A study of two complementary encoding strategies based on learning by demonstration for autonomous navigation task

Antoine de Rengervé* **Florent D'halluin**** **Pierre Andry***
Philippe Gaussier* **Aude Billard****
*ETIS, CNRS ENSEA University Cergy-Pontoise F-95000 Cergy-Pontoise
**LASA, EPFL, CH-1015 Lausanne, SWITZERLAND

Abstract

Learning by demonstration is a natural and interactive way of learning which can be used by non-experts to teach behaviors to robots. In this paper we study two learning by demonstration strategies which give different answers about how to encode information and when to learn. The first strategy is based on artificial Neural Networks and focuses on reactive on-line learning. The second one uses Gaussian Mixture Models built on statistical features extracted off-line from several training datasets. A simple navigation experiment is used to compare the developmental possibilities of each strategy. Finally, they appear to be complementary and we will highlight that each one can be related to a specific memory structure in brain.

1. Introduction

Human development is clearly influenced by interactions performed during the whole life. One of the natural way to teach someone how to do something is simply to demonstrate what is expected from him. Programming by demonstration [Billard et al., 2008] is a key approach for development in autonomous robots. It does not require any technical knowledge from the teacher as it tries to provide the robot with learning abilities similar to those present in human beings. This is particularly interesting when dealing with robotic systems that must perform a wide range of tasks. Approaches requesting to program every possible behavior happen to be a dead-end.

In this work, demonstrations enable the robot to build behaviors based on the learning of sensorimotor associations. The system can then infers what to do when presented with some given sensory information (proprioception, vision). However, there are different ways to encode such sensorimotor associations, and different ways to learn them, which will influence the developmental capacities of the robot. In this paper, we will focus on two strategies that give distinct answers to the question of encoding and learning. The Neurocyber team of ETIS lab developed a system based on artificial Neural Networks that allows a user to teach a mobile robot how to navigate robustly using visuo-motor association [Lagarde et al., 2010]. Meanwhile, the LASA lab uses a statistical approach based on Gaussian Mixture Models to learn the sensorimotor coupling. This can be employed for teaching gestures to robotic arms or humanoid robots [Calinon et al., 2009].

A simple U-shaped navigation task has been chosen in order to compare these two approaches. The robotic platform used in this work, a Robulab from Robosoft, can select a direction to go forward. When the robot goes away from the right path, its direction can be modified by using a joystick. In order to compute its position, the robot can use a monocular pan moving camera. With pan rotation, the camera provides the robot with visual panorama for self-localization. An odometer can be used for recording the trajectory in the Cartesian space.

The Neural Network (NN) model and the Gaussian Mixture Models (GMM) that can perform navigation task will be respectively developped in Section 2 and Section 3. The presented experiment illustrates similarities and differences between these two approaches. These two systems enable a robot to learn actions in the context of learning by demonstration in interaction with a human teacher. Both systems are based on state-action associations but they do not learn and encode information in the same way. Section 4 discusses the consequences of these different encodings for the capacities of the system while considering memory-cost, adaptation through long time learning, interaction and quality of the trajectory. In Section 5, we discuss the complementary aspects of these two approaches and how they could be related to different kinds of memories as human cognition is concerned. They show similarities with the Hippocampus and the Neocortex as described in [McClelland et al., 1995].

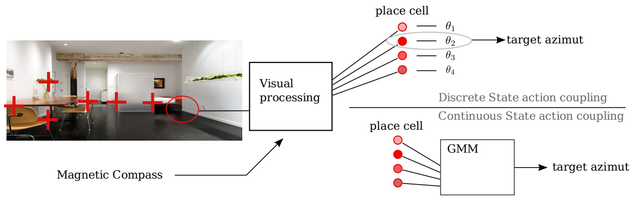

Figure 1: Overview of the system: Once the visual processing is done on data, it activates continuously the place cells neurons.

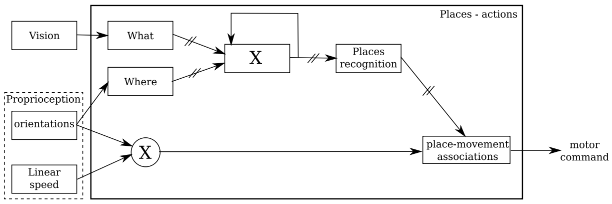

Figure 2: Model of place-movement associations learning. "what" and "where" information is extracted from vision and proprioception. They are merged and compressed in place codes allowing place recognition. These place codes are associated to proprioceptive informations.

