Learning Movement Sequences from Demonstration

R. Amit

Computer Science Department University of Southern California Los Angeles, CA 90089

Abstract

This work presents a control and learning architecture for humanoid robots designed for acquiring movement skills in the context of imitation learning. Multiple levels of movement abstraction occur across the hierarchical structure of the architecture, finally leading to the representation of movement sequences within a probabilistic framework. As its substrate, the framework uses the notion of visuo-motor primitives, modules capable of recognizing as well as executing similar movements. This notion is heavily motivated by the neuroscience evidence for motor primitives and mirror neurons. Experimental results from an implementation of the architecture are presented involving learning and representation of demonstrated movement sequences from synthetic as well as real human movement data.

1 Introduction

Imitation learning has recently gained great interest in several reserach communities, including neuroscience, ethology, psychology, and robotics. Attention has been brought to it due to growing evidence that forms of mimicry and imitation play significant roles in the developmental stages of animals and humans, respectively. Mikolasi [11] provides an extensive ethological perpective on priming, response facilitation, and finally imitation. Humans are the only species that are believed to be capable of true imitation, the ability to acquire and develop novel motor skills not already within their repertoire. From a robotics perpective, imitation is of great interest because it provides a potential means of automatically programming complex systems such as dexterous anthropomorphic robots, without extensive trials [17]. It also provides a means for more natural human-robot interaction [13].

In this paper, we propose a control and learning architecture that provides a framework for abstracting movements at different levels and represent movement sequences toward learning new motor skills. The framework operates in the context of imitation; and is based on the notion that all observed movement is mapped to a set of Maja Matarić

Computer Science Department University of Southern California Los Angeles, CA 90089

primitives, which is then used to learn as well as recognize entirely novel skills. The components in the architecture provide generality of representation, thereby resulting in the ability to classify and identify similar movements in the presence of noise and subtle variations.

The rest of the paper is organized as follows. Section 2 provides a brief account of the motivation behind the choice of mechanisms used in this work. Section 3 describes the framework and the control and learning architecture. Section 4 discusses the experimental test-bed used for validation and the implementation of the system. Section 5 gives the results of the experiments. Section 6 presents related work. Finally, Section 7 discusses the approach and our continuing work.

2 Motivation

Two lines of neuroscience evidence provides the motivation and grounding for much of the organization of our proposed architecture: so-called motor primitives and mirror neurons.

No longer novel is the evidence of spinal force fields in frogs [8] and more recently in rats as well. These are a small set of motor programs that are suitably combined, through supraspinal inputs, for synthesizing various complete movements, and thereby account for the frog's motor repertoire. Similarly, Mussa-Ivaldi et al [12] have shown that vertebrates make use of motor programs, called motor primitives, for generating movements. The presence of Central Pattern Generators (CPGs) in cats, monkeys, and humans [6] provides additional support for the use of combinations of innate parametric motor programs to create more complex movements.

The evidence for mirror neurons is more recent and has instigated much new reserach into the neuroscience of imitation. Rizzolatti et al [16] have shown the presence of neurons in primates that are active both during observation and execution of similar movements or tasks. Iacoboni et al [9] have reported studies in human subjects. These mirror neurons appear to be an important link that may explain the learning of facial expressions, mannerisms, skills, etc., in infants and adults.

Based on these lines of evidence, the substrate of our architecture consists of visuo-motor primitives [10], a notion combining ideas and functionality from both motor primitives and mirror neurons. They are motor programs that form the building blocks for more complex movements, while being able to ascertain the parameters of similar movements from observation. Higher layers make use of this property and combine lower ones in order to represent and have a setup of the parameters.

Our framework operates in the context of a demonstrator and an imitator. The demonstrator is a teacher, either human or robot, that can perform desired movements and tasks, while the imitator is the humanoid robot that is expected to learn, represent, and execute the observed movements. We assume that both the demonstrator and the imitator have the same kinematic limb structure, and have similar degrees of freedom (DOF) in the corresponding joints.



