High-level representations through unconstrained sensorimotor learning

Ozgur Baran Ozturkcu
Computer Science Department
Ozyegin University
Istanbul, Turkey
baran.ozturkcu@ozu.edu.tr

Emre Ugur
Department of Computer Engineering
Bogazici University
Istanbul, Turkey
emre.ugur@boun.edu.tr

Erhan Oztop[12]
[1]Ozyegin University, Istanbul, Turkey
[2]Osaka University, Osaka, Japan
erhan.oztop@ozyegin.edu.tr

***Abstract*—How the sensorimotor experience of an agent can be organized into abstract symbol-like structures to enable effective planning and control is an open question. In the literature, there are many studies that start by assuming the existence of some symbols and 'ground' those onto continuous sensorimotor signals. There are also works that aim to facilitate the emergence of symbol-like representations by using specially designed machine learning architectures. In this paper, we investigate whether a deep reinforcement learning system that learns a dynamic task would facilitate the formation of high-level neural representations that might be considered as precursors of symbolic representation, which could be exploited by higher level neural circuits for better control and planning. The results indicate that without even explicit design to promote such representations, neural responses emerge that may serve as the basis of abstract symbol-like representations.**

Keywords—Symbol Emerge, Reinforcement Learning Symbol Generation, Symbol grounding

I. INTRODUCTION

The term 'concept' or 'symbol' corresponds to internal representation of classes of things. Understanding symbolic manipulation is important as it enables humans with the means to classify, understand, predict and communicate [1]. From a neural evolutionary point of view, it is not yet known when and how high level, symbol-like representations emerge and start to be utilized by living systems for action and planning. As a general trend neural evolution utilizes what is available rather than reinventing a better version of an existing neural circuit [2]. So, it is conceivable that the symbolic, conceptual representation that humans use for effective action execution and planning are based on more primitive neural circuits evolved earlier.

Mostly, the AI literature does not consider this evolutionary symbol formation view. The mainstream use of symbols in AI dates back to times of the first intelligent robot Shakey [3], and can be linked to Newell and Simon's work [4], who both adhered to the notion that a physical symbol system has the necessary and sufficient means for general intelligent action. From this perspective, symbols can be seen as the main ingredient of intelligent behavior. However, the developmental psychologist and roboticists argue that the symbols are not innate and emerge by the dynamic interactions of the agent with the environment [5, 6, 7]. Symbol formation thus should be facilitated by the formation of intermediate neural representations for abstractions and concepts.

II. RELATED WORK

In the literature, there are many successful studies that have started by assuming the existence of symbol-like representations and discovered the continuous sensorimotor signals to ground them to the real world. In [8] Precup et al. showed that temporally learned abstract knowledge and the action series (options) make learning more efficient. Options were used as sub-goals with the aim of improving themselves. Once a policy selection method was chosen, it requires following the internal policy to the end. In addition, this was extended to concurrent activities, multi-agent coordination and hierarchical memory for addressing partial observability [9]. In [10], Saxe et al. brought a new look to macro actions (series of primitive actions), which ran several macro actions simultaneously to solve new tasks. While the aforementioned studies assumed the existence of predefined symbols, others have adopted the notion that symbols should be formed by the experience obtained through the sensorimotor apparatus of the agents [11]. In this vein, Ugur and Piater showed how symbols and symbolic rules can be formed in the continuous sensory space of a robot that explored the objects with its push, poke, grasp and release actions [12,13]. While these studies have explored symbol emergence in a forward predictive model learning framework, the work of Konidaris et al. [14,15] considered symbol formation in a Reinforcement Learning (RL) framework. To be concreate, Konidaris et al. showed the formation of symbols to be used as preconditions and effects of actions for deterministic [14] and probabilistic [15] plans in simulated environments. Subsequently, they showed that this framework can be applied to physical robotic systems for discovering symbolic and rule-based representations from robot sensorimotor data [16]. While aforementioned studies assumed existing feature extractors and focused on generating compact representations of symbols for planning, there are other studies that aimed at facilitating the emergence of such symbols using specially designed neural networks or classification techniques. For example, Stolle et al. [17] studied macro actions for creating logic and evaluation perspectives. Their intuition was that states that are frequently visited could provide a useful goal. Pierre-Luc Bacon et al. [18] showed that option-critic architecture was capable of learning both the internal policies and the termination conditions of options without additional rewards or sub-goals. Finally, Ranchod et al. [19] utilized inverse reinforcement learning to discover reusable skills with the segmentation of unstructured trajectories by applying a Bayesian nonparametric approach.

