HIGH-LEVEL REPRESENTATIONS THROUGH UNCONSTRAINED SENSORIMOTOR LEARNING

A Thesis

by

Özgür Baran Öztürkçü

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
March 2021

HIGH-LEVEL REPRESENTATIONS THROUGH UNCONSTRAINED SENSORIMOTOR LEARNING

Approved by:

Professor Erhan ÖZTOP, Advisor
Department of Computer Science
Özyeğin University

Assistant Professor Reyhan AYDOĞAN
Department of Computer Science
Özyeğin University

Assistant Professor Emre UĞUR
Department of Computer Engineering
Boğaziçi University

Date Approved: 15 January 2020

To my parents for their endless support

ABSTRACT

Living organisms, in particular mammals are adept at learning complex tasks that may require basic planning such as tool use and manipulation. This ability is manifested by the central nervous system; in particular by the brain, which has evolved from neural circuits that perform low level signal processing for action and perception. Artificial systems can learn to perform low level signal processing for tasks requiring basic perception and action. A particular learning mechanism for this is Reinforcement Learning (RL), in which the sensorimotor experience of an agent that involves a sequence of action decisions and returned 'rewards' from the environment are used to build a 'policy' for a given task. The complex task execution capabilities of higher-level animals may be facilitated by the abstractions formed based on the sensorimotor life cycle of lower-level neural circuits during learning. In a similar vein, the learning experience of an RL agent can be used to form symbol-like structures to enable more complex task executions, by for example explicit planning in symbolic space. The emergence of such high-level structures and their relationship with the environment and the agent dynamics may be revealed by analyzing the neural outputs during learning and actual deployment in different environments. In the literature, there are a range of studies that start by assuming the existence of symbolic structures that 'ground' them onto continuous sensorimotor signals experienced by the agent. In addition, many works aim to facilitate the emergence of symbol-like representations by using specially designed machine learning architectures. In this thesis, we take a different approach and 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 representations.

For our experiments, we use a simulated robot whose sole task is to learn a policy to stand up under velocity dependent force perturbations. The policy is represented and learned by a neural network that is composed of simulated neurons whose responses are recorded during learning and, later, when testing the policy in different environments. The recorded data are then analyzed to detect formation of high-level abstract representations and their relations to the task dynamics. The results indicate that without even explicit design to promote abstract representations, intriguing neural responses emerge via learning, which may serve as the basis of abstract symbol-like representations.

ÖZETÇE

Canlı organizmalar, özellikle memeliler basit planlama gerektiren alet kullanımı ve maniplasyon gibi karmaşık görevleri yerine getirmede ustadırlar. Bu yetenek, merkezi sinir sistemi tarafından, özellikle de aksiyon ve algı için alt seviye sinyal işleme gerçekleştiren nöral devrelerden gelişen beyin tarafından ortaya konmaktadır. Yapay sistemler, temel algılama ve aksiyon alma gerektiren görevler için alt seviye sinyal işlemeyi öğrenebilir. Özellikle pekiştirmeli öğrenme (RL), bir aktörün sensör deneyimi ile bir dizi aksiyon alması ve çevreden dönen "ödülleri" toplayarak belirli bir görevi yerine getirmek için gerekli karar mekanizmasını (policy) oluşturduğu öğrenme şeklidir. Daha gelişmiş hayvanların karmaşık görevleri gerçekleştirme yetenekleri, öğrenme sırasında alt seviye sinir devrelerinin sensör deneyimi döngüsüne dayalı olarak oluşturulan soyutlamalar ile gerçekleştirilir. Benzer bir şekilde bir RL aktörünün öğrenme deneyimi, sembol benzeri yapılar olusturularak, sembol ile planlama gibi karmaşık görevleri yerine getirmesini sağlamak amacı için kullanılabilir. Bu tür yüksek seviyeli yapıların ortaya çıkışı ve bunların çevre ve aktör dinamikleri ile ilişkileri, öğrenme sırasında nöron çıktıları ve muhtemel farklı ortamlarda konuşlandırılmasının analiz edilmesi ile ortaya çıkartılabilir. Literatürde pek çok çalışma bu sembollerin halihazırda var olduğu varsayımı ile başlar ve bu sembolleri sürekli motor sensör çıktıları üzerine oturtur. Diğer bir yaklaşım ise bu sembollerin ortaya çıkmasını sağlamak amacı ile özel olarak dizayn edilmiş sinir ağları kullanmaktır. Bu tez çalışmasında ise dinamik bir görevi öğrenen bir derin pekiştirmeli öğrenme sisteminin üst seviye bir sinirsel yapı oluşturma olasığı incelenmiştir ki bu yapılar sembolik temsillerin öncülleri olarak düşünülebilir. Deneylerimiz için, görevi hız vektörüne bağlı bir bozucu kuvvet altında ayağa kalkmayı öğrenmek olan simüle edilmiş bir robot kullandık. Simule

edilmiş nöronlardan oluşan ayağa kalkma mekanizması bir sinir ağı ile öğrenilebilir ve temsil edilebilir. Deneylerimizde böyle bir sinir ağının çıktıları öğrenme sırasında ve sonrasında farklı ortamlarda kaydedilmiştir. Kaydedilen veriler daha sonra üst seviye soyut temsillerin oluşumu ve bunların görev dinamikleri ile ilişkilerini tespit etmek için incelenmiştir. Sonuçlar, soyut yapılar için özel bir tasarım kullanmadan, sembol benzeri gösterim olarak görülebilecek ilgi çekici sinir çıktılarının oluşabileceğini göstermiştir.

