Incremental Learning and Memory Consolidation of Whole Body Human Motion Primitives

Dana Kulić¹, Yoshihiko Nakamura²
¹ Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada
² Department of Mechano-Informatics, University of Tokyo, Japan

The ability to learn during continuous and on-line observation would be advantageous for humanoid robots, as it would enable them to learn during co-location and interaction in the human environment. However, when motions are being learned and clustered on-line, there is a trade-off between classification accuracy and the number of training examples, resulting in potential misclassifications both at the motion and hierarchy formation level. This article presents an approach enabling fast on-line incremental learning, combined with an incremental memory consolidation process correcting initial misclassifications and errors in organization, to improve the stability and accuracy of the learned motions, analogous to the memory consolidation process following motor learning observed in humans. Following initial organization, motions are randomly selected for reclassification, at both low and high levels of the hierarchy. If a better reclassification is found, the knowledge structure is reorganized to comply. The approach is validated during incremental acquisition of a motion database containing a variety of full body motions.¹

Keywords incremental learning · motion primitives · whole body motions · humanoid robots

1 Introduction

As robots move to human environments, the ability to learn and imitate from observing human behavior will become important. This area of research has received considerable attention in the literature (Breazeal & Scassellati, 2002; Schaal, Ijspeert, & Billard, 2003). However, many of the approaches proposed thus far consider the case where there is an off-line initial learning stage, followed by a static execution and recognition stage. In this case, the number of motions to be learned can be specified a priori, and the designer can ensure that each motion example is correctly labeled to the appropriate motion group. However, to ensure adaptability to the changing environment as well as the possibility of changing interaction partners, a robot should be capable of continuous learning over its entire lifespan. We have been working toward algorithms that enable the robot to observe, segment and classify demonstrated actions on-line (Kulić, Takano, & Nakamura, 2007a, 2008a, 2008b) during co-location and interaction with the (human) teacher. During this type of learning, the number of motion primitives is not known in advance and may be continuously growing, and must be determined autonomously by the robot, as it is observing the motions. In addition, as the number of observed motions and learned motion primitives increases, the robot must be able to organize the
acquired knowledge in an efficient and easily searchable manner.

In the proposed approach (Kulić et al., 2007a; Kulić, Takano, et al., 2008a, 2008b), a hierarchical tree structure is incrementally formed representing the motions learned by the robot. Each node in the tree represents a motion primitive, which can be used to recognize a similar motion, and also to generate the corresponding motion for the robot. Because motions are being sorted incrementally, the algorithm may produce errors as compared with off-line clustering and organization. Two types of errors are possible: errors in classifying individual motions and errors in the structure formation of the motion hierarchy. However, it can still be advantageous for the robot to quickly learn a rough model of a new motion, rather than waiting for a large number of examples to become available, as the learned motion may then be further refined through other learning modalities, such as practice (Bentivegna, Atkeson, & Cheng, 2006) and feedback from the teacher (Nicolescu & Matarić, 2005), which may be more effective than repeated observation alone.

In this article, we propose a mechanism for incremental, on-line correction of initial clustering and organization errors during on-line motion observation and learning. Following initial model formation, the model is changed over time to stabilize and improve the initial representation. Our approach is based within the scope of statistical modeling of human whole-body motion patterns for on-line and long-term learning. This goal leads us to consider the knowledge and literature on the human memory system. Our design of the computational architecture is motivated by recent biological studies, which show that the structure of motion memory changes following acquisition, in a process termed memory consolidation (Krakauer & Shadmehr, 2006; Shadmehr & Holcomb, 1997).

1.1 Related Work


Learning from imitation is a complex problem, with many open challenges which need to be addressed before a functional system can be deployed in the human environment. Alissandrakis, Nehaniv, and Dautenhahn (2002b) identify the main problems which must be solved by the imitator as follows: who to imitate, when to imitate, what to imitate, how to imitate, and how to evaluate a successful imitation. In particular, these last three issues have received significant attention in the literature.

What to imitate addresses the issue of which aspect of the demonstrator’s behavior should be reproduced. The selected aspect may differ with the type of task being demonstrated, for example, for dance movements, joint angles should be imitated, whereas for assembly tasks, object positions, rather than exact arm movements, may be more important. The question of which task representation may be optimal has also received consideration in the biological research, particularly for the case of reaching movements (Bekkering & Pratt, 2004; Mussa-Ivaldi, Hogan, & Bizzi, 1985). Billard, Calinon, and Guenter (2006) propose an approach for autonomously determining what to imitate based on the variance in each signal type across demonstrations.

When the number of observed variables in the environment is potentially very large, it may not be possible to extract the variables to be imitated quickly through statistical means. Demiris and Khadhouri (2006) describe an approach for directing the attention of the imitator to the appropriate aspect of the demonstration using the action models themselves. In their approach, actions are modeled as forward/inverse models, which can be used both for action recognition and generation. During the perception of a demonstration, action models are active in parallel, and request state information required for that action from the perception system. The priority of the requests is proportional to the level of similarity that the given action model has with the current demonstration. Therefore the attention of the imitator will be directed to those perceptions with the strongest corresponding action hypothesis, providing a principled way to provide top down control of how the observer’s attention should be focused.

A second aspect of the what to imitate question concerns the level of hierarchy to be imitated, that is, should the imitator be learning the details of the motion primitives being demonstrated, or the correct
procedure for combining and/or sequencing the primitives. Both levels of hierarchy have been considered in the literature; for example, Billard et al. (2006), Nakanishi et al. (2004), Inamura, Toshiba, Tanie, and Nakamura, (2004), and Dillmann et al. (1999) consider the imitation of the motion primitives themselves, while Breazeal, Berlin, Brooks, Gray, and Thomaz (2006), Lockerd and Breazeal (2004), Nicolescu and Matarić (2001), and Dominey, Metta, Nori, and Natale (2008) consider how a system may learn a sequence of primitives which are defined a priori. Takano, Yamane, Matarić, Nehaniv, and Dautenhahn (2002a) propose an algorithm for learning the appropriate correspondence by building a library of known correspondences and using the imitator’s own history of imitations. Billard et al. (2006) solve the correspondence problem by formulating an optimization problem to find the trajectory that best fits the demonstrator data, subject to the robot’s kinematic constraints. Alissandrakis, Nehaniv, and Dautenhahn (2007) introduce a generic approach for the correspondence problem, in terms of body mapping for both state (joint position) and action (change of state) matching, based on the use of correspondence matrices between the degrees of freedom of the imitator and demonstrator. Correspondence matrices can be designed based on the task and embodiment of the actors to perform partial, mirror, or a variety of other mappings. Imitation metrics can then be defined making use of the correspondence matrices, also addressing the issue of how to evaluate a successful imitation.

It is also important to note that imitation is only a subset of the learning skills that may be employed by the learner. In addition to learning from observation, the learner may also learn through self-observation or practice (Erlhagen et al., 2006; Keysers & Perrett, 2004), interaction with the teacher (Dominey et al., 2008; Lockerd & Breazeal, 2004; Nicolescu & Matarić, 2005) or a combination of these approaches (Bentivegna et al., 2006).

One aspect of the learning problem which has received less attention in the literature has been the question of temporal extent over which the learning takes place. As noted by Breazeal and Scassellati (2002), the majority of algorithms discussed in the literature assume that the motions to be learned are segmented a-priori, and that the model training takes place off-line. Within these off-line approaches, several different techniques have been applied to modeling the motion primitives, including dynamical models (Nakanishi et al., 2004), neural networks (Ogata, Sugano, & Tani, 2005) and stochastic models (Billard et al., 2006; Inamura et al., 2004; Taylor, Hinton, & Roweis, 2006).