Overall, in all the review works above, there are always design biases or choices that promotes the formation of high-level representations. In this paper by contrast, we investigate the formation of symbol-like representations with no assumptions on the existence of such representations and with no explicit design choices to promote abstractions. To be concrete, our work is focused on the key consideration of whether a simple learning mechanism with no neural network engineering generates neural activities that may be considered symbol-like or abstract high-level representations. A positive answer would be invaluable, as one can envision higher level circuits that exploit these neural activities for further planning thereby giving a computational account of possible neural organization for movement control and learning.

To realize the aforementioned no-bias scenario, a RL framework is adopted for the sensorimotor learning of the 'squat-to-stand' task, where a robot needs to learn to generate joint torques to change its posture from squat to stand without falling over. In this setup, we sought to investigate whether symbol-like representations emerge during learning. This was done by analysing the properties of the neurons in the policy representing neural network during and after learning.

III. METHOD

To investigate whether high-level abstract representations may emerge through sensorimotor learning, we adopt a RL framework to teach a simulated robot to stand up in the face of postural perturbations. The task is adapted from our ongoing work on human adaptation to postural perturbations [20,21]. The critical consideration here is to see whether a simple learning mechanism with no neural network engineering generates neural activities that encode abstract high-level representations some of which may be considered precursors of symbolic representation.

A. Task

We use a simulated three degrees of freedom robot that has to learn from a neural network controller squat-to-stand movement under perturbation. The robot is modeled as a three-link chain attached to the ground (Fig. 1). In particular, the robot is composed of an upper leg, lower leg and a torso, with lengths of 0.61m, 0.39m, 0.61m and masses of 0.10kg, 17kg, 32.44kg respectively. The equations of motion are generated by the PyDy package (https://www.pydy.org). For the RL setup, the task state (Eq. 1) is defined as the vector of joint angles (∂_i) and angular velocities (φ_i):

$$S = \{\partial_1, \partial_2, \partial_3, \varphi_1, \varphi_2, \varphi_3\} \quad (1)$$

The action parameters (Eq. 2) of the robot controller are defined as the vector of torques (τ_i) applied to each joint at each simulation time step:

$$\upsilon = \{\tau_1, \tau_2, \tau_3\} \quad (2)$$

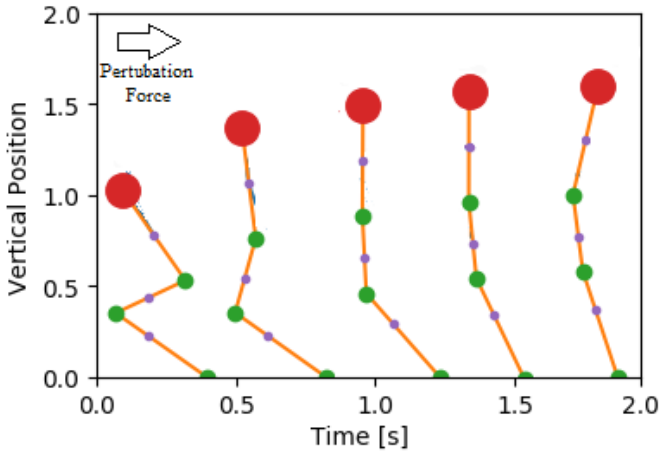

Fig. 1 Simulation movements evolving over time after training a neural network successfully.

An episode is defined as 2 seconds in which the robot has to complete the squat-to-stand task. The RL control frequency is set to 50Hz, therefore one episode generates 100 data points unless the episode ends due to one of the termination conditions (see below). The goal of the squat-to-stand task is considered to be satisfied when the height of the robot (i.e., endpoint/head vertical position) can be kept over 1.5 m for a duration of at least 0.2 seconds.

To make the task more challenging, the squat-to-stand task includes a non-trivial perturbation which pushes the robot in the posterior direction with a force F_{pert} proportional to the vertical velocity of the center of mass V_{com} of the robot (Eq. 3):

$$F_{pert} = C \,.\, V_{com} \quad (3)$$

In all simulations, the perturbation constant C is taken as C=300, except for the environment change experiment (see Section III.C) where it is reduced to induce a change in the environment. The perturbation constant and dynamic parameters for the model are chosen to mimic the parameters used in an ongoing experimental study where human adaptation in full body movement is studied [20,21].