ACKNOWLEDGEMENTS

I want to thank my thesis advisor Dr. Erhan Oztop for his inspiring lessons and guidance through the academic studies. I also thank Dr. Emre Ugur for his guidance in preparing our conference paper and presentation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 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. Symbol grounding problem defined by Harnad [5] as semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads. However symbols are not static and an intelligent agent cannot have a full set of symbols predefined. So, as an extension to symbol grounding Plunkett et al. [6] defines symbol emergence as result of developmental interactions between actor and environment which replaces symbol grounding for intelligent actions. 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 [7, 8, 9]. Symbol formation thus should be facilitated by the formation of intermediate neural representations for abstractions and concepts.

1.1 Related Work

1.1.1 Deep Reinforcement Learning

All RL tasks based on environment, state and action principle. Basically, an actor takes an action a_i in a state S_t which will take a (immediate or delayed) reward R from the environment and transit to a new state S_{t+1}. These definitions actually coming from Markov decision process (MDP). Main goal of the actor is maximizing the reward while the environment consists of the $s \in S$ states which can be continuous or discrete. Also actions which actor chooses has a space A. So, policy is defined as the action selected in each state $\pi(s)$. With the given definitions, optimal policy (generally given with the $\pi^*(s)$) is the policy which maximize the possible total reward from the environment. On the other hand, total reward should be handled with a discounting otherwise actor may choose the longest path available. So discount factor can be expressed as $R = \sum_{t=0}^{N} \gamma^t r_{t+1}$ where N represents number of steps and γ represents discount factor. In general discount factor is between 0 and 1, 0 means myopic actor and 1 means farsighted actor. These definitions carry us in a new set of definitions value function $V_\pi(s)$ and action-value $Q_\pi(s, a)$ function. Value function defines how good is the state for the actor and action-value function gives us the best action in a given state. Also, another important topic is the model of the environment itself. RL methodologies has two different branches in that direction i. model free approaches which model is not fully known ii. model based approaches which model is fully known by the system. The model is the transition function where next state and reward is known while taking any action. In RL literature there are some works aiming to find the optimal value function (or action-value function) and also some

works aiming to find the optimal policy directly. Finding the optimal value function has a major problem as it is indirectly related with the optimal policy as the main goal of the RL is finding the optimum policy. On the other hand, policy-based solutions finds the optimal policy directly.

Main target of value-based approaches are finding the optimal value function $V^*(s)$ or $Q^*(s,a)$ in order to find the optimal policy. Examples of this approach TD(0) [10], Qlearning [11] and Sarsa [12]. As an example of how function approximation works, Sarsa Q function update rule is given in (1). This update rule should be applied after every transition in order to find the optimal Q function.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s,t)) \tag{1}$$

After many studies on complex neural networks, new idea of approximating the Q function from high dimensional input has become possible. In [13] Mnih et al. prove than a RL agent can play an Atari game by approximating the Q function and Deep Q-networks (DQN) born with many other similar studies. Loss function in the paper (2) describes the general idea: γ is the discount factor, θ_i represents parameters of the Q Network at iteration i, θ_i^- is the target network parameters which are updated in fixed number of intervals with the θ_i.

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')} \sim U(D) \left[\left(r + \gamma \underset{a'}{max} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right] \tag{2}$$

On the other hand, there are some studies focused on finding the optimal policy directly, instead of finding the value function or action value function. Advantage of this approach is an optimal policy found as a result of the calculations without finding the value function. In order to gain the optimal policy, policy gradient theorem [14] should be utilized to find the gradients (3).

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H} \nabla_\theta log\pi_\theta(a_t|s_t)\overbrace{\left(\sum_{t=t'}^{H} R(s'_t, a'_t) - b(s_t)\right)}^{Advantage function}\right] \tag{3}$$

Advantage function or return estimate expresses the discounted rewards minus a baseline which can be find by Monte Carlo sampling methods or using value function approximate technique. Advantage function opens a new area in RL methods as actor-critic algorithms. TRPO method offered by [15] Schulman at al. introduce a new idea "limiting the policy update" at each step by using KL divergence. Also, [16] Schulman at al. proved that PPO can also solve more complex problems than TRPO by using the "clipping" which limits the change in the policy $[1-\epsilon, 1-\epsilon]$ which makes learning much simpler and more stable. Also, there are some works aiming to combine DQN and policy gradient as DDPG [17]. In their study Q function and policy learned at the same time which learns the Q function and uses the Q function to gather the policy. So, it can be said that DDPG is both value-based and policy-based approach at the same time.

1.1.2 Symbolic Representations in RL

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 [18] 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 [19]. In [20], 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 [21]. 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 [22, 23]. While these studies have explored symbol emergence in a forward predictive model learning framework, the work of Konidaris et al. [24, 25] 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 [24] and probabilistic [25] 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 [26]. 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. [27] 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. [28] 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. Daniel [29] at al. used options in semi Markov decision process (SMDP) learning by fixing the number of options as a hyper-parameter for continuous real-world tasks. Finally, Ranchod et al. [30] utilized inverse reinforcement learning to discover reusable skills with the segmentation of unstructured trajectories by applying a Bayesian non-parametric approach. In addition Kulkarni et al. [31] proved "intrinsic behaviors" i.e options improves hierarchical deep RL in a complex discrete stochastic decision process with stochastic transitions.

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.

1.2 Contribution of the Work

Remarkable contribution of this work is a step to understand how continuous sensorimotor signals reflect into the agent's neural network. As humans or animals are learning by using the symbols in their mind, an artificial system we should find a way to understand the emergence mechanism of the symbols and categorize the symbols in order to achieve multi dimensional high level tasks. There are many studies using symbol like structures in their neural network design but they never tried to reveal the mechanism of symbol like structures emergence.