2. Using Neural Networks for navigation

Neurocyber team has developed a controller for mobile robots which is able to associate visual information (a panorama of the environment) with self orientation (using a compass representing the direction of the actual movement). This controller is designed as a sensorimotor loop based on a neurobiological model testing some of the spatial properties of the hippocampus [Giovannangeli et al., 2006].

2.1 Place cells definitions

To be able to localize itself and navigate, the robot uses the recognition of place cells based on visual cues (Figure 1 and [Giovannangeli et al., 2006] for more details about visual features extraction). A place is defined by a constellation of visual features (landmark-azimuth couples) extracted from a panorama (Figure 3) compressed into a place code. The visual system extracts local views centered on points of interest (landmark recognition) that provides information of "what". A magnetic compass acting as proprioceptive information (spatial localization in the visual field) provides information of "where". The place code results of the merging of "what" and "where" information.

The merging of the information is performed in a product space (*i.e.* a matrix of product neurons $m_k(t)$ called *merging neurons*) defining a place code $M(t)$. More details about the definition of the place code and the merging neurons can be found

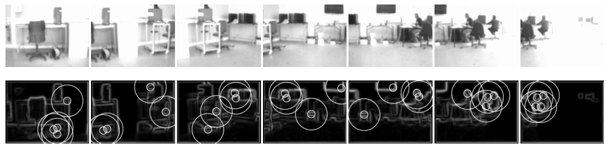

Figure 3: Example of visual features extraction on half visual panorama. The system computes a gradient on each view. 4 visual features are extracted from each gradient.

in [Lagarde et al., 2010]. The place-cell activities are built as the result from the computation of the distance between the learned place code and the current place code. The activity $PC_p(t)$ of the p^{th} place cell is:

$$PC_p(t) = \frac{1}{W_p}\left(\sum_{k=1}^{n_M} \omega_{kp}^{PC}(t) m_k(t)\right) \tag{1}$$

where $\omega_{kp}^{PC}(t)$ expresses the fact that the landmark-azimuth couple k (*i.e.* the k^{th} merging neuron which activity is $m_k(t)$) has been used to encode the place cell p. The number of couples used by the p^{th} place cell is given by $W_p = \sum_{k=1}^{n_M} \omega_{kp}^{PC}$, with n_M the number of recruited neurons in the landmark and azimuth matrix. A place cell learned in the location A responds maximally in A and creates a large decreasing field around A. Such a system is able to learn several regions of the environment. It can perform visual localization.

2.2 Place-movement association to define trajectories

Originally, the robot follows a random direction. When the robot takes a wrong direction, the human teacher can use a leash to drag the robot toward the desired path. When doing so, the user modifies the dynamics of the robot and the motor command of the wheels orientates the robot in the desired direction. The resulting change in proprioception (orientation change) triggers the learning of the association between the movement being done and the visual panorama (the current location). The orientations and the speeds are discretized so that each neuron S_i corresponds to a given orientation and a given speed. Their activities are calculated with (2).

$$s_i(t) = \sum_{p=1}^{n_{PC}} \omega_{pi}^{S}(t) \cdot PC_p(t)$$
$$S_i(t) = V(t) \cdot S_i^d(t) + (1 - V(t)) \cdot \left(\frac{s_i(t)}{s\max(t)}\right) \tag{2}$$

When the teacher modifies the dynamics of the robot, the change is detected and the vigilance signal

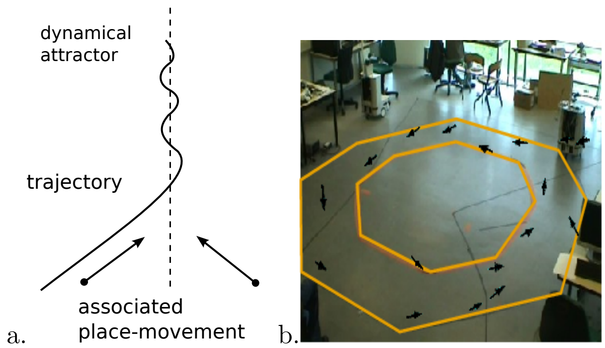

Figure 4: Learning of a path by correction of the learned dynamics. Each arrow represents a correction applied by the teacher. Hence, the robot learns the association between the place and its orientation. a.) A straight line trajectory can be defined by only two state-action associations. This trajectory is an attractor as from every point in the space, the robot will reach the line. b.) An example of trajectory that can be learned. After 3 rounds of learning, the robot is fully autonomous, the professor does not need to correct it anymore.