Figure 1: Architecture

The framework uses a hierarchical structure in which higher layers make use of the information from lower layers in the process of learning. The architecture is composed of three computational layers, as shown in Figure 1. The base primitives form the lowest computational layer. These are visuo-motor primitives that encode motor programs for executing simple movements, and are also capable of recognizing similar movements when observed in a demonstration. During a demonstration, the base primitives extract the parameters of the observed movement and pass them to movement specializers, at the next layer. During development, movement specializers become associated with specific movements, based on statistics of their occurrence. Finally, sequence learners reside at the top of the hierarchy. They operate using the activity of the specializers below them, i.e., utilizing those specializers as a vocabulary in order to learn probabilistic models of movement sequences from multiple demonstrations.

The details of the mechanisms involved in each of the three layers are provided in the following subsections.

3.1 Base primitives

The base primitives are modules at the bottom level of the control and learning hierarchy. Each base primitive is a visuo-motor primitive that encodes motor programs for execution of a class of movements [10]. The representation of the motor programs is in a parametric form such that different movements in the same class can be generated as a result of changing the parameters. The visuo-motor property of the primitives means that they also perform the function of recognizing movements similar to those they encode, when they are observed (e.g., in a demonstration). Recognition of movements consists of ascertaining the values of the parameters that would result in the execution of the most similar movement, i.e., the closest approximation of what was observed to what is known.

In our architecture, base primitives are assumed to be innate. Thus, they do not evolve over time but are hardwired. Each base primitive is in control of a predefined set of joint degrees of freedom (DOFs). It controls those joint DOFs to produce movements belonging to the class of encoded movements. In addition, it recognizes observed movements from the same class, which involve the same set of joint DOFs. Conceptually, base primitives are dynamical systems. They encode a forward model capable of synthesizing movements, and also an inverse model that ascertains the parameters of observed movements. In a robotics implementation, such as ours, the choice of base primitives and what movement classes they encode is a responsibility of the designer. The set of primitives chosen should span a set of desirable movement classes desired for the robot to perform as well as recognize. Although the innate and fixed nature of the base primitive layer imposes a set of constraints on the learnable movements of the system, a well-designed (or evolved) set can provide an effective substrate for humanoid movement.

In our current implementation, we restrict ourselves to the encoding of kinematic variables in the base primitives. Therefore, an underlying controller is required that maps the kinematic variables to the dynamics of the movement. Our base primitives use a linear model for the velocities of joint DOF synergies. This choice of implementation does not greatly reduce the capability of the base primitives. Nonlinear movements are represented with a cominatin of multiple locally linear models.

The linear model mechanism in base primitives is implemented in the following fashion. During a demonstration, when relevant joint DOFs are involved, a base primitive first records the *initial* configuration of the DOFs. It then tracks the velocity of those DOFs as they are executing the observed movement. A hyperplane is fit to the initial estimated values. For subsequently gathered velocity values, the distances to the hyperplane are computed. As long as the cumulative distance is within an (impirically determined) error bound, the movement is considered to fit within the perticular linear model. Once the error bound is exceeded, the configuration of the DOFs during the last valid set of velocities is recorded as the *final* configuration. Subsequent movement is treated as a new movement, and a new model is fit to it. The set of parameters, Φ , for a given movement are the initial and final configurations, and the slope coefficients of the hyperplane.

3.2 Movement specializers

Movement specializers form the next level in the hierarchy and are named for their function of specializing for particular frequently observed movements. Unlike the base primitives, which are hard-wired, movement specializers are flexible and learn to represent movements based on experience. Each base primitive encode a generic class of movements, while each movement specializer learns a specific movement. For example, a base primitive may encode reaching movements by the right hand while an associated movement specializer could learn specific reaches, such as reaching the nose with the same hand.

Thus, each movement specializer is associated with a single base primitive and specializes for a particular movement that belongs to the class of movements encoded by that primitive. During a demonstration, observed movements result in the activation of the associated primitives. These primitives compute the parameters of the observed movement. This parameter space is the operational space for the specializers. Each specializer eventually represents a point in this space. Multiple specializers are linked to a single primitive. Each of them specialize for a different movement, but all movements belong to the same class – the one encoded by the primitive. Competition among the specializers that are linked to a single primitive result in their specializing for different movements.

