Compressed Sparse Code Hierarchical SOM on Learning and Reproducing Gestures in Humanoid Robots

Georgios Pierris and Torbjørn S. Dahl

***Abstract*— Compressed Sparse Code Hierarchical Self-Organizing Map (CoSCo-HSOM) is an extension of ideas existent in the gesture classification and recognition research area. Building on Hierarchical Self-Organizing systems and cognitive models introduced by neuropsychologists, we present the CoSCo-HSOM algorithm introducing novel features to the previously published sparse encoding HSOM model. During the training phase we use *activity lists*, i.e., ordered lists of recently activated nodes on each level, instead of activity level based encoding of short term memory. Furthermore, we present how HSOMs can be used to learn and reproduce a generalized task on the Nao humanoid robot, using only the initial posture of the robot. The effectiveness of CoSCo-HSOM is supported through a comparative analysis with the Gaussian Mixture Model approach, on the same task using the same training data.**

I. Introduction

Nature has always been giving answers to our problems, but it seems that researchers have been selective on which models to use. Biologically inspired robots [1] bypass the slow process of evolution by replicating animal, or human embodiments for free. On the other hand, even though cognitive science has not reached the point yet to reverse engineer human brains, recent studies hypothesize the learning processes of infants in great detail [2]. In this work, we model robots as infants with no prior experience, assuming both organisms utilize “blank-memory”, or ”no-experience” systems. Infant cognition can be modeled as a hierarchical bottom-up structure and the usage of neural networks as layers in the hierarchy is a natural selection [3].

According to the infant cognition model, infants build complete behaviors in hierarchies. During the execution of a gesture, or a behavior in general, infants start from the bottom layer and go higher, until the brain gets confused. The source of the confusion is not clear yet, but it is assumed to be due to the addition of irrelevant noise, or similar problems. The higher we go, the more information we gather. As soon as the brain cannot go in a higher layer, we fall back and from that point the gesture is reproduced going towards the bottom layer executing the motion primitives.

The research leading to these results has received funding from the European Commission’s Seventh Framework Programme (FP7/2007-2013) under grant agreement number 231500.

G. Pierris is with the Cognitive Robotics Research Center, University of Wales, Newport, Allt-yr-yn Campus, Newport NP20 5DA, United Kingdom. georgios.pierris@newport.ac.uk

T. Dahl is a Reader at the University of Wales, Newport and head of the Cognitive Robotics Research Center, University of Wales, Newport, Allt-yr-yn Campus, Newport NP20 5DA, United Kingdom torbjorn.dahl@newport.ac.uk

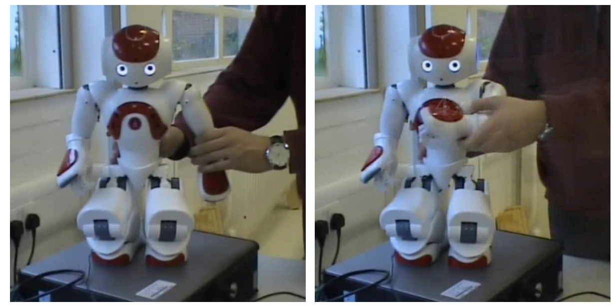

Fig. 1. Aldebaran Nao’s initial/final posture (left) and “reading watch” posture (right).

II. Overview of CoSCo-HSOM

The proposed Compressed Sparse Code Hierarchical Self-Organizing structure (CoSCo-HSOM) is intended for learning and reproducing gestures on humanoid robots. CoSCo-HSOM is an extension of a Hierarchical Self-Organizing Map (HSOM) [4], introducing a novel approach in compressing the sparse code signals. These signals are used as inputs in layers higher than the bottom. A common approach is to create in every layer long patterns of the last active nodes, i.e, a single node match of the input during the training phase. Depending on the memory size of the structure, the pattern tracks the last n nodes and sets the *activity level* to zero on all other nodes. A decreasing value is assigned on the active nodes in every step. A naive model is to linearly decrease this value[1]. The absolute values do not affect the structure; it is important though to follow a decreasing pattern to track the temporal information.