Dehaene, Changeux, and Nadal (1987) describe an early connectionist approach for learning temporal sequences, modeled on the observations of the acquisition of song by birds. In the proposed approach, elementary devices are used consisting of synaptic triads which yield short term modification of synaptic efficacy through heterosynaptic interactions based on a local Hebbian learning rule. A three-layer network constructed from such elementary devices exhibited spontaneous productions of pre-representations which
Hidden Markov models have been a popular technique for human motion modeling, and have been used in a variety of applications, including skill transfer (Dillmann et al., 1999; Yang, Xu, & Chen, 1997), robot assisted surgery (Ekvall, Aarno, & Kragic, 2006; Kragic, Marayong, Li, Okamura, & Hager, 2005), sign language and gesture modeling (Bernardin, Ogawara, Ikeuchi, & Dillmann, 2005; Iba, Paredis, & Khosla, 2005; Startner & Pentland, 1995) and motion prediction (Bennewitz, Burgard, Cielniak, & Thrun, 2005; Ho, Yamada, & Umetani, 2005). Developing from the earlier work on **showing by teaching** (Kuniyoshi & Inoue, 1994), a common paradigm is **programming by demonstration** (PbD; Bernardin et al., 2005; Dillmann, 2004; Dillmann et al., 1999; Kang & Ikeuchi, 1993). While PbD is a general paradigm, in many of the systems demonstrated thus far, the number of motions is specified a priori, the motions are clustered and trained off-line, and then a static model is used during the recognition phase.

**Calinon, Guenter, and Billard (2007)** describe a system for programming by demonstration. In their approach, the data, including both joint angle data and absolute and relative object position and hand position data, is first analyzed via principal component analysis (PCA) to determine the relevant subspace for the task (named the latent space). The reduced dimensionality data is then temporally aligned using dynamic time warping (DTW), which is a type of non-linear scaling that minimizes the error between the signals and a reference signal. Once the data has been temporally aligned, it is abstracted into a set of Gaussian mixture models (GMMs). The number of mixtures is selected via the Bayes information criterion. Once the number of mixtures is known, the EM algorithm is used to find the GMM parameters (the prior, mean, and covariance matrix). The GMM also includes the time parameter as an additional output variable. To evaluate the success of the imitation, an evaluation metric is defined in the reduced dimensionality subspace (the latent space), as a weighted square error between the candidate position and the desired position (the desired position is generated by the GMM model). By defining the cost function as a quadratic function, the problem can be reduced to finding an optimum trajectory subject to kinematic and environmental constraints. This system is also extended to an on-line, interactive approach by developing a method for incremental training of the GMM structure (Calinon
& Billard, 2007b) and developing an interactive training approach combining demonstration and kinesthetic training (Calinon & Billard, 2007a).

In order to learn from on-line demonstration, the system must group together similar motions on-line, without a priori specifying the number of motions. While many off-line clustering approaches have been proposed in the literature (for a review see Jain, Murty, & Flynn, 1999), fewer algorithms consider on-line, incremental clustering of multidimensional time-series data. One early example of on-line incremental clustering was the COBWEB algorithm for discrete concept attributes (Fisher, 1987). Zhang, Ramakrishnan, and Livny (1996) proposed a hierarchical algorithm for clustering large data sets incrementally, while explicitly taking the size of available memory into account. However, this algorithm also assumes the data consists of multidimensional points, which can be compared and summed based on the Euclidean or Manhattan distance.

Recently, attention has also turned to the clustering of continuous time-series data, named data streams. Aggarwal, Han, Wang, and Yu (2003) describe an exemplar system for clustering continuous data streams, with a partly on-line approach. However, with their approach, the number of clusters that can be learned is fixed.

Another recent example on data stream clustering is the work of Rodrigues, Gama, and Pedroso (2008) describing the on-line divisive-agglomerative clustering approach for time-series data streams. Unlike most clustering approaches, which attempt to cluster examples, the proposed algorithm tries to group continuous data streams, and is able to handle dynamic changes in the cluster structure. Unlike Aggarwal et al. (2003), this algorithm does not assume a constant, a priori specified number of clusters, and can adjust the number of clusters based on the current characteristics of the data.

However, in the case of motion primitives, the data type to be clustered is not a multidimensional point in Euclidean space or a continuous data stream, but a short segment of multidimensional time-series data. Few clustering approaches have been proposed for this type of data. In addition, we seek to preserve enough information in the cluster description so that the cluster can be used for subsequent motion generation of the motion primitive. One approach is to model each segment as an HMM, and generate a set of distance measures between each model. These distances can then be used to generate a Euclidean space of motions, based on multidimensional scaling (Takano, Tanie, & Nakamura, 2005). Clustering can then proceed in the generated Euclidean space. A similar approach is to convert the motions to Euclidean space using a temporal extension of Isomap (a type of non-linear dimensionality reduction; Jenkins & Mataric, 2004). However, these approaches require that all the data is known before clustering can begin, as the addition of new data can change the structure of the generated Euclidean space.

This article is based on the incremental clustering approach proposed by Kulić et al. (2007a) and Kulić, Takano, et al. (2008b). An on-line, incremental learning algorithm is used to build an initial model of the motion space, as motions are perceived. In this approach, a hidden Markov model based representation is used to abstract motion patterns as they are perceived. Individual motion patterns are then clustered in an incremental fashion, based on intra-model distances. The resulting clusters are then used to form a group model, which can be used for motion generation. Errors in the initial placement of both individual motions and motion groups are then also corrected on-line at a later time, in a process analogous to memory consolidation in humans.

The article is organized as follows: In Section 2, we overview the neuroscience findings on motion primitives and learning, as well as memory consolidation in humans and primates. In Section 3, we provide a review of the initial learning mechanism (initially proposed in Kulić, Takano, et al., 2008b), which incrementally learns motion primitives from observation of human motion. In Section 4, we propose a computational mechanism for implementing memory consolidation for incremental correction and improvement of the initially learned memory structure. Section 5 provides experimental results which validate the proposed approach. In Section 6, we discuss the relationship of the proposed model to biological findings, the current limitations of the model and how the model might be extended to address additional learning strategies and scenarios. Section 7 provides conclusions and directions for future work.
Connections to Biological Models of Motor Memory Formation and Consolidation

Our motion model is a computational realization of the fundamental functionality of the mirror neuron system, found in humans and other primates (Rizzolatti & Craighero, 2004; Rizzolatti, Fogassi, & Gallese, 2001). The mirror neuron system is believed to be a neural mechanism which links action observation and execution. The same neuronal representation is used for both motion recognition and motion generation. Similarly, in our model, the same data structure, the factorial hidden Markov model, is used for both motion recognition and motion generation. In macaque monkeys, mirror neurons activate only when goal directed actions are observed, for example, grasping an object, or biting into food, but not when the demonstrator mimics the action without the object being present. However, mirror neurons also activate when the monkey cannot observe the action visually, but other means of inferring the action are available (e.g., sound or previous knowledge), indicating that, in monkeys, the mirror neurons are used for action understanding, and not primarily for imitation learning. For humans, on the other hand, mirror neurons fire for both goal directed actions and for non-goal directed movements (Rizzolatti & Craighero, 2004). In addition, brain imaging studies indicate that human mirror neurons code for both the action and for the movements forming an action. In our model, the same structure encodes both the movement to accomplish the action, and the final goal state, encoded in the final state of the HMM.