V becomes equal to 1. $s_{\max} = \max_{i=1..n_S}(s_i)$ is used for an output normalization with n_S the number of actions. The output S_i can either be the action predicted by the place cells or the desired action $S_i^d(t)$ (orientation and speed) that is determined from the action performed by the robot with or without the intervention of the teacher. The weight ω_{pi}^S of the connection between the p^{th} place cell and the i^{th} action is adapted according to a learning rate $\epsilon(t)$ and the learning rule (3) which is inspired of Widrow&Hoff gradient descent rule:

$$\frac{d\omega_{pi}^S}{dt} = (S_i^d(t) - s_i(t)) \cdot PC_p(t) \cdot V(t) \cdot \epsilon(t) \qquad (3)$$

Building sensorimotor attractors by associating movements to different regions of the environment ([Giovannangeli and Gaussier, 2010], Figure 4) enables the robot to reproduce the learned trajectory. A more sophisticated version of this associative learning exists which is not described in this paper [Giovannangeli and Gaussier, 2010]. In the case of the short learning of the navigation experiment described in Section 2.3, that version shows little difference with the presented version.

2.3 Application of the model to a U-shaped trajectory

The robot follows a direction which is corrected by the human teacher whenever it is too far from the desired trajectory. Modifications of the dynamics of the robot imply the learning of new place cells (Figure 5). The Neural Network model enables the reproduction of this simple trajectory after only three runs,

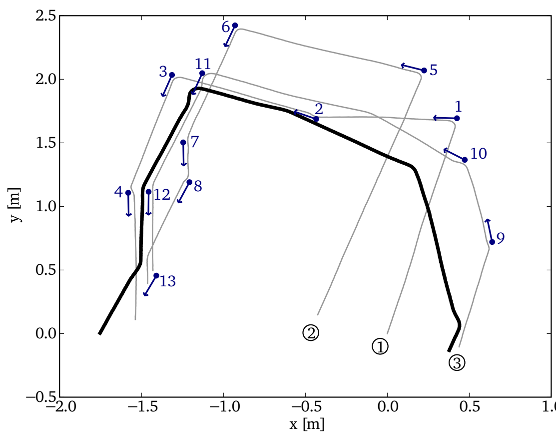

Figure 6: Experimental data corresponding to a simple U-shaped path. Top: trajectories in the Cartesian space (using odometry, reset at the beginning of each run). Grey lines are the trajectories during which learning occurred. The successive starting position are numbered. The reproduction (no correction) trajectory is in black. Dark blue arrows are the learned place-orientations.

using only the corrective interaction from the human teacher (Figure 6). During each run, the following data are recorded: activations of place cells (outputs from the Neural Network), the current position and orientation of the robot (using both odometry and magnetic compass). These data are used to train the GMM-based system for the navigational task.

3. Gaussian Mixture Model approaches for robot navigation

In this section we propose two uses of Gaussian Mixture Models in order to learn trajectories in navigation task. The first implementation is based on an ideal situation with relevant data directly available. Secondly, we propose an implementation which uses subjective visual cues in a more autonomous and realistic approach.

3.1 A direct transposition of the manipulation model using Cartesian coordinates

A Gaussian Mixture Model defines a probability density function on the state space of the robot.

$$p(\xi) = \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} e^{-\frac{1}{2}\left((\xi-\mu_k)^\top \Sigma_k^{-1} (\xi-\mu_k)\right)} \qquad (4)$$

D is the dimension of the state space and k is the number of states. π_k are prior probabilities, μ_k are mean matrices, Σ_k are covariance matrices and ξ are points of the state space. The GMM is characterized by the three parameters π_k, μ_k, Σ_k. Given

