Robotics and Autonomous Systems 74 (2015) 97–107

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations

Simon Manschitz [a,b,*], Jens Kober [c], Michael Gienger [b], Jan Peters [a]

[a] Institute for Intelligent Autonomous Systems, Technische Universität Darmstadt, 64289 Darmstadt, Germany

[b] Honda Research Institute Europe, 63073 Offenbach, Germany

[c] Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands

HIGHLIGHTS

- We present an approach for learning sequential skills from kinesthetic demonstrations.
- A sequential skill has the ability to sequence movement primitives (MPs) correctly.
- Learning the transition behavior between MPs is treated as a classification problem.
- The goals of the MPs are learned from demonstrations.
- The approach is validated in three experiments using a Barrett WAM robot.

ARTICLE INFO

Article history:
Received 2 February 2015
Received in revised form
26 June 2015
Accepted 3 July 2015
Available online 21 July 2015

Keywords:
Human–robot interaction
Kinesthetic teaching
Learning from demonstration
Programming by demonstration

ABSTRACT

We present an approach for learning sequential robot skills through kinesthetic teaching. In our work, finding the transitions between consecutive movement primitives is treated as multiclass classification problem. We show how the goal parameters of linear attractor movement primitives can be learned from manually segmented and labeled demonstrations and how the observed movement primitive order can help to improve the movement reproduction. The improvement is achieved by restricting the classification result to the currently activated movement primitive and its possible successors in a graph representation of the sequence, which is also learned from the demonstrations. The approach is validated with three experiments using a Barrett WAM robot.

1. Introduction

Adapting skills to new situations is arguably one of the key elements for robots to become more autonomous. Learning from demonstration (LFD) or imitation learning therefore has received a lot of attention in robotics research in the past years. The goal of LFD is to learn skills based on demonstrations of a teacher [1]. While most work in this domain concentrate on learning single movement skills, sequencing such learned skills in order to perform more sophisticated tasks is still an open research topic. There are two cases where such sequential skills are particularly useful. First, there are tasks which are not representable in a non-sequential way at all. As an example, consider a robot standing in front of a door. Without any additional knowledge, the system does not know whether the robot has to open the door or if the robot just closed it. The reason is that the same state is perceived for both options. This problem is often referred to as perceptual aliasing [2]. Dissolving perceptual aliasing requires either the previous movement history to be encoded in the perceived state or a policy which activates movements based on the history of movements. Such a policy is what we call a sequential skill. Second, even though a task may be representable using a single movement, it may be beneficial to decompose it into smaller (sub-)tasks first. Such a decomposition bounds the complexity of each (sub-)task and the resulting movements are often more intuitive and easier to learn.

We aim at learning sequential skills where the currently activated movement cannot be solely determined from the perceived state, but may also depend on the history of movements. The goal is to learn when to activate each movement, based on kinesthetic demonstrations. Kinesthetic teaching is a widely used teaching

* Corresponding author at: Institute for Intelligent Autonomous Systems, Technische Universität Darmstadt, 64289 Darmstadt, Germany.

E-mail addresses: manschitz@ias.tu-darmstadt.de (S. Manschitz), j.kober@tudelft.nl (J. Kober), michael.gienger@honda-ri.de (M. Gienger), mail@jan-peters.net (J. Peters).

http://dx.doi.org/10.1016/j.robot.2015.07.005
0921-8890/

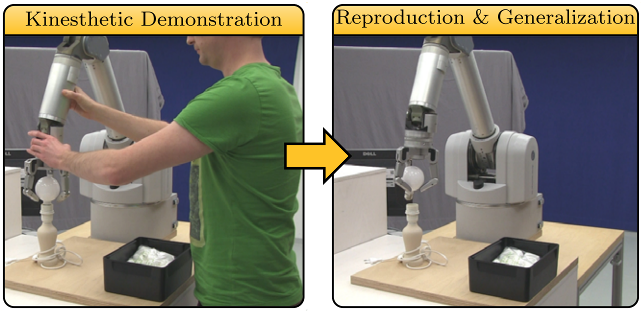

Fig. 1. The system is supposed to learn how to unscrew a light bulb from kinesthetic demonstrations. We evaluate our approach in this example using a real seven degrees of freedom (DOF) Barrett WAM robot with a four DOF hand.

method in robotics. Here, a teacher guides a robot through movements by physically moving the robot's arm, similar to parents teaching tasks to their children (see Fig. 1).

1.1. Related work

Single elementary movements are often referred to as movement primitives (MPs) in the literature [3,4]. The traditional way of sequencing MPs was inspired by the subsumption architecture [5], where the behavior of a system is represented by a hierarchy of sub-behaviors. A sequential skill is usually composed by a two-level hierarchy, whereby the lower-level MPs are activated by an upper-level sequencing layer. The sequencing layer is usually modeled as graph structure, finite state machine (FSM) or Petri net and the activation of an MP is interpreted as discrete event in a continuous system [6–8]. An alternative view is treating the overall system as continuous entity. For example, Luksch et al. [9] model a sequence with a recurrent neural network. In that architecture, MPs can be concurrently active and inhibit each other. Therefore the sequence is defined implicitly. Although this structure leads to very smooth movements, the model is hard to learn and has to be defined mostly by hand.