In the previously published work there is a computational overhead. In every training step, we have to update the sparse signal by accessing every node in all corresponding layers. The complexity might be linear, but doing it in every training step makes it time consuming. Instead of using activity levels in every node, we indirectly monitor the activity in every layer. A fixed-size FIFO queue is being updated in every layer keeping track of the recently active nodes’ IDs. Each queue comprises a signal, a sequence of IDs, which can be fed up in higher layers as input, instead of sending a longer version of the same information. The complexity of updating the queue is constant, as we only push and pop the best and the oldest matching node respectively in every step, without running through all nodes in every layer.

[1]For example, in a system keeping track of the last four nodes, the activity levels can be 1.0, 0.75, 0.5, 0.25, and 0.0 in older, or inactive nodes, with 1.0 being the most recently activated node in a layer.

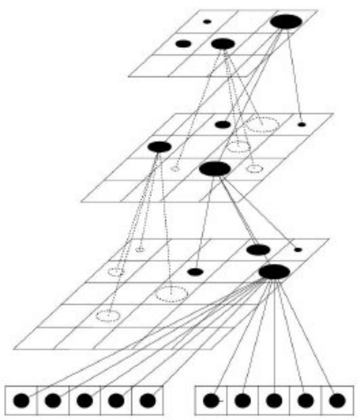

Fig. 2. Short term memory as activation in a hierarchical SOM.

The algorithm supports through the use of a single memory structure the integration of both short term memory (STM) and long term memory (LTM) [5]. The weights of the HSOM constitute the LTM. During the learning phase, the activity lists are used as inputs to nodes on higher levels in the hierarchy and influence how the weights are adjusted, producing a transferral of information from STM to LTM. As a result, different sets of node IDs come to encode different observed sequences. A single node represents a finite sequence of observations and, as such, may be reused several times in a longer sequence or by different sequences. This reuse will be encoded by different non-zero weights in the map, where the node has the role of an input, i.e., the map on the layer above in the hierarchical structure. Figure 2 presents a CoSCo-HSOM structure illustrating activity levels as ovals of different size. In CoSCo-HSOM, each node represents a principal component of the observed sequences of states and actions. Figure 2 also illustrates how the most recent actions are recorded in low level STM, while sequences of more abstract actions are represented on higher levels. The dashed ovals indicate sequences of a loose state-action model, i.e., nodes, that are referred to by nodes that are present in STM, but are not themselves present, i.e., the abstract sequences containing these nodes are currently active, but the nodes themselves no longer have a raised activity level. A more thorough description of the CoSCo-HSOM and its distinctive features will be discussed in Section III-C.

III. Experimental Set-Up

A. System Architecture

Hard-coded robotic behaviors introduce many problems to researchers, such as the lack of generalization and the absence of reusability. In order to compensate these problems, focus has been transfered to Imitation Learning, Programming by Demonstration and other machine learning solutions [6]. In the context of this work, we are mostly interested on the gesture production, and particularly on generalizing a motion by learning gesture patterns.

In the proposed approach, we develop a novel Hierarchy of Self-Organizing maps to reproduce gestures on humanoid robots, by learning timed sequences of robotic postures. A simplified version of robotic motion is the execution of these robot posture sequences. The posture execution is a control-based transition (linear, smooth etc.) from one pose to another in the configuration space. In turn, a single posture can be fully described by a time independent sequence of joint values, whereas motions are a parallel, simultaneous transition of all robot joints. From that point of view, the timed sequence of values on each joint is a time series. In our architecture, we propose an open-loop, off-line training system for humanoid robots that learn gestures through supervised demonstrations. The postures are being recorded and then fed unprocessed to train the hierarchy. Once the hierarchy has converged to its final weights in all layers, we can then save it in a binary, or plain text format. The memory needed to load the map in robots is fixed, independent of the gesture's length, and minimal compared to large dictionaries of generalized gesture versions. Finally, by loading the structure to the robot, it is able to reproduce complete gestures using only its initial posture.