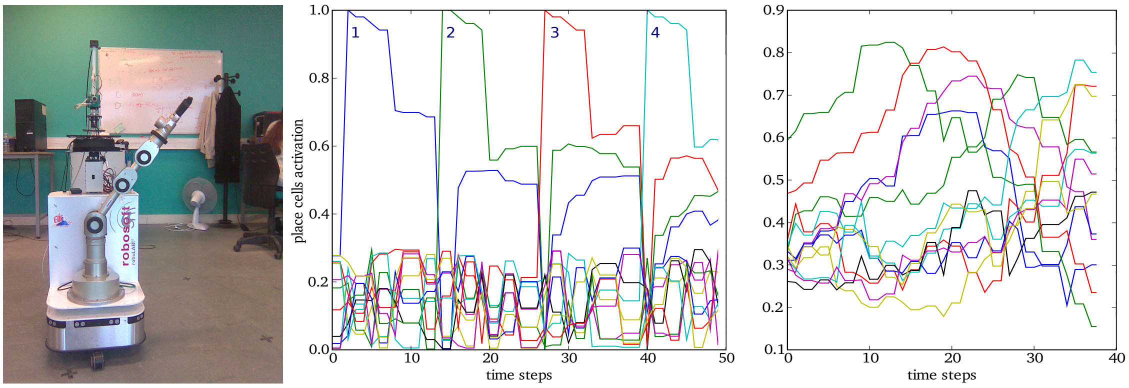

Figure 5: Experimental data corresponding to a simple U-shaped path. Left: Robulab mobile platform used for the experiments. Middle: place cell activation during trajectory while learning phase. Peaks correspond to the creation of new place cell, when user sets a new target azimuth (sharp turns in the Cartesian space) Right: place cell activation during the reproduction phase.

a training set of n datapoints (ξ_i), an Expectation-Maximization algorithm is used to find the parameters π_k, μ_k, Σ_k that maximizes the likelihood (6) of this training set. ***Expectation:*** We first estimate for each point of the training set the probability that this point ξ_i is generated by each Gaussian (or state k) $p(k|\xi_i)$.

$$p(k|\xi_i) = \frac{p(\xi_i|k)p(k)}{\sum_{j=1}^{K} p(\xi_i|j)p(j)} \qquad (5)$$

where $p(\xi_i|k) = \mathcal{N}(\xi_i, \mu_k, \Sigma_k)$ and $p(k) = \pi_k$. The overall likelihood of observing the given training set with the given model parameters is:

$$\mathcal{L}(x) = \prod_{i=1}^{n} \left(\sum_{k=}^{K} p(\xi_i|k)\pi_k \right) \qquad (6)$$

This is used as measure of convergence and performance of this algorithm. ***Maximization:*** Means and covariances of each Gaussian are recomputed by weighting each data point by the probability $p(k|\xi_i)$. These two steps are iterated until convergence.

The Gaussian Mixture Regression (GMR) enables to probabilistically complement partial information. Let $\xi = [\xi^{\mathcal{O}} \xi^{\mathcal{I}}]$ be a point in the state space with $\xi^{\mathcal{I}}$ that is known. The GMR enables to estimate $\xi^{\mathcal{O}}$ by taking the mean of the expectation distribution (7) (see [Cohn et al., 1996] for more details).

$$p(\xi^{\mathcal{O}}|\xi^{\mathcal{I}}) \sim \sum_{k=1}^{K} p(k|\xi^{\mathcal{I}})\ p(\xi^{\mathcal{O}}|\xi^{\mathcal{I}}, k), \qquad (7)$$

We now consider that the state space is $\xi = [\theta\ v\ x\ y]$ with θ the absolute azimuth, v the linear velocity and x, y the Cartesian position of the robot. The data recorded during the three training runs of the NN system (see Section 2.3) are used to generate the training sets (ξ_i) for the Gaussian Mixture Model (GMM). The training is off-line. A simulation of a robot using the orientation commands retrieved from the GMR has been realized. The resulting trajectory with a 4 states GMM is shown in Figure 7. This simulation tells us that it makes sense to use such a continuous state action model to learn trajectories for a differential drive robot.

The drawback of this approach is that the absolute Cartesian position of the robot must be known, which in this case was obtained by the carefully recalibration of an odometer. This approach can be used only if the robot has access to the absolute Cartesian position. Odometry is not reliable for that purpose. Without a regular recalibration, the possible drift in the computation of the position can lead to an important inaccuracy. A robust localization system is necessary, such as a statistical localization system (Kalman Filter, Particle filter [Thrun et al., 2001]), or an external visual tracking system with accurate calibration. These methods are still costly to settle, since they require either extra computation, or extra hardware.