In the chapter 3 and 4 our study shows that neural network responses are grouping in different clusters in learning, after the learning and even this kind of a specialization

can be seen with the environment change which neural network never experienced before. These groupings show us that a learning experience based on continuous sensorimotor signals generates symbol like structures which can be utilized for better learning experience.

1.3 Thesis Organization

The analysis given in three different sections (2) deep RL details and results on the robot, (3) the specialization of neuron populations that are formed via learning, and the response of the learned squat-to-stand controller network to changes in the environment, (4) the emergence of neural encoding of the physical robot state. 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.

CHAPTER II

PROBLEM STATEMENT AND TASK SETUP

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 an ongoing research on human adaptation to postural perturbations [32, 33]. The critical consideration here is to investigate whether a sensorimotor learning mechanism with no neural network engineering generates neural activities that encode abstract high-level representations, which may be considered precursors of symbolic representations.

2.1 Squat-to-Stand Task and the Simulation Details

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 Figure 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). PyDy package gives us an ability to formalize the complex multi body movements also at Ozyegin University there are some similar studies uses this PyDY package to calculate the robotic body movements. For the RL setup, the task state (4) is defined as the vector of joint angles (∂_i) and angular velocities (φ_i).

$$S = \{\partial_1, \partial_2, \partial_3, \varphi_1, \varphi_2, \varphi_3\} \tag{4}$$

The action parameters (5) of the robot controller are defined as the vector of

torques (T_i) applied to each joint at each simulation time step:

$$\upsilon = \{T_1, T_2, T_3\} \tag{5}$$

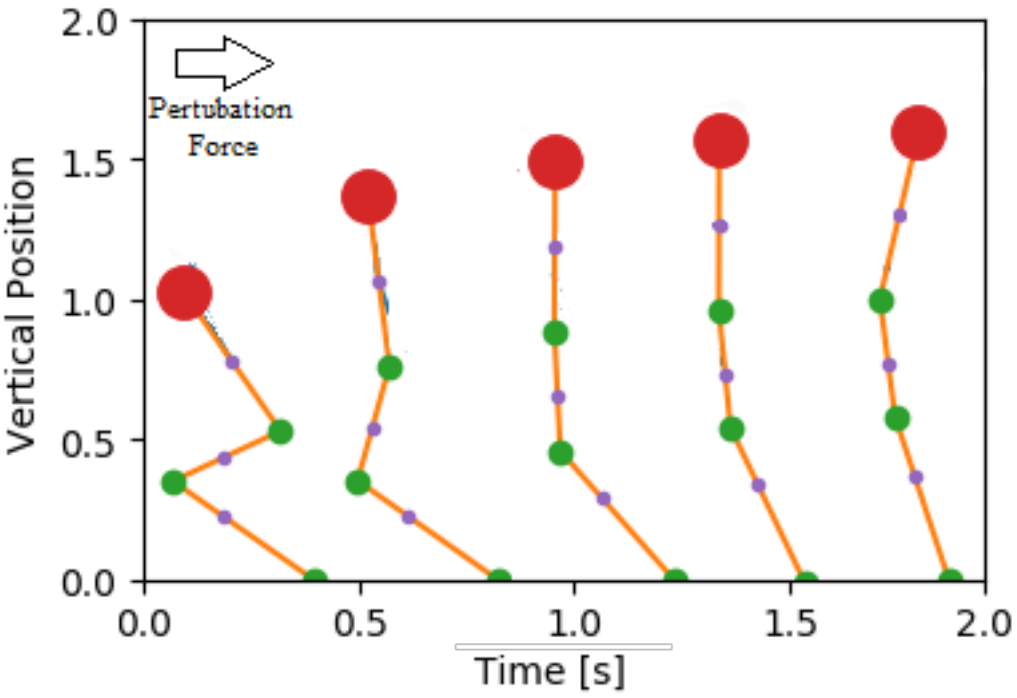

Figure 1: Simulation movements evolving over time after training a neural network successfully.

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 (6):

$$F_{pert} = C \times V_{com} \tag{6}$$

In all simulations, the perturbation constant C is taken as C=300, except for the environment change experiment (see Section 3.2) 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 [32, 33].

Hyper parameters used for learning can be found at Table 1 as a summary. All of these hyper parameters found empirically, learning rate found by trying many different values to gain a effective learning with the less number of episodes. Leaky relu chosen for activation function as normal relu function cannot create negative results but neural network should create some negative values for effective learning. Adam optimizer work really well at this robot's task as other optimizer are not so effective, so also optimizer found empirically. Lastly, initializer of the neurons found empirically with trial of the other initializers.

Parameter Name	**Parameter Value**
Learning rate	0.01
Activation Function	Leaky Relu
Optimizer	Adam Optimizer
Total episode time	2 seconds
Frames per second	50 frame
Initializer for neuron weights	Xavier Initializer

Table 1: Hyper parameters for the robot's neural network

2.2 Reinforcement 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 $r(h) = (\frac{h}{2} + 0.5)^2$ if $h > 0$ otherwise $r(h) = 0$. An episode is terminated when the allowed time of 2 seconds has elapsed or the robot falls ($h < 0.5m$) or the robot hits the joint limits $|\partial_i| \geqslant \pi$ 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 (7), 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).

$$\pi \sim N(\mu, \sigma) \tag{7}$$

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 (8).

$$\nabla U(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \nabla_\theta log \pi_\theta(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R\left(s_k^{(i)}, u_k^{(i)}\right) - V_\phi^\pi\left(s_k^{(i)}\right) \right) \tag{8}$$

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 [34]. 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 (9):

$$\phi_i + 1 \leftarrow argmin_\phi \frac{1}{m} \sum_{i=0}^{m} \left(R - V_\phi^\pi\left(S_i\right)\right)^2 \tag{9}$$

The standard deviation σ controls the exploration- exploitation trade off through action sampling (7). 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 $\sigma(t+1) = \sigma(t) \times 0.999$, and the decay constant is chosen empirically.