B. Hardware

Nao [7] is a relatively small humanoid robot. It has been been developed by Aldebaran Robotics based in Paris, France and is a 58 cm tall robot weighing 4.8 Kg, utilizing 25 degrees of freedom (Academics Version). The commercial release of Nao has been planned in the near future, however Aldebaran has achieved to promote Nao as an educational robotic platform and a family entertainment robot affordable to most budgets. Even though Nao's capabilities cannot be compared to those of other humanoid platforms, it can potentially be considered as a benchmark platform, due to its large popularity especially among European Institutes and RoboCup competitions. The Nao programming environment is based on the proprietary Naoqi framework, which serves as a middle-ware between the robot and high-level languages, such as C, C++, and Python [8]. Serving the generality of our approach, this framework is being used only in terms of strictly sending motor commands to the Nao. In that sense, the architecture is platform, and application independent.

The training data are captured from encoders on each joint. Joints are set in passive mode to safely move them, or can be dynamically guided by the demonstrator through touch interpretation. During demonstrations the "expert" user manipulates the joints of the robot in the desired direction, and speed, while recording data in high frequency (40Hz). It is a good practice to record gestures using high frequency, but in practice our approach has been proven effective even when using way too low recording frequency. Finally, taking advantage of CoSCo-HSOM's nature, there are no restrictions on what training data to use, in terms of their semantic meaning (e.g. joint values, velocities), as long as there is a controller to inverse the learned data to a motion on the reproduction phase.

C. Compressed Sparse Code HSOM

Research has been wide on the generalization of complex robotic gestures [9], [10], [11]. From a different perspective, we hypothesize that biological models consist a comprehensive approach to the problem. In this work, following the constructive model of cognition on infants [2], we have built an hierarchy of Kohonen Self-Organizing maps [12]. Every layer has a different set-up, in terms of the learning factor, learning radius, dimensions etc. and we randomly initialize the weights in every map. The bottom layer is used for learning different body postures with a variation of separating the state-action input in two different inputs. The state is the complete posture, or a part of it, whereas the action is considered to be the state (posture) in the next timestep. For example, at timestep t the input will be $S_t = \{\cup x_t^i \quad \forall i\}$, where x^i is the value of joint i (or other type of variable depending on the context of the problem). The second input is the $S_{t+1} = \{\cup x_{t+1}^i \quad \forall i\}$. The action is then a linear transition $S_t \rightarrow S_{t+1}$ in the configuration space, which is a single step prediction at the bottom layer, i.e., a gesture primitive. Following the path from the top to lower levels, we retrieve complex gestures expressed as a temporal sequence of primitives.

1) Training Phase: The CoSCo-HSOM algorithm has to be trained using training data in a sequential way, in contrast to usual random feeding of Kohonen maps. The order of the data implies the steps to accomplish the task. The distinctive characteristic in our approach is that in every layer we update a fixed size FIFO activity list tracking the IDs of active nodes. An ID is the (x, y) position in the discrete 2D grid of every map. The ID of the best matching unit in every training step in the bottom layer is pushed in the activity list, and the ID of the oldest node is popped, if the list is full. The size of the list is set by the user and we call it $memory$. An important step is to clear all the activity lists every time a complete gesture has been fed to the algorithm. Otherwise, all the training sessions are considered as a long gesture causing replications of the actual gesture. In general, this approach is sensitive to replicating the same motion during the reproduction phase.

