Programming by Demonstration: A Taxonomy of Current Relevant Methods to Teach and Describe New Skills to Robots*

Jordi Bautista-Ballester[1,2], Jaume Vergés-Llahí[1], and Domènec Puig[2]

[1] ATEKNEA Solutions,
Víctor Pradera, 45, 08940 Cornellà de Llobregat, Spain
{jordi.bautista,jaume.verges}@ateknea.com
[2] Department of Computer Engineering and Mathematics,
Universitat Rovira i Virgili, Tarragona 43007, Spain
jordi.bautista@estudiants.urv.cat,
domenec.puig@urv.cat

Abstract. Programming by Demonstration (PbD) covers methods by which a robot learns new skills through human guidance and imitation. PbD has been a key topic in robotics during the last decade that includes the development of robust algorithms for motor control, motor learning, gesture recognition and the visual-motor integration. Nowadays, PbD deals more with learning methods than traditional approaches, and frequently it is referred to as Imitation Learning or Behavioral Cloning. This work will review and analyse existing works in order to create a taxonomy of the elements that constitute the most relevant approaches in this field to date. We intend to establish the categories and types of algorithms involved so far in PbD and describing their advantages and disadvantages and potential developments.

Keywords: Mobile Robotics, Programming by Demonstration (PbD), Imitation Learning, Learning from Demonstration (LfD), Taxonomy.

1 Introduction

Programming by Demonstration (PbD) covers methods by which a robot learns new skills through human guidance and imitation. Also referred to as *imitation learning, lead through teaching, tutelage, or apprenticeship learning*, PbD takes inspiration from the way humans learn new skills by imitation in order to develop methods by which new tasks can be transmitted to robots. In this paradigm, the programmer become an *instructor* where both the decomposition and the programming of a skill is performed through the observation of a demonstration done by the instructor, which can be either a human being or another robot.

* This research has been partially supported by the *Industrial Doctorate* program of the Government of Catalonia.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_21,

1.1 Motivation

The motivation of this paper is connected with our current research which is focused on envisaging the most relevant techniques that allow us to teach and share skills to a group of autonomous outdoor mobile robots called **VinBot**[1].

The purpose of the project is to provide the necessary navigation and behavioral skills to a number of agriculture robots so they can perform tasks such as monitoring the growth of crops and estimating very valuable information such as the yield. These robots will be networked, that is, robots connected to a cloud-based service which will provide the off-board computational resources and all the necessary tools for communication, storage, process and share of the data obtained by the on-board sensors.

Robots in VinBot must be capable of learning certain skills involved in autonomous outdoor navigation, such as the creation of maps of the fields and how to cope with changes in this mutable environment while moving through them in different periods of the time. Also the avoidance of unknown obstacles and potential risks to the integrity of the robot will require learning new skills or, at least, sharing sets of already acquired strategies from other sources. Finally, we need natural ways to specify and control the missions, as well as learning certain tasks to tend a specific crop will also be necessary.

Here the importance of autonomously learning skills from demonstrations is even more evident since the interaction will mostly be performed with users unaware of robot programming who might not even be located in the same place as the robot. Additionally, we intend this newly acquired knowledge can benefit any other robot in the system. This implies that the representation of skills must be adequate both for transferring not only to similar but also to not strictly identical robots, as well as shared and replicated onto a number of units.

Consequently, we intend to employ the PbD paradigm for the tasks of skill learning and transference in the context of networked autonomous mobile robots. As it will be shown in this paper, PbD is a natural approach to deal with both the problems of learning skills from demonstrators and the representation of skills among different robotic *embodiments*. Despite most of the approaches analysed here were usually applied to more human-like platforms, such as humanoids or robotic arms, we also want to investigate what type of approaches best fit our specific mobile robot platform.

1.2 Outline of the Paper

This paper will first review the state of the art in the area of imitation learning techniques, analysing both the defining elements that compose the most relevant approaches and review the **taxonomy** of techniques describing their advantages and disadvantages. The objective of this paper is dual: *summarizing* and *standardizing* the current state of this problem and its constituent *elements*, and

[1] VinBot is an FP7 European project starting in 2014 whose main object is to develop a cloud-based mobile robotic system for agricultural applications.

also analysing the scope of each of them. Such categorization will be useful for a better understanding of the techniques and deciding the most effective methods and posterior research lines in the development of a system using PbD.

