PALADYN Journal of Behavioral Robotics

VERSITA

Research Article · DOI: 10.2478/s13230-013-0109-5 JBR · 3(4) · 2012 · 181-187

Learning a DFT-based sequence with reinforcement learning: a NAO implementation

Boris Durán[1]*, **Gauss Lee**[1]†, **Robert Lowe**[1]‡

1 Interaction Lab
University of Skövde
Skövde, Sweden

Received 18-12-2012

Accepted 27-03-2013

Abstract

The implementation of sequence learning in robotic platforms offers several challenges. Deciding when to stop one action and continue to the next requires a balance between stability of sensory information and, of course, the knowledge about what action is required next. The work presented here proposes a starting point for the successful execution and learning of dynamic sequences. Making use of the NAO humanoid platform we propose a mathematical model based on dynamic field theory and reinforcement learning methods for obtaining and performing a sequence of elementary motor behaviors. Results from the comparison of two reinforcement learning methods applied to sequence generation, for both simulation and implementation, are provided.

Keywords

sequences · neural dynamics · reinforcement learning · humanoid

1. Introduction

All agent actions from the simple to the complex, and across time scales of milliseconds to many minutes and beyond, involve sequences of sub-action elements. Sequences of such elements functional to 'desired' (e.g. goal- or reward-based) outcomes are not arbitrary, however. Individual elements must be organized such that an agent can move from an initial state or action to one that yields the desired outcome. Such organization in the context of embodied systems must be subject to ongoing revisions in order that desired outcomes can be consistently achieved.

Recently, the subject of much investigation into the embodied generation of sequences of sensorimotor activity has been the neural-dynamic approach with a particular focus on dynamic field theory, or DFT (cf. [1], [2]). In such systems, actions are selected as a function of the intrinsic dynamics of spatially continuous sensory and motor surfaces as subsymbolically represented by neural fields. A field is comprised of a population of neurons and has the same structure as a fully recurrent neural network where connections exert local excitatory or global inhibitory effects depending on the relative location within the network. Fields are used to represent perceptual features, motion or cognitive decisions, e.g. position, orientation, color, speed. The dynamics of these fields allow the creation of attractors (supra-threshold peaks) which are the units of representation in DFT [2]. The result of activating this type of network is a continuously adaptive system that responds dynamically to changes in sensed external stimuli. Through the use of field attractors, robotic behaviour can not only be responsive to, but also be be robust to, temporary sensory occlusions of stimuli [3]. Attractors also enable robustness to action durations [2] that may vary consequent to actuator inconsistencies or minor changes in the environment.

Sequence generation algorithms have been produced that exploit the embodied properties of neural fields. For example, [4] used a DFT architecture that enabled robots to infer goal-directed intentions from other robots in a joint action task utilizing the stable attractor states as various decision points. Action chains were learned using a hebbian rule following teacher demonstration of the sequence. [5] provided an architecture consisting of a number of ordinal nodes whose 'winner-take-all' selected activations represent effective intentions to produce a particular (sub-)action in a sequence. Sub-actions, which we refer to as 'elementary behaviours' (EBs - see [6]) are registered according to site activation on a particular 'action field'. Completion of a particular sub-action is registered following supra-threshold activation in a 'condition of satisfaction' (CoS) field. This activity in turn provides the key to destabilizing the current action 'intention' (active ordinal node) and enabling successor actions to be carried out.

Notwithstanding the successful deployment of such neural-dynamic architectures in mobile robots the two aforementioned models rely on an initial supervised hebbian learning phase prior to a secondary sequence generation phase. In the domain of autonomous mobile robotics, reinforcement learning processes, under the guidance of affective signals, are necessary for robots to continually update the learning units within a given sequence in an 'online' manner Reinforcement learning is a paradigm that has been employed for both autonomous, non-supervised learning in robots and for learning sequences of states or state-action pairs (cf. [7]). TD-based reinforcement learning robots, following initial random search, chance upon a sequence of behaviours that leads to a reward (positive reinforcer). In subsequent learning trials, the robots learn the temporally discounted value of states or state-actions pairs back-chained from the final rewarding state to the robot's start state (e.g. initial position in a learning trial).