3.2 Navigation with GMM training using place-cell activity

In a more developmental and autonomous approach, the robot should rely on subjective visual cues to localize itself. This would allow more robustness and adaptation as the robot would not need a predetermined map of the environment to localize itself. The raw landmark-azimuth pairs introduced in the neural approach in Section 2.1 could be used. Though, as the dimension of the inputs increases, the computational time for the GMM model increases like $O(n^3)$. Moreover, the structure of the visual information with noise and occulting implies that the number of pertinent information may not be stabi-

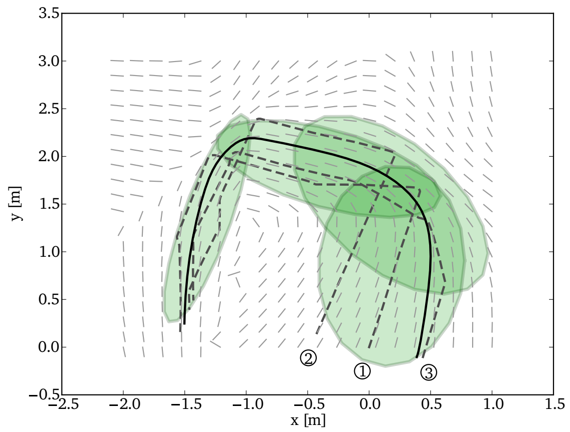

Figure 7: Generated trajectory (black), and model using information on the absolute Cartesian position ($p(\theta, v|x, y)$). Dashed lines are demonstration data, GMM states are represented with green ellipses and the light gray field lines represent target azimuths at given position.

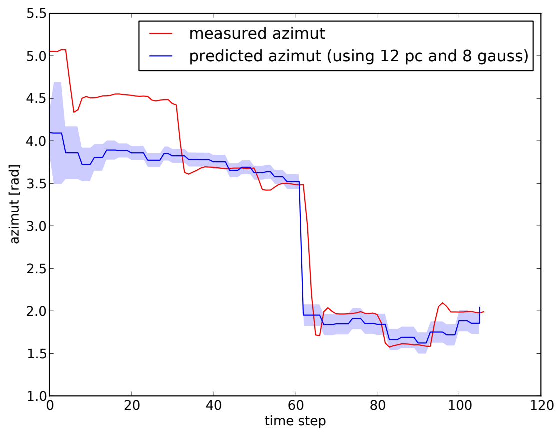

Figure 8: Predicted azimuths using place cell activation based on Equation 8 and using the data from 12 place cells. The surface in light blue represent the incertitude on the target angle given by the probability density ($\pm\sigma$).

lized which makes the training more difficult.

Place cells offer an efficient pre-processing of the visual inputs for the training. In the Neural Network model, they have proved to be robust to noise and occultation (keeping the same ordering), and they represent a smaller amount of information. The module that calculates the place-cell activities is added to the Gaussian Mixture Model system so that the state-space is now defined as $\xi = [\theta(t)\ PC(t)]$ with $PC(t) = (PC_1(t), \cdots, PC_N(t))$. The number of place cells is chosen arbitrarily. At every time step we can retrieve a continuous value for the target azimuth using (7) which becomes equation (8). It defines a continuous and probabilistic mapping between sensory inputs $PC(t)$ and the motor command $\theta(t)$.

$$\theta(t) \rightarrow p(\theta|PC(t)) \qquad (8)$$

The three training runs realized using the Neural Network system (Section 2.3) provide the training data. At the end of these runs, twelve place cells had been learned. The GM Model is trained using 8 Gaussians and these 12 place cells. A Gaussian noise of variance 0.1 is added to place cell activations during the training to simulate noisy visual inputs. Place-cell activities from the test set with the Neural Network model are given to the GMM to predict azimuth target. In Figure 8, a comparison between this azimuth and the real one from the test run with the Neural Network system shows that using place-cell activations is acceptable for retrieving a target azimuth from the GMM. Because place cells are generated during turns, there are no place cells at the beginning of trajectories. That can explain why the model performs badly at the beginning.