These two important differences seem to indicate that in humans, mirror neurons are used both for action understanding and imitation learning. In animal and human studies, two important classes of mirror neurons have been found: strictly congruent and broadly congruent mirror neurons, depending on the specificity of the action being encoded (Gallese, Fadiga, Rogassi, & Rizzolatti, 1996; Rizzolatti & Craighero, 2004). In our approach, the variable structure HMM emulates the mirror neuron function, and the location of the model in the hierarchy emulates the congruence properties of mirror neurons. Leaf nodes (most specific models) correspond to strictly congruent mirror neurons, while upper level nodes correspond to broadly congruent mirror neurons.

A key question regarding motion primitives is the representation used to encode the motor information. This question has received considerable attention in the literature, particularly for the case of arm movements. Early research focused on the question of joint angle versus spatial representation (Mussa-Ivaldi et al., 1985). Neurological and experimental evidence seems to indicate that both extrinsic and intrinsic coordinates are used, with simultaneous optimization at different levels of control (Rosenbaum, Meulenbroek, Jansen, & Vaughan, 2001). More recent research has also considered the location of the origin of the reference frame. Evidence that both space-based (origin centered with the person) and object-based (origin centered with the action target) representations are used has been uncovered (Bekkering & Pratt, 2004). An additional issue is how objects and other environment constraints influence the execution and imitation of motion (Alissandrakis et al., 2002a, 2007; Demiris & Khadouri, 2006; Erlhagen et al., 2006; Petrovskaya & Ng, 2007; Rosenbaum, Meulenbroek, Jansen, et al., 2001; Rosenbaum, Meulenbroek, & Vaughan, 2001). Rosenbaum and colleagues (Rosenbaum, Meulenbroek, Jansen, et al., 2001; Rosenbaum, Meulenbroek, & Vaughan, 2001) postulate that movement primitives are superimposed to handle such a hierarchy of constraints.

A second key question in biology and cognitive science is the model of learning, that is, how do the mirror neurons acquire their mirror-like properties. Heyes and Ray (2000) and Heyes (2001) propose the associative sequence learning (ASL) theory of imitation. They postulate that learning is based on action units, which are the basic units of the majority of actions being observed. Learning proceeds via two sets of associative processes, resulting in horizontal and vertical links. The horizontal process forms sequence associations between sensory representations of the action units forming the demonstrated action. The vertical process forms associations between the sensory representation for each action unit and the associated motor representation of the same component. Meltzoff (2002, 2005) proposes the active intermodal mapping theory, which postulates that infants possess an innate ability to map from perception to motor acts, and that initially captured motions get progressively refined through repeated practice. Based on a review of the neuroscience findings for macaque monkeys, Keysers and Perrett (2004) also propose an associative process.
based on Hebbian learning, but postulate that for hand movements, the training process is achieved through self-observation, while imitation games with the care-taker are primary for mouth and face movements. Öztop and Arbib (2002) hypothesize that the mirror neuron system initially evolved to provide appropriate hand-state feedback during grasp selection. The mirror neuron system was then later also adapted for understanding the actions of others by generalizing from one’s own hand to another’s hand. They also develop a computational model for the visual neural processing, reach and grasp motor areas, and the core mirror neuron system in grasping, based on associative learning. These theories all imply that the development of the imitation mechanism is highly experience dependent, consisting of correlation links between sensory and motor data which are formed over time.

As also noted by Calinon and Billard (2007c), our approach of modeling the sensory data flow as a set of state vectors linked temporally with a stochastic model (the HMM) and obtained incrementally over time, based on the experiences observed by the robot, conforms with these theories. The HMM model learning corresponds to the horizontal process of the ASL model, which forms sequence associations between sensory representations of the action units.

However, these theories of imitation learning from observation of action primitives are not universally accepted. Byrne and Russon (1998) postulate that many of the animal behaviors classified as imitation can more accurately be explained by priming mechanisms, such as stimulus enhancement, emulation and response facilitation. They argue that the main imitation mechanism used by primates and humans is not at the action level, but rather at the program level, that is, primates learn to imitate the efficient organization of the actions, while the individual actions comprising the complex behavior are learned by other means, such as trial and error. They argue that while action-level imitation does occur, for example during mimicry and play, it is not an efficient way of learning, and its purpose is primarily social rather than functional. Similarly, Wohlschlaeger, Gattis, and Bekkering (2003) propose the goal directed imitation (GOADI) theory, which proposes that imitation is guided by cognitively specified goals. According to the theory, the imitator does not imitate the exact movements produced by the demonstrator, but instead extracts from the demonstrator a hierarchy of goals and subgoals present in the task.

These goals and subgoals are then accomplished by motor programs already in the imitator’s repertoire which are most strongly associated with the achievement of the given goal. However, Hayes, Ashford, and Bennett (2008) describe experiments on human subjects which seek to elucidate circumstances under which the means and not only the goal is imitated. In their study, 10- and 11-year-old children are asked to imitate a juggling behavior. The motor acts of the juggling behavior are highly specific, and not already in the learners’ motor repertoire. A control group is given only verbal instructions on how to perform the juggling act. The experimental group achieved significantly better results in being able to successfully execute the task. Their results show that when the task to be learned is not fully constrained by the end-goal, and is not achievable by already known movements, learners have a difficult time learning through trial and error alone, and learn far better by copying observed movements. In our model, the means and goal are given equal weight during recognition and learning, however, the proposed framework could be adapted to consider the goal preferentially by using only the final state during motion comparison, rather than the entire trajectory.

In addition to the learning at the motion primitive level, higher order learning processes have also been studied. These studies consider the question of how motion primitives may be combined to form more complex actions and behaviors. Byrne (1999) presents the string parsing theory of imitation, and argues that this type of imitation can account for a large proportion of the learning observed in animals. In this type of learning, the animal observes many examples of the task to be learned, and acquires a program-level plan for the task based on statistical correlation learning. In order to achieve this type of learning, the animal must be able to decompose the observed time sequence into a series of primitives, either in terms of behavior or environment states. It is postulated that the mirror neurons serve to provide the behavior primitive segmentation. If the temporal sequence can be decomposed into a series of primitives, the structure, consisting of goal and subgoal states, will statistically emerge through multiple observations of a hierarchically organized task. There is also evidence from neuroimaging studies (Chaminade, Meltzoff, & Decety, 2002) that goal and action learning are coupled, and that imitating a gesture also activates neural processing
to infer the intention (or goal) underlying the observed action.

