4th International Conference on
Development and Learning and on Epigenetic Robotics
October 13-16, 2014. Palazzo Ducale, Genoa, Italy

Reinforcement and Shaping in Learning Action Sequences with Neural Dynamics

Matthew Luciw*, Yulia Sandamirskaya†, Sohrob Kazerounian*, Jürgen Schmidhuber*, Gregor Schöner†
* Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Manno-Lugano, Switzerland
† Institut für Neuroinformatik, Ruhr-Universität Bochum, Universitätstr, Bochum, Germany

***Abstract*—Neural dynamics offer a theoretical and computational framework, in which cognitive architectures may be developed, which are suitable both to model psychophysics of human behaviour and to control robotic behaviour. Recently, we have introduced reinforcement learning in this framework, which allows an agent to learn goal-directed sequences of behaviours based on a reward signal, perceived at the end of a sequence. Although stability of the dynamic neural fields and behavioural organisation allowed to demonstrate autonomous learning in the robotic system, learning of longer sequences was taking prohibitedly long time. Here, we combine the neural dynamic reinforcement learning with shaping, which consists in providing intermediate rewards and accelerates learning. We have implemented the new learning algorithm on a simulated Kuka YouBot robot and evaluated robustness and efficacy of learning in a pick-and-place task.**

I. Introduction

The current strive for intelligent robotic systems, which are able to operate autonomously in unconstrained real-world environments calls for adaptive and self-organising cognitive systems, which may autonomously initiate and terminate actions and, more over, autonomously select which action to activate next. A sequence of such action-selection decisions should bring the robotic agent to its behavioural goal. Reinforcement learning [1] is a learning paradigm, in which a computational agent learns to select correct actions in a goal-directed sequence from a numerical reward, typically received at the end of a successful sequence. Application of RL in robotics is challenging, however, because of large and continuous state-action spaces, which correspond to real-world perception and actions of a robotic agent. Moreover, the robot has to not only select the best next action, but also initiate this action and ensure that it is brought to the end. Each action may need coordination between different sensory inputs and motors of the robot and require stabilisation in the presence of noise and distractors. Thus, actions in physical robots become more than simple transitions between states, as typically assumed in RL algorithms.

In order to apply RL to cognitive robotics, we have recently used Dynamic Field Theory (DFT) [2] – a conceptual and computational framework, which allows to stabilise sensory representations and link them to motor control, while performing elementary cognitive operations, such as memory formation, decision making, and selection among alternatives. Following an inspiration from behavioural robotics [3], we have introduced a concept of elementary behaviours (EBs) in DFT, which correspond to different sensorimotor behaviours that the agent may perform [4], [5]. To the contrary to purely reactive modules of the subsumption and other behavioural-robotics architectures, the EBs in DFT are endowed with representations. In particular, a representation of the *intention* of the behaviour provides a control input to the sensorimotor system of the agent, activating the required coordination pattern between the perceptual and motor systems, and eventually triggers generation of the behaviour. A representation of a *condition of satisfaction* (CoS), in its turn, is pre-shaped by the intention to be sensitive to an expected outcome of the behaviour and is activated when the behaviour has reached its objectives. The CoS inhibits the EB's intention, which now gives way to the next behaviour in the behavioural sequence. We have previously discussed the need for this stabilised representations within EB and how they contribute to autonomy and robustness of behaviour [5], [6].

Further, we have demonstrated how goal-directed sequences of EBs can be learned from reward, received by the agent at the end of a successful sequence [7]. The latter architecture has integrated the DFT-based system for behavioural organization with the Reinforcement Learning algorithm (RL; [8], [9]), SARSA(λ). This system could autonomously learn sequences of EBs through random exploration, and the model operated in real-time, continuous environments, using the categorising properties of EBs.

Obviously, when applying RL approach in a robotic domain with a high number of available behaviors, waiting for such an agent to randomly explore action sequences becomes untenable. Thus, in this paper, we introduce a study of how the concept of shaping [10], [11], from the field of animal learning can be used in order to speed up training for the cognitive agent operating in our learning framework. Shaping has been previously applied to robot learning [12], [13], [14], [15], [16] and consists in providing feedback to the agent about success of behaviour execution before the final goal of the to-be-learned sequence is reached. In the architecture, presented here, the DFT-based framework for behavioural organisation trough EBs provides a robust interface to noisy and continuous environments, RL provides for autonomous learning through exploration, and shaping accelerates learning. Here, we have used the recently developed T-learning RL algorithm which fits the neural-

dynamic framework for behavioural organisation better than the SARSA.