Time-compression of long gestures is being achieved by training every layer exponentially. In every training step, the bottom layer is being trained, but the upper layers are trained every $updatePeriod^{level}$ steps, where $updatePeriod$ is again hard-coded by the user, and $level$ is the level of each layer counting from 0 (the bottom layer)[2]. Eventually, even long sequences can be encoded in a compact architecture without loss of valuable information. Finally, the use of activity lists, instead of complete activity maps of sparse signals in every layer, reduces the memory requirements, and running cycles of the hierarchy.

2) Reproduction Phase: Gesture recognition [13], rather than reproduction in robots, is where related work focus on.

[2]For example, if the $updatePeriod$ is 2 in a three layer architecture, after training step 7, the bottom will have been trained 7 times (at steps 1, 2, 3 ...), the second layer 3 times (at 2, 4, 6) and the third only 1 (at 4)

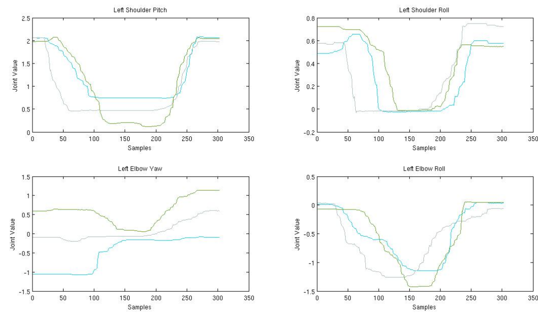

Fig. 3. Pre-processed training data for GMM

In this work, we propose a novel approach of reproducing a learned gesture in humanoid robots. The gesture reproduction cannot start though, before the controller builds a short history. The history is a small number of occurrences at the bottom layer, that will allow confidently to go higher in the hierarchy to gather more gesture related information. Clearly the history dependence is a drawback, but our goal is with no prior knowledge and only using the trained HSOM, and the current posture of the robot to reproduce a complete gesture.

CoSCo-HSOM starts with a prediction of the next posture using the initial as input. The output at the bottom layer will be the next posture and an occurrence of the winning node will be added to the bottom layer's activity list. We follow the same process, until we have built a "considerable" amount of matches at the bottom layer. "Considerable" is expressed as a manually set $offset$ of the maximum $memory$ defined for the system. For example, if the $memory$ is four, then a node in the second layer represents four nodes at the bottom. Having three matches at the bottom layer, we can confidently predict the fourth by visiting the layer up. In that level, the best matching unit is found using only the three occurrences, ignoring the fourth. As soon as we predict the fourth node, we execute it and continue in the same way to the upper layers. The $offset$ is not static and allows us to study the effect of more aggressive predictions.

One could argue, that the first steps of the reproduction are naive gesture primitives, resulting in poor reproduced gestures. This is partially true, but we expect to get better predictions in the short future, after having built the required history to find occurrences in higher levels. The higher we go, the more confidently we predict. As soon as we decide to move downwards, the predicted node points to the directly lower layer in $memory$ different, or same nodes. In turn, each node points to other nodes in lower levels. The prediction of a sequence of primitives is a tree traversal problem. Particularly, we follow a Depth First Search approach starting from the oldest node. The older nodes are in the end of every activation list (we insert from the beginning and pop from the back). Thus, the oldest primitive can be found by following the rightmost node in every activation list, and the newest the leftmost. Then, we simply append the pattern of postures to the so far predicted gesture and eventually execute the transition from posture to posture.

All weights in the Kohonen maps have been initialized with real numbers. In the bottom layer, this is not a problem, as the posture of the robot has the same structure. In higher layers though, where we learn activity lists with IDs, these will be natural numbers (indexes in a 2D matrix, or the position in the topology). In order to solve that practical problem, we use the rounded signal (might need the floor of the ID, if it didn't converge fast) to get the indexes. Another solution could be to find the closest node, according to the euclidean distance of the prediction, and the nodes in the lower layer. The last solution though, introduces a computational overhead, wasting the advantage of predicting gestures with linear complexity.