The previous experiment allows us to compare the retrieved orientation command on a test trajectory. We now simulate place cells by Gaussian activations. The parameters of the Gaussian activations of the place cells can be estimated from the recorded data provided by the learning and test runs for the Neural Network system. It is then possible to associate place-cell activities to every position in the environment. These activities are given to a simulated robot to reproduce the learned trajectory. Three extrapolated place cells are used. The different parts of the trajectory correspond to different dominant states of the GMM. It appears that the states defines a new topology that is built on top of the place cells. The result of the simulation is given in Figure 9. Using such an approximation shows that the GMM based system is well able to reproduced the desired trajectory by using radial basis activation for place cells.

4. Encoding and learning strategies

Both approaches succeed in solving the task. However they present differences in encoding that influences the abilities of the system. The Neural Network system is based on vision. It gathers information (azimuth, landmarks) into place cells which are associated to desired orientations when the robot trajectory is corrected. The Gaussian Mixture Model encodes statistically pertinent information. Using Cartesian coordinates as inputs is efficient for navigation, but place cells can also be used.

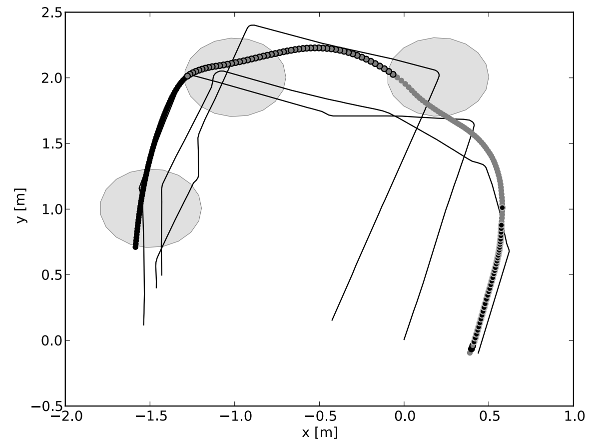

Figure 9: Simulated trajectory with a GMM system using place-cell activities approximated by Gaussians. The light gray ellipes display simulated place cells. Black thin lines are training trajectories. The reproduction trajectory is represented by successive points. For each point, the most probable Gaussian among the four in the Gaussian Mixture Model gives its design to the point. There are four different designs, one for each Gaussian state.

4.1 Optimized encoding and long time learning

With several runs and corrections whenever the NN system is not showing the required behavior, a dynamical attractor can be constructed (Figure 4). Each time the trajectory of the robot is corrected, another place cell is learned. But there is no guarantee that the new place cell will only correct the mistake. As place cells have a wide recognition activity, new place cells can interfere with older ones. The attractor stability and generalization rely on the number and the positions of place cells recruited to learn the actions [Giovannangeli and Gaussier, 2010]. Several place cells can be needed to stabilize the dynamics over the desired path, especially under strong constraints (if small variance on the resulting trajectory is needed). With many place cells, there can be a redundancy in the encoded information. A specific implementation for adaptive orientation cells can enhance the model [Giovannangeli and Gaussier, 2010]. Before recruiting new place cells, it is possible to try to adapt the orientation associated to an already known cell. However, the optimized repositioning or pruning of place cells is currently an unsolved problem. As learning continues, with changes of the environment, more and more place cells may be recruited leading to an oversampling of the space.

Generalizing from several samples enables the GMM system to optimize the encoding of the trajectory by computing Gaussian states that minimize redundancy and maximize available information. There is no risk of bad interference as every state participates during learning. With the tested implementation, the number of Gaussians that will be used by the system must be given in advance. Other specific implementations can solve this problem and enable incremental learning [Calinon et al., 2009]. As learning continues, the first datasets may become obsolete. If the environment changes too much, the system requires a complete re-learning from new correct data.

4.2 Interaction

The presented Neural Network implements on-line learning by correction of the behavior. The robot can reproduce immediately a learned walk, even if it may not exactly be the one that is shown. As the teacher can directly see what has been learned by the robot, bad orientation can be instantaneously corrected. This new association immediately modifies the dynamics of the robot which can be corrected again, etc. This incremental approach enables the system to focus on ill-learned part of the trajectory.