II. Background

A. Dynamic Field Theory

Dynamic Field Theory (DFT; [2]) is a mathematical and conceptual framework for modelling embodied cognition, which is built on an approximation of dynamics of neuronal populations, first analysed by Amari [17]. While many neural network architectures use individual neurons as computational units, DFT uses description of population activity, expressed in continuous activation functions. The activation function of a DF obeys the following differential equation (analyzed by Amari [17]):

$$\tau \dot{u}(x,t) = -u(x,t) + h + S(x,t) + \int \omega(x - x')\sigma(u(x',t))dx', \quad (1)$$

where the dimension x represents the space of a behaviourally relevant variable, such as color, visual space, or motor commands, which characterises the sensorimotor system and the tasks of the agent; t is time, $h < 0$ is a negative resting level, and $S(x,t)$ is the sum of external inputs, integrated by the DF along with the lateral interaction, defined by the sum-of-Gaussians interaction kernel $\omega(x - x')$, which has a the local excitatory and a long-range inhibitory part. Output of the DF is shaped by a sigmoidal non-linearity, σ, i.e. the regions of a DF with activity below activation threshold are silent for the rest of a DF architecture, only regions with suprathreshold activation have impact on other DFs, the motor system, and the DF itself.

The non-linearity and lateral connectivity in the DF's dynamics lead to stable localized *peaks* of activation to be attractor solutions of the dynamics. These peaks build-up from distributed, noisy, and transient input in an instability, which separates a quiescent, subthreshold state of the DF from the activated state. The activation peaks represent perceptual objects or motor goals in the DFT framework. The peaks may be sustained through lateral interactions even if the initial input, which induced them ceases.

This ability to form and stabilize robust categorical outputs makes DFT architectures particularly well suited for robotic control systems, in which motor control relies on perceptual (visual) information. Multiple coupled DFs spanning different perceptual and motor modalities can be composed into complex DFT architectures to organize robot behavior. The building blocks of these architectures are Elementary Behaviors (EBs), each consisting of stabilised representations of intention and condition of satisfaction.

B. Elementary Behaviors and Behavior Chaining

An Elementary Behavior (EB) is an organizational structure in DFT which not only defines the actions associated with a behavior, but also the mechanisms for initiating and terminating that behavior.

In the original work on behavioural organisation with attractor dynamics [18], attractor dynamics define values of behavioral variables (e.g., heading direction), which characterise the state of the robot and control the robot's effectors. The dynamics may integrate both attractive and repulsive 'forcelets', which determine behaviour of the system, controlled by the given dynamics.

The limitation of the original attractor dynamics approach was inability of EBs to switch themselves off and activate another attractor-based controller. Thus, only appearance of a new target would reset the attractor for the heading direction dynamics, no explicit mechanisms is defined to trigger selection or search for a different target when the previous one was reached. This setting worked well in navigation scenarios, due to an ingenious engineering of the attractive and repelling contributions to the dynamics (e.g. leading the robot to turn away from the reached target). However, in more complex scenarios, which include object manipulation, the system cannot rely on the immediately and continually available sensory information, but needs to leverage the sensory information into usable representations, upon which autonomous decisions may be made and the flow of actions may be autonomously controlled by the agent itself.

Thus, the concept of elementary behaviour (EB) was introduced in DFT [4]. The key elements in a DFT EB are the intention and the condition of satisfaction. Both the intention to act and the condition of satisfaction are represented by dynamic fields over some behavioural (perceptual or motor) dimension, as well as by the associated with these dynamic fields CoS and intention nodes (zero-dimensional DFs), which are the elements, on which learning may operate (see further). Because the amount of time needed to complete an action may vary unpredictably in dynamic and partially unknown environments, the intention to achieve the behavior's goal is maintained as a stable state until completion of the goal is signaled. The condition of satisfaction is activated by the sensory conditions that index that an intended action has been completed. The CoS serves two roles: on the one hand, it terminates the associated intention, and on the other hand, as a stable state, it serves to select the behavior to do next.

In previous work, we have shown how EBs may be chained according to rules of behavioral organization [4], [5] or serial order [19]. In order to introduce learning into the scheme, adaptive weights can be placed between CoS node of one EB and the intention nodes of other EBs, representing transitions between a just completed behavior (CoS), and possible next behaviors (intentions). These weights serve as values in the RL sense and may be learned, as was introduced recently [7].

C. Reinforcement Learning.

We use a particular RL method, called T-learning [31], in which value is associated with transitions between states, instead of states or state-action pairs. Assigning value to transitions is appropriate in our framework because an elementary behavior itself is an attempted transition between

two stable states (attractors), and the agent's decision constitutes choosing the next elementary behavior when it is in one of these stable states. In other words, the agent needs to pick the next most likely optimal transition. T-learning is the simplest RL method for this type of setup. With T-learning and EBs, there is no need to define a separate action set.

Further, T-learning is a more efficient learning method than standard SARSA or Q-learning when some actions are failure-prone, as is the case in many robotic learning environments. In our setup, the rewarding transitions are reinforced when accomplished successfully, but when they fail, a different transition is credited — one which goes to a 'failure' state, whereas if state-action values were used, a failure would de-value the action itself (or the state-action pair). T-learning suits the learning of 'skilled' behavior, where learning to make difficult transitions is highly rewarding. By difficult, we mean that only a small percentage of possible actions reliably transition to a state associated with high reward. The T-learning agent will continue to try to make that transition even after a failure.

