (trajectory encoding) (Billard *et al.*, 2008). Hybrid approaches combining trajectory and symbolic learning have also been proposed in the literature, for example, by Ogawara *et al.* (2003). An illustration of different learning levels in PbD is given in Figure 1.10.

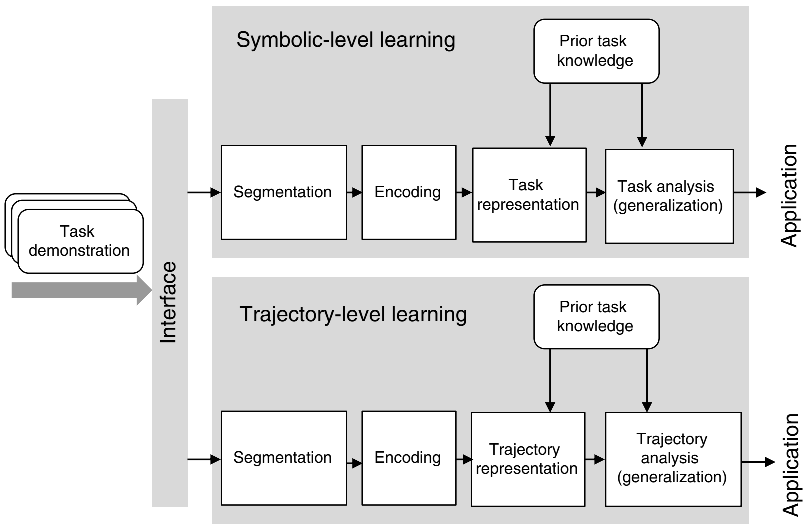

Figure 1.10 Learning levels in PbD.

1.4.2.1 Symbolic Level

Symbolic task representation (Figure 1.10) is based on high-level machine learning methods for skills acquisition as a hierarchical sequence of predefined behaviors, referred to as elementary actions or motion primitives (Friedrich *et al.*, 1998; Aramaki *et al.*, 1999; Dillmann, 2004; Saunders *et al.*, 2006). Thus, high-level tasks can be learned through hierarchy rules among a pool of acquired low-level actions (Figure 1.11). Simple elementary actions can be of the form "end-effector moves forward," whereas an example of a goal-directed behavior is "grasp the red box." Symbolic-level learning often uses graph-based or first-order logic for knowledge representation and learning. The encoding involves description of a sequence of known and given primitives. For instance, Nicolescu and Mataric (2001) used a behavior-based network to construct tasks representation, where the nodes in the network represent the behaviors, and the links represent the preconditions and postconditions dependencies. The basic behaviors (i.e., elementary actions) were associated with the state of the environment required for their activation and the predecessor behaviors (preconditions), and also with the effects of the behaviors on the environment and the successor behaviors (postconditions). For instance, the authors evaluated the approach in a task where the goal is to pick up a small box, pass it through a gate formed by a blue and a yellow object in the scene, and drop off the carried box next to an object with orange color. A mobile robot with a manipulator arm was employed for learning

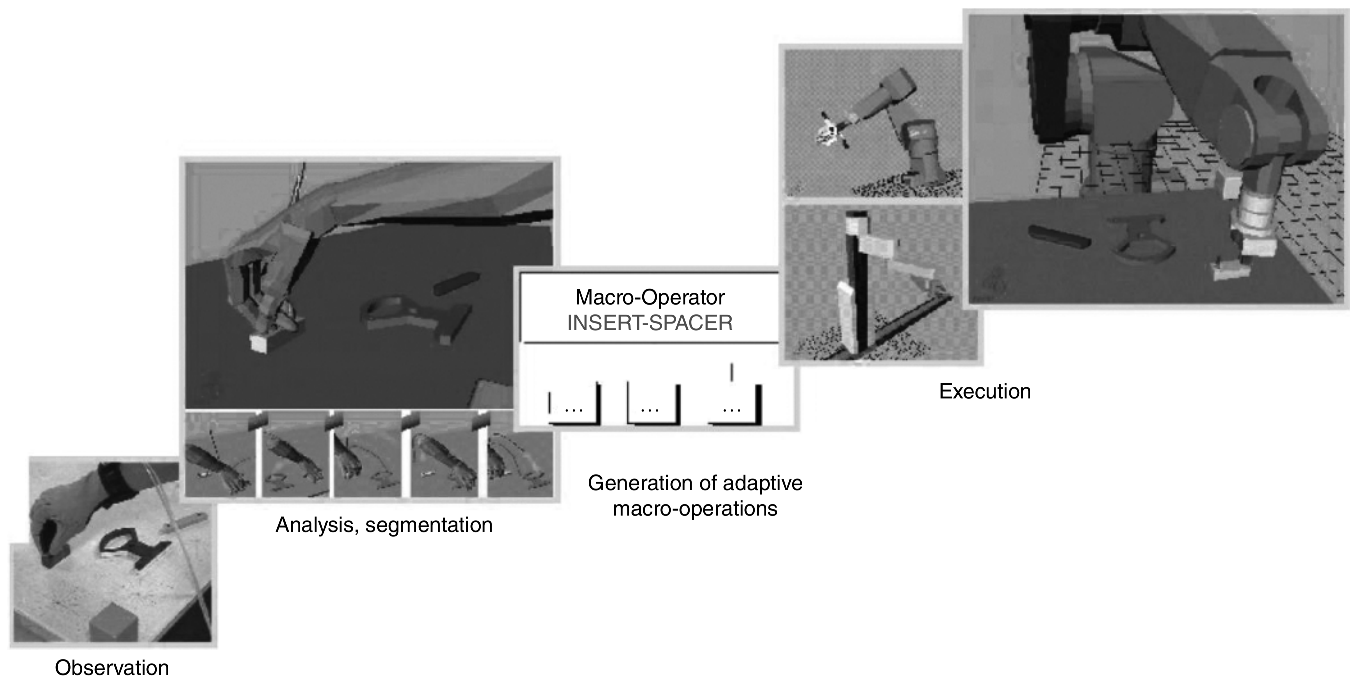

Figure 1.11 Learning at a symbolic level of abstraction by representing the decomposed task into a hierarchy of motion primitives. (Friedrich *et al.* (1998). Reproduced with permission of Springer.)

the task from human demonstrations. The robot learner built the network links based on sequential dependencies of the basic behaviors extracted from the observations. The library of basic behaviors included pick up, track green object, track yellow object, track orange object, and drop. The preconditions for picking up the box require that the drop behavior is active, i.e., the gripper is empty, and that the box is detected. Similarly, postconditions for dropping off the box are that the robot has passed through the gate and has reached the orange object.

The symbolic form of task representation allows predefined behaviors to be reused for different tasks. However, the task representation can fail upon occurrence of a behavior that does not have a corresponding match within the library of preprogrammed behaviors. The main drawbacks of the symbolic task representation are the requirement for efficient segmentation of demonstrations into elementary actions, the requirement to predefine a large set of basic controllers for reproduction of the task sections related to the elementary actions, and their limited applicability with tasks that demand high level of accuracy.

1.4.2.2 Trajectory Level

The trajectory task representation (Figure 1.10) entails encoding of demonstrated actions as continuous signals in the Cartesian space or in the joint angle space. An example of trajectory learning is shown in Figure 1.12. Trajectory-level learning is a low-level learning that relies on the observed trajectories to approximate the underlying demonstrator's policy directly and generalize the trajectories. Representing the demonstrations at a trajectory level is convenient for specifying the velocities and accelerations at different phases of the demonstrated tasks, as well as for defining the spatial task constraints. This type of task representation also allows encoding of arbitrary gestures and motions, conversely to the symbolic-based representation, where the task representation requires prior knowledge about the elementary components that comprise the demonstrated motions. Encoding often involves the use of techniques (i.e., statistical models) to reduce the dimensionality of the segmented signals.