D. *Configuration*

The system described is fully customizable, through a XML file. $memory$, $updatePeriod$, $state - action$'s size, $trainingSteps$, the height of the hierarchy, the size and topology of every layer, the learning radius, and the learning factor are customizable, allowing for experiments with various setups. It worths mentioning, that the learning factor has to be different in every layer using a small learning factor at the bottom layer and more aggressive factors as you go higher. It is obvious that the higher layers are being trained infrequently. Different learning factors eliminate the trade-off of finding a good learning factor for all layers, which is not possible especially in our work, where we tend to train hierarchies, even to the height of six or more.

IV. Experiment

We present an experiment with the Aldebaran Nao robot learning a simple gesture. We also compare our motion with the Programming by Demonstration approach, which uses Gaussian Mixture Models [10]. With the assumption that the CoSCo-HSOM algorithm is model-free, we argue that this method is generic, and transferable to other types of robots. A simple gesture similar to raising the left arm at the "reading watch" position is used to verify the effectiveness of the reproduction phase. The initial, and final postures are the same, making the experiment even more complex.

The task consists of training the Nao robot with the help of an expert supervisor to learn, and reproduce the desired gesture (Figure 1). We use three similar training sessions of one gesture. For this experiment, we have only used the left arm's six DOFs (we present the four though, as the Wrist and Fingers are not used) as a training set, and not the complete configuration of the robot. The time series of all three training sessions for each joint are shown in Figure 4. Obviously, the training data have not been processed in any way, and this is one of the advantages of this approach. CoSCo-HSOM does not need any pre-processing steps, and raw data can be used without problems. All the details concerning the exact configuration used in this experiment is in Table I (all layers have Orthogonal topology; Hexagonal is also an option). The reproduced gesture of the trained hierarchy is the black line in Figure 4. A thorough discussion of the results follows in Section V.

TABLE I
CONFIGURATION OF HIERARCHICAL SOM

Hierarchy	Learning Factors / Grid Size
Height - number of levels : 6	Level 5 : 0.8 / 1 x 1
Memory - size of STM : 4 (activation lists)	Level 4 : 0.6 / 2 x 2
Update Period : 2	Level 3 : 0.5 / 6 x 6
Training Steps : 5000	Level 2 : 0.3 / 8 x 8
State Length : 6	Level 1 : 0.2 / 10 x 10
Action Length : 6	Level 0 : 0.1 / 16 x 16

A. *Future Experiments*

The presented experiment is only a part of our initial results and serves the purpose of a qualitative comparison of the reproduced gestures against other methods. We currently explore the effect of different configurations in various gestures and how the CoSCo-HSOM can handle multiple gesture production within a single hierarchy. We expect to encounter difficulties in the reproduction phase, where having learned different gestures, sharing similar, if not the same initial postures, will create perturbations in the motion.

Taking advantage of the modular architecture, we also research methods of building hierarchies of HSOMs. This work can potentially be used on learning behaviors from sequences of complete gestures. Our future plans, include an extensive analysis of the limits of this work in various domains, where the CoSCo-HSOM could be proved useful. We are also confident that sensorimotor control applications will take advantage of this approach. This area is open now and we are looking forward to fitting our work in the reproduction of cognitive models in robotics.

V. Discussion and Related Work

The cost of humanoid robots has been dropped significantly in the last years, attracting more research groups, and even independent researchers explore methods similar to gesture learning. The idea of using neural networks to learn, generalize, and reproduce gestures, or behaviors is not novel. A simple way to represent robotic motions is the use of temporal sequences, or time series. Self-Organizing Incremental Neural Networks (SOINN) and Dynamic Time Warping(DTW) have been successfully used in the recognition of temporal gesture data from humans [14]. The classification of different gestures is an integral part in cognitive robotics to allow for conscious reproductions. It has been shown that SOINN-DTW method outperforms approaches based on Hidden Markov Models.