The T-learning update rule is

$$T(s,s') \leftarrow T(s,s') + \alpha\left[r + \gamma T(s',s'') - T(s,s')\right], \quad (2)$$

where s, s', and s'' are three successive stable states (either CoS of an EB, or the failure state), r is a reward, α is the learning rate, and γ is the discount factor. In our implementation, T-learning is combined with eligibility traces [32], [33], in the same way as when SARSA becomes SARSA(λ) [8], to deal with non-Markovian nature of the task (a whole sequence is reinforced by a single reward, received at the end of the sequence).

D. Shaping

Shaping, first introduced by B.F. Skinner [10], [11], is well-known in both the psychological and reinforcement learning literature as a method of conditioning. Shaping involves teaching a desired behavior by *successive approximations*, where the teacher or trainer invents and rewards subgoals, which bring the agent's behavior closer to that of the desired behavior. Critically, one of the defining characteristics in shaping, is successive and shifting *positive* rewards, rather than the use of negative punishments.

In RL, an undirected exploration methods, such as ϵ-greedy or purely random search [22], [8], which rely on random actions, result in (a) redundancy in the search due to lack of a memory structure and (b) search bias centralized on the starting position, making it more difficult to discover far away rewards. This problems are particularly relevant in RL implemented on autonomous robots, which are slow, prone to breakage, and cannot sustain long exploration periods.

Informed, directed exploration methods (such as optimistic initialization or artificial curiosity [23]) may accelerate learning. But the learning speed with directed exploration is still considerably lower than when using guided learning, wherein knowledge about how to achieve rewards is transferred to the agent during exploration. Guided learning manifests in RL under various guises and names, including chaining [24] or chunking of actions into macro-actions [25], manipulating the reward function to guide the agent [12], [13], and knowledge transfer over tasks [26], [27].

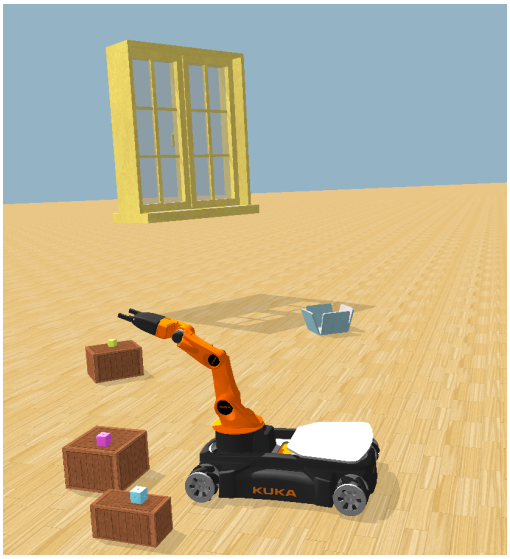

Fig. 1. The simulated YouBot and the environment, in which learning takes place. The task of the robot is to bring colored objects to the grey box.

Dorigo and Colombetti [16] introduced the term *robot shaping*, wherein a trainer, providing guidance and support, was found to be greatly effective in speeding up the robot's learning. Various methods have been introduced to allow the teacher to reinforce the robot in a timely and useful manner, such as a reinforcement sensor [28], "good" and "bad" buttons [29], as well as related methods such as Learning from Easy Missions (LEM; [14]).

III. Methods and Experimental Setup

Robot and Environment. We used the Kuka Youbot in our experiments, implemented in the Webots simulator [30]. The Youbot combines an omnidirectional mobile base with a one degree of freedom (DOF) rotating base platform and a standard three DOF arm with a two pronged gripper with force feedback. The Youbot provides flexibility to move around untethered on a flat surface, and to reach for and grasp small objects. The Youbot was enhanced with a RGB and kinect sensor on the front, to detect and localize targets for reaching, and infrared range (IR) sensors around the robot, to detect obstacles. The Youbot is placed in an environment with a few differently colored blocks upon boxes, some obstacles, and a deposit location, the container. A reward is given when the robot transports an object of a specific color into the container at the deposit location (see Fig. 1). Our implementation used seven different elementary behaviors (EBs), which when chained appropriately lead to a find-pick-and-place action. We introduce these elementary behaviours next.