Statistical models can capture the inherently stochastic character of human demonstrated trajectories, and thus many authors employed such models to approximate the demonstrator's policy. These approaches offer compact parametric representation of the observed movements, through probabilistic representation of the variability in the recorded signals. In the work of Calinon and Billard (2008), a Gaussian mixture model (GMM) was exploited for encoding observed tasks, by representing the recorded continuous trajectories as mixtures of Gaussian distributions. Gaussian process regression (GPR) (Schneider and Ertel, 2010) and

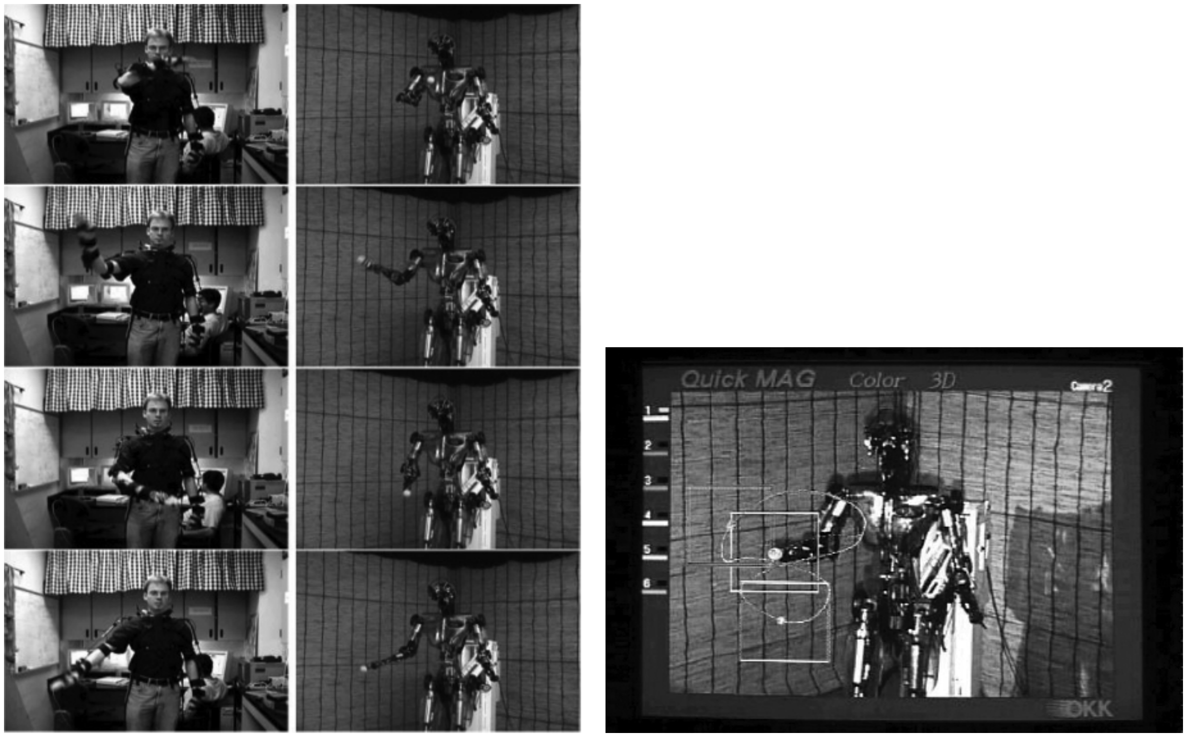

Figure 1.12 A humanoid robot is learning and reproducing trajectories for a figure-8 movement from human demonstrations. (Ijspeert *et al.* (2002b). Reproduced with permission of IEEE.)

Gaussian mixture regression (GMR) (Calinon *et al*., 2007) have also been used for encoding the trajectories. HMM has been employed by a number of authors (Tso and Liu, 1997; Yang *et al*., 1997; Aleotti and Caselli, 2006) to model a set of multiple demonstrations, captured as either Cartesian or joint angle trajectories. Due to its robustness to spatiotemporal variations of the observed time sequences, in recent years, HMM has become one of the preferred methods for modeling and analysis of human motions. Alternatively, dynamic systems such as dynamic motion primitives (DMPs) (Ijspeert *et al*., 2002a) can be utilized where differential equations are introduced to generate 1D movements, and their shapes are approximated by weighted Gaussian basis functions. Such DMPs are used as building blocks for more complex tasks. The main disadvantage of trajectory-level task representation is its inability to reproduce high-level skills.

Initially, PbD had focused on position and velocity trajectories for reproducing efficient task-oriented motions. However, these trajectories might be problematic when tasks involve contact with the environment or a human collaborator. Thus, both force and motion trajectories need to be learned (Kronander and Billard, 2014) for tasks involving compliant motions. The research work on compliant PbD has focused on the following three categories: (i) design of compliant mechanisms with active (e.g., Barret WAM arm), passive (Jafari *et al*., 2011), or hybrid (Grebenstein *et al*., 2011) robot joints to provide adaptive robot joint impedance for safe implementation of PbD; (ii) online tuning of impedance for given tasks using accurate model of robot–environment interactions (Chan and Liaw, 1996); and (iii) impedance adaptation skills learning through demonstrations of human's adaptive compliance (Ajoudani *et al*., 2012) to enable variation of robot (joint or task) stiffness during physical interactions with a human collaborator. Sensory data from electromyography (EMG) signals (Peternel *et al*., 2014) and haptic information (Kronander and Billard, 2014) can be used for learning the impedance regulation.

1.4.3 Task Analysis and Planning

The task analysis and planning phases in PbD consist of establishing a mapping between the demonstrated actions and the corresponding actions the robot learner should undertake to achieve the task goals.

1.4.3.1 Symbolic Level

For tasks represented at a symbolic level, the task analysis is often related to recognition of the relevant relationships between the teacher's hand, the manipulated objects, and the environmental objects. Hence, it is important in this phase to establish relations between the states of the environment and the corresponding actions, for example, conclusions about which states and actions must occur before other states. The task analysis can also involve elimination of suboptimal demonstrations, extraction of task constraints, aggregation of the demonstrations into different patterns, etc.

The task planning refers to generating a sequence of robot actions for attaining the desired states. It involves interpretation of the spatial relationships of the relevant objects and generation of an appropriate sequential behavior that will accomplish the task goals without violation of the task constraints. For instance, in the behavior-based network architecture reported by Nicolescu and Mataric (2001), the basic predefined behaviors were activated when all the preconditions (observed states in the environment) and the postconditions (action effects) were satisfied. Similarly, in the article by Ekvall and Kragic (2008), a variant of the Stanford Research Institute Problem Solver (STRIPS) planner was employed for teaching a robot the essential order of the subtasks for accomplishing the task goals.

1.4.3.2 Trajectory Level

The task analysis and planning in trajectory-based approaches imposes generation of a trajectory for accomplishing the task goals. A body of literature applied imitation of a single demonstrated trajectory by a robot learner (Asada *et al.*, 2000). However, these frameworks for transfer of knowledge do not provide cognitive abilities to the learning system. A type of learning system that possesses advanced cognitive abilities relies on learning from multiple demonstrated trajectories and selecting a single trajectory from the demonstrated set, which is the most adequate for reproducing the task objectives (Tso and Liu, 1997; Calinon and Billard, 2004). The drawback of these approaches is that the reproduction trajectory is retrieved from only one of the demonstrated trajectories, which is selected as the most consistent across the demonstrated set. The PbD systems with a higher level of cognition are capable of generalizing from a set of repetitive demonstrations in creating a trajectory for task reproduction.

The step of planning a reproduction strategy at a trajectory level of representation is often based on regression techniques. For instance, the work by Calinon *et al.* (2007) used GMR to obtain a generalized version of tasks modeled by GMM. This method produces smooth generalized trajectories by taking into consideration the covariances of the Gaussian distributions, and is also suitable for representing the spatial task constraints based on the variance across the demonstrated motions. Third-order spline regression (Calinon and Billard, 2004) and nonuniform rational B-splines (NURBS) (Aleotti and Caselli, 2005) have also been used for creating a generalized trajectory from sets of extracted key points in the observations (Inamura *et al.*, 2003) based on statistical task modeling with HMM. Atkeson *et al*. (1997) proposed to use locally weighted regression for skill acquisition, whereas Schaal and Atkeson (1998) employed receptive field weighted regression in the context of incremental learning of a fitting function.