A related approach to CoSCo-HSOM uses a three layer Kohonen neural network on a mobile robot to learn temporal sequences of behavior primitives to acquire behavioral plans [15]. The hierarchy constitutes from the motor layer (motor drives) at the bottom, the sense layer (energy consumption, brightness sensor, range sensor etc.) at the second level, and at the top level the temporal layer tracking the behavioral sequences in the lower layers. Even though this approach has been successfully used in the simulated environment of the WAMOEBA-1R mobile robot, it cannot

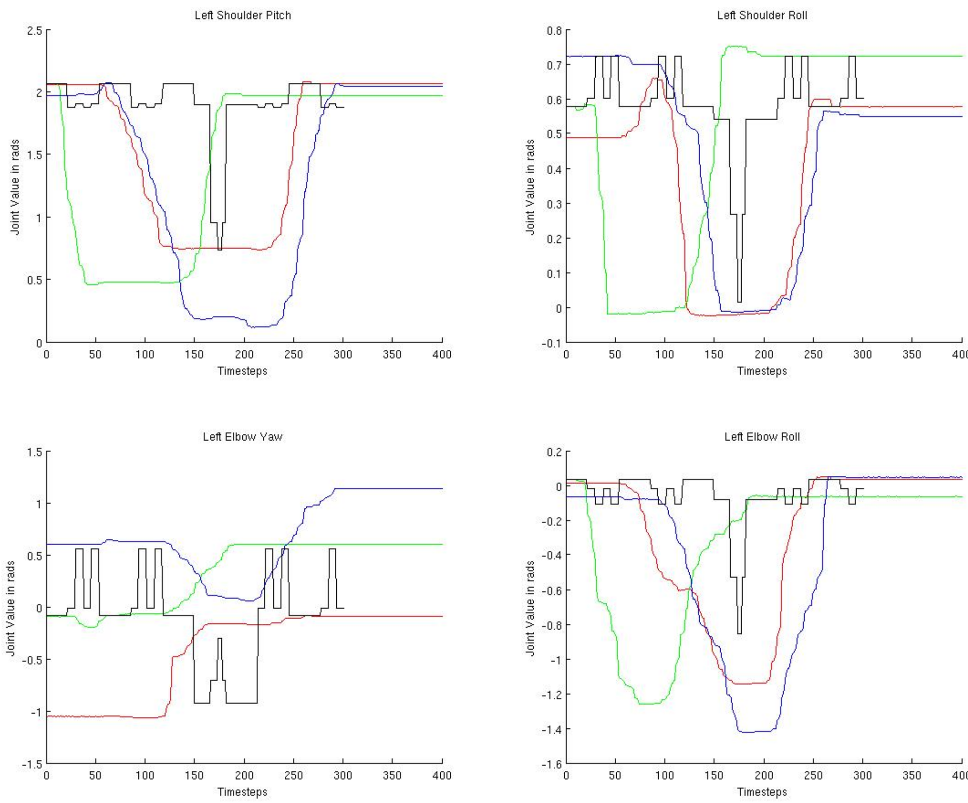

Fig. 4. Raw training data from three training sessions. The black line represents the reproduced motion.

be easily transfered to more complex environments in humanoid robots as the sensor layer is divided and hard-wired to platform specific inputs. Furthermore, the lack of temporal compression during the learning phase makes it impossible to learn longer behaviors than the size of the map at the temporal layer.

Recurrent neural networks (RNN) have been investigated for learning multiple temporal sequences [16] and later extended to learning more complex tasks of ball, and blocks manipulations on a humanoid robot [17]. Even though RNNs have been successfully used in various applications, they suffer a number of drawbacks [18] such as the high computational training costs, the possibility of getting stuck in local minimum, their slow convergence, and the problem of learning long term dependencies [19]. Echo State Networks (ESN) is a recent approach leading to simpler training algorithms for RNNs performing well in time series prediction [20]. Another application is in mobile robotics, where learning behavioral patterns is essential [21]. ESNs have been praised for their simplicity, but setting them up is an important step for the behavior of the system. Besides ensuring the "echo state property" to avoid chaotic behaviors [22], recent studies point out the importance of the asymptotic stability of ESNs running with output feedbacks [23].