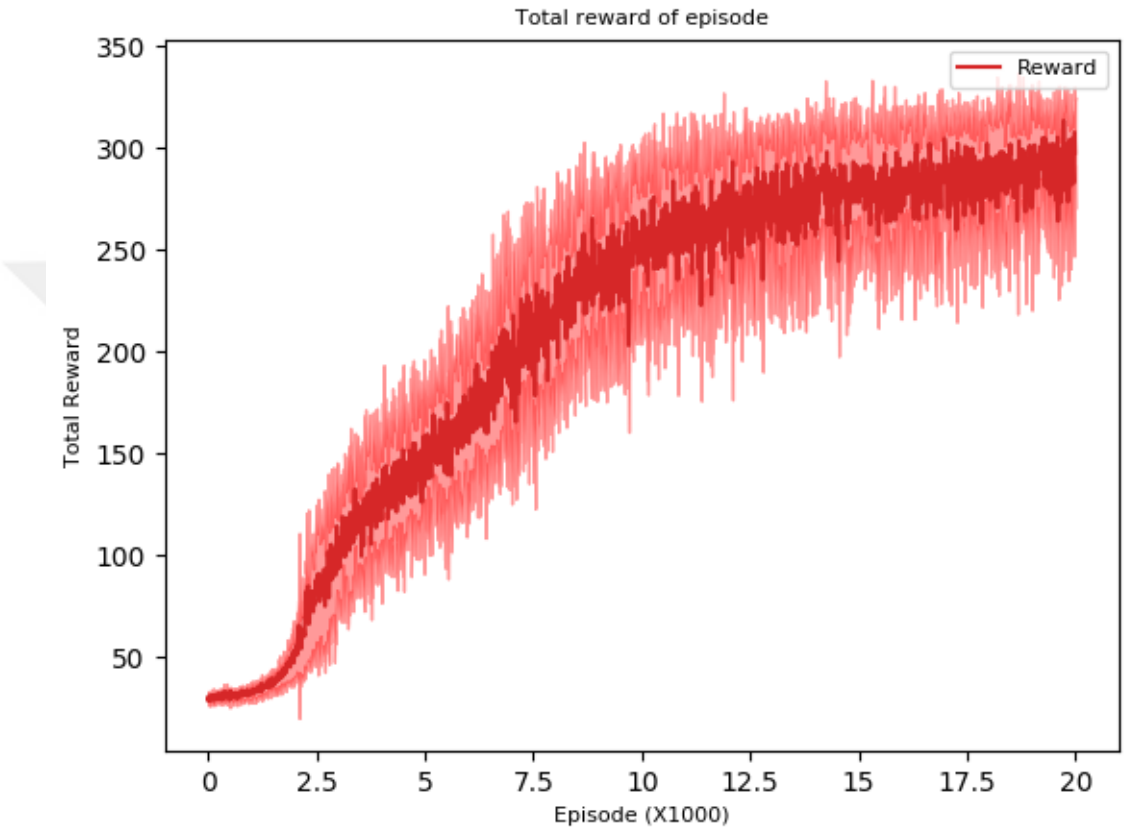

Figure 2: Mean total reward collected as a function of the number of learning episodes, over 20 repeats are shown. The shades around the mean indicates standard deviation.

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.

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 Figure 2.

CHAPTER III

NEURAL SPECIALIZATION VIA LEARNING

3.1 Specialization over Neural Populations

This analysis focused on detecting functional specialization of neurons during the learning process. Functional specialization cannot come up all of a sudden, it should be forming during the learning process. Therefore, analysis started by development process itself as we are waiting these formations starts to emerge during the learning. 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 (8) & (9)). To assess the number of distinct neural response patterns in the policy network, the number of clusters are estimated by using X-means algorithm [35] 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.

X-Means parameters are given in the Table 2 which gives the input and output dimensions and also other hyper parameters. Analysis is done for every 1000 episode starting from episode 1000 to 20000 which makes 20 different X-Means analysis in every development pause operation.

Parameter Name	**Parameter Value**
Initial Cluster Size	3
Maximum Cluster Size	16
Whole duration number of data points	100
First 0.5 seconds number of data points	25
Middle 1 second number of data points	50
Last 0.5 seconds number of data points	25
Number of repeat for X-Means	10

Table 2: Hyper parameters for the X-Means Analysis

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 sub-populations 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.