In addition, several works employed the theory of dynamical systems for reproduction of trajectories modeled as a critically damped mass-spring-damper system (Ijspeert *et al*., 2003; Gribovskaya *et al*., 2010). The advantage of such systems is the independence of explicit time indexing in creating a reproduction strategy, in a sense that the system dynamics can evolve toward achieving a discrete goal or toward maintaining a periodic motion (Ijspeert *et al*., 2002b) without temporal dependence on the demonstrated data (Figure 1.12).

1.4.4 Program Generation and Task Execution

In the program generation step, the problem solution strategy from the task planning is translated into an executable robot program, which is afterward transferred to the robotic platform for execution of the desired motions.

For tasks represented by a sequence of symbolic cues, the planned actions for achieving the task goals are mapped onto a repertoire of preprogrammed robot primitives, i.e., elementary actions (Nicolescu and Mataric, 2001; Dillmann, 2004). The resulting program code in the native robot language is deployed on the robot learner platform and the planned sequence of elementary actions is executed in the actual environment. Translation of the generated plans for task reproduction onto a different robotic platform is straightforward, provided that the required elementary actions are predefined with the other robot, and it supplies the basic capabilities for achieving the task goals, such as degrees of freedom (DoFs) and workspace.

Generating a program for tasks represented at a trajectory level largely depends on the used parameters for generation of reproduction strategy. For plans described with Cartesian poses of robot's end-effector or manipulated objects, the inverse kinematics problem is solved for calculating the robot joint angles, which are sent as command signals to the robot controller. In fact, most often an inverse differential kinematics algorithm is used, since these algorithms provide a linear mapping between the joint space variables and the operational space variables. Examples are the least norm (i.e., pseudoinverse) method (Whitney, 1969), weighted least-norm method (Whitney, 1972), damped least-squares method (Nakamura and Hanafusa, 1986; Wampler, 1986), etc. On the other hand, in some PbD approaches, the joint angles of the demonstrator's hand are recorded, and the planned strategy for task reproduction is expressed in the desired joint angles for the robot learner. In that case, the program generation may be preceded by scaling of the joint angles trajectories, to accommodate for the different kinematic parameters between the teacher and learner agents (Calinon *et al*., 2005). The task reproduction step involves deployment of the executable program onto the robot's low-level controller and execution of the program.

Some PbD systems endow validation of the generated robot programs via a simulation phase before the actual execution by the robot. For instance, in the work of Ehrenmann *et al*. (2002), the task abstraction and the generated reproduction plans were shown on a graphical interface, and a human supervisor was prompted to accept or reject the proposed sequence of actions. Only after the approval by the end-user, the code was transferred to a robot and the task was executed in the real-world conditions.

1.5 Applications

From application point of view, PbD will not be limited to programming industrial manipulators. An important growing area of application is service robotics, where the service recipients are often novice users, for example, use of homecare robots by the elderly. Entertainment and security robotics are other important areas of growth, where a PbD capability will extend quality and quantity of such applications. PbD has also been extended for use in humanoid robots.

Various applications of PbD under development have been summarized as follows:

Industrial robots—These robots are usually referred to as robotic manipulators with special-purpose end-effector. However, automated guided vehicles (AGVs) can also be treated as industrial robots. The working environment of industrial robots is mostly structured. The main industrial application is for manufacturing automation across a variety of applications including material handling, assembly, welding, dispensing, processing (e.g., sealing, cutting, casting, and surface finishing), and inspection. Due to the increased demand for flexibility and reduced structured environments, programming costs of industrial robots contribute significantly to the production costs. Therefore, PbD could play a major role in reducing the time and cost of production. Examples include robotics assembly and surface finishing.

1) *Robotic assembly*—PbD has been used for teaching difficult task of peg-in-hole for assembly operations (Yang *et al*., 2014). In new directions of manufacturing (i.e., just-in-time manufacturing), the assembly

operations need to be highly flexible and adaptive to many tasks. Additionally, programming of small parts assembly is difficult as the clearance may exceed the robot motion accuracy. The robot must master different control modes from free to compliant motions and their coordination. Therefore, to lower the cost and time of programming, the use of PbD is very encouraging.

2) *Robotic surface finishing*—Variations in the objects geometry, surface roughness, and required tolerances require highly skilled operation knowledge to adapt to such changes and adjust tool paths and parameters. Subsequently, research is underway to use PbD to capture suitable finishing parameters and to adjust industrial robots tool path for a given surface finishing task (Ng *et al.*, 2014).

Humanoid robots—Large number of DoFs and kinematic redundancies in humanoid robots make their programming quite challenging (Figure 1.13). Therefore, PbD approaches have been adopted to teach various tasks such as visuomotor coordination (Lemme *et al.*, 2013), and natural interactions with a human by a social robot (Liu *et al.*, 2014).

Service robots—Robots are expected to work closely with humans to help them in their daily lives. An important feature of these robots is their close interaction and even collaborations with a human being. For this purpose, a common understanding of the collaborative tasks is a key element of the future service robots (Rozo *et al.*, 2016). While in some scenarios, such collaboration is meant to increase process efficiency, in many tasks such as those in houses and hospitals, physical contact is necessary. Additionally, such robots will face many variations in their scenes on a day-to-day basis. Many users of service robots will also lack programming skills. Obviously, conventional programming approaches cannot be adopted for service robots. Alternatively, PbD approaches hold great potential for ease of programming through capturing the task models by demonstrations. One such example is shown in Figure 1.14.

Medical robots—Many medical procedures require high dexterity and specialized skills which are usually acquired through intensive training and pose high financial burden to healthcare systems. In the meanwhile, many medical procedures follow similar steps and tasks. Besides, in the

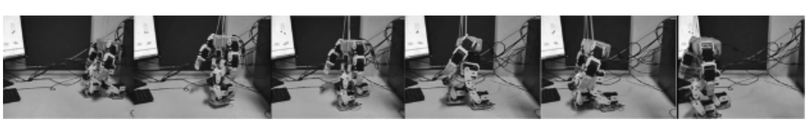

Figure 1.13 Control of a 19 DoFs humanoid robot using PbD. (Field *et al.* (2016). Reproduced with permission of IEEE.)

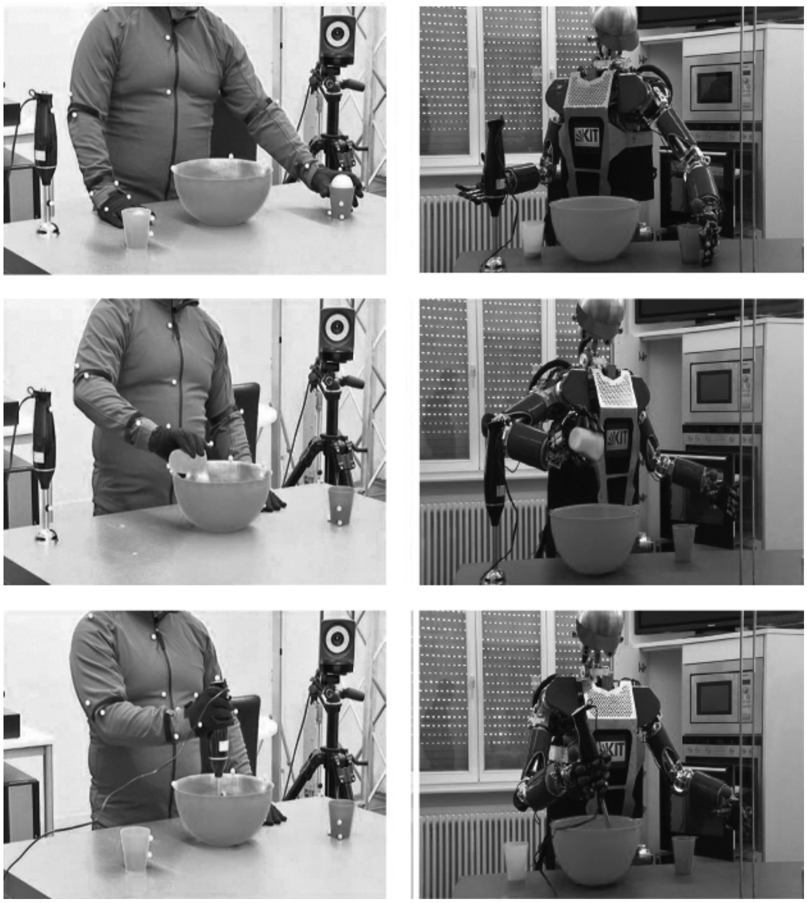

Figure 1.14 A kitchen helping robot learns the sequence of actions for cooking from observation of human demonstrations. (Wachter *et al.* (2013). Reproduced with permission of IEEE.)

absence of a specialist, robotic intervention could prove vital. Therefore, medical robotics are good candidates for integrating PbD (van den Berg *et al*., 2010). For instance, the extended focused assessment with sonography in trauma (eFAST) has been proved very effective in identifying the internal bleedings in prehospital settings like ambulances. Despite its advantages such as portability and noninvasiveness, the lack of experienced sonography operators has limited its applications. In Mylonas *et al.* (2013), a lightweight robot is programmed by expert demonstrations for eFAST scanning and is shown in Figure 1.15.