2 Programming by Demonstration (PbD)

Since the 80s research in this topic has grown steadily and has become a central topic in robotics. Learning robot skills for complex platforms that interact in complex and variable environments is faced with two key challenges.

First, the complexity of the task to learn is such that learning by only *trial-and-error* would be impractical. In this way, PbD appears as a strategy to speed-up and to facilitate the process of learning by reducing the search space and yet allowing the robot to refine its model of demonstration by trial-and-error. Also PbD permits the robot to incorporate usual tasks by means of a non-specialized instructor.

Second, PbD favours a closer relation between the learning process and control stage, so the latter can be adapted in real time to perturbations and changes that will likely happen in the environment.

In this section we describe the building elements encounter in PbD and organize them in Tables 1, 2, and 3.

2.1 Elements in Programming by Demonstration

The challenges faced by PbD were enumerated in [4] as a set of key questions: *What to imitate? How to imitate? When to imitate? Whom to imitate to?* To date only the first two questions have actually been addressed in PbD. Fig. 1 shows an example of the first two questions.

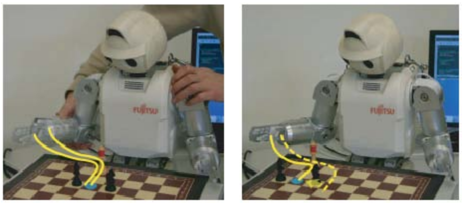

Fig. 1. Observation of multiple demonstrations and reproduction of generalized motion. In this case, extracted from [57], moving the chess queen forward. The robot records its joint's trajectories and learn to extract (what to imitate?) and after that it reproduces the skill (how to imitate).

In a similar manner, Breazeal & Scassellati in [5] went in further detail into the ideas of how, in fact, a robot is able to know what elements to imitate when attempting to learn the movements of a human being, what perceptual aspects are relevant to this task and how a robot is able distinguish the parts that will be emulated once a relevant action is perceived, as well as the moment the robot must convert this perception into a sequence of motor responses in order to achieve the same result.

According to the work by Argall et al. [6], PbD can be divided in two stages, namely, *collection of examples*, where all the information forming a demonstration is collected, and *derivation of a policy*, also known as *mapping*, where a set of examples are used to define a group of actions able to reproduce the behavior outlined in the demonstration.

What to Imitate: Collection of Examples. In this first stage, a set of informations from the demonstrator, be it a robotic or human, and possibly also from the environment, is collected from the readings of a capturing system. This can be a device mounted either on the demonstrator or on the learner, the commands of a remote control operated by the demonstrator, or a sensor located externally in the environment, like a camera.

Due to the *correspondence problem*, i.e., how to correspond actions in different embodiments and robotic platforms [4] (Fig. 2), in the collection stage we must be aware of the particular structure for both the demonstrator and the robot learner. Consequently, two successive mapping steps are required in order to cope with this problem, namely, *record mapping* and *embodiment mapping*, where the first is a mapping between sensor readings and motor commands and the latter, a mapping between motor commands from the demonstrator's body onto that of the learner.

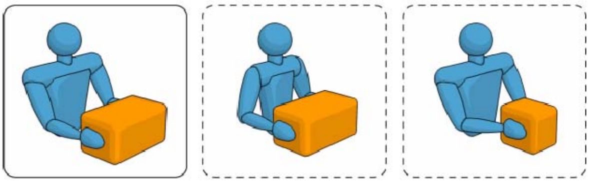

Fig. 2. Illustration of the correspondence problem, extracted from [57]

Hereafter we describe the sort of techniques summarized in Table 1 showing their requirements with respect of record and embodiment mappings. Two main groups of techniques can be separated depending on whether the demonstrator conveys directly the learner. Also the main related works are referred in Table 1.