A second area of study is the neural processes occurring during and immediately following learning. Neuroscience studies of motor memory formation have shown that following learning, the motor memory does not remain constant in the brain, but rather changes over time in a process termed memory consolidation (Diekelmann & Born, 2007; Krakauer & Shadmehr, 2006; Shadmehr & Holcomb, 1997; Stickgold, 2005). There appear to be two complementary memory consolidation processes: an initial stage of stabilization, through which the memories become resistant to interference, and a second stage, where system wide reorganization is performed (Diekelmann & Born, 2007; Stickgold, 2005). The first stage (the waking stage) occurs in the hours immediately following motor memory acquisition, while the subject is awake, and can be detected by evidence of disruptions corrupting the memory during a limited time window following initial acquisition (Krakauer & Shadmehr, 2006). Krakauer and Shadmehr (2006) review the literature on this type of consolidation of motor memory. The second stage (sleep-dependent stage) occurs during sleep, and can be detected by measuring performance improvements following sleep, without any further practice of the motion (Diekelmann & Born, 2007; Stickgold, 2005). During this type of consolidation, brain imaging studies show that brain regions active during memory formation are repeatedly reactivated during sleep as the motion representation is gradually redistributed to different networks and brain regions, thought to signify a system-level consolidation (Diekelmann & Born, 2007). Neuroscientists also hypothesize that the content of dreams reflects aspects of memory consolidation taking place during the different stages of sleep (Payne & Nadel, 2004).

Motor performance can also be improved through the process of mental practice, also called motor imagery or rehearsal. Mental practice is defined as the covert, mental rehearsal of a task in the absence of actual physical rehearsal (Driskell, Copper, & Moran, 1994). This type of rehearsal is often used to aid athletic performance, and has also been proposed for use in rehabilitation (Jackson, Lafleur, Malouin, Richards, & Doyon, 2001; Liu, Chan, Lee, & Hui-Chan, 2004). Analysis of the effects of mental training show that mental training using motor imagery improves global motor performance, such as muscular strength (Yue & Cole, 1992), reduction of variability, and increases in temporal consistency (Driskell et al., 1994; Jeannerod, 1995). The mechanism by which mental rehearsal improves performance is not clearly understood, but researchers postulate that mental rehearsal improves performance by training the higher level models of motor muscle activation patterns, that is, learning the internal model of an action (Jeannerod, 1995; Yue & Cole, 1992).

There have also been many studies concerned with modeling the underlying computational processes governing memory formation and learning. McClelland, McNaughton, and O’Reilly (1995) investigate the roles of the hippocampus and the neocortex during learning, and propose a computational model of the learning process. They postulate that there are two complementary processes for learning in the brain: the neocortex and the hippocampus. The neocortex learns very slowly, with many examples, while the hippocampus learns a compressed version quickly, with few examples. They show that in order to correctly learn the correlation, the learning rate must be slow, gradual (gradient based) and requires many examples. They also show that the presentation of the learning examples must be interleaved to avoid catastrophic interference. On the other hand, the hippocampus is a system for fast learning with few examples. The hippocampus is used to provide the neocortex with additional examples of the concept to be learned in a gradual, interleaved fashion, during off-line consolidation. McClelland et al. (1995) highlight the importance of interleaved learning, whereby a particular item is not learned instantly, but is acquired gradually, through a series of presentations interleaved with exposure to other items.

In our approach, following observations of similar motions, a model of the abstracted motion is learned, corresponding to the waking stage. During this stage, an initial model of the motion primitives and the motion primitive hierarchy is formed, which may not be the optimal organization of memory, and may be susceptible to errors in clustering as a result of presentation order. Then, at a later time, system-wide reorganization is performed by selecting learned motions and reclassifying, therefore correcting initial clustering and hierarchy formation errors, in a process analogous to rehearsal memory consolidation found in the sleep-dependent stage. In this second stage, the system repeatedly reactivates the clustering and recognition process on previously observed patterns. Since, during this
stage, the system can select motions to be rehearsed in an arbitrary manner, motions can be presented in an interleaved manner, allowing the system to reduce its susceptibility to presentation order, and optimize the memory storage structure and each motion primitive model over time.

In this article, we focus on memory consolidation of motion primitives encoded as HMMs, however, the fundamental idea is general and applicable to a wide range of learning tasks which occur continuously and cannot be performed in an off-line fashion, after all the data has already been collected. Namely, the initially learned structure, learned when less data was available, can be fine tuned and improved by an incremental process, which takes previously observed data and repeatedly reapplies the same learning algorithm used initially on the data at a later time. This process allows the system to refine and correct the cluster boundaries once more information is available, and also to reduce the influence of presentation order, by allowing the system to be exposed to interleaved learning even when the system may not have access to such a presentation order in the actual environment.

3 Initial Incremental Behavior Learning

In this section, we provide a brief review of the initial motion primitive learning system, which is used as the base for applying an implementation of computational memory consolidation, as described in the following sections. Here, we provide an outline of the system necessary for developing the discussion in the following sections; additional details and analysis of the basic algorithm performance is provided in Kulić, Takano, et al. (2008b). In the proposed framework (Kulić, Takano, et al., 2008a, 2008b) the task of the learning system is to autonomously extract and learn motion primitives from time-series data obtained through on-line observation of a human demonstrator. A motion primitive is defined as full-body motion segment (i.e., an action unit; Heyes, 2001; Heyes & Ray, 2000) which is reused multiple times during task or behavior execution. A motion primitive may be described in terms of the joint or Cartesian coordinates. In this article, we focus on the learning of the motion primitives themselves, and do not address the issue of temporal segmentation. In a full real-time application, motion primitives can be segmented autonomously from the continuous time-series data stream via stochastic segmentation (Kulić & Nakamura, 2008b).

Each time a new motion sequence is observed, the robot must decide if the observed motion is a known motion, or a new motion to be learned. In addition, over the lifetime of the robot, as the number of observed motions becomes large, the robot must have an effective way of storing the acquired knowledge for easy retrieval and organization. In the proposed approach, a hierarchical tree structure is incrementally formed representing the motions learned by the robot. Each node in the tree represents a motion primitive, which can be used to recognize a similar motion, and also to generate the corresponding motion for the robot.

An overview of the algorithm and the incremental hierarchy formation is shown in Figure 1. The algorithm initially begins with one behavior group (the root node). Each time a motion is observed from the teacher, it is encoded into a hidden Markov model (Figure 1a). The encoded motion is then compared with existing behavior groups via a tree search algorithm, using the symmetric model distance measure (Kulić, Takano, & Nakamura, 2007b; Rabiner, 1989; see Figure 1b), and placed into the closest group (Figure 1c). Each time a group is modified, a hierarchical agglomerative clustering algorithm (Jain et al., 1999) is performed within the exemplars of the group (Figure 1d). If a cluster with sufficiently similar data is found, a child group is formed with this data subset (Figure 1e,f). The time-series data of the motion examples forming the child group is then used to generate a single group model, which is subsequently used for both behavior recognition and generation. Therefore the algorithm incrementally learns and organizes the motion primitive space, based on the robot’s lifetime observations. The algorithm pseudocode is shown in Figure 2.

Each newly acquired observation sequence is encoded into a hidden Markov model. Each motion is initially represented by a simple, single-chain, front-to-back HMM. If a better model is required, additional chain(s) are added as described below.

Once the newly observed behavior is encoded as an HMM, it is compared with existing groups (if any). Here, the distance between two models can be calculated (Rabiner, 1989) by Equation 1.

\[
D(\lambda_1, \lambda_2) = \frac{1}{p} \left[ \log P(O^{(2)}|\lambda_1) - \log P(O^{(2)}|\lambda_2) \right]
\] (1)
where $\lambda_1$, $\lambda_2$ are two models, $O^{(2)}$ is an observation sequence generated by $\lambda_2$, and $T$ is the length of the observation sequence. Because this measure is not symmetric, the average of the two intra-distances is used to form a symmetric measure.