Figure 1.15 The experimental setup used for teaching (on the left) includes an ultrasound machine, an ultrasound phantom model and a handheld ultrasound transducer with force sensing and built-in 3D position markers for optical tracking system. The robotically controlled ultrasound scanning is also shown (on the right). (Mylonas *et al*. (2013). Reproduced with permission of IEEE.)

In addition to motion trajectory and task learning in the aforementioned domains of applications, PbD platform can be used for control and planning purposes as well. From this perspective, typical applications include motion planning, grasp planning, and compliance planning.

Social robots—The presence of robots in human populated environments will inevitably increase in the years to come, thereby dictating the development of complex interactions between robots and humans. PbD can provide a platform for learning and building social mechanisms between humans and machines. For instance, transferring knowledge to a robot can also be perceived from the educational point of view. That is, such transfer of knowledge would have similar effects as the transfer of knowledge to the children or other humans, and this can stimulate interest in the humans to observe and support the learning progress by the robot. Thus, such interactions can cause emotional involvement in the interaction with the robot learner (Calinon and Billard, 2007a).

Robot motion planning—With increased number of DoFs and presence of obstacles in human–robot collaborative workspaces, motion planning problem could become quite challenging. In particular, some of the obstacles might change from each task to another task (Seidel *et al*., 2014). PbD approach can be used to create incrementally a graph-based representation of the demonstrated obstacle-free task space which can be utilized for planning of safe paths. A real-time trajectory modification algorithm can also be integrated to change the learned trajectory by PbD when additional task constraints such as short distance and obstacle avoidance constraints are introduced (Kim *et al*., 2015).

Figure 1.16 Robot grasp planning application. (Aleotti and Caselli (2010). Reproduced with permission of Elsevier.)

Robot grasp planning—Planning quality grasps for given objects and tasks with complex robotic hands could be quite challenging. PbD can be used to plan effective human-like grasps (Figure 1.16). After recognition of contact states and each finger approach direction taught by an expert as the basis of understanding grasp, it can be mapped onto a robot gripper device which is usually very different from human hand. Task knowledge is then applied via high-level reasoning to choose among the applicable grasps (Aleotti and Caselli, 2010).

Robot compliance planning (*impedance learning*)—In addition to motion trajectory learning, impedance learning can enable robots to reproduce many collaborative tasks and compliant motions with the environment. Various sensors or combinations of them can be used. For example, EMG signals can be used to teach various compliance levels to a robot for a given task (Peternel *et al.*, 2014). In another work (Kronander and Billard, 2014), a teacher shook and also firmly held the robot and the haptic information was used to teach when low and high stiffness gains were required.

1.6 Research Challenges

Despite the significant advances in PbD and robotic technology, there are open issues that remain to be addressed. At the present time, transfer of skills to robots with PbD poses many challenges. In order to learn from demonstrations, the robotic system must possess certain level of cognitive skills. It must be able to reliably perceive the states and actions of the environment, to create an abstract representation of demonstrated tasks, and to generate a plan for reproduction of the demonstrated actions. Furthermore, the learning system should have the ability of generalizing the observed task solutions across different initial arrangements of the scene objects. Examples of open issues include the following:

- Creating robot programs from demonstrations for generic tasks
- Correct interpretation of the intent of the demonstrator when accomplishing a task
- Compensating for the differences in kinematic and dynamic parameters, and differences in the DoFs when the teacher and the learner have dissimilar bodies
- Relating the model of the human kinematics to task learning and reproduction
- Dealing effectively with nonoptimal and ambiguous demonstrations, or with the failures in executing the task
- Developing robust learning methods for achieving the desired performance under disturbances and changes in the environment
- Evaluating the performance of the robot in skills acquisition, and the performance of the teacher in skills transfer
- Provision of prior knowledge in a balanced way to speed up learning

The remaining text in this section discusses several of the above research challenges in robot PbD.

1.6.1 Extracting the Teacher's Intention from Observations

Certainly, one crucial element in learning from observation of demonstrations is the interpretation of the teacher's intention by the robot learner. One aspect of it is to interpret the interactions among the subtasks, that is, to determine for which of the subtasks the order of execution is important, and for which subtasks it is not. For example, it is important to know whether achieving the final goal of the task is sufficient for successful task reproduction (e.g., peg-in-hole task), or whether the task requires achieving many subgoals (e.g., trajectory following for a welding task). With regard to these questions, most of the studies in PbD relied on one of the following: (i) the teacher instructs the robot about the possible intentions during the demonstrations, or (ii) the teacher's intentions are extracted from multiple observations of demonstrations.

An example of the first set is the work of Wu and Kofman (2008) who proposed the teacher to give a brief task description to the learner before the demonstration phase. The description provided information about the overall structure of the task and its subgoals, which helped the learner in creating the task hierarchy before the demonstrations. In addition, during the demonstrations, the teacher provided short voice descriptions in the transitions between the subtasks, which alleviated the task segmentation by the robot. Several works proposed to use pointing and gazing cues during the demonstrations (Scassellati, 1999; Calinon and Billard, 2006). In that scenario, the teacher gazes toward the object of interest and/or points with his hand toward the object, so that the learner can infer the intents of the teacher's actions.

The second type of approaches revolves around extracting the teacher's intention from multiple demonstrations of the same skill. For instance, the methods for robot PbD using statistical models as a mathematical tool for task representation (e.g., HMM) employ latent states in describing the intention of the teacher during the demonstrations. The mental states of the teacher, that is, his/her intentions in performing particular tasks, are observed through a measurable process, which are the teacher's executed actions (Yang *et al*., 1997). The learner observes the demonstrations and attempts to extract the teacher's intentions from the constraints on the demonstrations described in a probabilistic framework. In the work of Calinon (2009), for demonstrations encoded with mixtures of Gaussians, the covariance matrix of the corresponding Gaussian distribution for each segment of the demonstrations was used to induce the task constraints. The segments of the task with small variability across demonstrations were associated with highly constrained motions. Thus, the learner could infer whether the teacher intended to perform precise or loose movements for the different parts of the trajectories.

The research work in the book concentrates on extracting the teacher's intentions from multiple demonstrations of the same task. Hence, the robot learner should possess abilities to identify the relevant task constraints and to extract task-specific knowledge from the observed task examples. Although augmenting the robot's task knowledge with teacher's instructions is also beneficial and leverages the learning, the topic is beyond the scope of the book.

1.6.2 Robust Learning from Observations

1.6.2.1 Robust Encoding of Demonstrated Motions

Robust encoding of human demonstrations relates to the selection of a set of training examples that conveys quantitative and qualitative information about a skill, which is sufficient for deriving a model of the skill. Efficient learning implies fast skill transfer from a few examples, since providing a large number of demonstrations for the same task can be irritating for the demonstrator, and it can cause fatigue and poor demonstrating performance. However, a limited number of demonstrations can lead to undemonstrated portions of the task. Therefore, an important question in robot PbD is the extent of training examples, i.e., how many demonstrations are enough to teach a skill. Within the PbD literature, generally between four and six demonstrations are employed for skill transfer to robots, although in some cases greater number of demonstrations are exploited (e.g., 10 demonstrations in the approach of Martinez and Kragic (2008) and 25 demonstrations in the work of Gams and Ude (2009)).