The use of reinforcement learning in the context of sequence generation promotes autonomy but also viability as it enables robots to adapt to changing environments (i.e. where 'conditions of satisfaction' are not met). [8] demonstrated that temporal difference (TD) based Actor-

*E-mail: boris.duran@his.se
†E-mail: gauss.lee@his.se
‡E-mail: robert.lowe@his.se

Critic algorithms could be potentially deployed for learning arbitrary sequence lengths; however, their approach (neural network state function approximation - the 'Critic') relied on the learning, over blocks of trials, of stimulus-action pairs successively back-chained from the rewarding pair to the initial pair. This may therefore be seen as a semi-supervised approach. The model was also only tested in simulation using abstract actions without a mobile robotics application. Learning stimulus (or state)- action pairs in the real world represents a challenge that must account for the embodied dynamics of real world interactions. More recently, TD-based reinforcement learning techniques have been applied to robots learning fine-grained sensori-motor patterns [9] though these are not conceived in terms of learning chains of elementary behaviours that may be reused according to different contexts.
In this paper, we present an extension of the sequence learning/generation architecture of [10] that uses TD-based reinforcement learning algorithms with simple discrete state-action representation. In this way, we hope to exploit: 1) the embodied properties of a neural-dynamic approach (i.e. provision of stable time invariant attractors robust to temporary sensory occlusions), and 2) the autonomy provisioning properties of reinforcement learning for exploring the space of elementary behaviours (EBs) that contribute to rewarding sequences. We compare the performance of a NAO robot on a simple sequence learning/generation task using SARSA and Q-learning (both with TD(0) or TD(λ), i.e. one sample backups of state-action pairs). Comparisons of these two particular TD algorithms is common practise in reinforcement learning tasks motivated both from the perspective of computational efficiency (e.g. Bellot et al. 2012) and also biological relevance (cf. [11]). The main contribution of our work can, therefore, be viewed as providing an investigation into how standard TD-based reinforcement learning algorithms may be applied to autonomous agents operating in the real world over long duration action sequences.
The paper breaks down as follows: In section 2, we discuss the model that learns and generates serially ordered EBs as well as the particular pre-given EBs that the NAO robot can use; in section 3, we apply the model to the NAO robotic platform and present results of the learning efficiency of the two TD algorithms; finally, in section 4 we offer some concluding remarks.

2. The Model

The model developed for the present project is based on the work done by [10]. A single sequence of motor behaviors should be performed in an specific order. After exploring its motor repertoire, the agent will be able to extract that right sequence. Seven different actions were used for the current application:

- Stand up
- Search (position)
- Approach
- Arm up (point)
- Arm down
- Go back
- Sit down

Each of the seven actions previously mentioned is represented by what is known as an *Elementary Behavior* (EB), [6]. An EB groups three discrete neurons (a memory, an ordinal, and a condition-of-satisfaction) and two dynamic fields (an intention field and a condition-of-satisfaction field). Not all behaviors need to include fields and in the specific case of NAO some behaviors are performed in a blocking call, i.e. the program will not continue to the next line until the robot finishes its current action. This means that the onset and offset of a EB has a discrete nature, therefore there is no need (or possibility) of using dynamic fields for those EBs.
The main components of the proposed model are described next. First the different components of each elementary behavior are described under the dynamics of serial order. Second the algorithms used for both reinforcement learning approaches, *Q-learning* and *SARSA*, are explained as pseudo-code.

2.1. Serial order

In order to simplify but at the same time exemplify the use of neural and field dynamics, the model makes use of dynamic fields only for the *Search* EB. There are other EBs that could make use of dynamic fields as well, for example in the case of *Approach*, *Go back* or *Point* the robot could be dynamically coupled to fields that represent distances and orientations. The present work focuses on the learning of serial order of the discrete representation of behaviors, i.e. the right sequence of EBs. The implementation of all distances and orientations for each degree of freedom of NAO falls out of the scope of the present work.

Ordinal nodes

An EB i is activated by its ordinal node, v_i^o. The dynamics of ordinal nodes are given by the following equation:

$$\begin{aligned}\tau^o \dot{v}_i^o = & - v_i^o + h^o + c^{o,o}\sigma(v_i^o) - c^- \sum_{i'(\neq i)} \sigma(v_{i'}^o) \\ & - c^{o,s}\sigma(v_i^s) + \sum_{i'} W_{i,j}^{o,m}\sigma(v_j^m) \end{aligned} \quad (1)$$

where τ^o is the time constant, h^o is a negative resting level, $c^{o,o}$ is a self-excitatory gain, c^- is the strength of the mutual inhibition between ordinal nodes. $c^{o,s}$ is an inhibitory input from the condition-of-satisfaction node of the EB, this helps deactivate the ordinal node when the EB is finished. $W_{i,i'}^{o,m}$ is a connection weight from the memory node i' to the ordinal node i, this matrix controls the sequential activation of ordinal nodes. Finally, $\sigma(\cdot)$ represents a sigmoidal non-linearity which gives the output of each node.

Memory nodes

Memory nodes, v_i^m, become activated when the associated ordinal node has been activated. They remain active through strong self-excitatory coupling even after the activation of the condition-of-satisfaction node. An active memory node provides excitatory input to the next ordinal node in a sequence. These memory-ordinal connections are the ones encoding the serial organization of EBs, therefore it is the matrix that represents them the one that will be the target of the reinforcement learning algorithms.

$$\tau^m \dot{v}_i^m = - v_i^m + h^m + c^{m,m}\sigma(v_i^m) + c^{m,o}\sigma(v_i^o) \quad (2)$$

notations is analogous to that of Eq.(1). Note that there's no mutual inhibition between memory nodes.