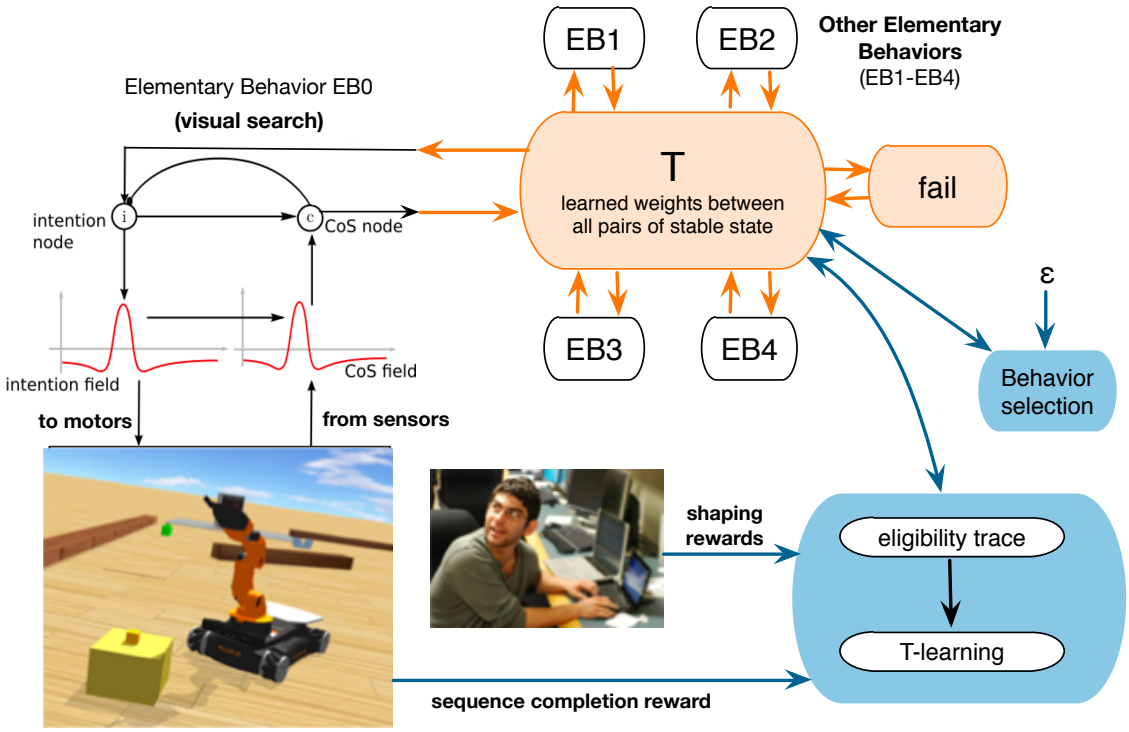

Fig. 2. System architecture. See text for details.

- **Visual search** is an EB, which controls the orientation of the robot's base (its heading direction) and which is 'satisfied' when the target object is central in the robot's vision, as detected in a perceptual DF. The perceptual DF's input is a color histogram from every input image column of the robotic camera [19], i.e. this is a 2D field over dimensions of color and angular space. The target color is biased in the perceptual field by a Gaussian-ridge shaped input, such that only the target color's appearance in the camera image can produce an activity peak in the perceptual DF. The output of the perceptual DF along the spatial dimension feeds into the CoS and motor intention fields. The motor intention field controls the heading direction of the robot, based on location of the target object in space, as represented in the perceptual DF. The CoS has a preshape over central image columns, and consequently the CoS only creates a peak if the target is centred in the camera image. If the target is not visible, a default pseudo-random movement behavior takes over.
- **Approach target** EB moves the robot towards the target, which must have been found with the visual search EB beforehand. This is expressed as a pre-condition of the approach EB [4]: if the object has not been found first (the CoS of visual search has not been activated), the approach EB transitions to the failure state (see further). Behavioral variables of the approach EB are heading direction and speed. The infrared sensors provide repulsive forces, which signal obstacles in the robot's vicinity. The CoS of this EB is activated when the distance between the target and robot is below a given threshold.
- **Orient arm to target** rotates the arm platform until the angle between the base-gripper vector and the base-target vector becomes nearly zero. This provides an optimal angle of approach for grasping.
- The **reaching** EB uses the Jacobian (which relates joint angle changes to the velocity of the end effector) of the three DOF RRR arm to continually move the central point between the gripper prongs to a point just above the target.
- The **close gripper** EB closes the gripper prongs until the force feedback, resulting from the gripper pressing on the object, surpasses a threshold.
- The **open gripper** EB moves the gripper prongs in the opposite direction until the joint limits are reached.
- **Approach deposit location** has the same dynamics as **approach target**, but uses the deposit location as the target.

Failure State. The elementary behaviors above can fail, either due to the lack of a precondition (e.g., the approach behaviour), or something going wrong during execution (e.g., the object not being grasped properly and falling down). We added a failure state for such cases. To detect failure, *conditions of dissatisfaction* (CoD) nodes are built into each EB. All behaviors use timer-based CoD's, set to 200 time steps. After failure, the robot pose is reset, as is the environment's state.

System Architecture. The system architecture is depicted in Fig. 2. The figure shows a set of elementary behaviors,

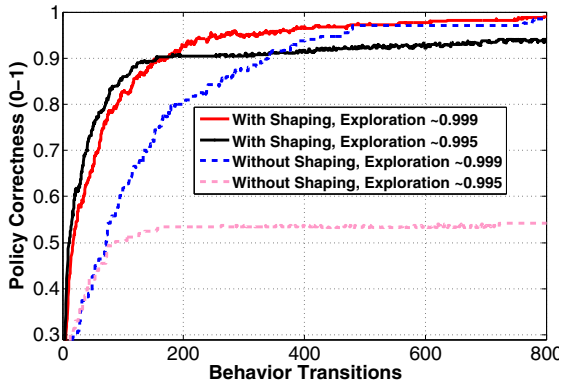

Fig. 3. Shaping improves the speed of policy learning, with an appropriate exploration setting.

some field-based (with intention and CoS dynamic fields), and some not (for which intention and CoS node suffice). The EBs take turns controlling the robot. Each behavior entails a stabilized representation of the condition of satisfaction, which corresponds to the state of the RL agent. There is another stable state associated with failure of any behavior. A set of weights T, between each pair of EBs defines the possible transitions between EBs. When a behavior completes, a behavior selector chooses the next behavior. This is performed either through a random choice (based on the parameter ϵ, as in ϵ-greedy action selection), or by taking the behavior with the largest weight e.g., the largest $T(s, s')$, given s. The weights are adapted through T-learning with an eligibility trace, using parameters α (learning rate), γ (discount factor), and λ (eligibility trace decay). The rewards come from the environment when a particular sequence is completed. Additional shaping rewards may arrive after the agent correctly completes each step of the sequence.

