Original Paper

Adaptive Behavior

Adaptive Behavior
1–15

DOI: 10.1177/1059712315608424
adb.sagepub.com
SAGE

Integrating learning by experience and demonstration in autonomous robots

Paolo Pagliuca and Stefano Nolfi

Abstract

We propose an integrated learning by experience and demonstration algorithm that operates on the basis of both an objective scalar measure of performance and a demonstrated behaviour. The application of the method to two qualitatively different experimental scenarios involving simulated mobile robots demonstrates its efficacy. Indeed, the analysis of the obtained results shows that the robots trained through this integrated algorithm develop solutions that are functionally better than those obtained by using either a pure learning by demonstration, or a pure learning by experience algorithm. This is because the algorithm drives the learning process toward solutions that are qualitatively similar to the demonstration, but leaves the learning agent free to differentiate from the demonstration when this turns out to be necessary to maximize performance.

Keywords

Learning by demonstration, learning by experience, autonomous robot, evolutionary strategies

1. Introduction

Humans frequently learn by using a combination of demonstration and trial and error. 'When learning to play tennis, for instance, an instructor will repeatedly demonstrate the sequence of motions that form an orthodox forehand stroke. Students subsequently imitate this behaviour, but still need hours of practice to successfully return balls to a precise location on the opponent's court' (Kober, Bagnell, & Peters, 2013, p. 1257).

From a machine learning perspective, the combined use of learning by demonstration (Argall, Chernova, Veloso, & Browning, 2009; Billard, Callinon, Dillmann, & Schaal, 2008) and learning by experience (Kober et al., 2013; Nolfi & Floreano, 2000; Sutton & Barto, 1998) provides advantages and drawbacks. Firstly, it enables the learning agent to exploit a richer training feedback constituted by both a scalar performance objective (reinforcement signal or fitness measure) and a detailed description of a suitable behaviour (demonstration). Secondly, it permits to combine the complementary strengths of the two different learning modes. Indeed, the possibility to observe proper behaviours frees the agent learning by demonstration from the need to find out the behavioural strategy that can be used to solve the task and, consequently, narrows down the goal of the learning process only to the discovery of how the demonstrated behaviour can be re-produced. Moreover, the possibility to rely on the objective scalar measure evaluating the functionality of the overall robot's behaviour enables the learning robots to select variations that represent real progresses and to exploit properties that emerge from the agent/environmental interactions.

On the other hand, the two learning modes have also drawbacks. Indeed, in learning by demonstration, the attempt to solve the task by approximating a demonstrated solution, rather than directly optimizing performance, exposes learning agents to the problems caused by the fact that minor differences between the demonstrated and reproduced behaviours might cumulate over time eventually causing huge undesired effects. In learning by experience, the selection of variations that represent genuine advance with respect to the current capabilities of the agent might drive the learning process toward local minima that might then prevent the discovery of better solutions. Furthermore, the combination of these two learning modes has potential shortcomings as well. In effect, channelling the learning

Institute of Cognitive Sciences and Technologies, National Research Council (CNR), Roma, Italia

Corresponding author:
Paolo Pagliuca, Institute of Cognitive Sciences and Technologies, National Research Council (CNR), Via S. Martino della Battaglia, 44 00185 Roma, Italia
Email: paolo.pagliuca@istc.cnr.it

process through demonstrated solutions is likely to be beneficial only when the demonstrated behaviour does represent a solution that is effective and learnable from the point of view of the learning agent. This is not necessarily the case when the learning agent is a robot and the demonstrator is a human, i.e., when the demonstrator and the learning agent have different body structures and different cognitive capabilities. Finally, the combined effects of two different learning methods can drive the learning process toward the synthesis of intermediate solutions, representing a sort of compromise between the changes driven by the two learning modes, which are not necessarily functionally effective.

Over the last few years, several researchers have proposed hybrid learning by demonstration and experience methods for robot training. In particular, Rosenstein and Barto (2004) extended an actor critic reinforcement learning (Sutton & Barto, 1998) with a supervisor agent that co-determines, together with the learning agent, the actions to be executed at each time step. These actions are generated by performing a weighted sum of the actions proposed by the supervisor and the learning agent. The parameter that trades off between the two actions is varied by taking into account the estimated efficacy of both the supervisor and the agent in specific circumstances and the need to increase the autonomy of the agent progressively during the learning process. Overall, the results obtained in a series of case studies demonstrate that the combination of the two learning modes can provide advantages. Nonetheless, the strategy of averaging the actions suggested by the supervisor and the agent might limit the efficacy of the algorithm to domains in which the negative effects caused by the summation of incongruent actions are negligible. In a related work, Judah, Roy, Fern, and Dietterich (2010) incorporated user advice into a reinforcement learning algorithm. More specifically, learners alternate between practice (i.e., experience) and end-user critique, where advice is gathered. The user analyses the behaviour of the learners and marks subset of actions as good or bad. The policy is optimized through a linear combination of practice and critique data. Results obtained in a series of experiments demonstrate that the approach significantly outperformed pure reinforcement learning. However, the study revealed usability issues related to the amount of practice and advice needed to achieve successful performance.

Kober and Peters (2009) investigated how a special kind of pre-structured parameterized policies called motor primitives (Ijspeert, Nakanishi, & Schaal, 2004; Schaal, Mohajerian, & Ijspeert, 2007) can be subjected to a combined learning by demonstration and experience training (Daniel, Neumann, & Peters, 2012; Paraschos, Daniel, Peters, & Neumann, 2013). Motor primitives are constituted by two coupled parametrical differential equations. The equations are hand-designed, while the equation parameters are learned. Learning occurs in two phases: first, the parameters are subjected to a learning by demonstration process; then, the parameters are refined on the basis of a reinforcement learning process. This method has been successfully applied to a series of distal rewarded tasks that could not be solved through learning by demonstration by itself. However, it operates appropriately only when the amount of deviation from the demonstrated trajectories is actively limited (Kober & Peters, 2009; Kober et al., 2013; Valenti, 2013). As a result, the method can only be successful when the learning by demonstration phase enables the learning robot to acquire a close-to-optimal strategy that only requires subsequent fine-tuning.

Argall, Browning, and Veloso (2008) proposed a method for enriching the demonstration data set with advice produced by the human demonstrator during the observation of the behaviour exhibited by the learning robot. Advice consist in required modifications such as 'turn less/more tightly' or 'use a smoother translational speed' that can be easily identified by a human observer and can be used to automatically generate additional demonstration data obtained through a corrected version of the behaviour displayed by the learning robot. The collected results indicate that this technique can outperform standard learning by demonstration methods. Consequently, as hypothesized, personalizing the demonstration depending on the characteristics of the learning robot can improve the outcome of the learning process. Nevertheless, this method only operates by attempting to reduce the discrepancy between the actual and the demonstrated behaviour and does not combine this with a learning by experience method driven by a direct measure of performance. Therefore, it does not provide a way to overcome the problem caused by the fact that minor inevitable differences between the demonstrated and the reproduced behaviour can lead to large negative consequences over time.

Other related works are those of Abbeel and Ng (2005), Cetina (2007), Chernova and Veloso (2007) and Knox and Stone (2009, 2010). Abbeel and Ng (2005) demonstrated the advantage of initializing the state-action pairs of a reinforcement learning based on a learning by demonstration method. Cetina (2007) implemented an algorithm in which the advice of the user is used to restrict the set of possible actions explored by the reinforcement learning. Chernova and Veloso (2007) proposed a learning by demonstration approach, in which the learning agent can progressively increase its autonomy by reducing the number of demonstrations required from the teacher. Knox and Stone (2009) proposed a reinforcement learning method that operates on the basis of short-term rewards provided by human trainers that predict the expected long-term effect of single actions or short sequence of actions.