In teleoperation (Fig.3), the demonstrator operates the robot learner and its sensors capture the motion. No record mapping is necessary since the sensory system is that from the robot learner. In shadowing, the demonstrator shows the task and the robot learner captures the motion with its sensors while attempting to match that of the instructor. In this case, a record mapping is necessary. Both

Table 1. What to imitate? Techniques in the first stage –Collection of Examples according to [6]–. First column tells the method the demonstrator employs to convey the learner. The use of record or embodiment mappings is shown respectively in 3^{rd} and 4^{th} columns. The most extreme cases are *teleoperation*, where no mapping is needed, and the case of *external observation*, where both mappings are required.

Demonstrator Methods	**Technique Names**	**Record Mapping**	**Embodiment Mapping**	**Related Works**
Robot Learner	Teleoperation	No	No	[7, 9–14]
	Shadowing	Yes	No	[15, 16]
Externally	Sensors on Instructor	No	Yes	[2, 17, 18]
	External Observation	Yes	Yes	[19–22]

in teleoperation and in shadowing no embodiment mapping is needed since the robot captures information directly from its sensors.

In case sensors are on the instructor, record comes directly from these sensors and need no record mapping. With this imitation technique, the demonstrator executes the task and the sensory system information is recorded. If sensor readings come from an external observation, the demonstrator executes the task and the external sensory system records the execution which will be translated by means of a record mapping onto the learner motor commands. In case no sensory device is located on the robot learner, an embodiment mapping is also required both in sensor-on-instructor and external-observation techniques.

Fig. 3. Teleoperation is one of the most common way for wheeled mobile robots. In this case, extracted from [58], a Pioneer 3-DX robot is teleoperated by a Phantom Premium haptic device.

How to Imitate: Derivation of a Policy. The second stage consists in executing a group of actions that allows the robot to reproduce the behavior that was demonstrated from a set of examples. The following are the three most common approaches: 1) learning a function mapping states to actions, 2) learning a model of the world dynamics, or 3) using a planner that produces the sequence of actions after learning the model of an action.

We relate the most interesting works corresponding to this section in Table 2, classifying them with respect to the learning approach utilized.

Table 2. How to imitate? Methods in the second stage –Deriving a Policy according to [6]–. Most of the works in PbD are within the *Mapping Function* category, either in Classification or in Regression.

<table>
<tr><th>Approaches</th><th colspan="2">Learning Techniques</th><th>Related Works</th></tr>
<tr><td rowspan="6">Mapping Function</td><td rowspan="3">Classification</td><td>Low Level Robot Actions</td><td>[23, 24, 29]</td></tr>
<tr><td>Robot Movement Primitives</td><td>[7, 19, 25, 28]</td></tr>
<tr><td>High Level Behaviors</td><td>[23, 26, 27]</td></tr>
<tr><td rowspan="3">Regression (Mapping Function Approx)</td><td>At Run Time</td><td>[13, 32]</td></tr>
<tr><td>Prior Run Time</td><td>[28, 33, 34]</td></tr>
<tr><td>Prior Execution Time</td><td>[14, 17, 20, 21]</td></tr>
<tr><td rowspan="2">System Models</td><td rowspan="2">Reward Based Learning</td><td>Engineering Reward Function</td><td>[35, 36]</td></tr>
<tr><td>Learned Reward Function</td><td>[9, 22, 37, 38]</td></tr>
<tr><td>Plans</td><td colspan="2">Using a Planner</td><td>[12, 39]</td></tr>
</table>

In the first approach, learning a mapping function, the algorithms can be grouped in two different families depending on whether the output is discrete –*classification*– or continuous –*regression*–.

In the second approach, learning a model of the world dynamics, a reward function is maximized. It can be a user-defined or learned in optimization process. It is typically formulated within the framework of Reinforcement Learning (RL).

In the third approach, goal executions can be represented as plans. Therefore, the planning framework represents a policy as a sequence of actions leading from an initial state to the target state.

2.2 Performance of Programming by Demonstration

Recent studies demonstrate that the performance achieved by the different exposed techniques may vary substantially. Different metrics to compute the imitation performance for each proposed approach can be used as in [41], where several metrics were utilized to evaluate a reproduction attempt with respect to the set of demonstrations (e.g. Root Mean Square Error (RMSE), Norm of Jerk, Learning Time and Retrieval Duration).