Most concepts for sequencing MPs concentrate either on segmenting demonstrations into a set of MPs and/or on learning the individual MP parameters [10–13]. Reproducing a sequence of learned MPs then serves as proof of concept for the segmentation. The actual sequence is not so important here, therefore the MPs are chosen randomly or are the same as in the demonstrations [14,15]. The transition behavior between MPs is also either deterministic (e.g., the succeeding movement depends only on the previous movement) or not learned at all [16]. For triggering transitions, often subgoals or sequential constraints of a task are used [17,18]. Sequential constraints (e.g., subtask A has to be executed before subtask B) can also be used to extract symbolic descriptions of tasks [19–21]. Such a description implicitly determines the MP sequence and is often intuitive. Indeed, symbolic approaches can perform sufficiently well for predetermined settings. However, they lack generality as they rely on predefined assumptions about the tasks. If these assumptions do not fully apply, they are likely to bias the system towards suboptimal decisions. Therefore probabilistic methods have become more popular, as they allow for a better generalization.

In [22], a nearest neighbor classifier is used to decide which MP to activate when the current movement has finished. Butterfield et al. [23] use a hierarchical Dirichlet process hidden Markov model as classification method for determining the next MP based on the sensor information and current MP. Niekum et al. [24] segment a demonstration with a beta process auto-regressive hidden Markov model in a set of MPs and build a FSM on the sequential level. The transition behavior is learned with k-nearest neighbor classification. The focus of our work lies on incorporating several demonstrations with varying MP sequences into one model of a task and learning the transition behavior between succeeding MPs. Basis for learning are the manually segmented and labeled sensor data traces from a set of kinesthetic demonstrations.

1.2. Proposed approach

In this paper, an MP is a dynamical system (DS) with a linear attractor behavior. A detailed description of the underlying MP framework can be found in [9]. Please note, however, that our methods are kept general and that they should be applicable to arbitrary MP frameworks and feature sets. Each MP has a goal $\boldsymbol{s}_g$ in task space coordinates that should be reached if it is activated. A goal can be a desired position of a robot body, joint angle, force or a combination thereof and can be defined relative between bodies using reference frames. MPs may be terminated before their goal is reached, for example, if a sensor reading indicates to the system that an obstacle is close to the robot. More generally, the transition behavior can be triggered based on the state of a feature set denoted as $\boldsymbol{x}_i$, with i indicating the time step. The features are not global but assigned to MPs, leading to one feature vector $\boldsymbol{x}_i^{(k)}$ per MP p_k. We assume a predefined set of K MPs denoted as $P = \{p_1, p_2, \dots, p_K\}$. All parameters of each MP in the library (such as the reference frames) are known, but the attractor goals are not.

Similar to most other approaches, the transition behavior between MPs is considered to be discrete in this paper. Therefore, only one MP is active at a time. At every time step, the system has to decide which MP to activate. A straightforward way of applying machine learning methods to this problem would be training a single classifier with the labeled demonstration data. The skill could be subsequently reproduced by choosing the classification result for the current feature values as next activated MP. Nevertheless, complex skills involve many different MPs and due to perceptual aliasing between the different movements the classification may yield unsatisfying results. As the number of MPs grows, resolving the perceptual aliasing with a better set of hand-crafted features for the classification becomes intractable.

Our proposed approach consists of three stages as depicted in Fig. 2. In the first stage, the goal parameters of the individual MPs are learned from the demonstration data (Section 2). In the second stage, a representation of the demonstrated sequences is learned by connecting the observed MPs in a graph (Section 3). Each node in the graph corresponds to an MP and each transition leads to a potentially succeeding MP. In the final stage, the MP transition behavior is learned (Section 4). One classifier is linked to each node in the graph. The task of the classifier is to decide when to transition to a new state in the graph during the reproduction of a skill, resulting in an activation of a different MP. These decisions are made based on the current state of the robot and its environment. The state is based on a set of features which are computed from raw sensor values. The graph structure therefore helps to improve the classification, as the overall classification problem is split into many smaller problems which may be easier to solve. Here, the reduced difficulty is due to the restriction of the possible classification results, which leads to less perceptual aliasing. An experimental validation of the approach is presented in Section 5, followed by a conclusion and a short outlook on future work in Section 6. Graph and transition behavior learning have been previously presented at two conferences [25,26]. In this paper, the approach is extended by learning the MP goal parameters and evaluated in substantially more experiments.

2. Learning movement primitive parameters

Learning the parameters of MPs directly from the demonstrations is crucial in order to avoid tedious parameter tuning. In our