A still unanswered question of all the presented approaches concerns how different training feedbacks can be effectively integrated. Indeed, the different learning modes (i.e., learning by experience and learning by demonstration) often drive the learning agents toward different outcomes, whose effect is not necessarily additive. For example, the combination of the two processes in sequence, that can be obtained by first subjecting the robot to a learning by demonstration phase and then to a learning by experience phase, might cause the washing out of the capabilities acquired during the first phase at the beginning of the second phase (Kober et al., 2013; Valenti, 2013). Similarly, a combination achieved by summing up the effects of the two processes might lead to undesirable consequences.

In this paper, we introduce a new method that integrates the learning by demonstration and by experience in a single algorithm. This algorithm operates by estimating the local gradient of the current candidate solution with respect to an objective performance measure and exploring preferentially variations that reduce the differences between the robot and the demonstrated behaviour.

The results obtained on two qualitatively different tasks demonstrate how the algorithm is able to synthesize effective solutions. Such solutions are qualitatively similar to the demonstrated behaviour with respect to the characteristics that are functionally appropriate, while deviate from the demonstration with respect to other characteristics.

In the next section, we illustrate the first experimental setting and the learning algorithm. In section 3, the obtained results will be described. The second experimental setting and the corresponding obtained results are described in sections 4 and 5. Finally, in section 6 we draw our conclusions.

2. Exploration experiment

In a first experiment, a simulated Khepera robot (Mondada, Franzi, & Ienne, 1993) has been trained for the ability to explore an arena surrounded by walls that contains cylindrical obstacles located in randomly varying positions (Figure 1). With the term explore we mean 'to visit the highest possible number of environmental locations'. To verify the advantages and disadvantages of different learning modes and of their combination we carried out three series of experiments in which the robot has been trained through a learning by experience, a learning by demonstration, and an integrated learning by experience and demonstration algorithm. The three algorithms are described in sections 2.1, 2.2 and 2.3.

The robot is provided with six infrared sensors located in its front side that enable it to detect nearby obstacles and two motors controlling the desired speed

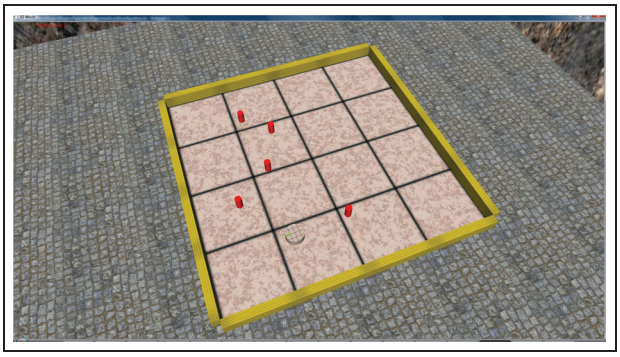

Figure 1. Exploration experiment. The robot and the environment.

of the two wheels. The robot's controller consists of a three-layer feed-forward neural network with six sensory neurons, which encode the state of the six corresponding infrared sensors, six internal neurons, and two motor neurons, which encode the desired speed of the two wheels. The activation of the infrared sensors is normalized in the range [0.0, 1.0]. The activation of the internal and motor neurons is computed based on the logistic function. The activation of the motor neurons is normalized in the range [−10.4, 10.4] cm/s. Internal and motor neurons are provided with biases.

The environment consists of a flat arena of 1.0×1.0 m, surrounded by walls, containing five cylinders with a diameter of 2.5 cm located in randomly varying positions. The robot and the environment have been simulated by using FARSA (Massera Ferrauto, Gigliotta, & Nolfi, 2013), an open software tool that has been used to successfully transfer results obtained in simulation to hardware for several similar experimental settings (e.g., De Greef & Nolfi, 2010; Sperati, Trianni, & Nolfi, 2008).

The performance of the robot, i.e., the ability to visit all environmental locations, is computed by dividing the environment into 100 cells of 10×10 cm and counting the number of cells that have been visited at least once over a period of 75 s during which the robot is allowed to interact with the environment. To select robots able to carry out the task in variable environmental conditions, the robot's performance is evaluated during 25 trials, each lasting 75 s. The position and the orientation of the robot and the position of the obstacles are randomly initialized at the beginning of each trial. The overall performance is calculated by averaging the performances obtained during the different trials. It is worth noting that the optimal performance is unknown since the time for exploration is limited and some of the cells are occupied by obstacles.

The 48 connection weights of the neural network and the eight biases are the free parameters that are initially set randomly and then learned by using the algorithms described in the next sections. The values of

these parameters determine the control policy of the robot, i.e., how the robot reacts to the experienced sensory states.

2.1 Learning by experience

In learning by experience methods, the learning robot discovers either the behaviour through which the task can be solved, or the control rules that, in interaction with the environment, lead to the production of the selected behaviour. The learning process is driven by a scalar measure of performance that rates the extent to which the robot is able to accomplish the task. The utilization of a performance measure that does not specify how the problem should be solved potentially enables the learning robot to find effective solutions that are often simpler and more robust than those identifiable by human designers (Argall, Browning, & Veloso, 2010; Coates, Abbeel, & Ng, 2008; Taylor, Suay, & Chernova, 2011). On the other hand, the use of such an implicit training feedback, that constraints only loosely the course of the adaptive process, can initially drive the learning process toward the synthesis of sub-optimal behavioural solutions that constitutes local minima (i.e., that cannot be progressively transformed into better solutions without producing performance drops).

A first way to achieve a learning of this kind consists in using a stochastic hill climber algorithm (Russell & Norvig, 2009), in which the free parameters of the robot's controller are randomly modified and the variations are retained or discarded depending on whether or not they produce an improvement of the robot's performance.

A second option consists in using a reinforcement learning (Sutton & Barto, 1998) that operates by calculating the estimated utility of possible alternative actions on the basis of received rewards and using this acquired information to preferentially select the actions with the highest expected utilities.

A third way entails the use of an evolutionary algorithm (Holland, 1975; Nolfi & Floreano, 2000; Schwefel, 1995) that operates on a population of candidate solutions. These solutions are selected based on their relative performance and varied through the production of offspring, i.e., copies modified through genetic operators such as mutation and crossing over.

Finally, a fourth option is provided by natural evolution strategies (NES) (Wierstra, Schaul, Peters, & Schmidhuber, 2008; also Rechenberg & Eigen, 1973; Schwefel, 1977) that operate by sampling the local gradient of the candidate solution (i.e., the correlation between variations of the free parameters and variation of performance) and modifying the candidate solution toward the most promising directions of the multi-dimensional parameter space. Samples are generated by creating varied copies of the candidate solution.

In our experiments, we used the xNES algorithm (Glasmachers, Schaul, Yi, Wierstra, & Schmidhuber, 2010), i.e., an algorithm of the latter type since, as we will see, it can be easily extended to incorporate also learning-by-demonstration feedbacks. The xNES algorithm has already been successfully tested on a series of domains including autonomous robot learning (Glasmachers et al., 2010).

More specifically, the xNES algorithm works as described in Algorithm 1. The notation used differs from (Glasmachers et al., 2010) in order to allow the reader to understand all the steps of the algorithm easily with plenty of details.

$$ind\tilde{\pi}(\bullet|\theta) \tag{1}$$

where π is a Gaussian distribution with parameters $\theta=(\mu,\sigma)$. μ represents the mean of the Gaussian distribution and it is set to 0.0. σ stands for the standard deviation of the Gaussian distribution and it is set to 1.0.

$$\lambda = 4 + floor(3 \cdot log(p)) \tag{2}$$

where p is the number of free parameters, and *floor* indicates the inferior integer part of the number.

$$\eta_{covM} = 0.5 \cdot max\left(0.25, \frac{1}{\lambda}\right) \tag{3}$$

where λ is the number of offspring.

$$u_k = \frac{max\left(log\left(\frac{\lambda}{2}+1\right) - log(k)\right)}{\sum max\left(log\left(\frac{\lambda}{2}+1\right) - log(j)\right)} \tag{4}$$

where λ is the number of offspring and k is the index of the kth offspring.