The repository of known groups is organized in a tree structure, so that the new observation sequence does not need to be compared with all known behaviors. The comparison procedure is implemented as a tree search. If the distance between the new observation and the cluster is larger than a threshold based on the maximum intra-observation distance $D^G_{\text{max}}$, this cluster will not be considered as a possible match for the new observation sequence. $D^G_{\text{max}}$ is the largest distance between existing observations in the cluster, based on the distance metric as defined in Equation 1. If there are multiple candidate clusters, the new sequence is placed in the closest cluster. If there are no candidates, the new sequence is placed in the parent cluster. In the case of a new motion pattern which is completely dissimilar to any existing motion patterns, the motion pattern will be placed into the root node.

When a new observation sequence is added to a group, a clustering procedure is invoked on that group, to determine if a subgroup may be formed. The complete link hierarchical clustering algorithm is used to generate the hierarchical tree structure within a group (Jain et al., 1999). Clusters are formed based on two criteria: number of leaves in the subgroup and the maximum proximity measure of the potential subgroup.

The maximum distance measure is based on the average of the inter-motion distances in the cluster:
where \( D_{\text{cutoff}} \) is the distance cutoff value (i.e., only clusters where the maximum distance is less than this value will be formed), and \( \mu \) is the average distance between observations. \( K_{\text{cutoff}} \) is a free parameter that determines how similar motions must be to each other before a cluster is formed. The selection of this parameter is discussed in more detail in Sections 4 and 5.

If a new subgroup is generated in Step 5, a new group model is trained using the raw observation sequences from all the group elements. The structure of the new group model is determined based on the maximum intra-observation distance for group, \( D_{\text{max}}^G \). The generated model is subsequently used by the robot to generate behaviors. The group model replaces the individual observations in the parent node.

The group model for the newly formed node represents the abstraction of the motion primitive. Since HMM is a generative model, the same model used for motion recognition is also used for motion generation (synthesis). When the generated motion sequence is to be used for robot motion commands, we do not want to introduce the noise characteristics abstracted by the HMM model. In this case, we use a greedy policy to estimate the optimum state sequence, by greedy sampling from the state transition model. At each state, the state duration is calculated based on the state transition matrix. Once the state sequence has been generated for each chain, the output sequence is calculated by summing the contribution from each chain at each time step, based on that chain’s current state value. Alternatively, if a motion most similar to a recently observed motion is required, the optimal state sequence could be generated by using the Viterbi algorithm, as described in Lee and Nakamura (2005). After the trajectory is generated, some low-pass filtering or smoothing is required as a post-processing step to eliminate the artifacts caused by discrete state switching and generate a smooth trajectory for use as a command input.

### 4 Incremental Memory Consolidation

The incremental clustering algorithm described above has been shown to produce reliable results, robust against presentation order. In all the experiments performed thus far, no false-positive errors have been reported at the leaf node (Kulić, Takano, et al., 2008b). However, depending on the presentation order, two types of errors can occur: false-negative errors and tree-structure errors. False-negative errors occur when the behavior is not classified at the correct hierarchy level, and tree-structure errors occur when the tree structure is not identical to the tree structure that would be observed during off-line clustering. These types of errors occur as a result of the incremental nature of the algorithm, where not enough information is available during early execution, when there are few examples, to find the correct segmentation boundary. In previous work, we focused on correcting false negative errors at the leaf nodes via the use of adaptive hidden Markov chains (Kulić, Takano, et al., 2008b). In the previous approach, a conservative threshold is selected for node formation, and similar motions at the leaf nodes are distinguished by using improved models. This results in a flatter tree structure and a reduced false-negative error rate. However, there are two major drawbacks with this approach: (a) the conservative threshold delays the formation of new nodes, which places increased burden on the human demonstrator, by requiring additional demonstrations before a primitive is learned; and (b) the resulting tree structure is not deep, thus the learned structure does not reveal much about the relationships between behaviors. If a deeper tree structure could be formed, the parent node would be more similar to a new child node, therefore allowing information from the parent node to be used to accelerate learning.

Herein, we propose to use a low threshold for node formation, such that nodes are formed quickly, and the resulting tree structure is deeper. As the presentation order is arbitrary, such an approach will lead to initial clustering errors. However, as more data become available, these initial mistakes can also be corrected in an incremental, on-line fashion, analogous to memory consolidation in biological systems. The process is carried out by reapplying the incremental clustering procedure multiple times to data that is already known, at a later, possibly off-line stage. This can be thought of as a type of rehearsal, similar to the idea proposed by Ogata et al. (2005). Similar to McClelland et al. (1995), we seek to improve performance through the benefits of interleaved learning. However, unlike the biological systems modeled by McClelland et al. (1995), in the proposed algorithm, a single structure is used to
implement both the fast learning and the slow learning mechanisms, and these processes operate concurrently on the same data structure.

Two types of corrections are proposed: corrections at the local level, dealing with misclassifications of individual exemplars, and corrections at the tree level, dealing with errors in structure formation. By considering corrections at both the local and more global level, unlike previous incremental clustering approaches (Fisher, 1987; Rodrigues et al., 2008), our algorithm is less prone to becoming trapped in local minima. For both types of corrections, the basic process is the same: a model is selected from the knowledge base, removed from its current location in the knowledge structure, and the incremental clustering algorithm used for classifying new motion, described in Section 3, is reapplied. In the case of local correction, an individual exemplar model is selected, while in the case of structure correction, a group model is selected. Various strategies can be considered for the selection of the next node to be analyzed for potential correction, for example, favoring recently classified models, or favoring those models where the group variability is large. In the experiments described below, a simple random selection strategy is employed. Figure 3 outlines the consolidation algorithm for exemplars, while Figure 4 shows the consolidation algorithm for the group models.

```
1: procedure MotionConsolidation
2: Select sample node $n_s$
3: Select sample motion $m_s$ from $n_s$
4: $n_b =$ TreeSearch($m_s$) Search for the best match for the sample motion
5: if $n_b = n_s$ then
6: return (No correction required)
7: else
8: Remove $m_s$ from $n_s$
9: if deletable($n_s$) then
10: if few motions remain in $n_s$, and none are group motions
11: Remove all $N$ motions from $n_s$
12: Delete $n_s$ from tree
13: for $i = 1 : N$ do
14: Call IncrementalCluster($m_i$)
15: end for
16: else
17: Call IncrementalCluster($m_s$)
18: end if
19: end if
20: end procedure
```

**Figure 3** Memory consolidation at the exemplar level.

```
1: procedure NodeConsolidation
2: Select sample node $n_s$, with parent node $n_p$
3: Remove $n_s$ from $n_p$
4: $n_b =$ TreeSearch($n_s$) Search for the best match for the group model of the node
5: if $n_b = n_p$ then
6: Return $n_s$ to $n_p$
7: return (No correction required)
8: else
9: Add $n_s$ to $n_b$
10: end if
11: end procedure
```

**Figure 4** Memory consolidation at the node level.
Exemplar reconsolidation can also trigger changes in the tree structure, if a node is emptied of exemplars as a result of reconsolidations. In this case, the node is removed from the tree, and the remaining exemplars reassigned by a call to the incremental clustering procedure.

During this process, the recognition of a sampled motion is carried out repeatedly through the use of the tree search algorithm, identical to the process which occurs during observation of a new motion. If a node is modified by the inclusion or removal of a motion or subnode, the node model is retrained with the modified data set and a new group model is generated, identical to the learning process that occurs when a node is modified as a result of the addition of a newly observed motion. Thus, the proposed process is analogous to memory rehearsal which occurs in the human brain, where the same neural areas active during learning are repeatedly activated during rehearsal and consolidation.