IV. Experimental Results

A. Effect of Shaping

In our previous work, the agent needed to discover the rewarding sequence through random search before any update of the value node weights due to reinforcement could be done. Learning in this case, while theoretically guaranteed, becomes too slow for real world agents. With shaping, the teacher provides positive reinforcement for successfully completed intermediate steps and thus modifies the reward function. Consequently, shaping should decrease the search time to discover the rewarding sequence and accelerate learning the policy that allows the robot to accomplish the sequence reliably.

To test this, we compared two RL setups, in which 100 learning trials were run in each. In both, there were four behaviors, and the rewarding sequence was 5 items long. In one case (with shaping), the agent received rewards after each correct step in the sequence (besides the first one completed). In the other (without shaping), the agent only received a reward after the entire sequence was completed. In each case, ϵ-greedy action selection was used, and we tested two exploration settings of $\varepsilon = \{0.999, 0.996\}$, where the random action chance starts from $\epsilon = 1$ (100%) and is multiplied by ε after each behavioral transition. In all cases, the learning rate was $\alpha = 0.1$, the eligibility trace parameter $\lambda = 0.6$, and the discount factor $\gamma = 0.9$.

In order to measure policy correctness, we measured the deviation of the optimal policy from the policy at every time step. To do this, the maximum valued action of each state was compared with the correct action. If they matched, the agent scored 0.25 points. An optimal policy, where the agent does the right thing in each state, scores 1, and the worst policy scores 0. Mean policy correctness for the four methods at each decision point are shown in Fig. 3. One can see that on average, the two variants with shaping learn much faster. The variants with a faster decreasing exploration factor are more prone to catastrophic failure runs, where the agent never learns the rewarding sequence. If one can tolerate a chance of failure, the variant with shaping and quicker exploration decrease can learn a bit faster than the other shaping variant. The variant without shaping and fast exploration decrease fails more than on the half of the trials. We have also tested performance of the models at different learning rates. In all cases, the shaping variants find the correct sequence faster.

B. Further RL Experiments

For further analysis of the performance of the neural-dynamic RL algorithm with shaping, we simulated the Youbot-Webots scenario numerically, in order to study learning in isolation. In these experiments, each behavior's success is endowed with a probability, instead of relying on the environment physics. Using this setting, we studied the effect of the choice of the learning algorithm (SARSA or T-learning), the effect of shaping, the effect of the eligibility trace, and the effects of all parameters.

First, we note that the scenario is designed to be the same as in the Webots simulation. The robot has to perform a sequence of seven elementary behaviors in the correct order to get the big reward. It has to transition from "Visual Search" to "GoTo Target" to "Orient Arm to Target" to "Reach" to "Grab" to "GoTo Deposit Location" to "Open Gripper". The states for the RL algorithms correspond to the stable states provided by the conditions of satisfaction of the seven EBs, with the addition of a failure state.

Each sequence of two behaviors was endowed with a probability of failure. For example, if the robot tries to go from the visual search CoS to the GoTo object CoS, there is a 5% chance of failure. If any behavior fails, the robot transitions to the failure state. These probabilities were mostly either very low (e.g., 10%) or high (100% for all impossible transitions).

We have established earlier that this problem is a POMDP, which can be learned using standard RL with shaping and an eligibility trace. Using both of these, we came to a set of parameters, with which SARSA and T-learning could solve the problem within about 12,000 behavior transitions: the discount factor $\gamma = 0.5$, the eligibility trace decay rate $\lambda = 0.95$, the learning rate $\alpha = 0.05$, the initial random action chance $\epsilon = 0.999$, and the random action chance decay rate $\varepsilon = 0.9999$.