2.2 Learning by demonstration

In learning by demonstration, the control policy is learned from examples or demonstrations provided by a teacher (Argall et al., 2009; Schaal, 1997). The objective of the learning process is therefore to reproduce the demonstrated behaviour. In particular, the learning algorithm should enable the discovery of the control mechanisms that, in interaction with the environment, yield the demonstrated behaviour (Billard, Balinon, & Guenter, 2006). The examples can be defined either as the sequences of sensory-motor pairs that are extracted from the teacher demonstration (offline demonstration learning) (Hayes & Demiris, 1994), or as the sequences of the desired action specifications that the teacher provides to the learning robot while it is acting (online demonstration learning) (Rosenstein & Barto, 2004).

In general terms, the availability of the demonstration reduces the complexity of the learning task and enables the use of potentially more powerful supervised

Algorithm 1: the xNES algorithm.

initialize:
  the candidate solution (*ind*) by selecting the best over 100 parameters' sets generated by using Equation 1
  the number of offspring (λ), see Equation 2
  the covariance matrix (*covMat*) with zeroed values. The covariance matrix encodes the correlation between free parameters of high utility samples. The utility of a sample is the relative performance advantage with respect to the other generated samples.
  the learning rate used to update the individual (η_{ind}). We use $\eta_{ind} = 1.0$.
  the update rate of the covariance matrix (η_{covM}), see Equation 3
repeat
  for $k = 1 \ldots \lambda$ do
    generate random variation vectors with a Gaussian distribution
      $s_k \tilde{\pi}(.|\theta)$
    generate samples by adding the product of the variation vectors by the exponential of the covariance matrix to the candidate solution
      $z_k = ind + e^{covMat} \cdot s_k$
  evaluate the fitness of the samples
      $f_k = F(z_k)$
  end
  sort z_k with respect to their performance f_k
    $ranking = z_1, \ldots, z_\lambda, F(z_1) \geq \ldots \geq F(z_\lambda)$
  calculate the utility of the samples u_k
    u_k, see Equation 4
  sum-up variations weighted by utilities
    $\nabla_{ind} = \sum u_k \cdot s_k$
  calculate the estimated local gradient
    $\nabla_{covM} = \sum u_k \cdot \left(s_k \cdot s_k^T - I\right)$, where I is the identity matrix, and T superscript represents the matrix transposition operation
  move the candidate solution depending on the local gradient with the given learning rate
    $ind = ind + \eta_{ind} \cdot e^{covMat} \cdot \nabla_{ind}$
  update the covariance matrix based on the estimated local gradient with the given update rate
    $covMat = covMat + \eta_{covM} \cdot \nabla_{covM}$
  for *500 iterations*

learning techniques. On the other hand, the behaviour demonstrated by the teacher might be impossible to learn from the robot's point of view (i.e., the demonstrator might be unable to identify behaviours learnable by the robot). Besides, the acquisition of an ability to produce a behaviour that is very similar but not identical to the demonstration does not guarantee the achievement of good performances since small differences might cumulate over time leading to significant differences.

In our experiment, we generated the demonstrations automatically through a demonstration robot provided with a hand-designed controller. This allowed us to generate a large number of demonstrations and to use exactly the same kind of demonstration in all experiments. The hand-designed controller of the demonstrator: (1) moves the robot straight at the highest possible speed when the infrared sensors are not activated, (2) makes the robot turn left or right when the two right or the two left infrared sensors are activated respectively, and (3) makes the robot turn left when both the two left and the two right infrared sensors are activated. This is made by setting by default the desired speed of the two wheels to the highest positive value and by proportionally reducing the speed of one wheel according to the average activation of the two opposite infrared sensors. It is worth noting that the optimal relation between the activation of the infrared sensors and the speed of the wheels, with respect to the ability of the robot to explore the arena, is not necessarily proportional and homogeneous for the two sensors located on either side. Nevertheless, since there is no clear way for determining this relation, we used a simple proportional and homogeneous relation.

The fact that certain characteristics are hard to optimize from the point of view of an external designer constitutes one of the reasons that explain why combining learning by demonstration and experience can be beneficial. Indeed, as we will see, the integrated learning by experience and demonstration (E&D) algorithm can find solutions that are better optimized in this respect.

In the offline demonstration mode, the sequence of sensory-motor pairs experienced by the demonstrator robot, while it operates in the environment for several

trials, are used to train the neural network controller of the learning robot. The training is performed by using the demonstrations as training set and learning the weights of the robot's neural controller through back-propagation (Rumelhart, Hinton, & Williams, 1986). More specifically, the sensory states experienced by the demonstrator robot and the motor actions produced by the demonstrator robot every 50ms are used to set the sensory state and the teaching input of the learning robot. Conversely, in the online demonstration mode, the learning robot is situated in the environment and can operate based on its own motor neurons. To generate the teaching input, the demonstrator robot is virtually placed in the same location (i.e., position and orientation) of the learning robot at each time step. The output actions produced by the demonstrator robot in this situation are used to set the teaching input of the learning robot.

In both cases, a learning rate of 0.2 was used and the learning process was continued for 1.875×10^7 time steps of 50 ms. This number was chosen in order to keep the overall period of evaluation constant with respect to the previous algorithm.

2.3 Integrated learning by experience and by demonstration

Here we propose a new algorithm that enables the realization of an integrated learning by E&D. The algorithm is so designed that the two learning modes can operate simultaneously by driving the learning process toward solutions that are functionally effective and similar to the demonstrated behaviour. The combination of the effects of the two learning modes is not realized at the level of the actions, as in the case of Rosenstein and Barto (2004), but rather at the level of the variations of the free parameters. In other words, the algorithm operates by combining variations that are expected to produce progress in objective performance with variations that are likely to increase the similarity with the demonstrated behaviour. As a whole, this implies that the algorithm tries to steer the learning by experience process toward solutions that resemble the demonstrated behaviour.

To obtain such an integrated algorithm, we designed an extended version of the xNES algorithm, described in section 2.1, which operates by iteratively estimating and exploiting both the local performance and the demonstration landscape. This has been made by training the current candidate solution for a limited time (i.e., 25 trials as in the case of the learning by experience algorithm) based on the demonstrated behaviour. More precisely, the integrated algorithm works as described in Algorithm 2:

Readers might replicate these experiments and the experiments described in section 4 by downloading and installing FARSA from https://sourceforge.net/projects/farsa/ and using the experimental plugins named KheperaExplorationExperiment and RobotBallApproachExperiment, respectively.

3. Results for the exploration experiment

In this section, we report the results obtained in three series of experiments performed by using the three algorithms described above. For each experiment, we ran 10 replications starting with different randomly generated candidate solutions.

The performance displayed by the best robots at the end of the training process (Figure 2) shows that the robots trained with the integrated learning by E&D algorithm outperform both the robots trained the learning by demonstration (D) algorithm (Mann–Whitney test, $df = 10$, $p < .01$) and the robots trained with the learning by experience (E) algorithm (Mann–Whitney test, $df = 10$, $p < .01$).