B. Learning setup

To solve this RL problem one can define several reward functions. In the reported experiments in this paper, no terminal cost is used, and the running reward function r is defined as an increasing monotonic function of robot height, *h(t)* that is the vertical position of the end effector of the robot's kinematic chain at time t. To be concrete, the reward function is given with $r(h) = (h/2+0.5)^2$ if $h>0$ otherwise $r(h)=0$. An episode is terminated when the allowed time of 2 seconds has elapsed or the robot falls ($h < 0.5$m) or the robot hits the joint limits $|\partial_i| \geq \pi$ for any joint *i*.

Since the action space is intrinsically continuous, we adopted a policy gradient method that can represent policies with continuous action spaces. Policy gradient RL finds a local optimal policy by following the gradient of the expected total reward over episodes. In this study, we used an actor-critic method that has a stochastic policy (Eq. 4), which is used to sample the actions for policy exploration and exploitation. The

critic, on the other hand, evaluates the goodness of the current policy by estimating the value function (V_π).

$$\pi \sim N(\mu, \sigma) \quad (4)$$

Thus, the actor and critic are two separate function approximators implemented as neural networks. In the current implementation, the actor network represents the mean of the policy and its parameters are updated according formula (Eq. 5).

$$\nabla U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\nabla_\theta log\pi_\theta\left(u_t^{(i)}|s_t^{(i)}\right)\left(\sum_{k=t}^{H-1}R\left(s_k^{(i)},u_k^{(i)}\right)-V_\phi^\pi\left(s_k^{(i)}\right)\right) \quad (5)$$

Where u, s and R(s,t) represents action and state and immediate reward; t, H, m represents time, rollout length and number of rollouts respectively; θ and ϕ denote the weights of the policy and critic networks respectively [22]. In turn, the critic network parameters, are updated by minimizing the squared loss to regress V against the average cumulative reward collected over the sampled trajectories (Eq. 6):

$$\phi_{i+1} \leftarrow argmin_\emptyset \frac{1}{m}\sum_{i=0}^{m}(R - V_\emptyset^\pi(S_i))^2 \quad (6)$$

The standard deviation σ controls the exploration-exploitation trade off through action sampling (Eq. 4). It is gradually decreased as learning progresses, as opposed to being learned by a neural network, which we found to work better for our task. The decay is implemented by the update rule where $\sigma(t+1) = \sigma(t)\ x\ 0.999$, and the decay constant is chosen empirically.

Both the actor and critic networks are designed as small networks to avoid redundancy and thus ease the analyses of neural responses. A policy network with a single hidden layer composed of 32 neurons is empirically found to be enough to learn a task. The input and output layers are automatically determined by the state and action spaces. Consequently, the policy network has 6 inputs and 3 outputs corresponding to the dimensions of state and action spaces, respectively. The critic network is implemented as a two hidden layer network. The number of neurons in the layers are set as 16 and 32 conforming to the small network desiderata.

C. Experiments and Data Analysis

The analysis addresses three basic issues of whether learning facilitates (i) the emergence of neural encoding of the physical robot state, (ii) the specialization of neuron populations that are formed via learning, and (iii) the response of the learned squat-to-stand controller network to changes in the environment. For the analysis, all the data generated during learning and testing were first stored in a database, which consisted of the generated actions, states, rewards, robot joint positions, and the neuron outputs at each time step. The neuron outputs were normalized within their corresponding layer so to be in the range 0 to 1. The details of the analyses are given in the following three subsections.

1) Neural coding of the robot state

For the control of the task of the squat-to-stand movement under perturbation, a key physically meaningful parameter is the position and velocity of the centre of mass of the robot (COM). In this analysis, we aimed to detect neurons that may capture COM dynamics, in particular the vertical distance of COM from the ground (COM Y) and the vertical velocity of the COM. For each neuron in the policy network, the linear correlation between neuron outputs and COM height/velocity was computed. Subsequently, correlations of the neuron outputs with COM height/velocity were obtained. This allows the visualization of the COM dynamics to be represented within the neuron population as histograms. The analysis was conducted after 20K training episodes, where the learning was stabilized, and the robot was observed to complete the squat-to-stand movement successfully. For detecting neural representation of the COM dynamics, single trials were performed with exploration turned off ($\sigma = 0$). The learning and testing were repeated 20 times to assess the variability in COM height and velocity representations formed.