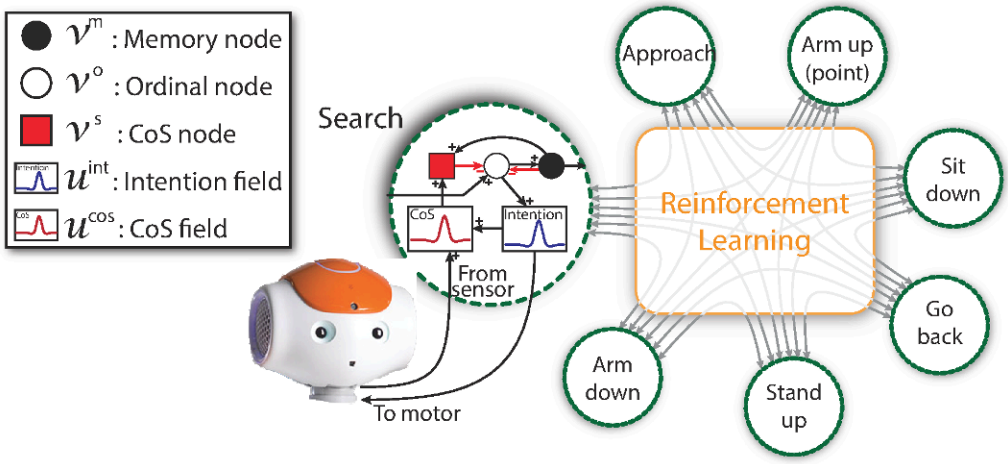

Figure 1. Schematics of the model for sequence generation based on dynamic field theory and reinforcement learning.

Condition-of-Satisfaction nodes

An active CoS node, v_i^s, signals that the behavioral goal of the EB have been realized. Its dynamics are described by the following equation:

$$\begin{aligned}\tau^s \dot{v}_i^s = & - v_i^s + h^s + c^{s,s}\sigma(v_i^s) - c^- \max_i(\sigma(v_i^s)) \\ & + c^{s,m}\sigma(v_i^m) + c^{s,cos} I_{cos} \end{aligned} \quad (3)$$

where the first three terms form the generic dynamics of an Amari node, the fourth term is a global inhibition within a layer. Input from the memory node of the EB is scaled by the constant, $c^{s,m}$. I_{cos} represents the information from sensors and for the case of on-or-off behaviors this would be a simple 1 or 0 respectively, but for sensors attached to continuous dimensions (fields) this term is given by $\int \sigma(u_i^s(x'))dx'$.

Intention field

An intention field, $u_i^{\text{int}}(x)$, receives localized input from the ordinal nodes. An active ordinal node thus specifies parameter values of the intended action. Since we are only using the *Search* behavior to exemplify the use of fields inside an EB, x, represents the orientation of NAO's head. For other EBs an intention field could be associated to other features (e.g., color, distance, area). The dynamics of the intention fields reads:

$$\begin{aligned}\tau^{\text{int}} \dot{u}_i^{\text{int}}(x) = & - u_i^{\text{int}}(x) + h^{\text{int}} + I^{\text{int}} \\ & + \int \sigma(u_i^{\text{int}}(x'))w^{\text{int}}(x - x')dx'. \end{aligned} \quad (4)$$

The intention field receives input, I^{int}, from the associated ordinal node, v_i^o, through a synaptic weights function, $W_i^{\text{int}}(x)$; or, as for the current application, from a gaussian input with mean at the position of the tracked object within the field of view of NAO's camera. The area of the tracked object is used to proportionally scale the amplitude of this input. Pushed by this input the intention field crosses the detection instability creating a peak and activating the pan motion of NAO's head.

Condition-of-Satisfaction field

A CoS field, $u_i^{\cos}(x)$, tells the system whether the intended action of the current EB has been achieved or not. This field receives input from sensors and from the intention field. Once both overlap, a peak forms in the CoS field and activates the CoS node, which in turns deactivates the current EB. The dynamics of the CoS field are given by the following equation:

$$\begin{aligned}\tau^{\cos} \dot{u}_i^{\cos}(x) = & -u_i^{\cos}(x) + h^{\cos} + I^{\cos}(x) \\ & + c^{\cos} \int \sigma(u_i^{\cos}(x'))w^{\cos}(x - x')dx' \\ & + c^{\cos,\text{int}} \int \sigma(u_i^{\text{int}}(x'))w^{\cos,\text{int}}(x, x')dx' \end{aligned} \quad (5)$$

2.2. Reinforcement learning

In order to verify the validity of reinforcement learning (RL) for a DFT-based sequence model, *Q-learning* (off-policy) and *SARSA* (on-policy) were applied and compared to each other. The major difference between these two learning mechanisms is that the maximum reward for the predicted state is not necessarily used for updating the q-values in *SARSA* but it is needed in *Q-learning*.