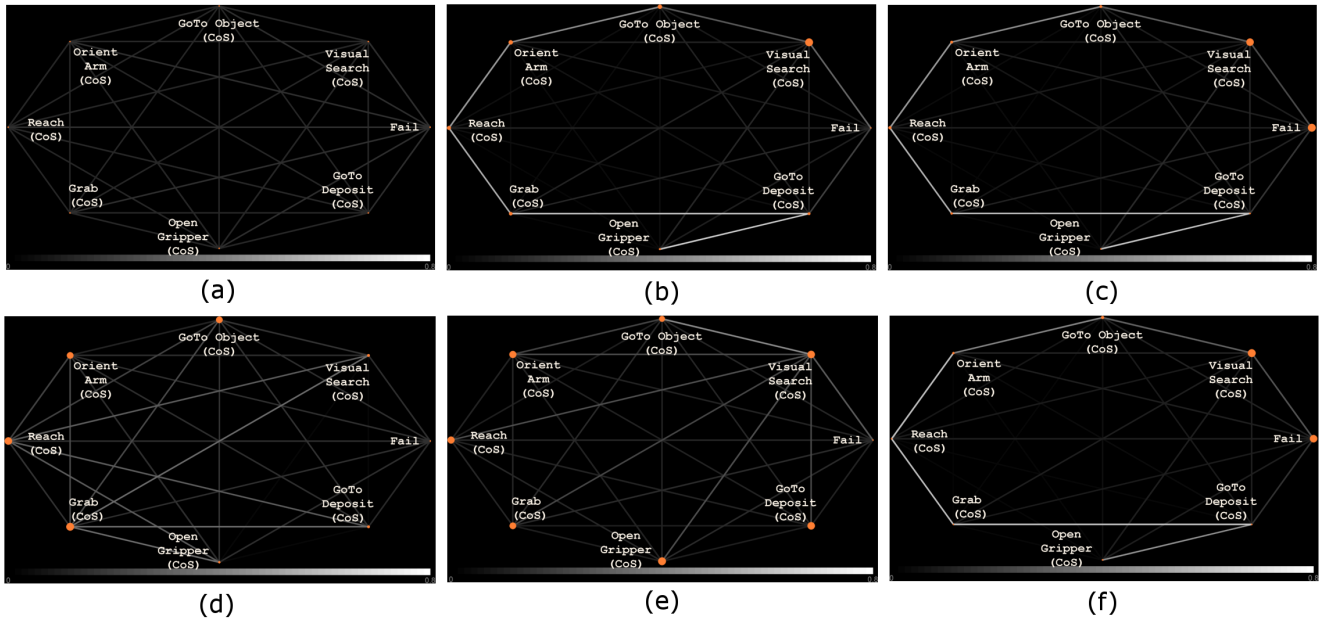

Fig. 4. Value functions in various conditions of learning viewed as graphs. All graphs were constructed in the following way. The weight for the edge shown between each two EBs is larger of the two directional transitions in the value function. In this way, a symmetric weighted adjacency matrix is constructed, which is row-normalized. (a). Initial value function using zeros. (b). State-action values of SARSA(λ) using shaping and memory (eligibility trace) after 20,000 transitions. (c). Transition values of T-learning using shaping and memory after 20,000 transitions. (d). State-action values when shaping was not used. (e). State-action values (outcome of SARSA) after the experiment where several EBs had a high chance of failure. (f). The result of T-learning using the same conditions as (e).

The shaping rewards were set to 1, after each successive correct step in the rewarding sequence, while the big reward at the end of the sequence was set to 100. When shaping was disabled, either with or without the eligibility trace, the agent could not find the rewarding sequence after 150,000 transitions. When shaping was enabled but the eligibility trace was disabled, the agent could find the rewarding sequence (within about 20,000 transitions), but the resulting state-action or state-state values were less robust than in the case where both shaping and the eligibility trace were used. Less robust means here, that the values of correct transitions differed less from the other values, and thus the sequence more easily got corrupted during exploration, although the agent usually converged to the correct sequence.

With T-learning, the learning algorithm does not need an action set. This is advantageous in our framework, since we have a direct mapping of RL states to the stable states (CoS/CoD nodes) of the neural-dynamic architecture. In DFT, there is no direct representation of classical RL actions – instantaneous transitions between states. Since in SARSA, the state-action transition values are learned, when implementing SARSA for comparison with the approach, presented here (T-learning with shaping), we use an attempted completion of a behavior as action in RL terms. In T-learning, we only need to represent that the agent tries to accomplish some transition, which corresponds to the neural-dynamic EB. Critically, if the agent fails, another transition gets the credit (to the failure state). In SARSA, the transition that was tried gets the credit, no matter where the agent actually ends up, which prevents the agent from trying a failed transition again.

Even though the main reason we used T-learning is to better match the conceptual framework of behavioural organisation in DFT, we can show a learning advantage as well. We used a "high-failure" environment, where the transitions from "Orient Arm to Target" to "Reach", and from "Reach" to "Grab" were both set to 70% failure rate. We also extended exploration by setting $\varepsilon = 0.99999$. In this case, T-learning, with both shaping and the eligibility trace, was able to find the correct sequence after 150,000 transitions. However, SARSA under the same conditions and parameters was not able to find the sequence.

Thus, T-learning enables the learning in high-failure environments. In the Webots simulation, neither the reach nor the grab fail anywhere close to 70%, the failure rate is close to 10% for this EBs in simulation. However, we expect this to change in an implementation with a real physical robot and consequently, the T-learning with shaping might be the key to application of RL in robotics.

With both shaping and memory, the 12,000 transitions are needed to find the seven item sequence in our simulation, most of the simulation being exploration, in which most behaviours are terminated due to passing the time limit. Without shaping, finding the sequence would take roughly 70 times longer.