2) Neural population specialization

This analysis focused on detecting functional specialization of neurons during the learning process. For this analysis, learning was conducted for 20K episodes until the robot learned to complete the squat-to-stand task. While training was taking place, in every 1,000 episodes, the exploration was temporarily turned off, and the policy network was put in the control of the robot with the current network weights. The analysis conducted in this part, not only considers the full squat-to-stand movement period, but also focuses on three predefined phases, being early (the first 0.5 seconds), mid (the middle 1 second) and final (the last 0.5 seconds) of movement segments that roughly corresponds to standing up, tuning balance, and balanced pose stages. As a single successful episode takes 2 seconds, and the data sampling is performed at 50Hz, a successful training episode generates 100 data points. Hence the defined phases of early, mid and final correspond to the neural activity vectors with sizes 25, 50 and 25 respectively. In case of early termination, the neural activity is taken as zero after the failure, and the learning update is performed with a shorter episode length (i.e., H is adjusted accordingly in Equations 5 & 6). To assess the number of distinct neural response patterns in the policy network, the number of clusters are estimated by using X-means algorithm [23] applied to the aforementioned activity vectors.

The analysis conducted after each 1,000 training episodes was used to obtain the evolution of the number of distinct neural activity patterns as a function of training time. All X-means clustering applications were repeated 10 times to assess the variation due to the stochasticity in the X-means algorithm. The X-means meta-parameters of minimum and maximum cluster sizes were set to 3 and 16 respectively. Once the number of clusters was estimated for each learning episode, the neural activity patterns corresponding to the cluster means were obtained by using standard K-means to detect potential abstract representations formed.

3) Response to environmental change

In this analysis, the response of the network to an environmental change with no additional learning was assessed. Like the earlier analysis, the system was trained for 20K episodes to learn the squat-to-stand movement under perturbation. After the learning task was finished, the perturbation force was slightly changed by reducing the perturbation coefficient from C=300 to C=290 (see Method Section A), which was sufficient to make the robot fall down. The response of the network to the aforementioned change in environment was analyzed according to the phases defined in the previous section.

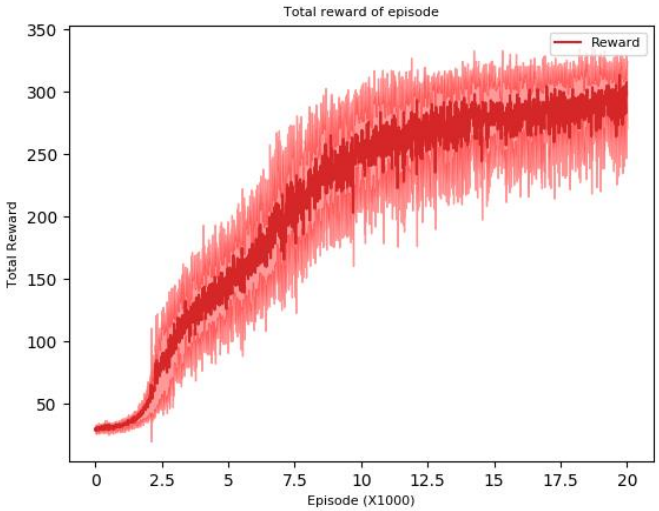

Fig. 2. Mean total reward collected as a function of the number of learning episodes, over 20 repeats are shown. The shades around the mean indicates standard deviation.

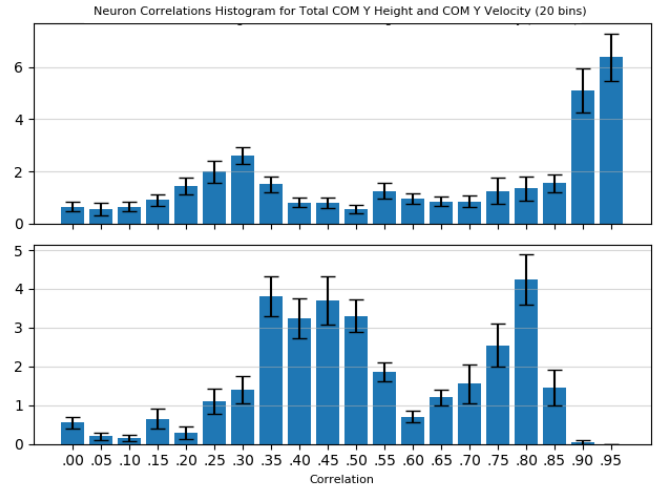

Fig. 3. The mean correlations of the robot's vertical COM position (upper panel) and velocity (lower panel) with the neural activities of the policy network are shown as histograms (y-axis indicates the number of neurons). Training and testing were repeated 20 times. The resulting standard errors also are shown superimposed on the mean bars.