According to the characteristics of the sequence model in section 2, the connection matrix $W^{o,m}$ (Eq.3) needs to be learned. In reinforcement learning, this matrix is represented by Q which is used to update the connection weights following a gradient change. Rows and columns of Q represent both the action taken a and the state space s representing finished actions, respectively. At the same time an arbitrary element $W_{i,j}^{o,m}$ represents the connection from an j-th EB into the next i-th EB. Therefore, a direct mapping between the $Q(a, s)$ matrix and $W^{o,m}$ can be made. The pseudo-code described below illustrates the process of reward/punishment sequence learning.

RL pseudo-code:

1. Initialize discount factor γ, learning rate $\{\alpha_k\}_{k=0}^{\infty}$ and Q matrix. Initialize the state s.
2. Select an action a according to the $\epsilon - greedy$ policy $\pi(a,s) \sim argmax\hat{Q}^{\pi}(a,s)$, where $\epsilon = 0.02$.
3. Observe its reward r and the next state s' . Calculate the prediction $Q(a',s')$ and update $Q(a,s)$, $W_{a,s}^{o,m}$:
   - For $Q\ learning$:
     - choose the maximun $Q(a',s')$ and update:

$$Q_{new}(a,s) = Q(a,s) + \alpha[r - Q(a,s) + \gamma max_{a'} Q(a',s')]$$
$$W_{a,s}^{o,m}(a,s) = F(Q_{new}(a,s))$$

   - For $SARSA$:
     - choose the $Q(a',s')$ according to $\epsilon - greedy$ policy and update:

$$Q_{new}(a,s) = Q(a,s) + \alpha[r - Q(a,s) + \gamma Q(a',s')]$$
$$W_{a,s}^{o,m}(a,s) = F(Q_{new}(a,s))$$

4. if the condition for state transition (see 3) is fulfilled then s = s', repeat from 2) until the sequence is memorized.
   *F is a function for extracting the diagonal and positive values from Q matrix.

Within the reinforcement learning process, the model of serial ordinal node is utilized as the memory which records the correctly-learned sequence of behaviors. The learning algorithm converges when the sequence of actions are fully memorized (The convergence is detected when the ArmDown node is up in neural sequence model). According to [12], RL is a function pertaining to dopamine systems in human brains. Hence, the implementation of RL with DFT-based models is actually the embodiment of a cognitive dynamic system involving the interaction of neural memory, dopamine systems and environment.
As to the efficiency of Q-$learning$ and $SARSA$, it is, in fact, a very application-dependent topic [7]. For the implementation on NAO, in order to make algorithms converge efficiently, an eligibility trace rule is applied for approximating action-state function and given by:

$$e(a,s) = \frac{exp(Q(a,s))}{\sum_{a_j} exp(Q(a_j,s))} \tag{6}$$

where $e(a,s)$ is eligibility for action-state pair (a,s). a_j is an arbitary state. $\sum_{a_j} exp(Q(a_j,s))$ is a sum with respect to all possible a. The advantage of using this moderate eligibility is: Firstly, it is associated with Q(a,s) with one-step backward back-up ($\alpha e(a,s)[r - Q(a,s) + \gamma Q(a',s')]$). If $Q(a,s) < 0$, the eligibility is reduced, increasing that of the other actions. If $Q(a,s) > 0$, the eligibility is increased by suppressing the other behaviors. Because of the requirement that each action can only be taken once in the sequence of the serial ordinal model, this eligibility can speed up the update. Secondly, $e(a,s)$ is bounded between 0 and 1. Unlike the incremental eligibility in [7], the increase of speed is also bounded, lest $W_{a,s}^{o,m}(a,s)$ can be abruptly updated over its boundary value. Finally, as for the normal eligibiltiy rule, $e(a,s)$ for all a,s decays with $\lambda = 0.8$ for each step if $Q(a,s)$ is not visited often anymore, the $e(a,s)$ decays with $\lambda e(a,s)$ if it is SARSA. It decays according to Watkin's rule [7] if it is Q-learning. On the other hand, the $e(a,s)$ in our work is not initialized with 0 as the backup of "penalties" is also necessary and useful in ruling out the negative behaviors.

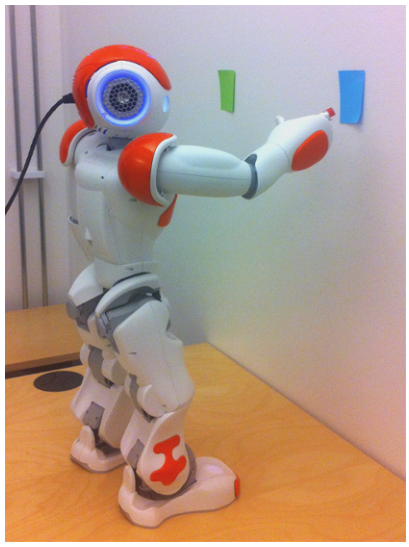

Figure 2. Snapshot of NAO after completing the learning process in the proposed scenario.