By analysing the behaviour displayed by the robots trained through the three learning algorithms (Figure 4) and the percentage of times these robots avoid obstacles by turning toward they preferred direction (Table 1), we can see that, as expected, the robot trained in the D and E&D conditions display a behaviour similar to the demonstration. The robots trained in the E condition, instead, exhibit a behaviour that differs qualitatively from the demonstration. In particular, they avoid the obstacles by turning always, or almost always, in the same direction. This qualitative difference plays a functional role with respect to the exploration task: indeed, avoiding obstacles by mostly turning in the same direction increases the probability to end up in previously explored locations of the environment compared with avoiding obstacles by turning in both directions.

The fact that the robots trained in the E experimental condition initially develop the ability to avoid obstacles by always turning in the same direction is not surprising. This simple strategy enables the robots to improve their performance with regard to the initial phase in which they are still unable to avoid obstacles. Yet, the acquisition of this strategy and its further refinement then blocks the learning process into a local minimum, i.e., a situation in which it is not possible to change the obstacle avoidance strategy without impacting negatively on performance.

The integrated learning by E&D algorithm overcomes this local minima problem thanks to its ability to drive the learning process toward solutions that are similar to the demonstration, i.e., toward solutions that avoid obstacles by turning either left, or right, depending on the relative position of the obstacle.

The behaviour of the robot trained in the E&D condition is similar to the demonstrated behaviour concerning the direction, but not with regard to the

Algorithm 2: the integrated learning by experience and demonstration algorithm.

initialize:
the candidate solution (*ind*) by selecting the best over 100 parameters' sets generated by using Equation 1
the number of offspring (λ), see Equation 2
the covariance matrix (*covMat*) with zeroed values
the learning rate used to vary the individual (η_{ind}). We use $\eta_{ind} = 1.0$.
the update rate of the covariance matrix (η_{covM}), see Equation 3
repeat
for $k = 1 \ldots \lambda$ do
generate random variation vectors with a Gaussian distribution
$s_k \tilde{\pi}(.|\theta)$
generate a special sample $<S_d>$ by training the candidate solution *ind* for 25 trials
for *25 trials* do
$w = w + backprop(ind)$
end
$\langle S_d \rangle = ind + w$
where the update of the free parameters (w) is computed by means of the back-propagation algorithm. In particular, the batch mode described in section 2.2 has been used
generate samples by adding the product of the variation vectors by the exponential of the covariance matrix to the candidate solution
$z_k = ind + e^{covMat} \cdot s_k$
evaluate the fitness of the samples
$f_k = F(z_k)$
end
sort z_k with respect to their performance f_k
$ranking = z_1, \ldots, z_\lambda, F(z_1) \geq \ldots \geq F(z_\lambda)$
calculate the utility of the samples u_k, $<S_d>$ is always ranked as second (the aim is at taking into consideration both the performance and the demonstration feedback)
u_k see Equation 4
sum-up variations weighted by utilities
$\nabla_{ind} = \sum u_k \cdot s_k$
calculate the estimated local gradient
$\nabla_{covM} = \sum u_k \cdot \left(s_k \cdot s_k^T - I\right)$, where I is the identity matrix, and T superscript represents the matrix transposition operation
move the candidate solution depending on the local gradient with the given learning rate
$ind = ind + \eta_{ind} \cdot e^{covMat} \cdot \nabla_{ind}$
update the covariance matrix based on the estimated local gradient with the given update rate
$covMat = covMat + \eta_{covM} \cdot \nabla_{covM}$
for *500 iterations*

trajectory followed to avoid obstacles (Figure 4). This trajectory depends on the relative reduction of the speed of the left or right wheel when the right or left infrared sensors are activated. In the case of the demonstrator robot, this relative reduction is proportional to the activation of the two left or of the two right infrared sensors. However, as you might remember, this relation is not optimized. It has been chosen simply because we, as the demonstrator designer, were unable to identify the best relation. The robots trained in the E&D experimental condition perform better than the demonstration in this respect (i.e., avoid the obstacles with a trajectory that maximize the exploration ability). Consequently, not only do they outperform the robots trained in the E condition, but they also have a more effective behaviour than the robots trained in the D condition (Figure 2).

Overall, this implies that the E&D algorithm leads the learning robots toward solutions that incorporate the functionally effective characteristics of the demonstrated behaviour, while deviate from the characteristics of the demonstrated behaviour that are not functional.

4. Spatial positioning with heading experiment

In this section we report the results of a series of experiments carried out in a more complex scenario in which a robot should reach a 2D planar target position with a given target orientation (i.e., a position and

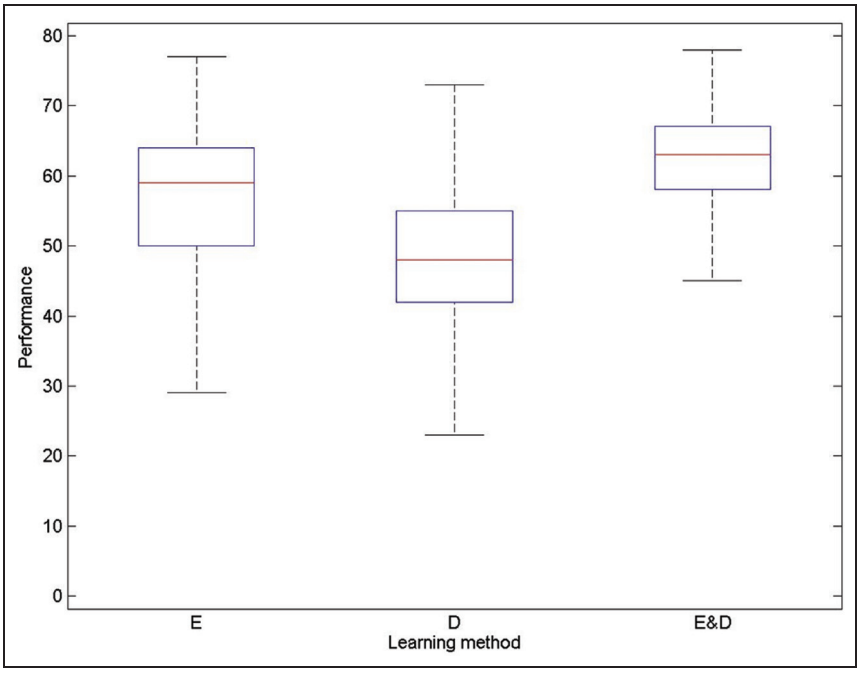

Figure 2. Boxplots of performance obtained, in the learning by experience (E), learning by demonstration (D) and integrated learning by experience and demonstration (E&D) experimental conditions. Each boxplot displays the performance obtained by post-evaluating the best 10 robots of each replication for 500 trials in environments including five obstacles. Boxes represent the inter-quartile range of the data, while the horizontal lines inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 times the inter-quartile range from the box. For the D condition, we only report the result obtained in the offline mode that yielded better results than the online mode.