IV. Results

A. Results: Neural coding of the robot state

To make sure that the results related to neural responses are not due to a peculiar reinforcement learning session, we first assessed the total reward collection regime by looking at repeated learning trials. To be specific, the mean total reward collected as a function of number of episodes was plotted together with the standard deviation to indicate the variation in learning (Fig. 2). The mean total reward averaged over 20 trials shows a monotonic increase as expected, as the robot could stand-up after each learning session.

Furthermore, the small standard deviation around the mean at any phase of learning suggests that the system learns the task in a similar and consistent fashion for each learning attempt. Thus it may be argued that the results given in the following section are general within the considered task domain, since consistent learning was observed as seen in Fig. 2.

We focused on the vertical (Y) axis COM dynamics in assessing the potential physical robot state representation by the neural activity after learning. The correlation analysis of the neural responses during the squat-to-stand task executions revealed that 6/32 neurons strongly encode COM vertical position (Fig. 3 top panel) indicated by the high mean correlation coefficient ($\rho>0.95$). Notably, 15/32 neurons also strongly correlated ($\rho>0.75$) with the vertical COM position. The small standard errors on the histogram bars indicate the consistency of the robot COM vertical position encoding by a significant subset of policy network neurons.

The number of neurons that strongly encode COM velocity was less with a lower correlation level compared to the position encoding (Fig. 3 bottom). On the average only 8/32 neurons had a strong correlation ($\rho>0.75$) with the COM vertical velocity. On the other hand, a large portion of the neurons (21/32) showed a mid-level ($0.75>\rho>0.30$) correlation, which was not the case for position encoding.

B. Results: Neural population specialization

As learning proceeds, it is putative that policy network neurons will attain certain functions that enable the robot to stand up. The question one might ask is whether a distributed functional property will be attained by all the neurons together, or whether some modularity will emerge. In case of the latter, we can try to understand the modular functional organization by investigating neuron subpopulations that share similar response characteristics, and therefore infer possible representations they may be endowed with by learning. Thus, we first assess the number of response clusters within the policy network.

When the neuron responses are considered for the whole duration of the squat-to-stand-up movement while learning, the number of clusters started at around 10 and converged to 3 as found by the X-means algorithm (see Fig. 5). The same X-means clustering also was repeated when the neuron outputs were constrained to different phases of the movement (i.e., early, mid, final). The early phase usually corresponds to standing up, whereas the mid phase corresponds to balanced posture for the successful trials. The final phase usually corresponds to the time when the robot starts to lose balance or fall. In these specific phases, the converged number of clusters was consistently found to be 5 on average (see Fig. 4). Therefore, we used K=5 for further analysis using K-means as presented next.

Although X-means gives us the number of neural response patterns formed, to see the individual patterns it is necessary to investigate the response profiles of the clusters found. For this, we applied the K-means algorithm at different points of learning progress, which corresponds to the detailed analysis of the blue curve in Fig. 4. To be specific, the clustering was applied after 2.5K, 7.5K, 12.5K and 20K learning episodes had taken place. The resulting mean neural response patterns, (i.e., the cluster means) are given in Fig. 5. As can be seen neural clusters are not very different at the beginning (Fig 5. left-top); specialization starts to appear after around 10K episodes of learning (Fig 5. top-right, bottom-left), and finally a mature robot controller with distinct responses is obtained when the task is successfully learned (Fig 5. bottom-right).