In addition, the approaches for robust learning should provide a means for dealing with ambiguity and suboptimality in the demonstrations. The demonstrated dataset can be ambiguous with respect to the possibilities for the teacher to achieve the same effect on the environment with different actions. Another source of possible ambiguities relates to the difficulties of the robot learner to distinguish two similar but different tasks, based on the recorded data with the available sensory system. Furthermore, the demonstrated set can be suboptimal in a sense that the individual demonstrations differ significantly, and the robot learner is not able to generate a successful reproduction plan.

The most intuitive approach for dealing with suboptimal and ambiguous demonstrations is to let the teacher select the demonstrated examples that will transmit enough information to the learner about the demonstrated task. This approach assumes that the teacher would speed up the learning by removing the nonimportant demonstration examples, using his/her natural abilities for generalization. Some authors proposed learning from demonstrations performed by several teachers, which can reduce the suboptimality of the performance by the individual teachers (Pook and Ballard, 1993). In the approach proposed by Chernova and Veloso (2008b) if the robot is not confident about some elements of the demonstrated task, it requests the teacher to perform additional demonstration(s). This way the robot incrementally builds the task model until the level of confidence for the entire task is above a certain threshold value.

Still, it is not clear how to automate the process of selection of training examples without relying on the generalization abilities of the teacher. Aleotti and Caselli (2006) proposed a distance metric for clustering the demonstrations into similar patterns, so that the clusters of trajectories can be treated as different skills. This approach can fail for sets containing large temporal variations across the demonstrations, or for sets with sparse solution space. On the other hand, enhanced robustness of the learning process can be achieved by eliminating the demonstrations that are too dissimilar to the set. Namely, once an appropriate model of the skill is created, the likelihood that a training example is generated by the learned model can be employed as a metric for deciding whether to use that specific training example for further processing and learning (Ogawara *et al.*, 2002b). Nevertheless, the model of the skill is only as good as the demonstrated examples so that deriving a model from a set of suboptimal demonstrations can result in a suboptimal model.

1.6.2.2 Robust Reproduction of PbD Plans

Another challenge in PbD is designing a controller that will ensure robust execution of the planned strategy for task reproduction.

In general, most of the PbD methods rely on strict temporal ordering of the states in the reproduction strategy. Hence, during the execution of the planned strategy, the robot attempts to attain the specified sequence of joint configurations, employing the explicit temporal ordering. The disadvantage of these PbD systems is the lack of flexibility in the task execution, that is, the systems lack robustness to perturbations and deviations from the ideal task configurations.

On the other hand, the design of robust controllers that will achieve the task goals in presence of perturbations (i.e., modeling errors and measurement noise) and changes in the environment is an open question in the PbD field. Within the literature, a body of work employed the dynamical systems approach (Ijspeert *et al.*, 2003) for generating reproduction strategies with enhanced robustness properties. For tasks where the only important goal is to attain a particular state at the end of the movement (so-called discrete tasks), the dynamical systems approaches ensure convergence toward the final state in presence of perturbations. However, for complex tasks (that involve accomplishing several subgoals), it is challenging to develop controllers that will execute under disturbances. Such advanced controllers should be able to perform replanning in real time by devising an alternative control strategy for achieving the desired goals (Gribovskaya *et al.*, 2010). Under significant level of perturbations that can cause large deviations from the initial strategy for task reproduction, it may be unclear for the robot how to proceed with the task execution in the new situation. The replanned strategy may require reconsidering the relative importance for execution of the different components of the task, and reformulation of the task constraints for obstacle avoidance, robot workspace limitations, etc.

1.6.3 Metrics for Evaluation of Learned Skills

The field of robot PbD currently lacks criteria for evaluation of the learning performance of the system. The difficulties arise from the fact that most of the studies in the literature focus on solving specific robotic applications, using different levels of task abstraction and task representation. The development of comprehensive evaluation metrics for the robot learning will enable comparison between the different approaches and application domains, and will provide a basis for solutions to generic tasks. This section briefly overviews several proposed techniques for evaluation of the skill acquisition in trajectory-based learning.

Pomplun and Mataric (2000) proposed a set of metrics based on the mean-square difference between the corresponding joint angles in the demonstrated and reproduced movements. The results showed that the metric based on segmented trajectories into sequences of high-level movements yields the most satisfactory results. Calinon and Billard (2007b) argued that assessment of the imitation performance should put more weight on those dimensions of the trajectories that are more constrained across the demonstrations, and thus are more important for successful reproduction. Calinon *et al.* (2005) introduced a metric that encapsulates goal-directed task constraints, that is, it enables the reproduced performance by the robot to be evaluated with a combination of functions involving distance metrics and task goals (Billard *et al.*, 2004). The method introduced a set of weighting factors for the level of importance of each of the several cost functions; however, the weights were manually selected by the demonstrator based on his understanding of the task goals.

A set of metrics for evaluation of the HRI in PbD was presented by Steinfeld *et al.* (2006). The proposed metrics pertain to assessment of the individual and joint efforts of a human and a robot in performing a task. Accordingly, the robot performance was evaluated through the degree of autonomous operation, abilities of self-assessment, awareness of human's presence, etc. The human operator performance was rated based on his/her level of knowledge of the task, mental abilities, knowledge of robot's abilities, etc. Evaluation of the performance of the human–robot as a team involved: effectiveness (percentage of the task that was performed with the designed autonomy); time efficiency; rate of utilization (e.g., percentage of request for help made by the robot, or by the human); etc. The authors also listed a number of biasing factors that have effects on the evaluation and many task-specific metrics.

1.6.4 Correspondence Problem

The process of skill transfer in PbD imposes the correspondence problem between the agent embodiments. In situations where both the teacher and the learner are robots with identical body structure, the transfer of knowledge infers direct mapping of the corresponding movements. On the other hand, learning of skills from demonstrations performed by a human teacher (or robots with different structures) involves solving the correspondence problems of workspace constraints, different DoFs, different kinematic and dynamic characteristics, etc.

Nehaniv and Dautenhahn (2001) examined the correspondence problem for a general case of imitation, which can be applied to biological or artificial agents. Their work developed a mathematical approach to model the mapping in task imitation, using the concept of relational homomorphisms. The level of success in imitation was exploited through the effects on the environment, in a sense that a successful imitation refers to accomplishing a desired effect using the affordances of the agent–environment coupling. Starting from the fact that a demonstrator defines the goal of the task, the authors held that the evaluation of the task reproduction is also demonstrator-dependent.

Alissandrakis *et al*. (2007) presented linear correspondence matrices for describing the mapping of the different DoFs between the teacher and the learner. State and action metrics for evaluation of the imitation matching were introduced, with the imitation process aiming to minimize the values of the states and action metrics. However, the proposed solutions based on linear mapping of the corresponding DoFs have limited applicability. In fact, in a PbD environment with a human teacher and a robot learner, the embodiment mapping is nonlinear and highly complex.

The research presented in the book is not concentrated on solving the correspondence problem in PbD, which nevertheless remains one of the open areas for future research.

1.6.5 Role of the Teacher in PbD

The level of HRI varies in different robot programming systems. For instance, traditional robot programming systems based on manually writing the program codes do not require interaction between the user and the robot, whereas the robot programming systems based on guiding the robot's links through required trajectories with a teach-pendant involve higher level of interaction. The HRI in PbD is more intense, and it can affect the speed of skill acquisition by the robot. Hence, one of the open questions in the PbD learning relates to the role of the demonstrator.

Several authors suggested that the human teacher should play an active role in the process of skill transfer to a robot. Starting from the fact that in transferring knowledge among humans both the teacher and the learner have active roles, the same concept can be implemented for the robot PbD problem by putting the teacher in the loop of learning. For instance, in the work of Friedrich *et al.* (1998), the teacher overviews the entire learning process, and either approves the execution of the final program code or rejects the proposed code and re-explains some parts of the task.