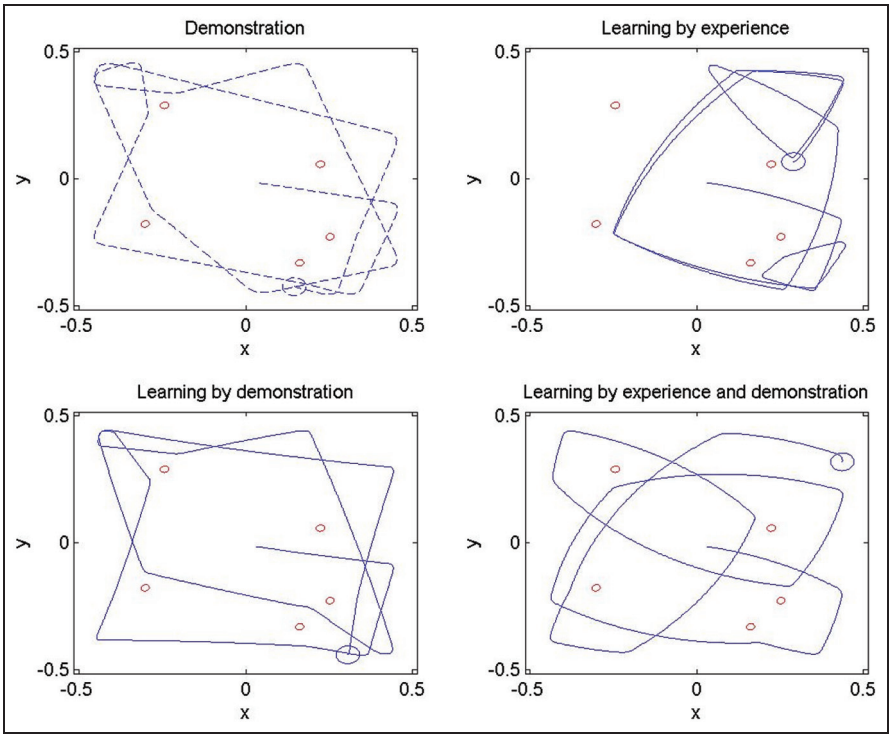

Figure 3. Trajectories displayed by the best robots of the three experimental conditions and by the demonstrator robot during a typical trial. To ensure that the trials shown are representative, they have been selected among the trials that produced a performance similar to the average performance obtained in each condition. The blue circle indicates the position of the robot at the end of the trial. The red circles indicate the position of the obstacles.

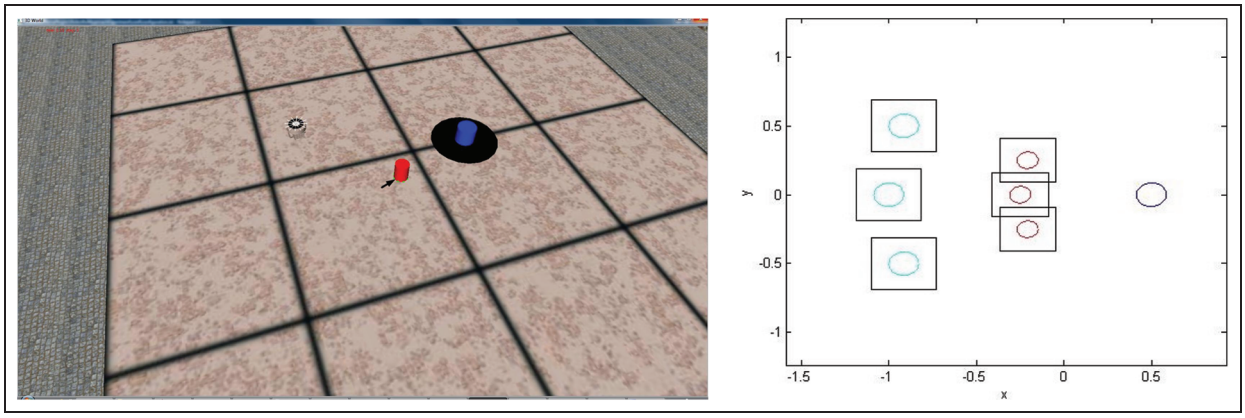

Figure 4. Spatial positioning with heading experiment. Left: The robot and the environment. The black arrow indicates the relative position and orientation that the robot should reach (i.e., the robot should reach the target with a relative position and orientation that could enable it to push the red object toward the blue object). Right: Exemplification of how the position of both the robot and the red cylinder varies among trials. The top circle indicates the blue cylinder. The intermediate and bottom rectangles indicate the areas in which the red cylinder and the robot are placed, respectively. The exact position of the red cylinder and of the robot is randomly chosen inside the selected rectangular area.

Table 1. Performance and percentage of times the best robots turn in their preferred direction in the three experimental conditions.

Best robot Best 10 robots	Performance Performance	Turns in the preferred direction (%) Turns in the preferred direction (%)
E	60.65	100.0
D	51.36	2.4
E&D	61.90	0.8
E	55.09	90.94
D	44.25	11.72
E&D	61.28	0.88

Some trained robots turn preferentially left, others preferentially right. Others turn both left and right, depending on the circumstances. These latter robots do not have a preferred direction, they turn left and right with similar probabilities. Top part: data for the best robot of all replications. Bottom part: average results of the 10 best robots of each replication.

orientation from which the red cylinder could be pushed toward the blue cylinder by simply moving forward; Figure 4).

More specifically, the experiment concerns a simulated marXbot robot (Bonani et al., 2010) placed in an arena containing a red and a blue cylindrical object. We used a different robotic platform in this second set of experiments since the marXbot is provided with an omni-directional colour camera that enables it to detect the relative position of the two cylinders.

The robot is equipped with 24 infrared sensors evenly spaced around the robot body, an omni-directional camera, and two motors controlling the desired speed of the two wheels. The camera image is pre-processed by calculating the fraction of red and blue pixels present in each of the 12 30°-sectors of a circular stripe portion of the image.

The robot's controller consists of a three-layer feed-forward neural network with 32 sensory neurons, six internal neurons and two motor neurons. In particular, eight sensory neurons encode the average activation state of eight groups of three adjacent infrared sensors and 24 sensory neurons encode the percentage of red and blue pixels detected in the 12 sectors of the perceived image. The two motor neurons encode the desired speed of the two wheels. The activation of the sensors is normalized in the range [0.0, 1.0]. The activation of internal and motor neurons is computed on the basis of the logistic function. The activation of the motor neurons, i.e., the desired speed of the two wheels, is normalized in the range [−27, 27] cm/s. Internal and motor neurons are provided with biases.

The environment consists of a flat arena of 5.0×5.0 m containing a red and a blue cylinder with a diameter of 12 and 17 cm respectively, randomly placed within 3×3 rectangular portion of the arena shown in Figure 4, right. In particular, the robot is initially placed within one of three possible rectangular portions

(Figure 4 right, circles in the bottom rectangular areas). Similarly, the red cylinder occupies one of three possible rectangular portions (Figure 4 right, circles in the central rectangular areas). The initial location of the blue cylinder has been kept fixed over all the simulations (Figure 4 right, top circle). As in the case of the previous experiment, the robot and the environment are simulated by using FARSA (Massera et al., 2013).

The performance of the robot is calculated by taking into account the offset between the target position and orientation of the robot and the actual position and orientation of the robot when it first touches the cylinder, or at the end of the trial. The target spatial parameters are defined as follows:

- the target position is located on the straight line passing over the centres of the bases of red and blue objects, outside the segment connecting these two points on the side of the red object, at a distance equal to the sum of the robot and red cylinder radius;
- the target orientation is given by the relative angle between the red and the blue cylinder.

More precisely, the fitness is calculated by averaging the product of the inverse of the position offset and the inverse of the angular offset, normalized in the range [0.0, 1.0], over trials:

$$Fitness = \frac{\sum_{t=1\ldots n_t} [(1-\Delta P_t)\cdot(1-\Delta O_t)]}{n_t}$$

where ΔP_t is the position offset normalized in the range [0.0, 1.0], ΔO_t is the angular offset normalized in the range [0.0, 1.0], t is the trial index, and n_t is the number of trials.

To select robots able to carry out the task in varying environmental conditions, we evaluated each robot for nine trials each lasting up to 50 s (trials end either when the robot touches the red cylinder, or after 50 s). In the case of the integrated learning by E&D algorithm, the back-propagation training has been also carried out for nine trials during each iteration. At the beginning of each trial the red cylinder and the robot are in random locations selected within one of the 3×3 rectangular areas shown in (Figure 4, right), that are respectively at a distance of approximately 0.75 and 1.5 m from the blue cylinder.