Argall et al. [6] also exposes the limitations of the dataset provided in the demonstration. The first limitation is caused by the dataset sparsity, which can occur when the demonstrator cannot demonstrate all possible states –*underdemonstrated states*–. To deal with this problem, [8, 39] proposed a generalization from existing demonstrations and [10, 14, 23], an acquisition of new demonstrations. The second, by the poor quality of a set of examples, which can happen whenever the instructor's demonstrations is ambiguous, unsuccessful or suboptimal in certain areas of the state space.

Different solutions were proposed to improve demonstration data when the demonstration is suboptimal or ambiguous [7, 11, 18, 23, 40] and attempts to learn from experience by means of feedback from the demonstrator or a reward function were tried, as in RL [8, 13, 17, 39, 40]. Table 3 shows main works dealing with the causes of low performance.

Table 3. Evaluation of the Apprenticeship. Two main reasons were identified in [6] as the cause of low performance: underdemonstrated states and poor quality data. We show some of the works dealing with this problems and the approaches followed.

Reasons	Approaches	Related Works
Underdemonstrated State	Generalization from Existing Demonstration	[8, 39]
	Acquisition of New Demonstrations	[10, 14, 23]
Poor quality data	Suboptimal or Ambiguous Demonstrations	[7, 11, 18, 23, 40]
	Learning from Experience	[8, 13, 17, 39, 40]

3 Taxonomy of Programming by Demonstration

Many approaches were proposed in relation with PbD. In this section we consider the most relevant ones, as enumerated in Table 4, and we intend to describe the advantages and drawbacks of each of them.

A great majority of the approaches in PbD mainly focuses on the spacial position and velocity of the end effector or the joint angles. First attempts depended on an explicit temporal indexing and virtually operated in an open loop. Main drawback of these techniques is that time dependence makes them very sensitive to both temporal and spatial perturbations.

To compensate these shortcomings, a heuristic to re-index the new trajectory in time is required, while simultaneously optimizing a measure of how well the new trajectory follows the objective one. This heuristic search is highly task-dependent and non-trivial and becomes less intuitive in high-dimensional spaces. One of these time-dependent approaches [46] employs Expectation Maximization (EM) and an extended Kalman's filter to follow a given trajectory. This algorithm also learns a local model of the robot's dynamics along the chosen trajectory.

EM was also used in [44] to optimize a Gaussian Mixture Model (GMM) for the estimation of the parameters of existing models. In order to find a statistical noise-free estimation of the dynamical model several approaches using either

Table 4. Approaches on Programming by Demonstration. There are two main sets of applications where PbD has been applied. For each set, several metrics were used to evaluate the performance of the techniques. Main approaches are related to their corresponding reference.

Applications	Metrics	Approaches	Related Works
Robot 3D position and velocity coordinates of end-effector and/or joint angles	RMS, RMSE, Learning Time, Norm of Jerk, Retrieval Duration, Max/Rel. Likelyhood, Stability, External Reward	HMM+GMR	[41]
		DMP	[42]
		MoMP	[3]
		BM	[30]
		SEDS	[29]
		RL	[37]
Robot position, orientation and velocity. Obstacles distances and discrete actions	Task Performance, Instructor Evaluation, Max. Reward Function	IRL	[9]
		RL	[49]
			[52]

Gaussian Process Regression (GPR) [31], Locally Weighted Projection Regression (LWPR) [33] or Gaussian Mixture Regression (GMR) [41] were proposed. GMR and GPR find a locally optimal model of the function by maximizing the likelihood for a complete model to fit the data, while LWPR minimizes the RMSE between the estimates and the data. One of the main drawbacks of these approaches is that they cannot guarantee a stable estimate of the motion since no stability constraint is forced near the optimization attractor point.