In our system implementation, each specializer is a vector, Φ , that corresponds to the parameters of the underlying primitive. Associated with each primitive is a set of specializers that initially do not represent any movement. The first parameter vector generated by a primitive on account of a demonstration is randomly assigned to a specializer. This then is a representation for the movement. A specializer that represents a movement is termed an *assigned* specializer. Subsequently for every presentation of a parameter vector, Φ^t , by the primitive, the minimum distance, d_{min}^t , from an assigned specializer, s_i^t , belonging to the set of all specializers, S, is found.

$$d_{min}^{t} = \min\{||\Phi^{t} - s_{i}^{t}||\}, s_{i}^{t} \in S$$

If the minimum distance, d_{min}^t , is within a threshold distance, d_{thresh} , then the corresponding specializer, s_a^t ,

is declared to be active. The active specializer is updated based on the following learning rule.

$$s_a^{t+1} = s_a^t + \alpha (\Phi^t - s_a^t)$$

The active specializer is thus moved closer to Φ^t with a learning factor, α . In the event that d_{min}^t is greater than d_{thresh} , an unassigned specializer is randomly picked and is assigned the parameter vector, Φ^t . This mechanism is conceptually similar to the biologically inspired ART networks [5]. The advantage of this type of mechanism is that it allows the development of a higher layer while still permitting learning of new movements at a given layer.

3.3 Sequence learners

The third layer, sequence learning, forms the highest level of the hierarchy. Sequence learners are capable of representing complex, composite movement sequences. They do so by encoding a probabilistic ordering over simpler movements. The ordering is done over the set of movements that the movement specializers represent, thus forming the growing vocabulary for complex movements.

Each sequence learner monitors the activity of all the movement specializers during a demonstration. Depending on the content of the demonstration, different sequences of specializers become active at different times. At all times, all sequence learners are computing the degree of match between the sequence they represent and the one being observed. The sequence learner that best represents the current sequence in the end becomes further reinforced for representing that sequence. Over multiple demonstrations, the sequence learners begin to represent the statistics behind the use of different movement sequences.

In our system, sequence learning is implemented using Hidden Markov Models (HMMs). An HMM is a collection of states, state transitions, and a fixed set of symbols. Associated with an HMM are three probability distributions: 1) symbol occurrence within any state, 2) state transition, and 3) any state being an initial state. Three problems are associated with an HMM: 1) finding the probability of an observation, 2) finding a likely set of state transitions for an observation, and 3) learning the probability distributions. Rabiner et al [15] provide a detailed account of HMMs, and the algorithms for solving the three problems.

In our implementation, each sequence learner is encoded as an HMM. Every movement specializer, in turn, is treated as a symbol in the set of symbols for the HMM. To a sequence learner, the observation sequence is the timeseries of active movement specializers during the course of a demonstration. All the sequence learners take the observation sequence as input, and each computes the probability for that observation. The HMM with the highest value for the probability is declared the winner. The probability distributions of the winning HMM are then updated for the observation sequence. Thus, with experience, sequence learners begin to stabilize and specialize for particular movement sequences.

4 Experimental Validation

To experimentally validate the described architecture and implementation, we developed a physics-based humanoid simulation test-bed and performed a series of learning trials using synthetic and human motion capture data as demonstrations.

4.1 The physics-based simulator test-bed

Since we are interested in control and learning in humanoid robots, we developed a synthetic humanoid testbed for validation. The test-bed is a physics-based simulator using the Vortex real-time advanced physics libraries¹. The simulation provides gravity as well as collision detection for all the objects in the environment. Graphical rendering is achieved by using the SGI OpenInventor libraries. Snapshots from the simulator environment are shown in Figure 2. Currently, the humanoid has actuated joints from the waist up, totalling 20 active DOFs. The lower body is not actuated, and the character is firmly attached to the ground at the feet. For all the actuated joints in the simulation, we made use of the RPRO joint in the Vortex library, which behaves like a Proportional-Derivative (PD) controller loop, and allows specification of desired joint orientations.



Figure 2: Snapshots of aerobic-style movement sequence in the simulator

4.2 Task Description

To validate the architecture, we presented our implementation, in the above-described physical simulation environment, with demonstrations based on synthetic and human motion capture data.

Motion capture data were obtained using the Vicon Metrics ² marker based capture system ³. Marker data for



Figure 3: Sequence from motion capture for reaching with both hands

the various joints were processed using BodyBuilder software and BodyLanguage scripting language of Vicon Metrics.