As for the previous experiment, a demonstrator robot that operates on the basis of a hand-designed controller is used to generate demonstration data. The controller works by calculating, at each time step, a destination point situated along the straight line (l) connecting red and blue cylinders. The distance between the destination point and the red cylinder is directly proportional to the offset between the current robot position and l:

$$distance = K \cdot d_l$$

where K is a constant set to 2.5 and d_l is the distance in metres between the robot and l line. This is because the time required for the robot to reach the target location depends on the initial angular offset between the robot orientation and the target orientation.

In particular, the coordinates of the destination point over planar surface are calculated according to the following equation:

$$destination = (distance \cdot \cos(\alpha), distance \cdot \sin(\alpha))$$

where α represents the relative angle between the red and the blue cylinder.

The action to be performed by the demonstration robot at each time step is selected in order to reduce both the distance between the robot and the destination point, and the offset between the current and the target orientation of the robot. In more detail, the desired speed of the robots' wheels is set on the basis of the following equations:

$$speedLeftWheel = \begin{cases} \frac{v_{max}}{2} if -180 \leqslant \theta \leqslant 0 \\ \frac{v_{min}}{2}\cdot\left(\frac{\theta}{180}\right) + \frac{v_{max}}{2}\cdot\left(1-\left(\frac{\theta}{180}\right)\right) if 0 < \theta \leqslant 180 \end{cases}$$

$$speedRightWheel = \begin{cases} \frac{v_{max}}{2}\cdot\left(\frac{\theta}{180}\right) - \frac{v_{min}}{2}\cdot\left(1+\left(\frac{\theta}{180}\right)\right) if -180 \leqslant \theta < 0 \\ \frac{v_{max}}{2} if 0 \leqslant \theta \leqslant 180 \end{cases}$$

where θ is the turning angle normalized in the range $[-180°,180°]$. θ is positive or negative depending on whether the destination point is located on the right or on the left of the robot's front.

This strategy enables the demonstrator robot to solve the task with a relative good performance. However, as for the previous experiment, the demonstration strategy is non-optimal. Indeed, in certain cases the robot might reach the destination point with a partially wrong orientation. Furthermore, in other cases it might spend an unnecessary amount of energy and time to reach the target position and orientation. Besides, we should consider that the learning robot might be unable to produce behaviours that match perfectly, or almost perfectly, the demonstrations as a consequence of limited precision of the robot sensory system. The control system of the demonstrator robot is not affected by this limitation since it operates on the basis of precise

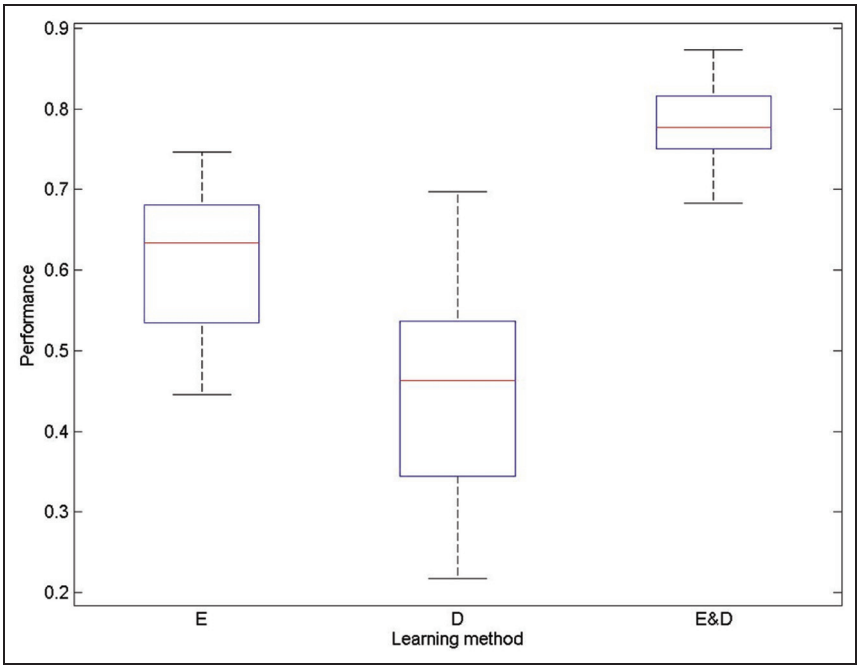

Figure 5. The boxplots display the performance carried out by the robots at the end of the training process in the learning by experience (E), learning by demonstration (D) and integrated learning by experience and demonstration (E&D). Boxes represent the inter-quartile range of the data, while the horizontal lines inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 times the inter-quartile range from the box. Data obtained by post-evaluating the best 10 robots of the 10 replications of each experimental condition for 180 trials. For the D condition, we only report the result obtained in the online mode that yielded better results with respect to the offline mode.

distance and angular measures extracted from the robot/environmental simulation.

The 212 connection weights and biases of the neural network are initially set randomly and then learned by means of the algorithms described above.

5. Results for the spatial positioning with heading experiment

Here we report the results obtained in three series of experiments in which the robots have been trained by using the algorithms described in sections 3.1–3.3. Each experiment was replicated 10 times.

By analysing the performance displayed at the end of the training process we can see how, as for the exploration experiment, the robots trained in the integrated learning by E&D condition outperform the robots trained in the E and D conditions (Mann–Whitney test, $df = 10$, $p < .01$). The difference between E and D conditions is also statistically significant (Mann–Whitney test, $df = 10$, $p < .01$). To verify robots' generalization capabilities, we post-evaluated the best robots for 180 trials during which the distance between the robot and the blue cylinder was systematically varied in the range [62.5, 300] cm (Figure 5). Also in this case the robots trained in the integrated learning by E&D condition outperform the robots trained in the E and D conditions (Mann–Whitney test, $df = 10$, $p < .01$). The robots trained in the E condition outperform the robot trained in the D condition (Mann–Whitney test, $df = 10$, $p < .05$). The best E&D robots outperform the hand-designed demonstrator robot as well (Mann–Whitney test, $df = 10$, $p < .01$). The fact that the difference in performance between the E&D condition and the D and E conditions is more marked in the case of this second and more complex experiment might indicate that the E&D algorithm scale better with respect to the complexity of the task.

To analyse the similarity between the behaviour displayed by trained robots and by the hand-designed demonstrator robot, we calculated the summed square error between the actions suggested by the demonstrator and the actions actually produced by the best robots trained in the three experimental conditions (Figure 6). As expected, the difference between the behaviour of the robots and the demonstration is larger in the E condition than in the D and E&D conditions. Surprisingly, the differences with respect to the demonstration behaviour are greater in the D condition, in which the robots are trained for the ability to reproduce the demonstrated behaviour only, than in the E&D condition, in which the robots are also trained for the ability to maximize performance. This could be explained by the fact that the local minima affecting the outcome of the