Figure 4 shows the value function for various learning conditions. Note, that only the variants with shaping succeed in learning the correct sequence of seven EBs after 20.000

transitions, including when the chance of failure is high (subplot f). A plot showing the time-courses of EBs during exploration and exploitation is shown in Fig. 5. Here, the different EBs take different amounts of time, making use of the stabilising properties of the neural-dynamic behaviours, in which the DF intentions sustain different durations, needed to finish an action in a physical environment. Overall, our results support the hypothesis that both shaping and memory (eligilbility trace), as well as the segregation of the state-action space into attractor-based neural-dynamic EBs, are very beneficial for learning sequences in partially observable environment.

C. Simulated Robot Experiments

We successfully applied the shaping-based RL system to the Youbot, to produce the behavior of depositing a yellow target object in the periwinkle container. A video of the Youbot performing the rewarding sequence is at http://www.idsia.ch/~luciw/videos/youbotreward.avi. A few behavior transition failures can be seen at http://www.idsia.ch/~luciw/videos/youbotfail1.avi (the robot tries to reach an object out of range), and http://www.idsia.ch/~luciw/videos/youbotfail2.avi (the robot tries to go to a target it has not located visually, yet).

D. Notes on a Comparison to the System of Duran, Lee, and Lowe

We should note the similarities and differences of our system to one of Duran, Lee, and Lowe, 2012 [34]. They also use reinforcement learning to adapt a set of weights, used to chain elementary behaviors together. However, their system uses a reward signal which is not sparse, namely, the distance of the robot's arm to the target. Using this information, the robot does not have to wait until the end of the sequence for any feedback about how well it is doing. In our case, the reward is delayed until the end of the behavioral sequence. This makes the learning problem more difficult since the information about the utility of each choice will not be available after each choice. One can think of our shaping rewards as a developmental mechanism to deliver this gradient information, to enable sequence learning in a feasible time frame.

V. Conclusions

In this paper, we have considered different variants of the reinforcement learning algorithm in a simulated robotic scenario in order to study theoretical challenges and possible solutions for application of RL in physical robots.

The state-action space of a robotic agent is high-dimensional and continuous in nature, which makes application of conceptually well-suited for robotics RL paradigm prohibitedly time- and resources consuming. We have argued that the neural dynamics, in particular the Dynamic Field Theory, offers means to segregate the continuous and high-dimensional state-action space of a robotic agent into discrete, attractor-based elementary behaviours. The dynamic neural fields and the DFT-based EBs have been previously demonstrated to successfully organise behaviour of real physical robots [4], [35]. Here, we have extended the recently introduced framework, which implements the RL algorithm SARSA in terms of the neural dynamics, by adding shaping. We have studied influence of the choice of the RL algorithm on the performance of the system, demonstrating that the T-learning enables learning in cases with a high probability of failure. We have demonstrated how shaping may be introduced in this framework, which facilitates learning of longer sequences in large spaces, discretised by the EBs.

In the animal learning literature, from which machine learning RL takes its inspiration, learning of long sequences from a single rewards at the end of a sequence is not typical. In most studies, RL in biological systems refers to learning of a single stimulus-response association only [36]. Thus, shaping – a successive learning of a sequence by rewarding intermediate stimulus-response transitions – is a way how RL may lead to learning long sequence in animals [10], [11]. Shaping is thus also a promising method to enable RL in robotics, where exploration of large behavioural spaces is too costly.

Reinforcement learning and shaping are not the only methods of learning, required to develop truly autonomous self-organising and adaptive agents. Indeed, these methods only lead to learning of what to do, i.e. which behaviours to activate in different situations, but not how to perform different actions. In some cases, more supervision and imitation learning may be needed [37], whereas in other cases motor babbling and self-organising mappings [38] may solve the "how" of the learning task.

Acknowledgment

This work was funded through the 7th framework of the EU in grant #270247 (NeuralDynamics project).

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: The MIT Press, 1998.

[2] G. Schöner, "Dynamical systems approaches to neural systems and behavior," in *International Encyclopedia of the Social & Behavioral Sciences*, N. J. Smelser and P. B. Baltes, Eds. Oxford: Pergamon, 2002, pp. 10 571–10 575.

[3] R. A. Brooks, "Do elephants play chess?" *Robotics and Autonomous Systems*, vol. 6, no. 1-2, pp. 3–15, 1990.

[4] M. Richter, Y. Sandamirskaya, and G. Schöner, "A robotic architecture for action selection and behavioral organization inspired by human cognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2012.

[5] Y. Sandamirskaya, M. Richter, and G. Schöner, "A neural-dynamic architecture for behavioral organization of an embodied agent," in *IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL EPIROB 2011)*, 2011.

[6] Y. Sandamirskaya, "Dynamic Neural Fields as a Step Towards Cognitive Neuromorphic Architectures," *Frontiers in Neuroscience*, vol. 7, p. 276, 2013.

[7] S. Kazerounian, M. Luciw, M. Richter, and Y. Sandamirskaya, "Autonomous reinforcement of behavioral sequences in neural dynamics," in *International Joint Conference on Neural Networks (IJCNN)*, 2013.

[8] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998, vol. 1, no. 1.