The motion capture data included movements involving the shoulders, elbows, and hips. The Movements consisted of reaching to various target positions, vertical and horizontal painting-like movements, and trajectory tracking including figure-8s, circles, flowers, etc. Each of the movements lasted 3-6 seconds. A total of about 200 seconds of capture data were available for the experiments. For our experiments we only used the data from the shoulder DOFs, and only those from the reaching movements. A sequence of snapshots of a reaching movement is shown in Figure 3.

Synthetic motion data were generated for simple aerobic-style movements involving all the DOFs of both shoulders. The movements involved stretching the arm sideways, taking it to a vertically upright position, etc. Each synthetic movement lasted about half a second. A set of snapshots from one such movement sequence is shown in Figure 2.

4.3 Experimental Design

For the experiments involving synthetic data, a total of 8 aerobic-style sequences were taken and zero mean Gaussian noise was added to the participating DOFs. The movements in the sequences involved transition between different poses of either arm. For the learning system implementation, two base primitives were used; each was assigned to the 3 shoulder joint DOFs, one primitive per shoulder. The movement specializer layer had a total of 30 specializers. These were divided equally between the two base primitives. A total of 10 sequence learners were implemented as HMMs with the scaled implementation for forward and backward variables, and log probability for the observation probability. These were implemented using the methods from Rabiner [14].

Each experiment involved the presentatin of the aerobic-style sequences, with a clear start and end markers for each sequence. Learning in the movement specializ-

¹Critical Mass Labs, http://www.cm-labs.com

²Vicon Motion Systems, http://www.vicon.com

³Motion capture data were generously provided by J. Hodgins, CMU.

ers occured on-line as the demonstration was in progress. The sequence learners were presented with the sequence of active movement specializers at the end of each demonstration, and updated so as to learn that sequence.

The human motion capture data consisted of presegmented short movements, such as reaching, not of sequences (such as the aerogic movements of the synthetic data). Thus, while synthetic data were used to evaluate the sequence learning ability of the architecture (the second and third layers of the system), the motion capture data were used for the validation of the first, base primitive layer of the system. Specifically, the human motion capture data were used to test how corectly and accurately the base primitives (using their combinations of linear models) were able to recognize and reconstruct the observed, entirely novel movements.

5 Results

Using the synthetic aerobic-style input data, the system was run for over 100 presentations of each of the synthetically generated movement sequences. The recognition performance of the system for these 100 trials is summarized in Table 1. During each trial, the winning sequence learner for each movement sequence was recorded. After the entire set of 100 trials was performed, we identified the sequence learner, for every movement sequence, which occurred the most times for that sequence presentation. This was taken to be the sequence learner representing the corresponding movement demonstration. All other instances of sequence learners associated with the particular presentation were considered to be incorrectly recognized. Thus, the table summarizes the percentage of correct sequence recognitions as per the criteria above.

This evaluation is reasonable because sequence specialization and recognition is an unsupervised process in our system. Hence, there is no definite "correct" label associated with each sequence learner. Consequently, the labeling of the sequence learners and their evaluation is based on their association with particular presentation sequences.

Sequence	Recognition (%)
1	94
2	93
3	98
4	99
5	100
6	98
7	88
8	92

Table 1: Synthetic data performance

Using the human motion capture data, Figure 4 shows

movement reconstruction plots obtained from the linear models at the base primitive layer. The graphs show the original data for the three DOFs of the left shoulder, their corresponding reconstructions, and the respective error plots. We note that this does not provide a substantial evaluation of the reconstruction. The absence of an execution stage in the implementation limits us from providing a better evaluation. The execution module is now under development.



Figure 4: Plots of the shoulder joint angles for target reaching, from motion capture(left column), the reconstruction from the base primitive linear models(center column), and the error(right column). The x-axis denotes time, and the y-axis the angle.

6 Related work

As mentioned before, imitation has gained increasing popularity in several fields, and thus the body of interdisciplinary related work is much too large to properly summarize in the scope of this paper. We provide only a brief summary of the work most similar to ours in terms of biological motivation and/or resulting methodology.