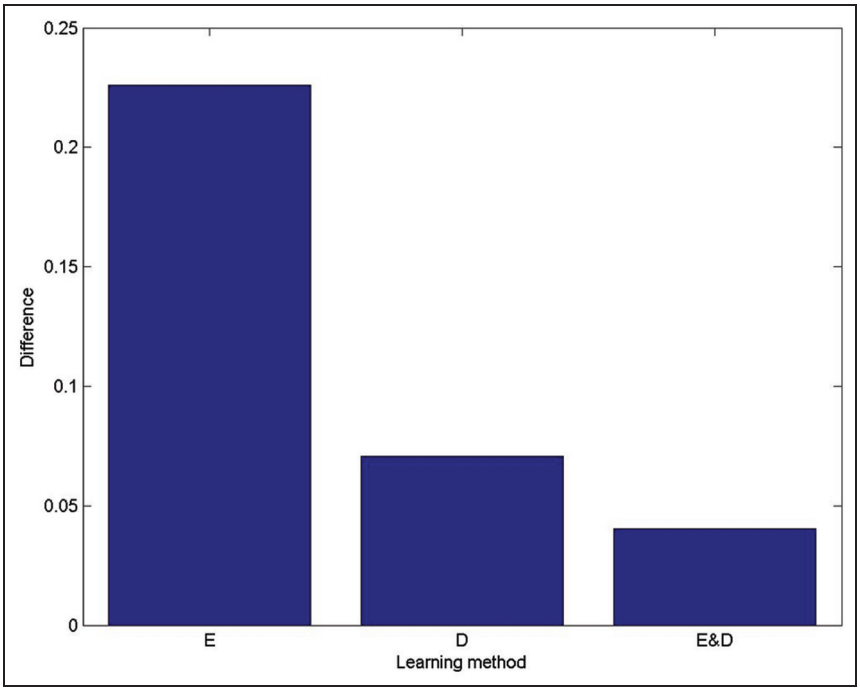

Figure 6. Average difference between the actions produced by the best robot trained in the three experimental conditions—experience (E), learning by demonstration (D) and integrated learning by experience and demonstration (E&D)—and the actions that would be produced by the demonstrator robot in the same robot/environmental conditions. Data are calculated by averaging the summed square difference between the actual and desired speeds of the two wheels, normalized in the range [0.0, 1.0], during 180 trials.

demonstration learning, that operates by minimizing the offset between the demonstrated and the reproduced actions, play a minor role when the learning process is driven primarily by the attempt to maximize the long term effect of actions (i.e., the ability to reach the target with the appropriate position and orientation).

Figure 7 shows the typical trajectories displayed by the three best robots obtained in the three corresponding experimental conditions.

Not surprisingly, the best D robot displays a behaviour that is qualitatively similar to the demonstration: the robot always approaches the red cylinder from the appropriate direction by producing a single curvilinear trajectory (Figure 7, left). Yet, the curvature of the trajectory of the D robot often differs significantly with respect to the curvature produced by the demonstrator robot, despite the learning error at the end of the training is very low (0.0042). This difference can be explained by considering that, as we mentioned above, small differences between the produced and the demonstrated behaviour tend to cumulate by producing significant differences over time. The other robots obtained in the remaining nine replications of the experiment display qualitatively similar behaviours (results not shown). The fact that the performance of the D robots are relatively low (Figure 5) can thus be explained by considering that the significant differences, that originate from the cumulative effect of small differences over time, have an effect on both the similarity with respect to the demonstrated behaviour and the performance. In some replications, the functional effect of the difference is small, while in other cases it is large (Figure 5). The analysis of the relation between learning error and performance demonstrates that these two measures are lowly correlated. In other words, the robots with the highest performance are not necessarily those with the lower residual learning by demonstration error. This can be explained by considering that, in learning by demonstration, performance does not play any role. The minimization of the error with respect to the demonstration, therefore, does not guarantee a maximization of performance.

The behaviour produced by the best E robot is better than the one produced by the best D robot and differs widely from the demonstration (Figures 6 and 7). Indeed, it produces a series of curvilinear trajectories followed by sudden changes of directions until the robot manages to reach a relative position and orientation from which it is able to move consistently toward the red cylinder. Since the number of back-and-forth movements is highly variable and dependent on the circumstances, in some cases the robot is unable to reach the target destination in time. The robots of the other replications of the experiment show qualitatively similar behaviours (results not shown).

The best E&D robots approach the target by producing single curvilinear trajectories that are qualitatively similar to that displayed by the demonstrator

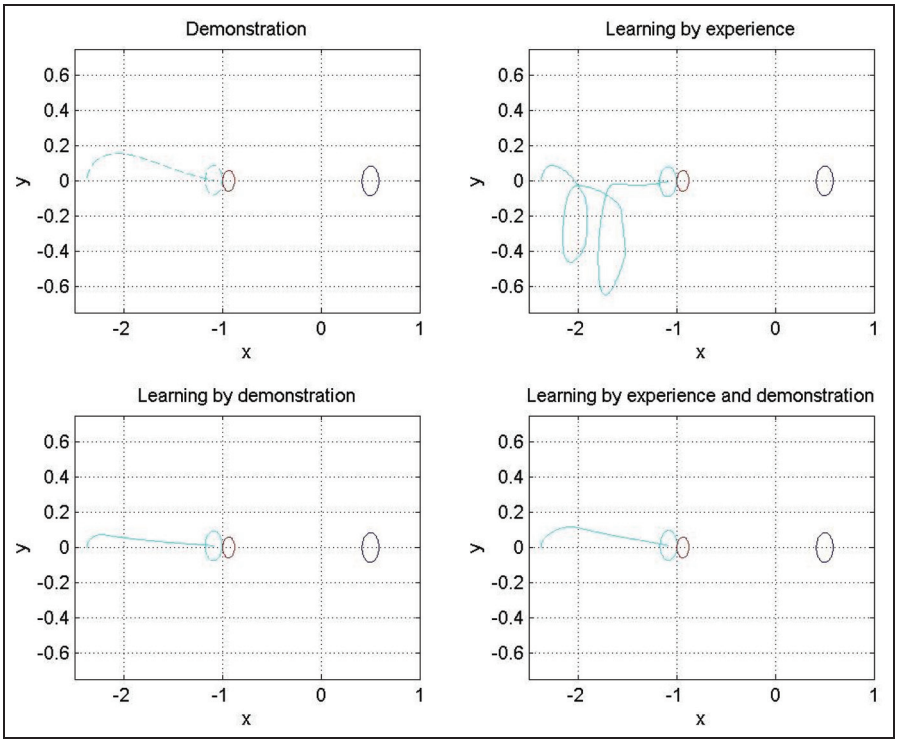

Figure 7. Trajectories displayed by the best robots obtained in the learning by experience (top-right), learning by demonstration (bottom-left) and learning by experience and demonstration condition (bottom-right) in a trial in which they start from the same initial position and orientation and in which the red cylinder is placed in the same position. The top-left picture shows the trajectory produced by the demonstrator robot. The red and blue circles represent the position of red and blue cylinder.

robot (Figures 6 and 7). Furthermore, their behaviours are even more similar to the demonstration than those shown by the D robots (Figure 6). Nonetheless, they manage to control the curvature of the trajectory in a way that enables them to achieve high performance and even outperform the demonstrator robot (the average performance of the demonstrator robot calculated over 500 trials is 0.79). Apparently, this is achieved by producing longer trajectories, with respect to the demonstrator, that allow the robot to reach the target destination with an orientation that matches better the target orientation. Overall, this implies that, also in this case, the E&D robots manage to find solutions that are qualitatively similar to the demonstrated behaviour, but deviate from the demonstration with respect to the aspects that are functionally sub-optimal.

6. Discussion and conclusion

Standard learning methods rely on a single type of learning feedback and simply discard all other relevant information. For example, reinforcement learning methods are based on reward signals that indicate how well the agent is performing, but neglect other relevant cues such as: (1) perceived states that provide indications on why the current exhibited behaviour had only partial success (e.g., the kicked ball went too far or too left with respect to the target), (2) advice provided by other agents (e.g., 'more slowly'), and (3) demonstrations of effective behaviours. Since the quality of the training feedback strongly affects the outcome of the learning process, the development of new learning methods capable of exploiting richer training feedbacks can potentially lead to significantly more powerful training techniques.

Although this research direction has started to be explored in the last few years, how different training feedbacks can be integrated in an effective manner still largely represents an open question.

As we stated in the introduction, in order to appreciate the complexity of the problem, we should consider that different learning modes (e.g., learning by experience and learning by demonstration) often drive the learning agents toward different outcomes and that their effect is not necessarily additive. This entails that identifying an effective way to combine heterogeneous learning modes is far from trivial.