Dynamic Movement Primitives (DMP) [45], originally proposed by Ijspeert et al. [2], offer a method by which a non-linear dynamical model can be estimated while ensuring global stability at the optimization attractor point, that is, robustness and precision encoding of complex dynamics. DMP is also robust against perturbations and allows changing parameters of the trajectory without altering the overall shape of the movement. These models are straightforward learned by imitation and well suited for reward-driven self-improvement. MoMP is an extension of these models recently proposed by Mülling et al. [3] to cope with complex motor tasks requiring several movement primitives. MoMP creates a framework based on the idea that complex motor tasks can frequently be solved using a relatively small number of movement primitives and do not require a complex monolithic approach to solely cope with an entire task.

A different robust approach complementary to DMP is that of SEDS by Khansari et al. [29], which intends to ensure time-independent learning generalized dynamics from multiple demonstrations. SEDS also outperforms BM, previously proposed by Khansari et al.[30], in that it ensures globally asymptotic stability instead of local stability and can generalize better the motion for trajectories far from those in the demonstrations. BM is more accurate, offers more flexibility and ensures the motion is locally stable. SEDS is more constraint because it fits a motion with a single globally stable dynamics. SEDS and DMP are complementary in the following way. DMP must be used whenever a motion is intrinsically time-dependent and only one single demonstration is available. In contrast, when the motion is time-independent and multiple demonstrations are available, SEDS would be the choice. A third time-independent approach based on HMM and GMR is proposed in [41]. This method evaluates the eigenvalues of each linear dynamical system and ensures that all of them have negative real parts (stable). Nevertheless, asymptotic stability is not guaranteed.

Considering objectives like position, orientation, and velocity of the robot, distances to the obstacles and a discrete set of actions, most of the techniques are *reward-based*, that is, a known reward function to guide the exploration is assumed [49–53]. A robot's policy will then consist in choosing actions that maximize an expected reward function. In order to avoid the former selection, Inverse Reinforcement Learning (IRL) [9] offers a framework to automatically determine the reward and discover the optimal control policy using a discrete state action space. Alternative approaches derive a cost function in a continuous space [22, 55].

Recent IRL works consider multiple experts and identify multiple reward functions [56]. The goal here is this multiplicity of policies will make the controller more robust by offering alternative ways to complete the task whenever the context no longer allows the robot to perform the task in the optimal way.

More recently, Grollman et al. [54] proposed an approach based on learning from a set of failures where the evaluation of metrics is a binary value representing the success of the learned task. This work offers an interesting alternative to approaches that combine IL and RL since no reward function needs to be explicitly determined.

4 Discussion

As seen in the previous sections of the paper, selection of a certain type of PbD approach is determined by a number of steps we must take into account. First of all, we must consider what information from the instructor during a demonstration will be recorded and employed to teach our robot learner. Choosing a certain approach will be determined, let's say, by the possibility or desire of using the capturing system directly on the instructor or instead obtaining the information from the images of a set of cameras looking at the scene. Several possibilities are available and the actual scenario of the task to learn will determine to a great point the most proper way to capture the movements.

The creation of a set of policies that will control the robot is a question of both the type of information we obtain from the sensors and the quality of such data, as well as the algorithm to perform the learning process. Data can be continuous or discrete, but also have different levels of quality, from motor commands to higher semantic behavioral levels, like in *'bring an apple from the kitchen'*. On the other hand, considerations about the construction of the learning algorithms –p.e., the mapping functions– and at what precise moment the resulting actions will be required also determine the approach to chose.

A series of drawbacks must be taken into account in designing the procedure for PbD. Since the learning process usually consists in iterative algorithms which will adjust the results according to some objective function until a goal is met, it is important to keep in mind several measures of performance in order to obtain an adequate solution, such as the total time and convergence speed of a particular algorithm. Also the robustness to perturbations and precision of the solutions obtained in the demonstrations and the possibility to include new examples to the learning process. In this aspect, stability is a key element for most current approaches. However, sometimes stability can only be maintained locally, while other times it can be globally. Two of the most important aspects that generate low performance are under-demonstrated states, which turn into poor generalizations and problems when incorporating new data, and data of poor quality from ambiguous and not relevant demonstrations.