The learning of the GMM system can be qualified of slow as the robot must first be driven completely passively before it can do anything. The whole trajectory must be shown several times so that the robot get enough training data. The learning is off-line and requests several samples of data so that statistics about the task can be built. Each sample requires the teacher to realize a trajectory from start to end so they are tedious for the human teacher. The encoding depends on the complete training. What was learned by the robot can not be known before the end of the training. If the robot must be corrected, another set of demonstrations by the human teacher is necessary to retrain the system.

4.3 Quality of the learned trajectory

The trajectory generated using GMM-based system is a smoother approximation of the U-shaped desired trajectory than the one generated with the Neural Network system. In the Neural Network system, a possible solution to avoid interferences is to use discretized orientations and a competition between the place cells. When using a strict competition, the trajectories often appear as straight concatenated lines, and do not seem very natural. A soft competition over the recognized place cells suppress this problem (a few winner place cells activate a mixture of associated directions allowing smoother trajectories). In the GMM system, since both inputs and outputs are continuous, when the system is between two known positions, it automatically combine the required behaviors yielding nicer trajectories. Because the actions are demonstrated by a human, the desired trajectory is hidden by variations. The statistical analysis of the datasets enables the GMM to retrieve the optimal trajectory and it can even determine the constraints and degrees of freedom for a given walk. According to the variance of the expectation distribution, the trajectory can be more or less constrained.

4.4 Explicit and implicit encoding, fast and slow learning

The difference in encoding can be summarized by explicit versus implicit encoding. The NN system directly encodes the state-action associations that are perceived during learning. The GMM system distributes the implicit encoding between different states. No state is specifically related to an event. This difference of encoding is bound to a difference in learning. Explicit associations can be learned rapidly and independently like corrections in the NN system. On the contrary, implicit associations require several data. The learning is slowed down by the need of several passive demonstration to create several training datasets. The system can not reproduce any action until the end of this training.

5. Discussion

The context of this study is learning in a situation of interaction. More specifically, we focused on two learning by demonstration approaches that are based on state-action associations. We do not tackle reinforcement learning as it does not really correspond to demonstration by a human teacher. The robot can not explore its environment to build reinforcement evaluation. It must use the information provided by the interaction. Linking interaction and reinforcement is an ongoing work (see [Hirel et al., unpublished]).

In [Zukowgoldring and Arbib, 2007], the authors defend the hypothesis that the main feature of interaction between an infant and its caregiver is not the direct transfer of knowledge but the reduction of the research space in order to help the infant to learn new tasks. The fast corrective learning in the Neural Network system corresponds quite well to this definition. As the lively space is reduced, the robot is led to reproduce what the human teacher wants it to do with more or less accuracy. But, this is limited. Rather than increasing more and more the number of corrective rules that define the action, a statistical analysis of the structure of the possible actions could optimize the encoding. In the case of a robot, the GMM system could complement the Neural Network system to enhance the learning abilities.

This reflexion driven by a developmental point of view is also based on an anatomical analysis. [Eichenbaum et al., 1994] stresses the complementarity of two distinct structures in the brain that are the Neocortex and the Hippocampus. These structures have specificities which are very similar to the specificities of the two encoding and learning strategies that we study in this paper. The Hippocampus is an explicit associative memory that can acquire rapidly information. As there is little inference from the encoded items, the interferences are limited. It is also related to novelty detection like changes of orientation in the case of our robot. This structure corresponds well to the place-cell associations learned in the Neural Network architecture. The Neocortex is an implicit memory as it does not directly encode specific events. It can discover gradually the statistical structures of experiences, by accumulating learning and data. This structure corresponds to the GMM process. Motor and PreMotor Cortex can be the location where the invariant features of movements and trajectories are retained. Some substructures and other structures of the brain participates in specific manners in the cognitive process. The Entorhinal Cortex interfaces the hippocampus and the neocortex. In [Arleo and Gerstner, 2000], Entorhinal Cortex is the location of the place cells. Place-cell activities can be used by both structures and enable exchanges between Hippocampus and Neocortex. This corresponds to the system we simulated in Section 3.2. The cerebellum provides interpolation and prediction abilities to the brain system. With interpolation and conditioning, it can improve the quality of movements. There exist a transfer of knowledge between the hippocampal episodic short term memory and the neocortical long term memory [McClelland et al., 1995]. This happens mainly during sleeping time. The cerebellum can play an important role in this process. Its predictive capacities can be used for internal rehearsal of episodic memories so that Neocortex can learn implicit representation. This long term representation can then be used by the Striatum to generate routine movements. During the transfer between short term and long term memory, a change in representation, ie. of encoding, can occur. This process is called memory consolidation. It has been observed in animals and in human [McClelland et al., 1995]. In [Kulić and Nakamura, 2009], memory consolidation is used with incremental learning so that the system can learn on-line with additive stability coming from consolidation memory which occurred both on-line during wake time and off-line during equivalent sleeping time. This consolidation enabled a better categorization of the action. This memory consolidation has also proved to make human-robot collaboration easier in [Ogata et al., 2004].