The computational cost of the proposed memory consolidation approach is dependent on the hierarchy level at which the consolidation occurs. For memory consolidation of individual exemplar motions, after selecting a motion to reconsolidate, the time required is identical to the time required to analyze a new motion. Since, in general, initial motion incorporation requires less time than the observation of a typical motion (Kulić, Takano, et al., 2008b), exemplar memory consolidation could be performed on-line, interleaved with new motion observation. This is because the influence of a single exemplar on the group model is small, so if an individual exemplar is reassigned to a new node, the node and parents of the previous parent nodes do not need to be updated immediately, and can be updated the next time the node is modified. On the other hand, at the node consolidation level, a child node has significant impact on the model of the parent node, since it contains multiple exemplars. If a node is removed from its previous parent node, a significant change in the parent model could take place, so nodes leading back from the old parent node to the root must be relearned. Therefore the time required to perform node consolidation will be dependent on the tree depth, and may require idle time (i.e., time during which observation of new demonstrations is not taking place) to complete.

5 Experiments

The proposed approach was tested on a data set containing a series of nine different human movement observation sequences obtained through a motion capture system (Kadone & Nakamura, 2005; Kurihara, Hoshino, Yamane, & Nakamura, 2002). The data set contains joint angle data for a 20 degree of freedom humanoid model from multiple observations of walking (WA, 28 observations), cheering (CH, 15 observations), dancing (DA, 7 observations), kicking (KI, 19 observations), punching (PU, 14 observations), sumo leg raise motion (SL, 13 observations), squatting (SQ, 13 observations), throwing (TH, 13 observations) and bowing (BO, 15 observations). Figure 5 shows an example of a walking motion from the dataset.

For this data set, motion primitives were segmented manually from the continuous time-series data, so that the performance of the clustering and memory consolidation algorithm alone could be verified. In a complete real-time system, the motion primitives must be segmented autonomously by the learning system. This can also be achieved with the use of stochastic models (Kulić & Nakamura, 2008b). Segmentation can also be combined with clustering approach proposed herein.
so that known motion primitives are used to improve segmentation performance (Kulić, Takano, et al., 2008a).

In the experiments, the performance of the incremental clustering and hierarchy formation algorithm with and without memory consolidation is compared. Motion sequences are presented to the algorithm in random order. Motion sequences are presented one at a time, simulating on-line, sequential acquisition. After each motion is presented, the incremental clustering algorithm is executed. When including memory consolidation, memory consolidation on one randomly selected node and one randomly selected motion is performed after each 10 new exemplars, and for 100 times at the end of the acquisition. In all of the tests performed, whether using reconsolidation or not, the algorithm correctly segments the behaviors such that the resulting leaf nodes represent the grouping that would be obtained with an off-line method. Out of 100 simulation runs performed, there were no cases of misclassification at the leaf nodes, showing that the final segmentation is robust to presentation order. Here, misclassification is defined as a false positive error. For example, a walk motion being misclassified as a punch motion is a type of false positive error. However, there were cases of false negative errors, where a motion which should have been recognized as a known motion was instead placed into a non-leaf node (for example, the root node).

Sample segmentation results for the non-consolidation algorithm are shown in Figures 6, 7, and 8 (Kulić, Takano, et al., 2008b). The left side of each image shows the final tree structure after all the motions have been presented, while the right side of the image provides a visualization of the clustering result in a 2-dimensional plane. Note that the actual order of node formation will vary depending on the motion presentation order. The average rate of false negative errors

Figure 6  Sample clustering result, $K_{\text{cutoff}} = 1.2$.

Figure 7  Sample clustering result, $K_{\text{cutoff}} = 0.9$. 
and the standard distribution of the false negative errors is shown for each motion in Tables 1 and 2. As noted before, no false positive errors were observed in any of the experiments performed.

The algorithm parameter $K_{\text{cutoff}}$ (the parameter which controls when a new cluster is formed from an existing cluster) determines the resultant tree structure. A high value for $K_{\text{cutoff}}$ (i.e., only clusters composed of a tight data set are formed) tends to result in a flat tree structure (as shown in Figure 6), while low values of $K_{\text{cutoff}}$ result in a deep tree structure, as shown in Figure 7. As the cluster formation parameter is relaxed, deeper trees tend to be formed. However, the resulting tree structure tends to be dependent on the presentation order. In the case of a high cutoff value (see Figure 6), the resulting tree structure is flat, and fairly insensitive to presentation order. The resulting structure is consistent with off-line clustering result. In about 9% of cases, the “dance” group fails to form, in contrast to the off-line clustering result, since this group contains the least examples. At the high cutoff value, the punch and throw motions are too similar to subcluster, resulting in a single hybrid generated motion (indicated as PU/TH in Figure 6; Kulić, Takano, et al., 2008b). This is also indicated in the high false negative error rates for the punch and throw motions in Table 1, as the motions do not tend to be recognized as belonging to a distinct motion type, but are instead placed in the group node. The generated motion resulting from that subcluster is shown in Fig-

---

### Table 1 False negative errors, $K_{\text{cutoff}} = 1.2$. (An error rate of 1 indicates that all motions fail to be recognized).

<table>
<thead>
<tr>
<th>Motion</th>
<th>Average</th>
<th>Std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>0.0068</td>
<td>0.0316</td>
</tr>
<tr>
<td>CH</td>
<td>0.0273</td>
<td>0.1447</td>
</tr>
<tr>
<td>DA</td>
<td>0.0843</td>
<td>0.2725</td>
</tr>
<tr>
<td>KI</td>
<td>0.0621</td>
<td>0.0789</td>
</tr>
<tr>
<td>PU</td>
<td>0.9879</td>
<td>0.0891</td>
</tr>
<tr>
<td>SL</td>
<td>0.0254</td>
<td>0.1247</td>
</tr>
<tr>
<td>SQ</td>
<td>0.0377</td>
<td>0.1332</td>
</tr>
<tr>
<td>TH</td>
<td>0.9900</td>
<td>0.1000</td>
</tr>
<tr>
<td>BO</td>
<td>0.0047</td>
<td>0.0358</td>
</tr>
</tbody>
</table>

### Table 2 False negative errors, $K_{\text{cutoff}} = 0.9$. (An error rate of 1 indicates that all motions fail to be recognized).

<table>
<thead>
<tr>
<th>Motion</th>
<th>Average</th>
<th>Std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>0.0168</td>
<td>0.0786</td>
</tr>
<tr>
<td>CH</td>
<td>0.0407</td>
<td>0.0918</td>
</tr>
<tr>
<td>DA</td>
<td>0.1929</td>
<td>0.3934</td>
</tr>
<tr>
<td>KI</td>
<td>0.0526</td>
<td>0.0876</td>
</tr>
<tr>
<td>PU</td>
<td>0.4093</td>
<td>0.2240</td>
</tr>
<tr>
<td>SL</td>
<td>0.0492</td>
<td>0.1744</td>
</tr>
<tr>
<td>SQ</td>
<td>0.0546</td>
<td>0.1084</td>
</tr>
<tr>
<td>TH</td>
<td>0.5638</td>
<td>0.3585</td>
</tr>
<tr>
<td>BO</td>
<td>0.0360</td>
<td>0.0896</td>
</tr>
</tbody>
</table>
As can be seen in the figure, the motion is an averaging of the two motions.