Looking at the taxonomy of the most relevant current approaches to PbD, we can appreciate that the two important steps to resolve are the measure of the performance of the result and the procedure to model the dynamic of the system with respect to the learning algorithm. For the first, most approaches use norms that measures the discrepancy with respect to the ideal demonstration provided by the instructor. The advantage of this approach is that we can rely on the instructor experience and goodwill, but the dependence on the instructor might also limit the overall performance of the system as the instructor become less competent or tasks grow in complexity and a correct demonstrations are more difficult to obtain. For the second problem, the main limitation is the scope of the task, i.e., a combination of primitives that can be learned independently and generalized or a longer scope task engulfing more extensive behaviors. This might compromise not only the stability of the solution but also the codification of the task into primitive ones, if we think of large differences between instructor and learner embodiments.

With respect to the future of PbD, Cangelosi et al. [48] proposed a *roadmap* of action learning research starting from 2010 till following 20 years. Until 2012, PbD was focused on how to solve *action learning*, where simplest actions or movements are considered, intended as complete motor primitives. Presently, we are on-going the second milestone in the roadmap, which refers to the flexible acquisition of action patterns and their combination to achieve more complex goals. For further details please refer to the work by Karaoguz et al. [47] or also to already mentioned Mülling et al. [3], who proposed to cope with complex motor tasks employing several movement primitives.

The acquisition of hierarchical and compositional actions is expected to be solved during the forthcoming years, and by 2016, the association between syntactic constructions and composite actions via social learning is likely to become the main focus of the investigation. On our behalf, future developments of PbD might also consider the semantic content of human commands, which can be found in natural language, but specially in visual content provided by cameras or instructional videos.

5 Conclusions

In this work we showed the categories and types of existing algorithms in PbD and discussed their advantages and drawbacks. We reviewed the state of the art in the area of imitation learning, analyzing both the building elements that compose the most relevant approaches and proposed a taxonomy of techniques describing. A summary and a standardization of the PbD techniques was accomplished in this paper, as well as the usage of each technique. Such categorization is expected to be useful for a better understanding of the PbD approaches and to decide the most effective methods and posterior research lines in the development of systems using PbD.

References

1. Calinon, S., Guenter, F., Billard, A.: On learning, representing, and generalizing a task in a humanoid robot. IEEE Trans. on Systems, Man and Cybernetics 32, 286–298 (2007)
2. Ijspeert, A., Nakanishi, J., Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots. In: IEEE Int. Conf. on Robotics and Automation, ICRA (2002)
3. Mulling, K., Kober, J., Kroemer, O., Peters, J.: Learning to Select and Generalize Striking Movements in Robot Table Tennis. The International Journal of Robotics Research 32(3), 263–279 (2013)
4. Nehaniv, C.L., Dautenhahn, K.: Like me? Measures of correspondence and imitation. Cybernetics and Systems 32(1-2), 11–51 (2001)
5. Breazeal, C., Scassellati, B.: Robots that imitate humans. Trends in Cognitive Sciences 6(11), 481–487 (2002)
6. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A Survey of Robot Learning from Demonstration. Robotics and Autonomous Systems 57(5), 469–483 (2009)
7. Pook, P.K., Ballard, D.H.: Recognizing teleoperated manipulations. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 1993 (1993)
8. Smart, W.D.: Making Reinforcement Learning Work on Real Robots Ph.D. Thesis, Department of Computer Science, Brown University, Providence, RI (2002)
9. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proc. of the 21st Int. Conf. on Machine Learning, ICML 2004 (2004)
10. Chernova, S., Veloso, M.: Multi-thresholded approach to demonstration selection for interactive robot learning. In: Proc. of the 3rd ACM/IEEE Int. Conf. on Human-Robot Interaction, HRI 2008 (2008)
11. Breazeal, C., Berlin, M., Brooks, A., Gray, J., Thomaz, A.L.: Using perspective taking to learn from ambiguous demonstrations The Social Mechanisms of Robot Programming by Demonstration. Robotics and Autonomous Systems 54(5), 385–393 (2006)
12. Rybski, P.E., Yoon, K., Stolarz, J., Veloso, M.: Interactive robot task training through dialog and demonstration. In: Proc. of the 2nd ACM/IEEE Int. Conf. on Human-Robot Interactions, HRI 2007 (2007)
13. Argall, B., Browning, B., Veloso, M.: Learning from demonstration with the critique of a human teacher. In: Proc. of the 2nd ACM/IEEE Int. Conf. on Human-Robot Interactions, HRI 2007 (2007)