3. Results

In order to successfully perform the whole sequence the physical robot needs to always start from $Stand\ up$ and end with $Arm\ down$ (the restart condition of one episode). For each step of one episode, the robot evaluates the behaviors according to its value function and adapts to environmental change (the position of the approaching target). With the implementation in two different situations (simulated in Matlab and non-simulated on the physical robot), the difference of supervised and non-supervised learning is observed.
NAO's repertoire of motor behaviors for this application consisted of 7 elementary actions. From this, for performing a simple task, a smaller set (three EBs) was chosen: search for an object, approach and point/reach, Fig. 2. A very basic color-segmentation algorithm was used to obtain the position of an object in the field of view of NAO. This information was represented in a dynamic field to drive the EB named $Search$.

3.1. Simulation: supervised

The reinforcement learning mechanisms here proposed take advantage of the discrete nature of the DFT sequential model. In this simulation, the condition for the transition between states is achieved when the CoS node of each elementary behavior is active and stable, i.e.

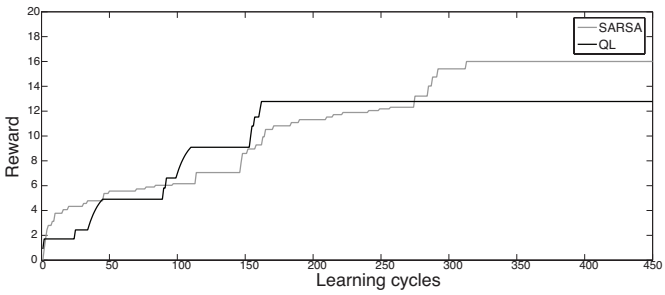

Figure 3. Comparing the performance of Q- and $SARSA$- learning in simulation.

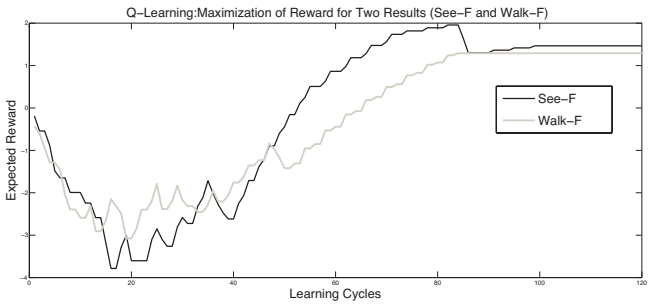

Figure 4. The curve of Q-learning for two converged behaviors: See-F and Walk-F.

$\sigma(v_i^s) \simeq 1.0$. At the same time the next potential action is selected from the activity of ordinal nodes, i.e. $\sigma(v_i^o) \simeq 1.0$. The reward signal r is a constant value set to 4 if a is equal to its correspondent action in the supervised set. It is equal to -10 (a penalty) if a is a repeated action in the last step. Otherwise it is set to zero.
Figure 3 shows the results (positive reward) of a supervised reward/punishment learning using Matlab. It is clearly observed that Q-*learning* outperforms $SARSA$ in terms of convergence speed. However, $SARSA$ has its own characteristics as well since its exploratory feature is more smooth and stable. Q-*learning* exploration happens densely during the end of every learned action which can be seen at the flat regions of Fig. 3. The sharp edges of the Q-*learning* curve represent the "greedy" quality of the exploitation of the selected actions.

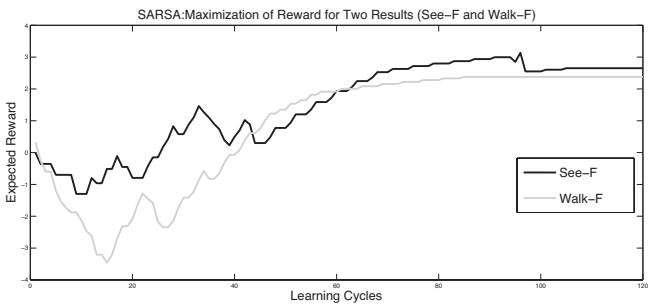

Figure 5. The curve of SARSA for two converged behaviors: See-F and Walk-F.