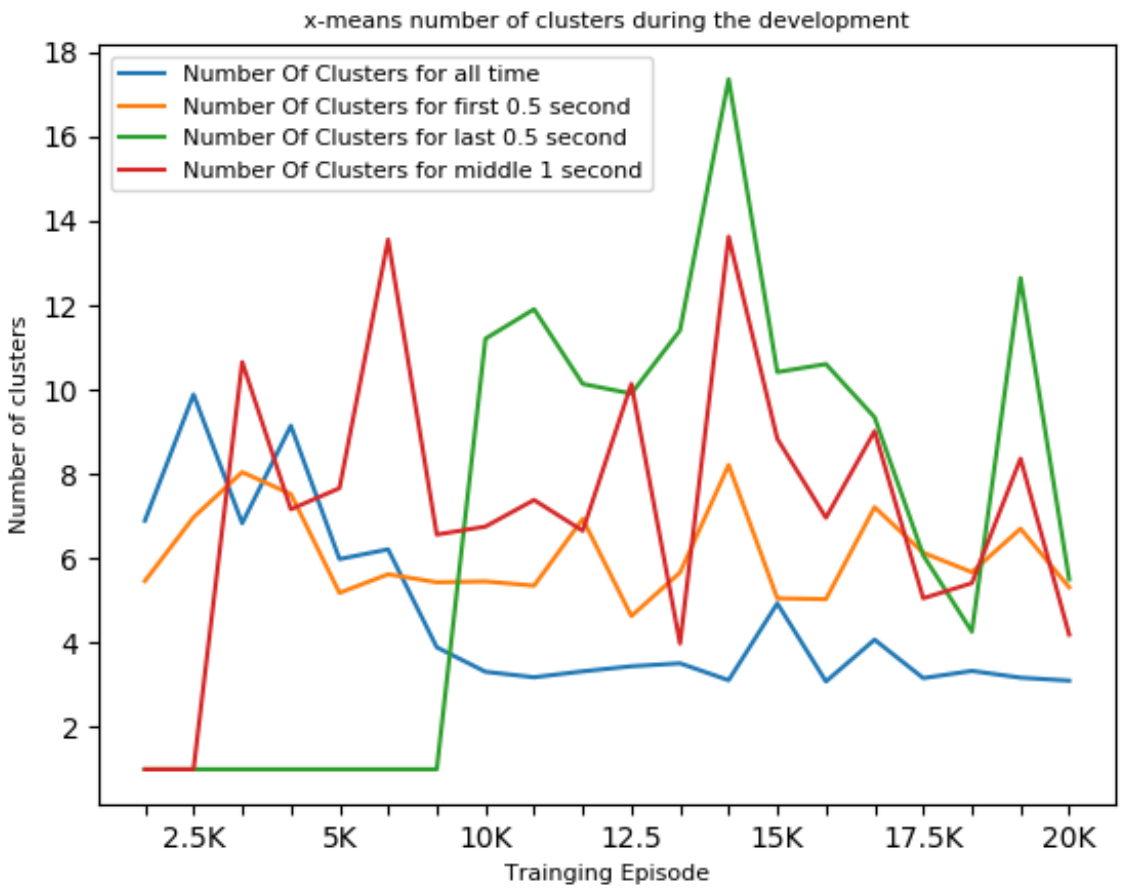

Figure 3: The mean correlations of the robot's vertical COM position (upper panel) and velocity (lower panel) with the neural activities of the policy network are shown as histograms (y-axis indicates the number of neurons). Training and testing were repeated 20 times. The resulting standard errors also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

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 Figure 3). 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. At the first 2.5K episode, number of the clusters for

the mid phase of the movement resulted as 1, considering that robot cannot reach the mid phase. Also till 10K training episodes, robot cannot reach the last 0.5 seconds, final 0.5 seconds, as robot falls (episode finishes) before robot reaches the 0.5 seconds, so number of clusters results as 1. In these specific phases, the converged number of clusters was consistently found to be 5 on average (see Figure 3). Therefore, we used K=5 for further analysis using K-means as presented next.