[9] L. P. Kaelbing, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

Fig. 5. Timings and Sequences of Elementary Behaviors of the Youbot. Left: During exploration. Right: Performing the Rewarding Sequence.

[10] B. F. Skinner, "The behavior of organisms: An experimental analysis." 1938.

[11] G. B. Peterson, "A day of great illumination: Bf skinner's discovery of shaping," *Journal of the Experimental Analysis of Behavior*, vol. 82, no. 3, pp. 317–328, 2004.

[12] V. Gullapalli, "Reinforcement learning and its application to control," Ph.D. dissertation, Citeseer, 1992.

[13] M. J. Mataric, "Reward functions for accelerated learning." in *ICML*, vol. 94, 1994, pp. 181–189.

[14] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposive behavior acquisition for a real robot by vision-based reinforcement learning," in *Recent Advances in Robot Learning*. Springer, 1996, pp. 163–187.

[15] L. M. Sasksida, S. M. Raymond, and D. S. Touretzky, "Shaping robot behavior using principles from instrumental conditioning," *Robotics and Autonomous Systems*, vol. 22, no. 3, pp. 231–249, 1998.

[16] M. Dorigo, *Robot shaping: an experiment in behaviour engineering*. The MIT Press, 1998.

[17] S. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybernetics*, vol. 27, pp. 77–87, 1977.

[18] E. Bicho, P. Mallet, and G. Schöner, "Target representation on an autonomous vehicle with low-level sensors," *The International Journal of Robotics Research*, vol. 19, no. 5, pp. 424–447, 2000.

[19] Y. Sandamirskaya and G. Schöner, "An Embodied Account of Serial Order: How Instabilities Drive Sequence Generation," *Neural Netw.*, vol. 23, no. 10, pp. 1164–1179, Dec. 2010.

[20] G. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering, 1994.

[21] S. Grossberg, "Behavioral contrast in short-term memory: Serial binary memory models or parallel continuous memory models?" *Journal of Mathematical Psychology*, vol. 3, pp. 199–219, 1978.

[22] S. B. Thrun, "The role of exploration in learning control," *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches. Van Nostrand Reinhold, New York*, 1992.

[23] J. Schmidhuber, "Curious model-building control systems," in *Proceedings of the International Joint Conference on Neural Networks, Singapore*, vol. 2. IEEE press, 1991, pp. 1458–1463.

[24] D. S. Touretzky and L. M. Saksida, "Operant conditioning in skinnerbots," *Adaptive Behavior*, vol. 5, no. 3-4, pp. 219–247, 1997.

[25] A. McGovern, R. S. Sutton, and A. H. Fagg, "Roles of macro-actions in accelerating reinforcement learning," in *Grace Hopper celebration of women in computing*, vol. 1317, 1997.

[26] O. G. Selfridge, R. S. Sutton, and A. G. Barto, "Training and tracking in robotics." in *IJCAI*. Citeseer, 1985, pp. 670–672.

[27] G. Konidaris and A. Barto, "Autonomous shaping: Knowledge transfer in reinforcement learning," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 489–496.

[28] M. Colombetti and M. Dorigo, "Training agents to perform sequential behavior," *Adaptive Behavior*, vol. 2, no. 3, pp. 247–275, 1994.

[29] J. Weng, "Developmental robotics: Theory and experiments," *International Journal of Humanoid Robotics*, vol. 1, no. 02, pp. 199–236, 2004.

[30] Webots, "http://www.cyberbotics.com," commercial Mobile Robot Simulation Software. [Online]. Available: http://www.cyberbotics.com

[31] V. Graziano, F. J. Gomez, M. B. Ring, and J. Schmidhuber, "T-learning," *CoRR*, vol. abs/1201.0292, 2012.

[32] J. Loch and S. Singh, "Using eligibility traces to find the best memoryless policy in partially observable markov decision processes," in *In Proceedings of the Fifteenth International Conference on Machine Learning*. Citeseer, 1998.

[33] M. R. James and S. Singh, "Sarsalandmark: an algorithm for learning in pomdps with landmarks," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 585–591.

[34] B. Durán, G. Lee, and R. Lowe, "Learning a dft-based sequence with reinforcement learning: a nao implementation," *Paladyn*, vol. 3, no. 4, pp. 181–187, 2012.

[35] W. Erlhagen and E. Bicho, "The dynamic neural field approach to cognitive robotics." *Journal of neural engineering*, vol. 3, no. 3, pp. R36–54, Sep. 2006.

[36] P. Dayan and B. Balleine, "Reward, Motivation, and Reinforcement Learning," *Neuron*, vol. 36, pp. 285–298, 2002.

[37] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning Movement Primitives," in *11th Iternational Symposium on Robotics Research (ISRR2003), Springer Tracts in Advanced Robotics Vol. 15*, P. Dario and R. Chatila, Eds. Springer Berlin / Heidelberg, 2005, pp. 561–572.

[38] R. Der and G. Martius, "From Motor Babbling to Purposive Actions : Emerging Self-exploration in a Dynamical Systems Approach to Early Robot Development," pp. 406–421.