3.2. Implemention on NAO: non-supervised

In order to glean the performance of Q- and $SARSA$- learning on the physical robot, we separately implement two algorithms in the same context and scenario. The robot learns the Q matrix every step in the episode. Different from the simulation, the transition condition is merely satisfied when the action taken is not a repetitive action in a previous sequence in each episode. Nevertheless the reward function is different and given by:

$$reward = r_{distance} + r_{penalty} \tag{7}$$

Where $r_{distance}$ is the only reward based on sensory information. For each action, if the distance from the arm to the target is closer, $r_{distance} = 20$; if the distance increases, $r_{distance} = -10$; otherwise, $r_{distance} = 0$. $r_{penalty} = -10$ when a repeated action is selected.
In the scenario, the robot needs to pick up proper actions and organize them in a correct sequence from a limited action space in order to finish the task through interaction with the environment. Specifically, the task is about finding a blue paper on the wall, approach (walk to), and point with lifting its arm. Since the distance is a dominant variable in this task, it is chosen to be the only reward-related information. With the NAO robot, it is able to sense the distance change in 3D space between two steps. The visual searching space is $[-\frac{\pi}{2}, \frac{\pi}{2}]$ in front of NAO, in which case the robot's working space is also limited. After finishing each episode, NAO is put back in the initial position.
Figure 4, 5 show the results of learning curves for two algorithms. Firstly, the efficiency advantage of Q-*learning* is not observed, which is distinct from supervised learning. Two algorithms both converge around 100 learning cycles. Secondly, in the first 30 learning cycles, there is a decrease of expected reward. Since the eligibility and $\epsilon - greedy$ policy both do not work well in the beginning, the robot needs to explore to get penalties in order to know which action to take. Those penalties are forward-chained with subsequent behaviors in a complete action sequence in one trial, gradually driving the emergence of a coherent action sequence. In dynamic systems theory, explorations of this type are "instabilities" which are important to the emergence of new behaviors [13]. Meanwhile, when the robot gets "punished" through interaction exploration, unappealling actions are suppressed, causing the emergence of attraction of certain behaviors. For instance, "arm down", as a neutral behavior, cannot be finally chosen without oppressing the other behaviors.

Interestingly, the sequence learning, most of the time, converges to two continuous behaviors which comprises an intriguing result: the dilemma of seeing (sensing) first versus walking (acting) first. One sequence is *Search* (seeing) - *Approach* (walking) - *ArmUp* - *ArmDown* and the other is *Approach* (walking) - *Search* (seeing) - *ArmUp* - *ArmDown*. It seems both of these two behaviors are correct. However, according to [14], sensorimotor systems are contingent on visual stimuli. In other words, these two systems cannot be separated. In the EB, these two actions are independent from each other, which may account for the 'dilemma' of arranging the sequence of seeing and walking. Meanwhile, the other behavior related to "forgetting of the correction behavior" is observed in the process of learning. For instance, when the robot chooses to walk backward (negative behavior) and then statistically select walking forward, the link between these two actions is considered as a positive connection (the correspondent Q(a,s) is updated above zero). It seems walking forward is the correction of walking backward. However, since walking backward is an unappealing action

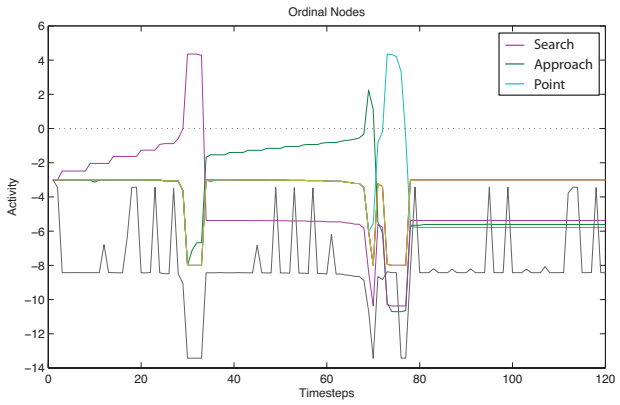

Figure 6. Activity of ordinal nodes for all elementary behaviors when the *Search* behavior is explored first.

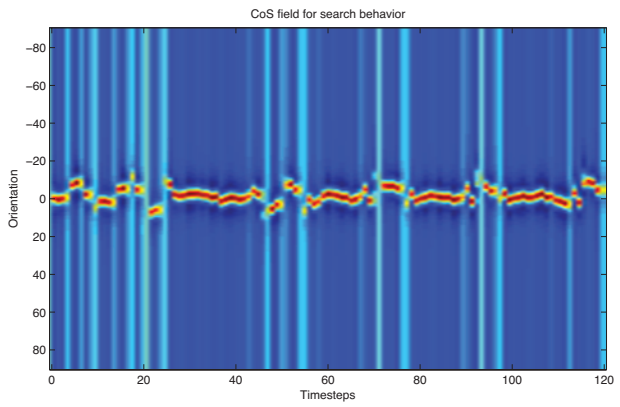

Figure 7. Activity of the CoS field for the *Search* behavior when the search behavior is explored first.