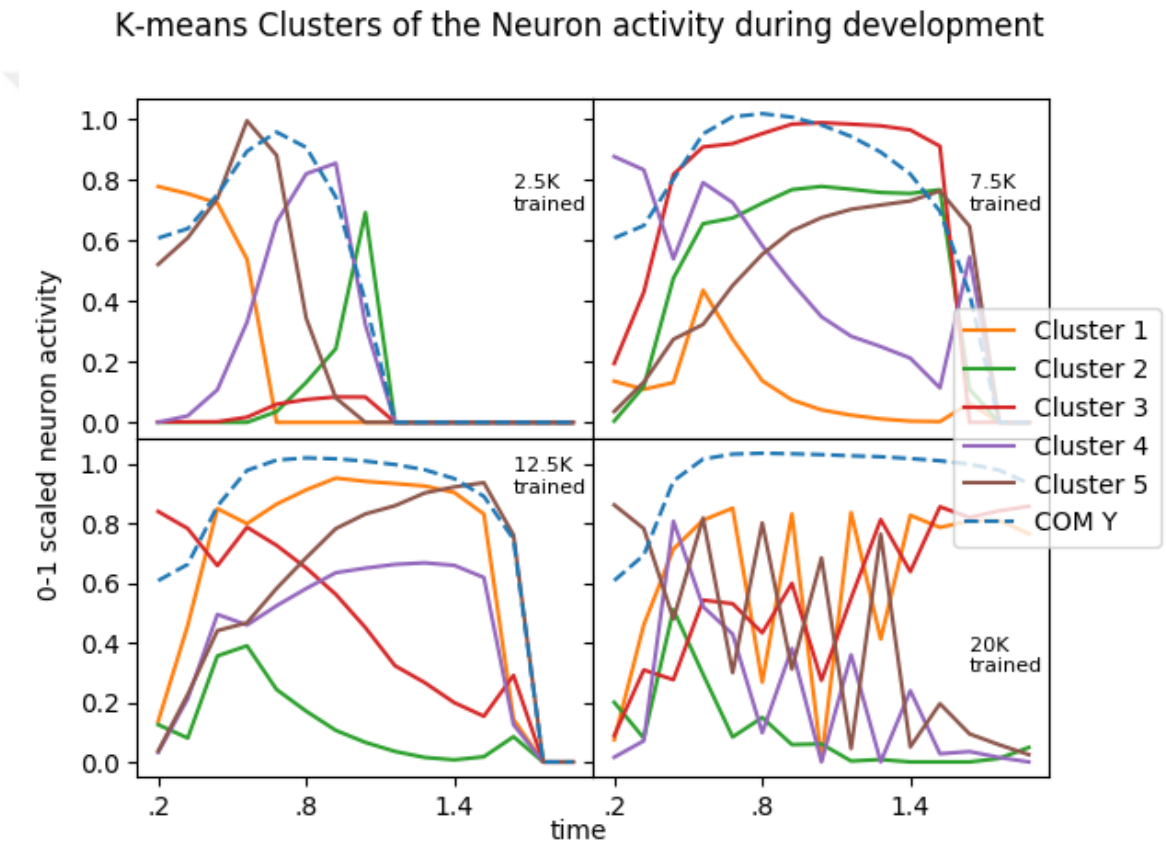

Figure 4: Neuron outputs are obtained by K-means clustering applied on the whole episode with the original training setup reported on different training stages 2.5K, 7.5K, 12.5K and finally 20Kth episode. Additionally, the dashed curve indicates the vertical COM position (COM Y). (Best seen in colours)

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 see Figure 3. To be specific, the clustering was applied after 2.5K, 7.5K, 12.5K and 20K learning episodes had taken place.

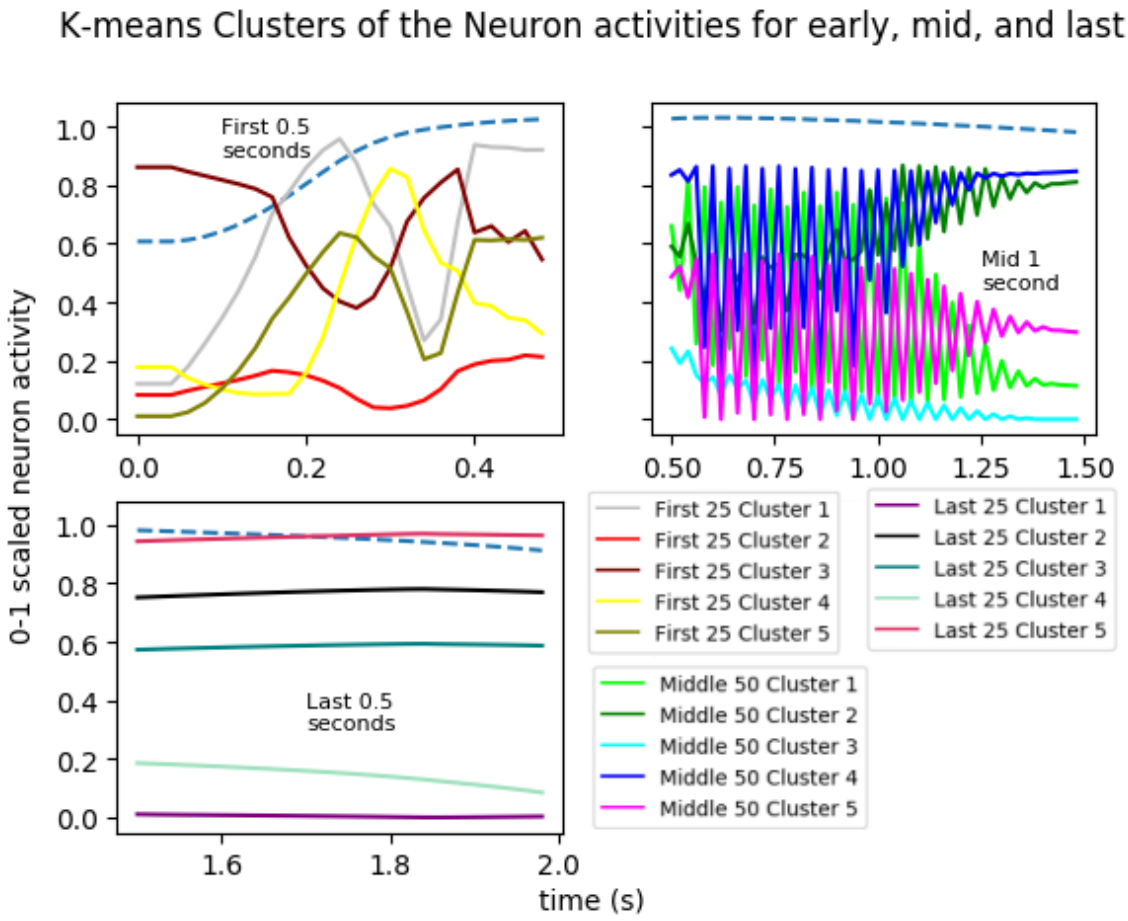

Figure 5: Neural response means obtained by K-means clustering applied on the neural response vectors corresponding to the early, mid, and final phases of a squat-to-stand movement. All outputs are gathered after 20K training trials. The blue dashed curve superimposed on the plot indicates the height of the robot during the movement. (Best seen in colours)

The resulting mean neural response patterns, (i.e., the cluster means) are given in Figure 4. As can be seen neural clusters are not very different at the beginning (Figure 4. left-top); specialization starts to appear after around 10K episodes of learning (Figure 4. top-right, bottom-left), and finally a mature robot controller

with distinct responses is obtained when the task is successfully learned (Figure 4. bottom-right).

Additional clustering analysis was performed that revealed an organizational relationship among the clusters as a function of simulation time (Figure 5). During the standing up phase (first 0.5 s), each neuron population had a specific response profile to actuate the robot without a fall (Figure 5, 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.