In this paper, the NN system and the GMM system have been studied separately. As a future work, we suggest that they are gathered in a whole architecture that benefits from the complementarity of the two strategies. Once the robot has acquired a new rough behavior, it can reproduce the trajectory over and over providing the necessary datasets for GMM training. Learning can be done in two times: a first rough learning of the task which is then refined using more data to determine the invariant features of the actions that solves the task. One model can take over the second one according to the situation. When facing already known situations, the GMM based system can produce an optimized adapted behavior and when facing new situations, the Neural Network enables interactive corrections to incrementally generate an adapted behavior.

The discussion provided here has been based on a navigation experiment, but the conclusion should be extended to action selection problem in general. We believe that the same explicit/implicit memory structures can be involved not only in navigation, but also in manipulation tasks or even higher complex tasks mixing different kinds of behaviors. Future works will study this hypothesis. As it can be seen in Section 4, the conclusions drawn may not be restricted to the algorithms studied in this paper. The conclusions should be extended to algorithms that could be classified either as fast learning, explicit encoding or as slow learning, implicit encoding.

In conclusion, the two strategies studied in this paper must be considered as two complementary encoding and learning strategies. One can not replace the other one, it is not a matter of trade-off between the two strategies. According to how evolution built our brain, both strategies must be present in order to provide the cognitive system with most efficiency as considering reactivity and interactivity during learning with good inferences to optimally retain any demonstrated knowledge.

6. Acknowledgement

This work was supported by the French Region Ile de France, the Institut Universitaire de France (IUF), the FEELIX GROWING european project and the INTERACT french project referenced ANR-09-CORD-014.

References

Arleo, A. and Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, 83(3):287–299–299.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Survey: Robot Programming by Demonstration. In *Handbook of Robotics*, volume chapter 59. MIT Press.

Calinon, S., Caldwell, D., Billard, A., and D'halluin, F. (2009). A probabilistic approach to learn and reproduce dynamics of human motion by imitation. In *Proceedings of 2009 IEEE International Conference on Humanoid Robots*.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *CoRR*, cs.AI/9603104.

Eichenbaum, H., Otto, T., and Cohen, N. J. (1994). Two functional components of the hippocampal memory system. *Behavioral and Brain Sciences*, 17(03):449–472.

Giovannangeli, C. and Gaussier, P. (2010). Interactive teaching for vision-based mobile robots: A sensory-motor approach. *IEEE Transactions on Man, Systems and Cybernetics, Part A: Systems and humans*, 40(1):13–28.

Giovannangeli, C., Gaussier, P., and Désilles, G. (2006). Robust mapless outdoor vision-based navigation. In *IEEE/RSJ International Conference on Intelligent Robots and systems*, Beijing, China. IEEE.

Kulić, D. and Nakamura, Y. (2009). Incremental learning and memory consolidation of whole body human motion primitives. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 17(6):484–507.

Lagarde, M., Andry, P., Gaussier, P., Boucenna, S., and Hafemeister, L. (2010). *Proprioception and Imitation: On the Road to Agent Individuation*, volume 264 of *Studies in Computational Intelligence*, pages 43–63. Springer Berlin / Heildelberg.

McClelland, J. L., Mcnaughton, B. L., and O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychol Rev*, 102(3):419–457.

Ogata, T., Sugano, S., and Tani, J. (2004). Open-end human robot interaction from the dynamical systems perspective: Mutual adaptation and incremental learning. In *Innovations in Applied Artificial Intelligence*, pages 435–444. Springer Berlin / Heidelberg.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141.

Zukowgoldring, P. and Arbib, M. (2007). Affordances, effectivities, and assisted imitation: Caregivers and the directing of attention. *Neurocomputing*, 70(13-15):2181–2193.