Calinon and Billard (2007a) suggested that both the teacher and the learner should have active roles not only during the demonstration and learning phases but also during the task execution phase. Namely, when the teacher observes the robot executing the task, he/she recognizes which parts of the task were not performed at a satisfactory level. Subsequently, the teacher would help the learner to improve the task reproduction by providing feedback and refining the robot performance. The same work also reviewed the physiological and sociological aspects of teaching robots and evaluation of the reproduction attempts. Accordingly, the degree of HRI during the teaching can be evaluated based on the human's involvement in the interaction and the enthusiasm in transferring knowledge. However, the quantification of teacher's involvement/enthusiasm in the teaching interaction is even more challenging in comparison to the evaluation of learner's ability to encode skills and to create a generalized version of the skill from a set of demonstrations. Using insights from pedagogy and developmental sciences, the authors suggested several benchmarks for evaluating the success of the knowledge transfer in PbD scenarios. Furthermore, it was argued that the teacher should be knowledgeable about the learner's abilities, in terms of the ways of learning skills, but also the range of motions, velocities limits, and similar characteristics of the learning agent. Consequently, he/she should adapt the teaching techniques to maximize the probability of fast and proper transfer of knowledge.

Investigation of the social mechanisms in HRI is not among the objectives of the book. However, with the increased presence of robots around us and the increased number of services that will be provided by robots in the near future, it will be inevitable that we learn how to interact with the robots in effective and safe ways. Many research resources are currently devoted to this topic outside the PbD field.

1.7 Summary

The chapter provides an introduction to robot PbD. The motivations for the development of PbD systems along with an overview of the existing literature related to the PbD paradigm are presented. Several important early works and approaches from the PbD domain are reviewed, followed by discussion on recent advancements and state of the art. The main steps in solving the PbD problem are classified into perception, task representation and modeling, task analysis and planning, program generation, and task execution. An overview of the methods for solving the individual phases of PbD is presented for task represented at both symbolic and trajectory level of abstraction. Examples of PbD applications in the domains of industrial and service robotics are presented. The major challenges and open questions within the robot PbD field are discussed in the last section of the chapter.

References

Aarno, D., Ekvall, S., and Kragic, D., (2005). Adaptive virtual fixtures for machine-assisted teleoperation tasks. *Proceedings of IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 1139–1144.

Ajoudani, A., Gabiccini, M., Tsagarakis, N., Albu-Schaffer, A., and Bicchi, A., (2012). Teleimpedance: exploring the role of common-mode and configuration-dependent stiffness. *Proceedings of 2012 IEEE-RAS International Conference on Humanoid Robots*, Osaka, Japan, pp. 363–369.

Aleotti, J., and Caselli, S., (2005). Trajectory clustering and stochastic approximation for robot programming by demonstration. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, pp. 1029–1034.

Aleotti, J., and Caselli, S., (2006). Robust trajectory learning and approximation for robot programming by demonstration. *Robotics and Autonomous Systems*, vol. **54**, no. 5, pp. 409–413.

Aleotti, J., and Caselli, S., (2010). Interactive teaching of task-oriented robot grasps. *Robotics and Autonomous Systems*, vol. **58**, pp. 539–550.

Aleotti, J., Caselli, S., and Regiani, M., (2004). Leveraging on a virtual environment for robot programming by demonstration. *Robotics and Autonomous Systems*, vol. **47**, no. 2–3, pp. 153–161.

Aleotti, J., Micconi, G., and Caselli, S., (2014). Programming manipulating tasks by demonstration in visuo-haptic augmented reality. *Proceedings of 2014 IEEE International Symposium on Haptic, Audio and Visual Environments and Games*, Richardson, USA, pp. 13–18.

Alissandrakis, A., Nehaniv, C.L., and Duatenhahn, K., (2007). Correspondence mapping induced state and action metrics for robotic imitation. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. **37**, no. 2, pp. 299–307.

Angelidis, A., and Vosniakos, G.-C., (2014). Prediction and compensation of relative position error along industrial robot end-effector paths. *International Journal of Precision Engineering Manufacturing,* vol. **15**, no. 1, pp. 66–73.

Aramaki, S., Nagasawa, I., Kurono, S., Nagai, T., and Suetsugu, T., (1999). The knowledge representation and programming for robotic assembly task. *Proceedings of IEEE International Symposium on Assembly and Task Planning*, Porto, Portugal, pp. 256–261.

Argall, B., Chernova, S., Veloso, M., and Browning, B., (2009). A survey of learning from demonstration. *Robotics and Autonomous Systems*, vol. **57**, no. 5, pp. 469–483.

Asada, M., Yoshikawa, Y., and Hososda, K., (2000). Learning by observation without three-dimensional reconstruction. *Proceedings of Sixth International Conference on Intelligent Autonomous Systems*, Venice, Italy, pp. 555–560.

Atkeson, C.G., Moore, A.W., and Schaal, S., (1997). Locally weighted learning for control. *Artificial Intelligence Review*, vol. **11**, no. 1–5, pp. 75–113.

van den Berg, J., Miller, S., Duckworth, D., and Hu, H., (2010). Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, Anchorage, USA, pp. 2074–2081.

Biggs, G., and MacDonald, B., (2003). A survey of robot programming systems. *Proceedings of Australasian Conference on Robotics and Automation*, Brisbane, Australia, pp. 1–10.

Billard, A., and Hayes, G., (1999). DRAMA, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, vol. **7**, no. 1, pp. 35–64.

Billard, A., Eppars, Y., Calinon, S., Schaal, S., and Cheng, G., (2004). Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, vol. **47**, no. 2–3, pp. 69–77.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S., (2008). Robot programming by demonstration. *Handbook of Robotics*. (Eds.) Siciliano, B., and Oussama, K., Berlin, Germany: Springer-Verlag, ch. 59.

Bischoff, R., Kazi, A., and Seyfarth, M., (2002). The MORPHA style guide for icon-based programming. *Proceedings of 11th International Workshop on Robot and Human Interactive Communication*, Augsburg, Germany, pp. 482–487.

Calinon, S., (2009). *Robot Programming by Demonstration: A Probabilistic Approach*. Boca Raton, USA: EPFL/CRC Press.

Calinon, S., and Billard, A., (2004). Stochastic gesture production and recognition model for a humanoid robot. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, pp. 2769–2774.

Calinon, S., and Billard, A., (2006). Teaching a humanoid robot to recognize and reproduce social cues. *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, UK, pp. 346–351.

Calinon, S., and Billard, A.G., (2007a). What is the teacher's role in robot programming by demonstration? Toward benchmark for improved learning. *Journal of Interaction Studies*, vol. **8**, no. 3, pp. 441–464.

Calinon, S., and Billard, A., (2007b). Learning of gestures by imitation in a humanoid robot. *Imitation and Social Learning in Robots, Humans and Animals: Social and Communicative Dimension*. (Eds.) Dautenhahn, K., and Nehaniv, C.L., Cambridge, USA: Cambridge University Press, pp. 153–178.

Calinon, S., and Billard, A., (2008). A probabilistic programming by demonstration framework handling constraints in joint space and task space. *Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 367–372.

Calinon, S., Guenter, F., and Billard, A., (2005). Goal-directed imitation in a humanoid robot. *Proceedings of IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 300–305.

Calinon, S., Guenter, F., and Billard, A., (2007). On learning, representing and generalizing a task in a humanoid robot. *IEEE Transaction on Systems, Man and Cybernetic—Part B*, vol. **37**, no. 2, pp. 286–298.

Chan, S., and Liaw, H., (1996). Generalized impedance control of robot for assembly tasks requiring compliant manipulation. *IEEE Transactions on Industrial Electronics*, vol. **43**, no. 4, pp. 453–461.

Chernova, S., and Veloso, M., (2008a). Multi-thresholded approach to demonstration selection for interactive robot learning. *Proceedings of ACME/IEEE International Conference on Human–Robot Interaction*, Amsterdam, the Netherlands, pp. 225–232.

Chernova, S., and Veloso, M., (2008b). Teaching multi-robot coordination using demonstration and state sharing. *Proceedings of International Conference on Autonomous Agents*, Estoril, Portugal, pp. 1183–1186.