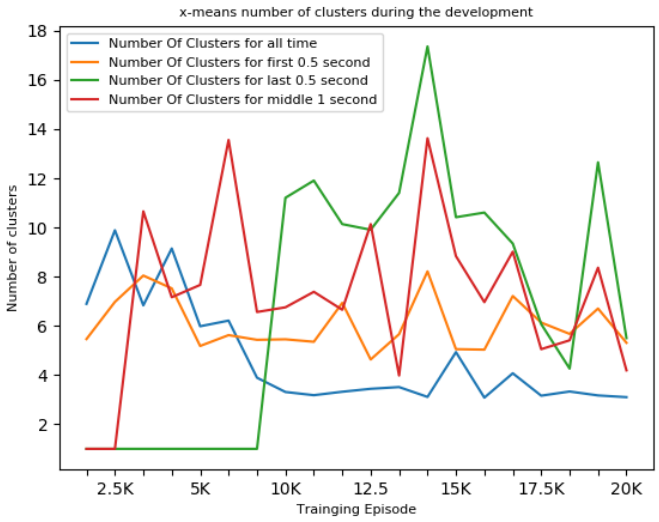

Fig. 4. Neuron output responses that were subject to X-means analysis are shown as a function of training episodes. Blue curve indicates the number of clusters found when the neural responses are taken as corresponding to the full episode of squat to standing. On the other hand, the orange, red and green curves correspond to the specific phases of early (first 0.5s), mid (0.5s-1.5s) and final (1.5s-2s) phases, respectively. (Best seen in colours)

Additional clustering analysis was performed that revealed an organizational relationship among the clusters as a function of simulation time (Fig. 6). During the standing up phase (first 0.5 s), each neuron population had a specific response profile to actuate the robot without a fall (Fig. 6, left-top); in the mid phase (0.5-1 s); the neuron populations act as distinct nonlinear feedback controllers to bring the robot to a standing posture; and finally, in the last phase the neuron populations produce almost constant neural output to counteract gravity.

From a general perspective the constant neural output can represent the concept of 'stability'; whereas the alternating output pattern of the mid-phase may represent the concept of 'keeping balance'.

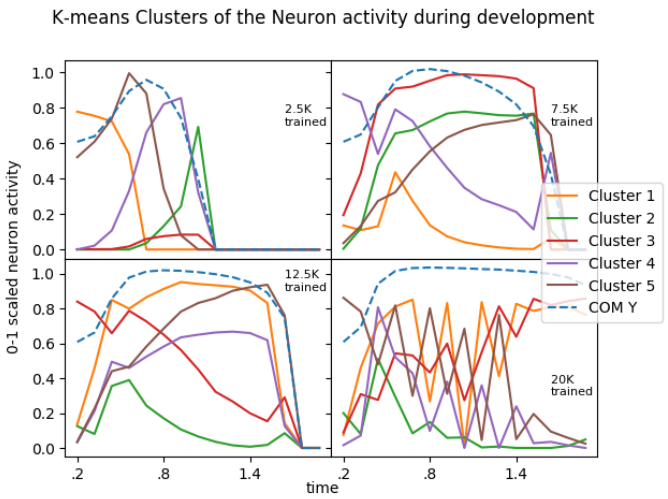

Fig. 5. Neuron outputs are obtained by K-means clustering applied on the whole episode with the original training setup reported on different training stages 2.5K, 7.5K, 12.5K and finally 20Kth episode. Additionally, the dashed curve indicates the vertical COM position (COM Y). (Best seen in colours)

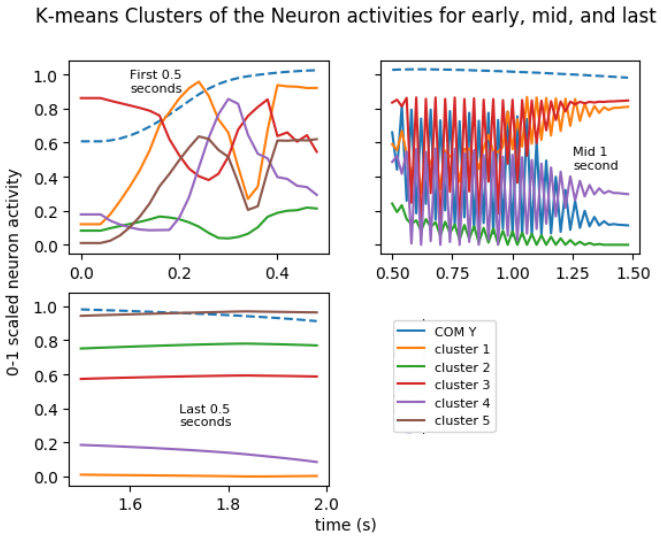

Fig. 6. Neural response means obtained by K-means clustering applied on the neural response vectors corresponding to the early, mid, and final phases of a squat-to-stand movement. All outputs are gathered after 20K training trials. The blue dashed curve superimposed on the plot indicates the height of the robot during the movement. (Best seen in colours)

C. *Result: Response to environmental change*

It is conceivable that a formed neural representation may display emergent patterns when the environment is changed beyond what the network has seen during its learning period. To assess the neural response in the policy network when the environment changed, the perturbation force constant, as previously noted, (see Eq. 3) was reduced from C=300 to C=290 so that the robot could no longer maintain balance with the control policy that was learned with the original perturbation constant. By using the number of clusters found in the previous section, the neuron responses were grouped with the K-means algorithm (K=5) for the specific phases of the early, mid and final phases of task execution. It was found that the neurons show similar patterns in the early and mid-phases to the ones obtained with the original setting, probably due to the fact that the perturbation constant change was small.