When low values of $K_{\text{cutoff}}$ are used, nodes are quicker to form, and the resulting tree structure becomes more dependent on presentation order. The similarity level at which nodes will form is highly dependent on presentation order. Figures 7 and 8 show two examples of different tree structures formed, from two simulation runs. Note that the identified leaf nodes remain the same. In addition, using the lower cutoff value makes it easier to subdivide the similar throw and punch motions. This can be seen from the lower false negative rate in Table 2, as significantly more of the punch and throw motions are correctly recognized. Note that some punch throw motions still remain difficult to recognize, and are instead placed in the punch/throw parent node, which is classified as a false negative error in Table 2. Even though the cutoff level was the same for both experiments, the similarity level of the nodes formed differed, based on the presentation order. The result in Figure 7 is consistent with global clustering, while in the result shown in Figure 8, one node is incorrectly assigned. The CH node is incorrectly assigned to the PU/TH/KI/SQ branch of the tree, whereas global clustering would have assigned the CH node to the WA/SL branch. This type of error is the result of the local nature of the algorithm, that is, clustering is being performed when only a part of the data set is available. Therefore, there is a trade-off when selecting the $K_{\text{cutoff}}$ value between facilitating quick node formation and differentiation and introducing misclassifications in the hierarchy tree.

If the relationship between motions is not important for the task, a flat tree result is acceptable, and a high value of $K_{\text{cutoff}}$ can be applied. However, it would be desirable to correctly extract both the leaf node groups, as well as a deeper tree, representing the hierarchy of motions. This tree information could then be used to analyze the relationships between motions and to accelerate learning of new motions belonging to the same branch of the tree. If memory consolidation is applied, errors in the hierarchy tree can be corrected incrementally, allowing the algorithm to take advantage of fast node formation, while reducing both exemplar and node classification errors. Figure 10 shows a sample tree structure formed following incremental acquisition and memory consolidation. Figure 11 shows the evolution of a sample tree structure as the interleaved construction and consolidation takes place. Table 3 shows the false negative error rate for each motion type when memory consolidation is applied. Figure 12 shows a comparison of the performance between negative errors.

<table>
<thead>
<tr>
<th>Motion</th>
<th>Average</th>
<th>Std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>0.0007</td>
<td>0.0050</td>
</tr>
<tr>
<td>CH</td>
<td>0.0007</td>
<td>0.0067</td>
</tr>
<tr>
<td>DA</td>
<td>0.0514</td>
<td>0.2192</td>
</tr>
<tr>
<td>KI</td>
<td>0.0326</td>
<td>0.0616</td>
</tr>
<tr>
<td>PU</td>
<td>0.1393</td>
<td>0.2397</td>
</tr>
<tr>
<td>SL</td>
<td>0.0100</td>
<td>0.0706</td>
</tr>
<tr>
<td>SQ</td>
<td>0.0300</td>
<td>0.1098</td>
</tr>
<tr>
<td>TH</td>
<td>0.2585</td>
<td>0.3639</td>
</tr>
<tr>
<td>BO</td>
<td>0.0027</td>
<td>0.0210</td>
</tr>
</tbody>
</table>
Adaptive behavior rates when no consolidation is applied, and negative error rates when the consolidation is applied. As can be seen from Tables 1, 2, and 3 and Figure 12, false negative errors are significantly reduced when memory consolidation is used, both for similar motions (punch and throw), and dissimilar motions. For the punch motion, the false negative classification error is reduced from 40% to less than 15%, while for the dissimilar motions, such as for example the sumo leg raise motion, the false negative classification error is reduced from 2.5% to 1%.

The resulting tree structures were also analyzed and compared with the global clustering result. Each automatically generated tree was compared with the

Figure 10  Sample clustering result, \( K_{\text{cutoff}} = 0.9 \), with memory consolidation.

![Figure 10](image)

Figure 11  Sample tree evolution over an interleaved construction and consolidation procedure, \( K_{\text{cutoff}} = 0.9 \). After the first five groups are formed, consolidation takes place for the first time to remove a double of the KI node (TreeId = 5). As more motions are observed, additional nodes are added to the tree via tree construction (TreeId = 9). Because of fast initial node formation, nodes SL, SQ, and CH are erroneously grouped together during initial node formation (TreeId = 11). At a later time, during consolidation, this initial mistake is corrected, and the SL, SQ, and CH nodes are returned to the better fitting parent node (TreeId = 12, 14). Finally, after repeated motion consolidation, the dance node is also correctly identified (TreeId = 19).

![Figure 11](image)
The error between two trees was computed based on the edit distance (Bille, 2005; Shasha & Zhang, 1997). The edit distance is defined as the minimum cost edit operations sequence that transforms one tree into the other (Shasha & Zhang, 1997). Edit operations consist of inserting a node or deleting a node; in our analysis each operation is assigned a cost of 1. Table 4 shows the results for the three cases considered: a high value of $K_{\text{cutoff}}$ (1.2) with no memory consolidation, a low value of $K_{\text{cutoff}}$ (0.9) with no memory consolidation, and a low value of $K_{\text{cutoff}}$ (0.9) with memory consolidation. As can be seen from the results, memory consolidation reduces the mean tree error (computed based on edit distance), while producing deeper resultant trees. The hierarchy structure error, as measured by the distance between the incremental and the off-line result, is reduced from 2.11 to 1.21.

These results show that the consolidation process is effective at improving both the motion model and the global memory structure. The repeated rehearsal of previously observed motions allows the system to reduce susceptibility to presentation order and correct initial mistakes in abstraction which may be the result of insufficient information at the start of knowledge acquisition. The consolidation of the system leads to improved performance in motion generation and recognition. This is analogous to recent results in biology, which have documented performance improvements following sleep-dependent memory consolidation (Diekelmann & Born, 2007; Stickgold, 2005). This type of consolidation is hypothesized to involve repeated reactivation of brain regions active during memory formation, in a process of system-wide consolidation of memory. Computational models of the process (McClelland et al., 1995) seem to indicate that this type of rehearsal allows the brain to take advantage of interleaved learning, allowing new knowledge to be added without catastrophic interference. Our results, while using a different learning and motion memory structure, confirm the benefits of this type of learning process.

### Table 4  Tree analysis results

<table>
<thead>
<tr>
<th>$K_{\text{cutoff}}$</th>
<th>Consolidation</th>
<th>Mean error</th>
<th>Mean depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>No</td>
<td>2.11</td>
<td>2.08</td>
</tr>
<tr>
<td>0.9</td>
<td>No</td>
<td>1.96</td>
<td>2.94</td>
</tr>
<tr>
<td>0.9</td>
<td>Yes</td>
<td>1.21</td>
<td>3.81</td>
</tr>
</tbody>
</table>

### Figure 12  Comparison of clustering performance in terms of false negative error.

The proposed approach focuses on the learning of the motion primitives themselves, that is, learning at the lowest level of the hierarchy. We model the motion primitives as atomic temporal sequences of joint angle data. This type of task representation is suitable for self-referenced motions, but may not be suitable for other tasks, such as tasks interacting with objects in the environment. However, the framework proposed is not representation dependent, and can be applied to any task representation, such as a Cartesian representation in the learner frame (i.e., space-based representation), or a representation in the target frame (i.e., object-based representation; Bekkering & Pratt, 2004), simply by selecting the appropriate representation and using the chosen representation as the observation vector. When using a relative frame representation, the generation method must also be modified to generate motions relative to the frame origin. Multiple representations could also be used in the same framework, by including a mechanism for automatically selecting the appropriate representation given the task, either through multiple observations by minimizing variance (Billard et al., 2006), or through attentional mecha-
nisms (Demiris & Khadhouri, 2006). The selected representation could also be potentially tuned and corrected by applying a consolidation algorithm similar to the one proposed here for the motion primitives themselves.