Dai, W., and Kampker, M., (2000). User oriented integration of sensor operations in an offline programming system for welding robots. *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA, pp. 1563–1567.

Dautenhahn, K., and Nehaniv, C.L., (2002). *Imitation in Animals and Artifacts*. Cambridge, USA: MIT Press.

Dillmann, R., (2004). Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, vol. **47**, no. 2–3, pp. 109–116.

Dillmann, R., Kaiser, M., and Ude, A., (1995). Acquisition of elementary robot skills from human demonstration. *Proceedings of International Symposium on Intelligent Robotic Systems*, Pisa, Italy, pp. 1–38.

Dufay, B., and Latombe, J.-C., (1984). An approach to automatic robot programming based on inductive learning. *International Journal of Robotics Research*, vol. **3**, no. 3, pp. 3–20.

Ehrenmann, M., Zollner, R., Knoop, S., and Dillmann, R., (2001). Sensor fusion approaches for observation of user actions in programming by demonstration. *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea, pp. 227–232.

Ehrenmann, M., Zollner, R., Rogalla, O., and Dillmann, R., (2002). Programming service tasks in household environments by human demonstration. *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, Berlin, Germany, pp. 460–467.

Ekvall, S., and Kragic, D., (2008). Robot learning from demonstrations: a task-level planning approach. *International Journal of Advanced Robotic Systems*, vol. **5**, no. 3, pp. 223–234.

Ekvall, S., Aarno, D., and Kragic, D., (2006). Task learning using graphical programming and human demonstrations. *IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, United Kingdom, pp. 398–403.

Fang, H.C., Ong, S.K., and Nee, A.Y.C., (2012). Interactive robot trajectory planning and simulation using Augmented Reality. *Robotics and Computer Integrated Manufacturing*, vol. **28**, no. 2, pp. 227–238.

Field, M., Stirling, D., Pan, Z., and Naghdy, F., (2016). Learning trajectories for robot programming by demonstration using a coordinated mixture of factor analysis. *IEEE Transactions on Cybernetics*, vol. **46**, no. 3, pp. 706–717.

Fischer, K., Kirstein, F., Jensen, L.C., Kruger, N., Kuklinski, K., Wieschen, M.V., *et al*., (2016). A comparison of types of robot control for programming by demonstration. *Proceedings of 11th ACM/IEEE International Conference on Human–Robot Interaction*, Christchurch, New Zealand, pp. 213–220.

Freese, M., Singh, S., Ozaki, F., and Matsuhira, N., (2010). Virtual robot experimentation platform v-rep: a versatile 3d robot simulator. *Proceedings of Second International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, Berlin, Germany, pp. 51–62.

Friedrich, H., Dillmann, R., and Rogalla, O., (1998). Interactive robot programming based on human demonstration and advice. *International Workshop on Sensor Based Intelligent Robots*, Dagstuhl Castle, Germany, pp. 96–119.

Gams, A., and Ude, A., (2009). Generalization of example movements with dynamic systems. *Proceedings of IEEE/RAS International Conference on Humanoid Robots*, Paris, France, pp. 28–33.

Grebenstein, M., Albu-Schaffer, A., Bahls, T., Chalon, M., Eiberger, O., Friedl, W., *et al*., (2011). The DLR hand arm system. *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 3175–3182.

Gribovskaya, E., Khansari-Zadeh, S.M., and Billard, A., (2010). Learning nonlinear multivariate dynamics of motion in robotics manipulators. *International Journal of Robotics Research*, vol. **30**, no. 1, pp. 80–117.

Hopler, R., and Otter, M., (2001). A versatile C++ toolbox for model based, real time control systems of robotic manipulators. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, USA, pp. 2208–2214.

Ijspeert, A.J., Nakanishi, J., and Schaal, S., (2002a). Movement imitation with nonlinear dynamical systems in humanoid robots. *Proceedings of 2002 IEEE International Conference on Robotics and Automation*, Washington, USA, pp. 1398–1403.

Ijspeert, A.J., Nakanishi, J., and Schaal, S., (2002b). Learning rhythmic movements by demonstration using nonlinear oscillators. *Proceedings of*

IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, pp. 958–963.

Ijspeert, A.J., Nakanishi, J., and Schaal, S., (2003). Learning attractor landscapes for learning motor primitives. *Advances in Neural Information Processing Systems 15*. (Eds.) Becker, S., Thrun, S., and Obermayer, K., Cambridge, USA: MIT Press, pp. 1547–1554.

Ikeuchi, K., and Suehiro, T., (1993). Toward an assembly plan from observation. Part I: task recognition with polyhedral objects. *IEEE Transactions on Robotics and Automation*, vol. **10**, no. 3, pp. 368–285.

Inamura, T., Tanie, H., and Nakamura, Y., (2003). Keyframe compression and decompression for time series data based on the continuous hidden Markov model. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA, pp. 1487–1492.

Jafari, A., Tsagarakis, N., and Caldwell, D., (2011). Awas-ii: A new actuator with adjustable stiffness based on the novel principle of adaptable pivot point and variable lever ratio. *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 4638–4643.

Kanayama, Y.T., and Wu, C.T., (2000). It's time to make mobile robots programmable. *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA, pp. 329–334.

Kim, J.-J., Park, S.-Y., and Lee, J.-J., (2015). Adaptability improvement of learning from demonstration with sequential quadratic programming for motion planning. *Proceedings of IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Busan, Korea, pp. 1032–1037.

Kjellstrom, H., Romero, J., and Kragic, D., (2008). Visual recognition of grasps for human-to-robot mapping. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 3192–3199.

Koenig, N., and Howard, A., (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, pp. 2149–2154.

Konidaris, G., Barto, A., Grupen, R., and Kuindersma, S. (2012). Robot learning from demonstration by constructing skill trees. *International Journal of Robotics Research*, vol. **3**, no. 1, pp. 360–375.

Kronander, K., and Billard, A., (2014). Learning compliant manipulation through kinesthetic and tactile human–robot interaction. *IEEE Transactions on Haptics*, vol. **7**, no. 3, pp. 367–380.

Kruger, V., Herzog, D.L., Baby, S., Ude, A., and Kragic, D., (2010). Learning actions from observations: primitive-based modeling and grammar. *Robotics and Automation Magazine*, vol. **17**, no. 2, pp. 30–43.

Lauria, S., Bugmann, G., Kyriacou, T., and Klein, E., (2002). Mobile robot programming using natural language. *Robotics and Autonomous Systems*, vol. **38**, no. 3, pp. 171–181.

Lemme, A., Freire, A., Barreto, G., and Steil, J., (2013). Kinesthetic teaching of visuomotor coordination for pointing by the humanoid robot icub. *Neurocomputing*, vol. **112**, no. 3, pp. 179–188.

Liu, S., and Asada, H., (1993). Teaching and learning of deburring robots using neural networks. *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, pp. 339–345.

Liu, P., Glas, D.F., Ishiguro, H., and Hagita, N., (2014). How to train your robot—Teaching service robots to reproduce human social behavior. *Proceedings of 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, UK, pp. 961–968.

Lozano-Pérez, T, (1983). Robot programming. *Proceedings of the IEEE*, vol. **71**, no. 7, pp. 821–841.

Luger, G.F., (2008). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (sixth edition). Boston, USA: Pearson Education Inc., pp. 442–449.

Martinez, D., and Kragic, D., (2008). Modeling and recognition of actions through motor primitives. *Proceedings of IEEE International Conference on Robotics and Automation*, Pasadena, USA, pp. 1704–1709.

McGuire, P., Fritsch, J., Steil, J., Rothling, F., Fink, G., Wachsmuth, S., *et al.*, (2002). Multi-modal human–machine communication for instructing robot grasping tasks. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp. 1082–1088.

Munch, S., Kreuziger, J., Kaiser, M., and Dillmann, R., (1994). Robot programming by demonstration (RPD)—using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. *Proceedings of 24th International Symposium on Industrial Robots*, Hannover, Germany, pp. 685–693.