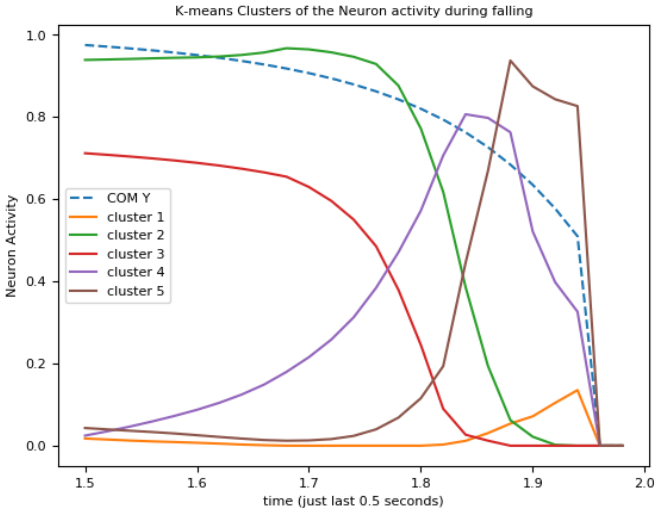

Fig. 7. Neural response means obtained by K-means clustering applied on the neural response vectors corresponding to the final phase of squat-to-stand movement under the changed environment (perturbation constant reduced to 290 from 300). The blue dashed curve superimposed on the plot indicates the COM vertical position of the robot during the movement. (Best seen in colours)

Therefore, further analysis is conducted on the final phase of the movement where the environment change has a dramatic effect of fall vs. no-fall in spite the small change in perturbation. The result of the K-means algorithm applied to the final phase of the movement revealed distinctive cluster means (Fig. 7). The results in Fig. 7 can be summarized as follows; (i) - cluster 1, 13 neurons- these neurons have minor activity during the final phase of the movement, they only show a slight increased activity towards the end of the execution of the task. (ii) - cluster 2, 7 neurons- these neurons decrease their activity in parallel to COM position (COM Y) decrease trend. However, the drop in activity seems predictive and is sharper than the vertical COM position. (iii) - cluster 3, 8 neurons - these neurons decrease their activities in parallel to the COM position decrease trend. Their response is a somewhat time-shifted and scaled-down version of Cluster 2 neurons. (iv) - cluster 4 and 5, 4 neurons- increase their activities while the vertical COM position starts to decrease. These neuron activities, especially cluster 4's, can be predictive of an upcoming fall. Therefore, these neural responses can be used as a symbolic representation indicating a pending fall.

V. CONCLUSION

It is an open question whether the neural circuitry of the brain is explicitly programmed to develop high-level constructs, concepts or symbols, or alternatively such representation may emerge through mere sensorimotor learning. If the latter is true then the evolution of neural circuitry in biological systems could be explained by a hierarchical mechanism, where already existing circuit capabilities such as abstract and symbol-like representations are exploited by 'newer' circuits. To check the plausibility of the latter alternative, we simulated a simple robot that must learn squat-to-stand movements in the face of systematic perturbations via reinforcement learning, where the policy/controller of the robot is represented as a neural network. The analysis on the neural responses of the policy network revealed that (1) parsimonious physical representation of body dynamics (COM dynamics) is, to a large extent, represented in the neural responses (Fig. 5), (2) certain neuron populations in the policy network that learned to implement a stand-up controller form functional units that can be used to represent symbol-like constructs during learning (e.g., 'stability' - Fig. 6), (3) a change in the environment that was not seen before or during learning that may yield discrete representations of the robot self (e.g., 'now falling down!' – Fig. 7). Thus, our study supports the idea that basic sensorimotor learning that allowed earlier biological systems to survive might also have facilitated the formation of symbol-like or high-level representations that were amenable to evolutionary exploitation by higher-level additional circuitry. Therefore, we believe the next step to advance the symbol emergence argument is to show how the formed representations in a policy network can be exploited by additional neural mechanisms for effective planning and execution. This we plan to address next. It would be also interesting to investigate the pros and cons of distributed vs. local representations of high-level concepts for biological and artificial systems. The former allows robustness whereas the latter allow energy economy in neural computations involving the formed representations.