Another approach is the use of Gaussian Mixture Models [10]. As our work is novel in gesture reproduction, we compare our solution with the GMM model. Even though the algorithms have nothing common to share, they both solve the same problem. Serving the objectiveness of the analysis, we use almost the same training data in both approaches. The raw data of the training sessions are in Figure 4 (excluding the black line).

One of the key features of CoSCo-HSOM algorithm is the ability to directly feed it with the training data, whereas in the GMM model a pre-processing step is required. We have also ignored the beginning, and the end of all training sessions, to partially align the signals[3]. After the alignment process, we resample the signals, in order to get a fixed number of points. The final training dataset for the GMM approach is in Figure 3, and can be compared with the data used in the CoSCo-HSOM (Figure 4).

We can point out three weaknesses of the CoSCo-HSOM. Firstly, there is some oscillation in the first steps of the reproduced gesture. As mentioned in Section III-C.2, the agent presents a reflexive behavior ($currentState \rightarrow nextState$) in the very beginning, until it builds a short history. Even later though, we get sharp edges. This is virtually unavoidable, due to the nature of HSOM, where in practice it detects the principal components of a gesture; therefore, we compress the gesture information to retrieve the gesture's key poses. Finally, a consequence of that is the absorption of the time variant, making the algorithm perfect for gesture recognition, but "problematic" for gesture production. The controller

[3]Demonstrators unconsciously tend to keep the same speed during the motions. Surprisingly, throughout every session we got large variance in the duration of inactivity, before the actual demonstration.

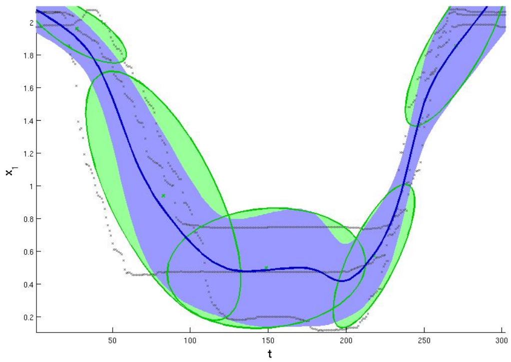

Fig. 5. Left Shoulder Pitch. The green areas are the Gaussians used to encode the signals, where the thick blue line is the final trajectory to reproduce. The light blue envelope, around the trajectory, represents the expected variations, and the dark points are the training sessions.

has to implicitly decide the length of the gesture while executing the transition between postures. Paradoxically, we take advantage of that "problem", as it allows incremental retraining across experts, with no prior knowledge of the gesture's temporal information. Researching more on different configurations, we expect to further improve the results and avoid any oscillations, or spikes during the reproduction phase. In the accompanied video, it is clear that even with these oscillations, the reproduced gesture remains smooth enough to be acceptable for execution.

In Figure 5, we present only the GMM's reproduction of the "Left Arm Shoulder Pitch" joint. It is obvious that the GMM approach, produces smooth motions and manages to remain smooth on the transition from one motion to another. It is also clear that the GMM algorithm manages to reproduce gestures without losing any temporal information. The memory requirements (centers, and covariance matrices of Gaussians) are even smaller. One of the drawbacks of this work, is that if we were using raw data, without aligning the signals using DTW [24] (the resampling process is unavoidable), the reproduced motions would not be nicely constraint [10] to get a meaningful reproduction through Gaussian Mixture Regression.