Mylonas, G.P., Giataganas, P., Chaudery, M., Vitiello, V., Darzi, A., and Yang, G.Z., (2013). Autonomous eFAST ultrasound scanning by a robotic manipulator using learning from demonstrations. *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, pp. 3251–3256.

Nakamura, Y., and Hanafusa, H., (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control. *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. **108**, no. 3, pp. 163–171.

Nathanael, D., Mosialos, S., Vosniakos, G.C., (2016). Development and evaluation of a virtual training environment for on-line robot programming. *International Journal of Industrial Ergonomics*, vol. **53**, pp. 274–283.

Nehaniv, C.L., and Dautenhahn, K., (2001). Like me?—Measures of correspondence and imitation. *International Journal on Cybernetics and Systems*, vol. **32**, pp. 11–52.

Ng, C.W.X., Chan, K.H.K., Teo, W.K., and I-Ming, C., (2014). A method for capturing the tacit knowledge in the surface finishing skills by demonstration for programming a robot. *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, Hong Kong, China, pp. 1374–1379.

Nicolescu, M.N., and Mataric, M.J., (2001). Experience-based representation construction: learning from human and robot teachers. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, USA, pp. 740–745.

Ogawara, K., Takamatsu, J., Kimura, H., and Ikeuchi, K., (2002a). Generation of a task model by integrating multiple observations of human demonstrations. *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, USA, pp. 1545–1550.

Ogawara, K., Takamatsu, J., Kimura, H., and Ikeuchi, K., (2002b). Modeling manipulation interactions by hidden Markov models. *Proceedings of IEEE/ RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp. 1096–1101.

Ogawara, K., Takamatsu, J., Kimura, H., and Ikeuchi, K. (2003). Extraction of essential interactions through multiple observations of human demonstrations. *IEEE Transactions on Industrial Electronics*, vol. **50**, no. 4, pp. 667–675.

Ogino, M., Toichi, H., Yoshikawa, Y., and Asada, M., (2006). Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping. *Robotics and Autonomous Systems*, vol. **56**, no. 5, pp. 414–418.

Payandeh, S., and Stanisic, Z., (2002). On application of virtual fixtures as an aid for telemanipulation and training. *Proceedings of 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, USA, pp. 18–23.

Peternel, L., Petric, T., Oztop, E., and Babič, J., (2014). Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human in-the-loop approach. *Autonomous Robots*, vol. **36**, no. 1–2, pp. 123–136.

Pomplun, M., and Mataric, M.J., (2000). Evaluation metrics and results of human arm movement imitation. *Proceedings of First IEEE-RAS International Conference on Humanoid Robotics*, Cambridge, USA, pp. 1–8.

Pook, P.K., and Ballard, D.H., (1993). Recognizing teleoperated manipulations. *Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, USA, pp. 578–585.

Rooks, B.W., (1997). Off-line programming: a success for the automotive industry. *International Journal of Industrial Robot*, vol. **24**, no. 1, pp. 30–34.

Rosenstein, M.T., and Barto, A.G. (2004). Supervised actor-critic reinforcement learning. *Learning and Approximate Dynamic Programming: Scaling Up to the Real World*. (Eds.) Si, J., Barto, A., Powell, W., and Wunsch, D., New York, USA: John Wiley & Sons, Inc., ch. 14.

Rozo, L., Calinon, S., Caldwell, D.C., Jiménez, P., and Torras, C., (2016). Learning physical collaborative robot behaviors from human demonstrations. *IEEE Transactions on Robotics*, vol. **32**, no. 3, pp. 513–527.

Saunders, J., Nehaniv, C.L., and Dautenhahn, K., (2006). Teaching robots by moulding behavior and scaffolding the environment. *Proceedings of ACM/IEEE International Conference on Human–Robot Interaction*, Salt Lake City, USA, pp. 118–125.

Scassellati, B., (1999). Imitation and mechanisms of joint attention: a developmental structure for building social skills on a humanoid robot. *Computation for Metaphors, Analogy and Agents*. (Ed.) Nehaniv, C., vol. **1562**, Springer Lecture Notes in Artificial Intelligence, Berlin, Germany: Springer, pp. 176–195.

Schaal, S., and Atkeson, C.G., (1998). Constructive incremental learning from only local information. *Neural Computation*, vol. **10**, no. 8, pp. 2047–2084.

Schneider, M., and Ertel, W., (2010). Robot learning by demonstration with local Gaussian process regression. *Proceedings of IEEE/RAS International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, pp. 255–260.

Seidel, D., Emmerich, C., and Steil, J.J., (2014). Model-free path planning for redundant robots using sparse data from kinesthetic teaching. *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, USA, pp. 4381–4388.

Shimizu, M., Endo, Y., Onda, H., Yoon, W.-K., and Torii, T., (2013). Improving task skill transfer method by acquiring impedance parameters from human demonstrations. *Proceedings of 2013 IEEE International Conference on Mechatronics and Automation*, Takamatsu, Japan, pp. 1033–1038.

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., *et al.*, (2006). Common metrics for human–robot interaction. *Proceedings of ACM/IEEE International Conference on Human–Robot Interaction*, Salt Lake City, USA, pp. 33–40.

Takahashi, T., and Sakai, T. (1991). Teaching robot's movement in virtual reality. *Proceedings of International Conference on Robotics and Automation*, Osaka, Japan, pp. 1538–1587.

Thamma, M., Huang, L.H., Lou, S.-J., and Diez, C.R., (2004). Controlling robot through internet using Java. *Journal of Industrial Technology*, vol. **20**, no. 3, pp. 1–6.

Thomaz, A.L., and Breazeal, C., (2006). Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance. *Proceedings of 21st National Conference on Artificial Intelligence*, Boston, USA, pp. 1000–1005.

Tso, S.K., and Liu, K.P., (1997). Demonstrated trajectory selection by hidden Markov model. *Proceedings of International Conference on Robotics and Automation*, Albuquerque, USA, pp. 2713–2718.

Vijayakumar, S., and Schaal, S., (2000). Locally weighted projection regression: an O(n) algorithm for incremental real time learning in high dimensional space. *Proceedings of International Conference on Machine Learning*, San Francisco, USA, pp. 1079–1086.

Voyles, R., and Khosla, P., (1999). Gesture-based programming: a preliminary demonstration. *Proceedings of IEEE International Conference on Robotics and Automation*, Detroit, USA, pp. 708–713.

Wachter, M., Schulz, S., Asfour, T., Aksoy, E., Worgotter, F., and Dillmann, R., (2013). Action sequence reproduction based on automatic segmentation and object-action complexes. *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, Atlanta, USA, pp. 189–195.

Wampler, C.W., (1986). Manipulator inverse kinematic solutions based on damped least-squares solutions. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. **16**, no. 1, pp. 93–101.

Webel, S., Bockholt, U., Engelke, T., Gavish, N., Olbrich, M., and Preusche, C., (2013). An augmented reality training platform for assembly and maintenance tasks. *Robotics and Autonomous Systems*, vol. **61**, pp. 398–403.

Whitney, D.E., (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, vol. **10**, no. 2, pp. 47–53.

Whitney, D.E., (1972). The mathematics of coordinated control of prosthetic arms and manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, vol. **94**, no. 4, pp. 303–309.

Wu, X., and Kofman, J., (2008). Human-inspired robot task learning from human teaching. *Proceedings of IEEE International Conference on Robotics and Automation*, Pasadena, USA, pp. 3334–3339.

Yang, J., Xu, Y., and Chen, C.S., (1994). Hidden Markov model approach to skill learning and its application to telerobotics. *IEEE Transactions on Robotics and Automation*, vol. **10**, no. 5, pp. 621–631.

Yang, J., Xu, Y., and Chen, C.S., (1997). Human action learning via hidden Markov model. *IEEE Transactions on Systems, Man, and Cybernetics—Part A*, vol. **27**, no. 1, pp. 34–44.

Yang, Y., Lin, L.L., Song, Y.T., Nemec, B., Ude, A., Buch, A.G., *et al*., (2014). Fast programming of peg-in hole actions by human demonstrations. *Proceedings of 2014 International Conference on Mechatronics and Control*, Jinzhou, China, pp. 990–995.