REFERENCES

[1] Medin, D.L. and Ross, B.H., 1992. Cognitive psychology. Harcourt Brace Jovanovich.

[2] Tosches, M.A., Developmental and genetic mechanisms of neural circuit evolution. Developmental Biology, 2017. 431(1): p. 16-25.

[3] Kuipers, B., Feigenbaum, E.A., Hart, P.E. and Nilsson, N.J., 2017. Shakey: From Conception to History. AI Magazine, 38(1), pp.88-103.

[4] A. Newell and H. A. Simon, "Completer Science as Emprical Inquiry: Symbols and Search," Communications of the ACM, vol. 19, no. 3, pp. 113–126, 1976.

[5] Taniguchi, T., Ugur, E., Hoffmann, M., Jamone, L., Nagai, T., Rosman, B., Matsuka, T., Iwahashi, N., Oztop, E., Piater, J. and Wörgötter, F., 2018. Symbol emergence in cognitive developmental systems: a survey. IEEE Transactions on Cognitive and Developmental Systems, 11(4), pp.494-516.

[6] J. Piaget,Play, Dreams and imitation in childhood. Newyork: W. W. Norton, 1962.

[7] Kaplan, B.S., 1963. Symbol Formation: An Organismic-developmental Approach to Language and the Expression of Thought. John Wiley & Sons.

[8] Sutton, R.S., Precup, D. and Singh, S., 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial intelligence, 112(1-2), pp.181-211.

[9] Barto, A.G. and Mahadevan, S., 2003. Recent advances in hierarchical reinforcement learning. Discrete event dynamic systems, 13(1-2), pp.41-77.

[10] Saxe, A.M., Earle, A.C. and Rosman, B., 2017, August. Hierarchy through composition with multitask LMDPs. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3017-3026). JMLR. org.

[11] Sun, R., 2000. Symbol grounding: a new look at an old idea. *Philosophical Psychology*, *13*(2), pp.149-172.

[12] Ugur, E. and Piater, J., 2015, May. Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning. In 2015 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2627-2633). IEEE.

[13] Ugur, E. and Piater, J., 2015, November. Refining discovered symbols with multi-step interaction experience. In 2015 IEEE-RAS 15th International Conference on Humanoid Robots (pp. 1007-1012). IEEE.

[14] Konidaris, G., Kaelbling, L. and Lozano-Perez, T., 2014, June. Constructing symbolic representations for high-level planning. In Twenty-Eighth AAAI Conference on Artificial Intelligence.

[15] Konidaris, G., Kaelbling, L. and Lozano-Perez, T., 2015, June. Symbol acquisition for probabilistic high-level planning. In Twenty-Fourth International Joint Conference on Artificial Intelligence.

[16] Konidaris, G., Kaelbling, L.P. and Lozano-Perez, T., 2018. From skills to symbols: Learning symbolic representations for abstract high-level planning. Journal of Artificial Intelligence Research, 61, pp.215-289.

[17] Stolle, M. and Precup, D., 2002, August. Learning options in reinforcement learning. In International Symposium on abstraction, reformulation, and approximation (pp. 212-223). Springer, Berlin, Heidelberg.

[18] Bacon, P.L., Harb, J. and Precup, D., 2017, February. The option-critic architecture. In Thirty-First AAAI Conference on Artificial Intelligence.

[19] Ranchod, P., Rosman, B. and Konidaris, G., 2015, September. Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 471-477). IEEE.

[20] Babic, J, Oztop, E and Kawato, M, 2016, Human motor adaptation in whole body motion, Scientific Reports, vol. 6, p. 32868

[21] Kunavar T, Camernik J, Kawato M, Oztop E, Babic J (2020.1) Failure as a reinforcement in motor learning. Mechanism of Brain and Mind: 19th Winter Workshop, Rusutsu, Japan

[22] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. and Kavukcuoglu, K., 2016, June. Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp. 1928-1937).

[23] Pelleg, D. and Moore, A.W., 2000, June. X-means: Extending K-means with efficient estimation of the number of clusters. In Icml (Vol. 1, pp. 727-734)