VI. Conclusion

We have presented the CoSCo-HSOM algorithm, an hierarchical Self-Organizing map that manages to learn, and reproduce gestures in humanoid, and general purpose robots. A simple experiment helped us understand the effectiveness of our approach, and its novel characteristics. We also made a comparative discussion against other neural network approaches, and the Gaussian Mixture Model algorithm. Our future plans include an extensive analysis on the limits of this work in humanoid gesture recognition, and reproduction. We are confident that sensorimotor control applications will take advantage of this approach and we expect to explore these areas. Finally, further testing in other domains will provide us with a thorough understanding of CoSCo-HSOM's behavior.

VII. Acknowledgment

The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 231500.

References

[1] N. E. Sharkey, "Biologically inspired robotics," *Handbook of Brain Theory and Neural Networks*, 2002.

[2] L. B. Cohen, H. H. Chaput, and C. H. Cashon, "A constructivist model of infant cognition," *Cognitive Development*, vol. 17, no. 3-4, pp. 1323–1343, 2002.

[3] S. Grossberg, *Neural networks and natural intelligence*. Cambridge, MA, US: The MIT Press, 1988.

[4] J. Lampinen and E. Oja, "Clustering properties of hierarchical self-organizing maps," *Journal of Mathematical Imaging and Vision*, vol. 2, no. 2-3, pp. 261–272, 1992.

[5] T. S. Dahl, "Hierarchical traces for reduced nsm memory requirements," in *the Proceedings of the BCS SGAI International Conference on Artificial Intelligence, Cambridge, United Kingdom*, 2010.

[6] Y. Demiris and A. Billard, "Special issue on robot learning by observation, demonstration,and imitation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 2, pp. 254–255, 2007.

[7] D. Gouaillier and P. Blazevic, "A mechatronic platform, the Aldebaran robotics humanoid robot," *32nd IEEE Annual Conference on Industrial Electronics, IECON 2006*, pp. 4049–4053, November 2006.

[8] G. Pierris and M. Lagoudakis, "An interactive tool for designing complex robot motion patterns," *IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, May 2009.

[9] A. Billard and R. Siegwart, "Robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 65–67, 2004.

[10] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.

[11] N. Otero, A. Alissandrakis, K. Dautenhahn, C. L. Nehaniv, D. S. Syrdal, and K. L. Koay, "Human to robot demonstrations of routine home tasks: exploring the role of the robot's feedback," in *HRI*, 2008.

[12] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

[13] A. Shimada and R. ichiro Taniguchi, "Gesture recognition using sparse code of hierarchical som," in *ICPR*, 2008, pp. 1–4.

[14] S. Okada and O. Hasegawa, "Classification of temporal data based on self-organizing incremental neural network," in *ICANN (2)*, 2007, pp. 465–475.

[15] T. Ogata, K. Hayashi, I. Kitagishi, and S. Sugano, "Generation of behavior automaton on neural network," in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 2, 7-11 1997, pp. 608 –613 vol.2.

[16] M. Ito and J. Tani, "Generalization in learning multiple temporal patterns using rnnpb," in *Neural Information Processing*. Springer, 2004, pp. 592–598.

[17] M. Ito, K. Noda, Y. Hoshino, and J. Tani, "Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model," *Neural Networks*, vol. 19, no. 3, pp. 323–337, 2006.

[18] C. Gallicchio and A. Micheli, "On the Predictive Effects of Markovian and Architectural Factors of Echo State Networks," 2009.

[19] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *Proceedings of the IEEE International Conference on Neural Networks*, 1993.

[20] H. Jaeger, "The echo state approach to analysing and training recurrent neural networks," 2001.

[21] C. Hartland and N. Bredeche, "Using echo state networks for robot navigation behavior acquisition," 2007.

[22] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach," *Fraunhofer Institute for Autonomous Intelligent Systems (AIS), International University Bremen*, 2002.

[23] Q. Song and Z. Feng, "Stable trajectory generator-echo state network trained by particle swarm optimization," in *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2009, pp. 21–26.

[24] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD Workshop*, 1994, pp. 359–370.