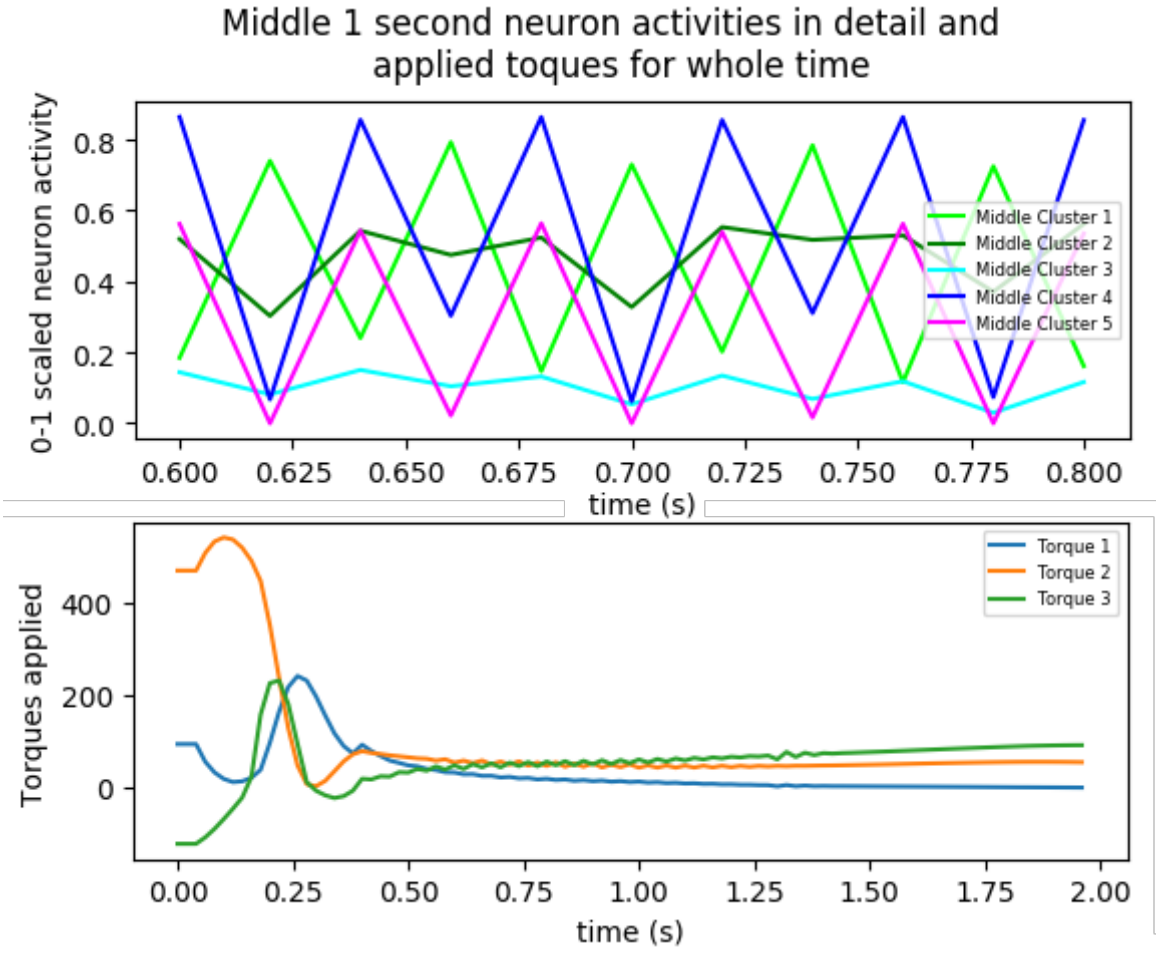

Figure 6: Middle phase neuron output K-means clusters between 0.6 and 0.8 seconds given in the upper panel. Applied torque values for each joint given in the lower panel.

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'. Robot's movements and the torques in mid and final phase is similar as it can be seen on Figure 6 lower panel, on the contrary neural outputs are very different. Middle phases neuron cluster outputs analyzed in Figure 6 upper panel in detail. As it can be seen in Figure 6, some cluster's crests are overlapping other cluster's troughs which cancels each other and makes the torque value produced in the middle phase near constant.