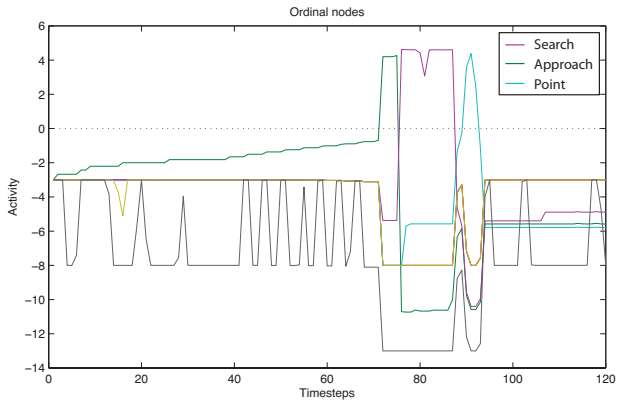

Figure 8. Activity of ordinal nodes for all elementary behaviors when the *Approach* behavior is explored first.

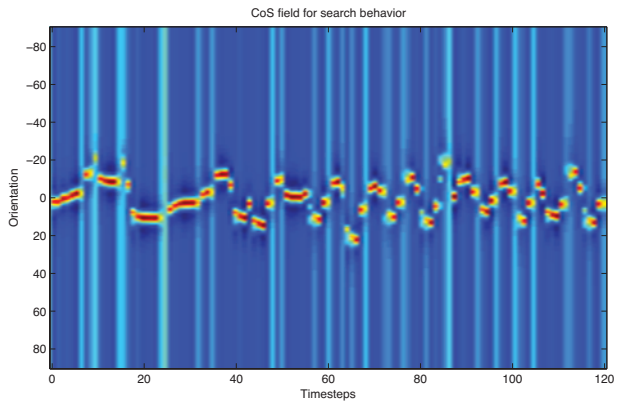

Figure 9. Activity of the CoS field for the *Search* behavior when the *Approach* behavior is explored first.

in our task, it is seldom chosen so that the positive link disappears, as if it is "forgotten".

Because of the random learning policy at the beginning, the system fully explores the action space in order to experience different negative and positive behaviors with respect to the reward function. This determines the volatile dynamics for each experiment. Since the aim of our work is to memorize the right sequence with a DFT-based model, the dynamics of the neural model during learning is not crucial as long as the ordinal nodes layer records the converged action sequence with three node activities above threshold (Figure 6 and 8). In both figures there is a gradual increase of the dynamics chosen by the system to converge into a first optimal behavior. Intriguingly, the onward dynamics is entirely distinct. In the learning process, the system explores the complete sequence as a target so that it is possible that some positive transitions have been found before or when the first converged behavior is being determined. As the sequential behavior can only be boosted one by one from each state to the next, the first unidentified behavior actually delays the boost of the subsequent nodes. This is why in Fig. 6, the activity of the *Pointing* behavior immediately reaches the zero threshold after the inhibition of the *Approach* behavior. In Fig. 8 the activities of both *Search* and *Point* behaviors reach the zero threshold immediately after the first converged *Approach* behavior. These behaviors are the ones learned before the first optimal one based on the forward-propagated punishment and back-chained reward.

The activity of the ordinal nodes for the case when the learning module took the *Search* behavior first is plotted in Fig. 6. Since this behavior was explored and executed first, the robot tries to locate the target in the center of the camera for later approaching it. A peak in the Condition-of-Satisfaction (CoS) field for the *Search* behavior (Fig. 7) inhibits the activity of the *Search* node allowing a new behavior (*Approach*) to be explored and executed. Once the *Approach* behavior has finished a CoS signal for this node inhibits its activity allowing the *Point* behavior to start its exploration and execution. Something similar happens when the learning module explores the *Approach* behavior first, Fig. 8. A new sequence of behaviors is learned from the same dynamics of exploration, execution and inhibition but this time the initial behavior is *Approach*.

The input to the CoS field comes from the camera of the robot and it represents the horizontal orientation of a colored object or target, Fig. 7, 9. The main activity of these fields stays around the center of the camera with some interruptions due to the non-smooth way of NAO ap-

proaching the target. By comparing these two plots it is easy to notice the advantages of selecting the Search behavior before starting to walk with the *Approach* behavior. In Fig. 7 it is possible to see a smoother tracking of the target compared to the one observed in Fig. 9. In other words, a sequence where the robot adopts 'Search' the target first before Approaching' it is more efficient than a sequence where the robot starts walking before knowing where to walk.