14. Grollman, D.H., Jenkins, O.C.: Dogged learning for robots. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 2007 (2007)
15. Nehmzow, U., Akanyeti, O., Weinrich, C., Kyriacou, T., Billings, S.: Robot programming by demonstration through system identification. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2007 (2007)
16. Ogino, M., Toichi, H., Yoshikawa, Y., Asada, M.: Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping. The Social Mechanisms of Robot Programming by Demonstration, Robotics and Autonomous Systems 54(5), 414–418 (2006)
17. Calinon, S., Billard, A.: Incremental learning of gestures by imitation in a humanoid robot. In: Proc. of the 2nd ACM/IEEE Int. Conf. on Human Robot Interactions, HRI 2007 (2007)
18. Aleotti, J., Caselli, S.: Robust trajectory learning and approximation for robot programming by demonstration. The Social Mechanisms of Robot Programming by Demonstration, Robotics and Autonomous Systems 54(5), 409–413 (2006)
19. Billard, A., Mataric, M.: Learning human arm movements by imitation: Evaluation of biologically inspired connectionist architecture. Robotics and Autonomous Systems 37(2-3), 145–160 (2001)
20. Ude, A., Atkeson, C.G., Riley, M.: Programming full-body movements for humanoid robots by observation. Robotics and Autonomous Systems 47, 93–108 (2004)
21. Steil, J., Rthling, F., Haschke, R., Ritter, H.: Situated robot learning for multimodal instruction and imitation of grasping. Robot Learning by Demonstration, Robotics and Autonomous Systems 2-3(47), 129–141 (2004)
22. Ratliff, N., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Proc. of the 23rd Int. Conf. on Machine Learning, ICML 2006 (2006)
23. Chernova, S., Veloso, M.: Confidence-based learning from demonstration using Gaussian Mixture Models. In: Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2007 (2007)
24. Saunders, J., Nehaniv, C.L., Dautenhahn, K.: Teaching robots by moulding behavior and scaffolding the environment. In: Proc. of the 1st ACM/IEEE Int. Conf. on Human Robot Interactions, HRI 2006 (2006)
25. Kober, J., Peters, J.: Imitation and Reinforcement Learning. Practical Learning Algorithms for Motor Primitives in Robotics 17(2), 1–8 (2010)
26. Rybski, P.E., Voyles, R.M.: Interactive task training of a mobile robot through human gesture recognition. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 1999 (1999)
27. Lockerd, A., Breazeal, C.: Tutelage and socially guided robot learning. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2004 (2004)
28. Calinon, S., Sauser, E.L., Billard, A.G., Caldwell, D.G.: Evaluation of a Probabilistic Approach to Learn and Reproduce Gestures by Imitation. In: IEEE Int. Conf. on Robotics and Automation (x), pp. 2671–2676 (May 2010)
29. Khansari-Zadeh, S.M., Billard, A.: Learning Stable Nonlinear Dynamical Systems with Gaussian Mixture Models. IEEE Trans. on Robotics 27(5), 943–957 (2011)
30. Khansari-Zadeh, S.M., Billard, A.: BM: Aniterative algorithm to learn stable nonlinear dynamical systems with Gaussian Mixture Models. In: Proc. Int. Conf. Robotics and Automation, pp. 2381–2388 (2010)
31. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. Springer, New York (2006)

32. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning rhythmic movements by demonstration using nonlinear oscillators. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2002 (2002)
33. Vijayakumar, S., Schaal, S.: Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space. In: Proc. of the 17th Int. Conf. on Machine Learning, ICML 2000 (2000)
34. Grollman, D.H., Jenkins, O.C.: Sparse incremental learning for interactive robot control policy estimation. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 2008 (2008)
35. Chersi, F.: Learning Through Imitation: a Biological Approach to Robotics. IEEE Trans. on Autonomous Mental Development 4(3), 204–214 (2012)
36. Merrick, K.: Intrinsic Motivation and Introspection in Reinforcement Learning 4(4), 315–329 (2012)
37. Guenter, F., Billard, A.: Using reinforcement learning to adapt an imitation task. In: Proc. of the IEEE/RSJ Int. Conf.onIntelligent Robots and Systems, IROS 2007 (2007)
38. Abbeel, P., Coates, A., Quigley, M., Ng, A.Y.: An application of reinforcement learning to aerobatic helicopter flight. In: Proc. of the Advances in Neural Information Processing, NIPS 2007 (2007)
39. Nicolescu, M.N., Mataric, M.J.: Methods for robot task learning: Demonstrations, generalization and practice. In: Proc. of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2003 (2003)
40. Chernova, S., Veloso, M.: Learning equivalent action choices from demonstration. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2008 (2008)
41. Calinon, S., Dhalluin, F., Sauser, E.L., Caldwell, D.G., Billard, A.G.: Learning and Reproduction of Gestures by Imitation. An Approach Based on Hidden Markov Model and Gaussian Mixture Regression 17(2), 44–54 (2010)
42. Ijspeert, A., Nakanishi, J., Schaal, S.: Trajectory formation for imitation with nonlinear dynamical systems. In: Proc. IEEE Intl Conf. on Intelligent Robots and Systems (IROS), pp. 752–757 (2001)
43. Vijayakumar, S., Dsouza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation 17(12), 2602–2634 (2005)
44. Dempster, A., Rubin, N.L.D.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Statist. Soc. B 39(1), 1–38 (1977)
45. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: Proc. Int. Conf. Robotics and Automation, pp. 1293–1298 (2009)
46. Coates, A., Abbeel, P., Ng, A.Y.: Learning for Control from Multiple Demonstrations. In: Proc. of the 25th Int. Conf. on Machine Learning (ICML 2008), pp. 144–151 (2008)
47. Karaoguz, C., Rodemann, T., Wrede, B., Goerick, C.: Learning Information Acquisition for Multitasking Scenarios in Dynamic Environments. IEEE Trans. on Autonomous Mental Development 5(1), 46–61 (2013)
48. Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., Tani, J., et al.: Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics. IEEE Trans. on Auton. Mental Develop. 2(3), 167–195 (2010)
49. Nicolescu, M.N., Jenkins, O.C., Olenderski, F.: Learning behavior fusion from demonstration. Interaction Studies 9(2), 319–352 (2008)
50. Grollman, D.H., Jenkins, O.C.: Incremental Learning of Subtasks from Unsegmented Demonstration. In: Int. Conf. on Intelligent Robots and Systems (2008)

51. Chernova, S., Veloso, M.: Interactive Policy Learning through Confidence-Based Autonomy. Journal of Artificial Intelligence Research 34, 1–25 (2009)
52. Argall, B., Browning, B., Veloso, M.: Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. Robotics and Autonomous Systems 59(3-4), 243–255 (2011)
53. Argall, B., Sauser, E.L., Billard, A.G.: Tactile Guidance for Policy Adaptation. Foundations and Trends in Robotics 1(2), 79–133 (2010)
54. Grollman, D.H., Billard, A.G.: Donut as I do: Learning from failed demonstrations. In: Int. Conf. on Robotics and Automation (2010)
55. Ratliff, N., Ziebart, B., Peterson, K., Bagnell, J.B., Hebert, H.: Inverse Optimal Heuristic Control for Imitation Learning. In: Proc. of the 12th Int. Conf. on Artificial Intelligence and Statistics (2009)
56. Choi, J., Kim, K.: Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In: Advances in Neural Info. Proc. Sys. 25, NIPS 2012 (2012)
57. Billard, A., Calinon, S.: Robot Programming by Demonstration. In: Handbook of Robotics, ch. 59 (2007)
58. Farkhadinov, I., Hwan, J.: A User Study of a Mobile Robot Teleoperation. In: Proc. of the 4th International Conference on Ubiquitous Robotics and Ambient Intelligence (2007)