In this paper we proposed a new method that enables to combine learning by demonstration and learning by experience feedbacks. The algorithm is shaped in a way ensuring that the learning process is constantly driven by both an objective performance measure, which estimates the overall ability of the robot to perform the task, and a learning by demonstration feedback used to

steer the learning process toward solutions similar to the demonstration.

The analysis of the results obtained by testing this method on two qualitatively different problems indicates that our new integrated algorithm is indeed able to synthesize solutions that are (1) functionally better than those obtainable by using a single mode only, and (2) qualitatively similar to the demonstrated behaviour. Besides, the integrated learning algorithm succeeds in synthesizing solutions that incorporate the aspects of the demonstration that are functionally useful, whereas deviate with respect to other non-functional aspects.

In future works we plan to investigate the efficacy of the proposed method in task involving significant larger search space (i.e., a greater number of parameters to be set) that are particularly hard to tackle through learning by experience only.

References

Abbeel, P., & Ng, A. Y. (2005). Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd International Conference on Machine Learning*. New York: ACP Press.

Argall, B. D., Browning, B., & Veloso, M. (2010). Mobile robot motion control from demonstration and corrective feedback. In: O. Sigaud & J. Peters (Eds.), *From motor learning to interaction learning in robots, SCI* (vol. *264*, pp. 431–450). Heidelberg: Springer.

Argall, B. D., Chernova, S., Veloso. M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*(5), 469–483.

Argall, B. D., Browning, B., & Veloso, M. (2008). Learning robot motion control with demonstration and advice-operators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)* (pp. 399–404). IEEE Press.

Billard, A., Callinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. In B. Siciliano & O. Khatib (Eds.), *Handbook of robotics*. New York: Springer.

Billard, A., Calinon, S., & Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. In The Social Mechanisms of Robot Programming by Demonstration, *Robotics and Autonomous Systems*, *54*(5), 370–384 (special issue).

Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G. & Mondada, F. (2010). The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4187–4193). Piscataway, NJ: IEEE Press.

Cetina, V. U. (2007). Supervised reinforcement learning using behavior models. In *Sixth International Conference on Machine Learning and Applications*. New York: IEEE Press.

Chernova, S., & Veloso, M. (2007). Confidence-based learning from demonstration using Gaussian Mixture Models. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)* (pp. 1–8). ACM New York.

Coates, A., Abbeel, P., & Ng, A. Y. (2008). Learning for control from multiple demonstrations. *Proceedings of the 25th International Conference on Machine Learning (ICML '08)* (pp. 144–151). ACM New York.

Glasmachers, T., Schaul, T., Yi, S., Wierstra, D., & Schmidhuber, J. (2010). Exponential natural evolution strategies. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO'10)*, Portland, OR, pp. 393–400. New York: ACM.

Daniel, C., Neumann, G., & Peters, J. (2012). Hierarchical relative entropy policy search. In N. Lawrence and M. Girolami (Eds), *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS 2012)* (pp. 273–281).

De Greef J., & Nolfi, S. (2010). Evolution of implicit and explicit communication in a group of mobile robots. In S. Nolfi, & M. Mirolli (Eds.), *Evolution of communication and language in embodied agents*. Berlin: Springer.

Hayes, G., & Demiris, J. (1994). A robot controller using learning by imitation. In A. Borkowski and J.L Crowley (Eds.), *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems* (pp. 198–204). IEEE Press.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Ijspeert, A., Nakanishi, J., & Schaal, S. (2004). Learning attractor landscapes for learning motor primitives. In S. Thrun, L. K. Saul, & B. Scholkopf (Eds.), *Advances in neural information processing systems* (vol. *16*). Cambridge, MA: MIT Press.

Judah, K., Roy, S., Fern, A., & Dietterich, T.G. (2010). Reinforcement learning via practice and critique advice. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Knox, W. B., & Stone, P. (2009). Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*. New York: ACM Press.

Knox, W. B., & Stone, P. (2010). Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (vol. 1)*. Richland SC: International Foundation for Autonomous Agents and Multiagent Systems.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research 32*(11), 1238–127.

Kober, J., & Peters, J. (2009). Policy search for motor primitives in robotics. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems 21*. Carmel, IN: Current Associates Inc.

Massera, G., Ferrauto, T., Gigliotta, O., & Nolfi, S. (2013). FARSA: An open software tool for embodied cognitive science. In P. Liò, O. Miglino, G. Nicosia, S. Nolfi, & M. Pavone (Eds.), *Proceeding of the 12th European Conference on Artificial Life*. Cambridge, MA: MIT Press.

Mondada, F., Franzi, E., & Ienne, P. (1993). Mobile robot miniaturisation: A tool for investigation in control algorithms. In: *Proceedings of the third International Symposium on Experimental Robotics*, Kyoto, Japan

Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. Cambridge, MA: MIT Press/Bradford Books.

Paraschos, A., Daniel, C., Peters, J., & Neumann, G. (2013). Probabilistic movement primitives. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (NIPS). Cambridge, MA: MIT Press

Rechenberg, I., & Eigen, M. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog.

Rosenstein, M. T., & Barto, A. G. (2004). Supervised actor-critic reinforcement learning. In J. Si, A. Barto, W. Powell, & D. Wunsch (Eds.), *Learning and approximate dynamic programming: Scaling up to the real world*. New York: John Wiley & Sons, Inc.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & I. L. McCelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (vol. 1)*. Cambridge, MA: MIT Press.

Russell, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach*. Englewood Cliffs, NJ: Prentice Hall.

Schaal, S. (1997). Learning from demonstration. In: M.C. Mozer, M. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems 9*. Cambridge, MA: MIT Press.

Schaal, S., Mohajerian, P., & Ijspeert, A. (2007). Dynamics systems vs. optimal control—A unifying view. *Progress in Brain Research*, *165*(1), 425–445.

Schwefel, H. P. (1977). *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie (vol. 26)*. Basel/Stuttgart: Birkhaeuser.

Schwefel, H. P. (1995), *Evolution and optimum seeking*. New York: Wiley.

Sperati, V., Trianni, V., & Nolfi, S. (2008). Evolving coordinated group behaviour through maximization of mean mutual information. Special Issue on Swarm Robotics, *Swarm Intelligence Journal*, *4*(2), 73–95.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Boston, MA: MIT Press.

Taylor, M., Suay, H., & Chernova, S. (2011). *Integrating reinforcement learning with human demonstrations of varying ability*, AAMAS, Taipei, Taiwan.

Valenti, M. (2013). *Learning of manipulation capabilities in a humanoid robot*. Master thesis, School of Engineering, Università degli Studi di Roma 'La Sapienza', Italy.

Wierstra, D., Schaul, T., Peters, J., & Schmidhuber, J. (2008). Natural evolution strategies. In *Proceedings of the Congress on Evolutionary Computation (CEC08)*. Hong Kong: IEEE Press.

About the Authors

Paolo Pagliuca received an MSc in engineering from the University of Roma Tre, Italy and is a PhD student at the CNR node of the University of Plymouth in Roma. His main research interests are within the domain of evolutionary robotics, adaptive behaviour and combination of learning and evolution in autonomous robots.

Stefano Nolfi (http://laral.istc.cnr.it/nolfi/) is a research director of the Institute of Cognitive Sciences and Technologies of the Italian National Research Council and head of the Laboratory of Autonomous Robots and Artificial Life (http://laral.istc.cnr.it/). Stefano conducted pioneering research in artificial life and is one of the founders of Evolutionary Robotics. His main research interest is in study of how embodied and situated agents can develop behavioural and cognitive skills autonomously by adapting to their task/environment. Stefano authored and co-authored more than 150 peer-review scientific publications.