4. Conclusion

DFT-based sequence generation was previously presented with supervised learning, i.e. using a Hebbian rule, [5], and implemented on a mobile robot. In this paper we sought to validate and extend that work to a non-supervised type of learning and to implement it on a humanoid robot platform. The advantages of using neural dynamics in sensory-motor loops are tangible when dealing with real environments and it was once more demonstrated in our current application. Moreover, the flexible, but robust, nature of the serial order dynamics was evident when stabilizing the transitions between elementary behaviors. The activity of ordinal and Condition-of-Satisfaction nodes was perfectly matched to the pair action-state of reinforcement learning algorithms fusing them to dynamic field theory effortlessly. From the reinforcement learning point of view DFT works as the "critic" part of an actor-critic configuration.
Reinforcement learning, as a dynamic learning process based on interaction, adaptively connects the environment and the memory (DFT-based sequence model) in searching for the task-oriented sequence with respect to a specific reward/punishment chain. From the perspective of dynamic systems theory, RL finds two "attractors" from the given primitive behaviors in order to finish a goal-directed task by balancing exploration and exploitation. The "instabilities" shown in the learning process not only insulates the importance of exploration but also the source of the emergence of new behaviors[13]. However, since the action and state space is limited with EB for ordinal-node model, the performance is highly dependent on the modelling of each basic action. This is the reason for the occurrence of the dilemma between seeing and walking.
The use of RL methods and DFT-based sequences was presented in [15], being the main difference the use on our side of punishment to rule out the "negative" behaviors. Moreover we thought important to include a comparison of RL methodologies in this type of tasks which is not done in [15].
The study of two very different learning processes in our model gave us important insights into the way sequences can be built depending on initial conditions. One of the sequences followed a more logical path starting from a search behavior for later approaching and finally for reaching or pointing. The other case did not follow a logical path and its effects were observed in the poor tracking of targets. This tells us that some trade off or mixed model must be implemented between supervised and unsupervised learning approaches in order to obtain efficient and dynamic learning architectures.
The following stages of our research involve the implementation of hierarchical serial order. After showing the feasibility of working with the NAO platform we are encouraged to create more complex scenarios involving different objects and routines using the same robot.

References

[1] S. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybernetics*, vol. 27, pp. 77–87, 1977.
[2] G. Schöner, *Cambridge Handbook of Computational Cognitive Modeling*. R. Sun, UK: Cambridge University Press, 2008, ch. Dynamical systems approaches to cognition, pp. 101–126.
[3] E. Bicho, P. Mallet, and G. Schoner, "Target representation on an autonomous vehicle with low-level sensors." *The International Journal of Robotics Research*, vol. 19, no. 5, pp. 424–447, May 2000. [Online]. Available: http://dx.doi.org/10.1177/02783640022066950
[4] W. Erlhagen, A. Mukovskiy, F. Chersi, and E. Bicho, "On the development of intention understanding for joint action tasks," 2007.
[5] Y. Sandamirskaya and G. Schöner, "An embodied account of serial order: How instabilities drive sequence generation," *Neural Networks*, vol. 23, no. 10, pp. 1164–1179, December 2010.
[6] Y. Sandamirskaya, M. Richter, and G. Schöner, "Neural dynamics of sequence generation and behavioral organization," in *Front. Comput. Neurosci.: Computational Neuroscience & Neurotechnology Bernstein Conference & Neurex Annual Meeting, BC11*, no. 0, 2011.
[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, Mar. 1998. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262193981
[8] R. E. Suri and W. Schultz, "Learning of sequential movements by neural network model with dopamine-like reinforcement signal," *Experimental Brain Research*, vol. 121, pp. 350–354, 1998, 10.1007/s002210050467. [Online]. Available: http://dx.doi.org/10.1007/s002210050467
[9] J. Modayil, A. White, and R. S. Sutton, "Multi-timescale nexting in a reinforcement learning robot," *CoRR*, vol. abs/1112.1133, 2011.
[10] Y. Sandamirskaya and G. Schöner, "Serial order in an acting system: a multidimensional dynamic neural fields implementation," in *Development and Learning, 2010. ICDL 2010. 9th IEEE International Conference on*, 2010.
[11] Y. Niv, "Reinforcement learning in the brain," *Journal of Mathematical Psychology*, vol. 53, no. 3, pp. 139–154, 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0022249608001181
[12] M. Wiering and M. van Otterlo, *Reinforcement Learning: State-Of-the-Art*, ser. Adaptation, Learning, and Optimization. Springer, 2012. [Online]. Available: http://books.google.com/books?id=YPjNuvrJR0MC
[13] E. Thelen and L. Smith, *Dynamic Systems Approach to the Develop*, ser. The MIT Press/Bradford Books series in cognitive psychology. Mit Press, 1996. [Online]. Available: http://books.google.com/books?id=kBslxoe0TekC
[14] J. K. O'Regan and A. Noë, "A sensorimotor account of vision and visual consciousness." *The Behavioral and brain sciences*, vol. 24, no. 5, Oct. 2001. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/12239892
[15] S. Kazerounian, M. D. Luciw, M. Richter, and Y. Sandamirskaya, "Autonomous reinforcement of behavioral sequences in neural dynamics," *CoRR*, vol. abs/1210.3569, 2012.