Our representation of motion primitives simultaneously encodes both the goal and the means of the motion primitive, by encoding the means in the intermediate states of the stochastic model, and encoding the goal in the final state. As several studies and proposed models have argued (Byrne, 1999; Byrne & Russon, 1998; Hayes et al., 2008; Meltzoff, 2002, 2005; Wohlschlaeger et al., 2003), when the task to be imitated contains a strong goal component, and an alternate means to achieving the goal is already known to the imitator, the goal may be imitated preferentially to the means. While our current implementation considers the means and the goal equally, the proposed learning structure can be easily modified to implement goal preference, by changing the distance metric when comparing observed motion primitives. The current distance metric (Equation 1) considers the similarity between known primitives and newly observed motions based on the entire trajectory. For tasks where the final goal is more important than the means of achieving the goal, the learning structure and memory consolidation approach presented here could also be implemented by using a different distance metric, for example considering only the goal state, or a weighted sum of goal and means.

The focus of this article is full body human motions imitated by a humanoid robot, that is, two systems with very similar body morphologies, where one-to-one correspondence for the major body parts can be assumed. The correspondence between the human motion and the humanoid motion is computed by measuring the positions of the various body parts of the human, and computing the joint angles of the robot kinematic model which best correspond to the observed data (i.e., an inverse kinematic correspondence model). For cases where the robot morphology is different from the demonstrator, the proposed system could be adapted to filter the observed motions through a correspondence matrix (Alissandrakis et al., 2007) prior to learning. In this case, the correspondence matrix could be considered as the innate mapping from observations to motor actions, as postulated by the AIM model (Meltzoff, 2002, 2005). Differences in dynamics were also not considered in the learning phase. In a physical implementation of the proposed system (Kulić, Lee, et al., 2008) on a humanoid robot, the learned motion primitives were given to the robot as trajectory commands, and then a low-level controller was used to implement the trajectory on the physical robot while ensuring dynamic and postural stability was maintained. An alternate approach would be to define motion primitives as dynamical systems (Nakanishi et al., 2004) rather than probabilistic trajectories; the approach proposed here could also be applied to this type of motion primitive definition as long as a distance metric could be defined between motion primitives in terms of the dynamical systems model.

In this work, we do not consider the question of how to learn the sequencing of the motion primitives. Some initial work in this direction shows promise that motion primitives learned autonomously via the proposed approach can also be used for on-line sequence learning (Kulić, Lee, et al., 2008). In the work of Kulić, Lee, et al. (2008), a simple network graph is used to learn the sequencing rules; however, the proposed approach could also be coupled with a more sophisticated algorithm for sequence learning, based on interaction with the teacher (Dominey et al., 2008), or based on goal extraction (Erlhagen et al., 2006). An interesting direction for future work would consider how these higher level learning processes might also be refined through a process of memory consolidation, by implementing memory consolidation at higher levels of hierarchy. A second interesting direction is the potential to combine the learned movement primitives simultaneously, in addition to sequentially, in order to address multiple constraints in the environment, as suggested by Rosenbaum et al.’s model (Rosenbaum, Meulenbroek, Jansen, et al., 2001; Rosenbaum, Meulenbroek, & Vaughan, 2001).

7 Conclusions and Future Work

This article develops an approach for incrementally correcting and improving the memory structure of motion primitives learned during on-line observation of human motion. In the initial learning process (Kulić, Takano, et al., 2008b), motions are organized into a hierarchical tree structure, where nodes closer to the root represent broad motion descriptors, and leaf nodes represent more specific motion patterns. Errors made by the incremental clustering algorithm, both in
defining individual motion boundaries, and in the memory structure itself are then corrected incrementally via a mechanism for memory consolidation. The memory consolidation algorithm selects learned motions and reclassifies them using the current tree structure, by repeatedly reapplying the same learning algorithm used to initially acquire the motions, analogous to rehearsal. Both individual motion exemplars, as well as motion groupings can be processed in the same manner. In this way, the memory consolidation algorithm corrects both individual motion misclassifications and hierarchy formation errors, resulting in both improved classification and an improved structure over time. By repeatedly considering previously observed data at a later time, this approach allows the learning system to lessen the influence of presentation order through gradual presentation and interleaved presentation, analogous to computational models of learning processes in the human brain (McClelland et al., 1995).

In future work, we will investigate how the obtained structure information can be further used to improve learning performance, by utilizing structure information, such as information about the parent node when learning new primitives. In addition, we will consider how the learning performance can be improved through incorporating higher order processes, such as the sequencing of primitives and considering goal-oriented behaviors.

Acknowledgment

This work is supported by the Japanese Society for the Promotion of Science grant 18.06754 and Category S Grant-in-Aid for Scientific Research 20220001.

Note

1 An early version of this article was presented at the 2008 International Conference on Epigenetic Robotics (Kulić & Nakamura, 2008a).

References


Petrovskaya, A., & Ng, A. (2007). Probabilistic mobile manipulation in dynamic environments, with application to


About the Authors

Dana Kulic received her combined B.A.Sc. and M.Eng. degrees in electromechanical engineering and her Ph.D. degree in mechanical engineering from the University of British Columbia, Vancouver, BC, Canada, in 1998 and 2005, respectively. From 2006 to 2009 she worked as a JSPS post-doctoral fellow and project assistant professor at the Nakamura-Yamane Laboratory, Department of Mechano-Informatics, University of Tokyo, Japan. She is currently an assistant professor with the Electrical and Computer Engineering Department at the University of Waterloo. Her research interests include human–robot interaction, robot learning, humanoid robotics, and mechatronics.

Yoshihiko Nakamura received his Ph.D. in precision engineering from Kyoto University in 1985. He was an assistant professor at Kyoto University from 1982 to 1987, and an assistant and later associate professor at University of California, Santa Barbara from 1987 to 1991. He joined University of Tokyo in 1991 and is currently a professor at the Department of Mechano-Informatics. Dr. Nakamura's research stems from kinematics, dynamics, and control. Humanoid robotics, cognitive robotics, neuro-musculoskeletal human model, biomedical systems applications, and their computational issues are the current fields of his research. His book publications include Advanced Robotics: Redundancy and Optimization (1991, Addison-Wesley), Building the Robot Brain (in Japanese 2003, Iwanami), and Robot Motion (coauthored with M. Uchiyama, in Japanese 2004, Iwanami). Dr. Nakamura was a recipient of King-Sun Fu Memorial Best Transactions Paper Award, IEEE Transaction of Robotics and Automation in 2001 and 2002. He received the Most Active Distinguished Lecture Award in 2007 from the Robotics and Automation Society of IEEE. He is a member of academic societies including IEEE and the Robotics Society of Japan. He is the Vice President of IFToMM, and the Chairman of the Japan Council of IFToMM. Dr. Nakamura is a Foreign Member of the Academy of Engineering Science of Serbia and a Fellow of the Japan Society of Mechanical Engineers and the World Academy of Art and Science. Address: Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.