Schaal et al [18] have demonstrated the use of oscillatory and discrete pattern generators, in combination, for various tasks like ball bouncing, drumming, "3-D" drawing patterns, etc. They used separate oscillators for each joint and a reference oscillator for coordination. Discrete movements were superimposed for positioning. Ijspeert et al [1] proposed the idea of using mixtures of nonlinear differential equations to represent kinematics of movements, thereby effecting trajectory generation. Bentivegna et al [2] used the idea of primitives for motor learning. They apply the idea for learning to play air hockey and marble maze in simulation and on a real robot. Billard et al [3, 4] used connectionist-based approaches to represent movements. They make use of a recurrent connectionist network that is able to learn oscillatory movements, and also discrete movements as a special cases of oscillatory ones. Fod et al [7] automatically derived primitives through an off-line process of segmentation and the application of principal component analysis.

The approaches above have typically dealt with the problem of finding appropriate representations for primitives, and possible means to learn them. In this work our approach has been to assume the presence of a set of base level primitives, and then design a learning system that can make use of those for learning more complex, sequential skills.

7 Conclusion

We presented an architecture capable of using an underlying substrate of primitives to learn and represent movement sequences. This is part of our ongoing work in using the notion of primitives to model motor control and learning by imitation. Future work will involve incorporating a motor counterpart to the architecture to allow execution of movements and thus more realistic evaluation.

Acknowledgements

This work was supported by DARPA Grant DABT63-99-1-0015 under the Mobile Autonomous Robot Software (MARS) program.

References

- Ijspeert A.J., Nakanishi J., and Schaal S. Trajectory formation for imitation with nonlinear dynamical systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (To appear). 2001.
- [2] D. Bentivegna and C. G. Atkeson. Using primitives in learning from observation. *First IEEE-RAS International Conference on Humanoid Robots (Humanoids* 2000), 2000.
- [3] A. Billard and Maja J. Matarić. Learning human arm movements by imitation: Evaluation of biologicallyinspired connectionist architecture. In *Robotics & Autonomous Systems*, (In press). 2001.
- [4] Aude Billard and Maja J Matarić. Betty: Robot, play with me! robot: O.k. how do we play? betty: You watch me and do like i do. look! In Proceedings, Workshop on Interactive Robotics and Entertainment (WIRE-2000), Pittsburgh, April 30-May 1 2000.
- [5] Gail A Carpenter and Stephen Grossberg. ART 2 : self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26(23):4919–4930, December 1987.

- [6] J. Duysens and W. A. A. van de Crommert. Neural control of locomotion; part 1: The central pattern generator from cats to humans. gait and posture. *Gait* and Posture, 7(2):131–141, 1998.
- [7] Ajo Fod, Maja J Matarić, and Odest Chadwicke Jenkins. Automated derivation of primitives for movement classification. *First IEEE-RAS International Conference on Humanoid Robotics (Humanoids-*2000), MIT, Cambridge, MA, 2000.
- [8] S. F. Giszter, F. A. Mussa-Ivaldi, and E. Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, 1993.
- [9] M. Iacoboni, R. P. Woods, M. Brass, H. Bekkering, J. C. Mazziotta, and G. Rizzolatti. Cortical mechanisms of human imitation. *Science*, 286:2526–2528, 1999.
- [10] Maja J. Matarić. Visuo-motor primitives as a basis for learning by imitation. In Kerstin Dautenhahn and Chrystopher Nehaniv, editors, *Imitation in Animals* and Artifacts. MIT Press, 2001.
- [11] Ádám Miklosi. The ethological analysis of imitation. *Biological Review*, 74:347–374, 1999.
- [12] Mussa-Ivaldi, F. A. Giszter, and E. Bizzi. Linear combination of primitives in vertebrate motor control. *Proc. Nat. Acad. Sci. USA*, 91:7534–7538, 1994.
- [13] Monica Nicolescu and Maja J. Matarić. Learning and interacting in human-robot domains. Special Issue of IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 31(5):419–430, 2001.
- [14] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition, 1989.
- [15] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, January:4–16, 1986.
- [16] G. Rizzolatti, L. Gadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.
- [17] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.
- [18] S. Schaal and D. Sternad. Programmable pattern generators. 3rd International Conference on Computational Intelligence in Neuroscience, Research Triangle Park, NC, pages 48–51, 1998.