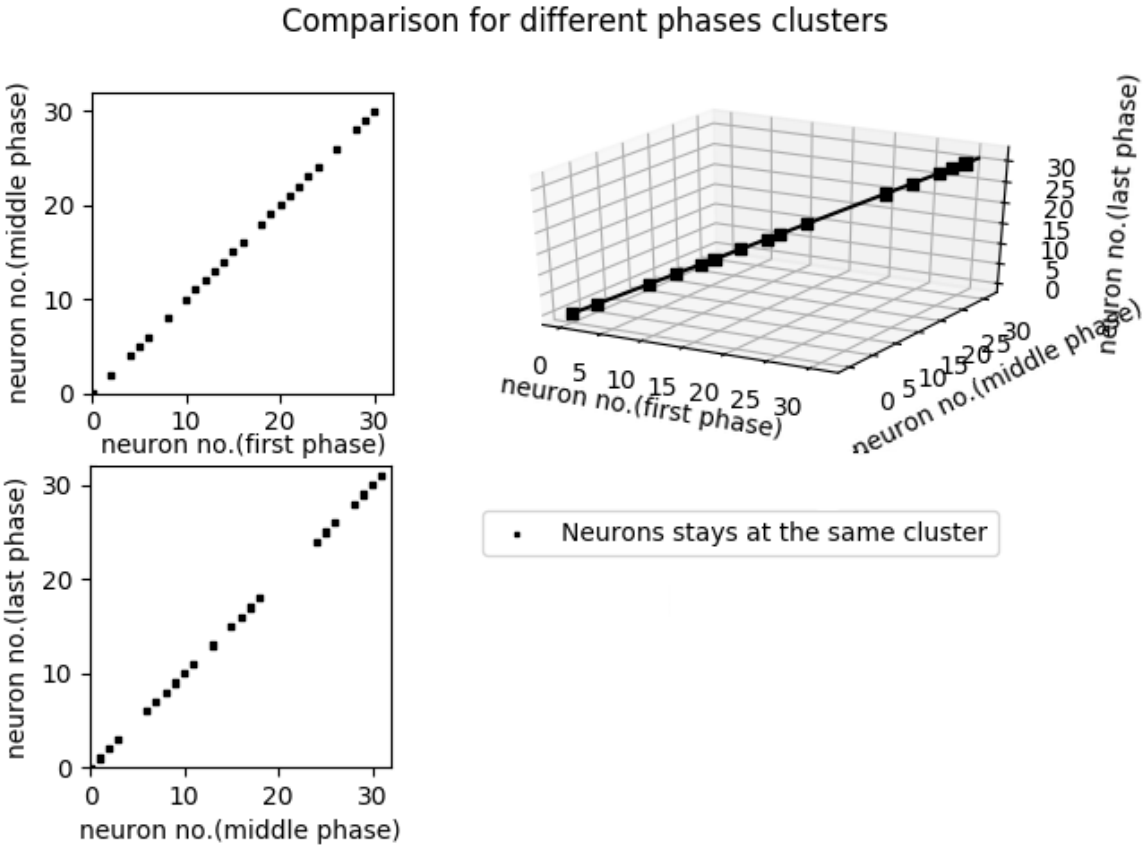

Figure 7: Neuron transition within the clusters of three phases (first, middle and last). Each point in the figure shows the neurons stays in the same cluster within the transition.

Following up the clusters given in the Figure 5, neurons are analyzed to understand that whether clusters in first, middle and last (three phases) are continuous or

		Correlations	
		COM Y	COM Y Velocity
First 25	Cluster 1	0.6637865678	0.4304447801
	Cluster 2	0.2520544412	-0.1214516327
	Cluster 3	-0.5301680901	-0.5850459112
	Cluster 4	0.7187061919	-0.0163198922
	Cluster 5	0.7991715075	0.3419619975
Middle 50	Cluster 1	0.506219085	-0.8131867861
	Cluster 2	-0.7879611842	0.4906160441
	Cluster 3	0.7030738352	-0.3010681074
	Cluster 4	-0.4162287129	0.7053547545
	Cluster 5	-0.0067657097	0.5102015711
Last 25	Cluster 1	0.8447178192	-0.7629440507
	Cluster 2	-0.7121175908	0.6370272593
	Cluster 3	-0.7491272935	0.6794211715
	Cluster 4	0.9990082822	-0.9499041494
	Cluster 5	-0.8195801115	0.7444290889

Table 3: Correlation values between found cluster centers and robot's COM Y value and COM Y velocity.

different than each other. In order to analyze the cluster resemblance, neurons with the each cluster of three phases are matched by using the most similar neuron groups. Later on, neighbor phase's clusters (first-middle and middle-last) are analyzed to see the correlations between phases. Between first phase and middle phase, 24 out of 32 neurons are stays in the same cluster as can be seen in Figure 7 top left panel. On the other hand, 22 out of 32 are continue to function in the same cluster between middle and last phases, Figure 7 lower left panel. Lastly all three phases clustering results are analyzed in Figure 7 top right panel. 15 out of 32 neurons stayed in the same cluster for all first, middle and last phase. The results show that most of the neurons are functioning synergistically as they stay together (i.e. appear in the same cluster) over the defined phases of the movement.

Table 3 shows the correlations between clusters in first, mid, and final phase and COM Y, COM Y velocity values. In first phase clusters, cluster 4 and cluster 5 has a strong correlation with COM Y value; however, no significant correlation between

COM Y velocity and cluster centers can be found. In the middle phase, clusters 2 and 3 has a strong correlation between COM Y and cluster 1 has a strong correlation with COM Y velocity. Lastly, in the last phase correlation between cluster neurons and the cluster centers are strong as both neuron outputs and COM Y values are flat at the final phase.

3.2 Neural Response to Change in Task Dynamics

In this analysis, the response of the network to an environmental change with no additional learning was assessed. We want to asses this response as neural network have to respond this environmental change which never been introduced before. This analysis can reveal some of the hidden specialization such as some neurons may try to resists against that kind of a change. 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 (2.1 Task), which was sufficient to make the robot fall down. The effect of the change found empirically by trying the minimum change which can cause loss of balance. A bigger change is causing irrelevant neuron activities as robot cannot stay at balance posture for a long time or not at all. The response of the network to the aforementioned change in environment was analyzed according to the phases defined in the previous section.

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. (6)) 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.

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 (Figure 8). The resulting pattern fall into four different groups in which has different behaviours regarding the robot activity. The first group which has 13 neurons (cluster 1), has minor activity during the final phase of the movement. They only showed minor activity at the end of the phase. We can say that this group has nearly no effect on the fall. The second group which has 7 neurons (cluster 2), decrease their activity level in parallel to COM Y position decrease trend. However, the drop in activity seems predictive and is sharper than the vertical COM position. So, this group lost it's relationship with COM Y position when the robot starts to fall. The third group (cluster 3) which has 8 neurons decrease their activities in parallel to COM position decrease trend. Their response is somewhat is a time-shifted and scaled down version of Cluster 2 neurons.

Lastly, but importantly forth group which has 4 neurons and cluster 4 and 5 in it, 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 an inevitable fall.

Analysis of the different environments for the learned actor may reveal the relationship between neuron activities and actor dynamics or environment which can be utilized for further development techniques. On the other hand, this result indicate

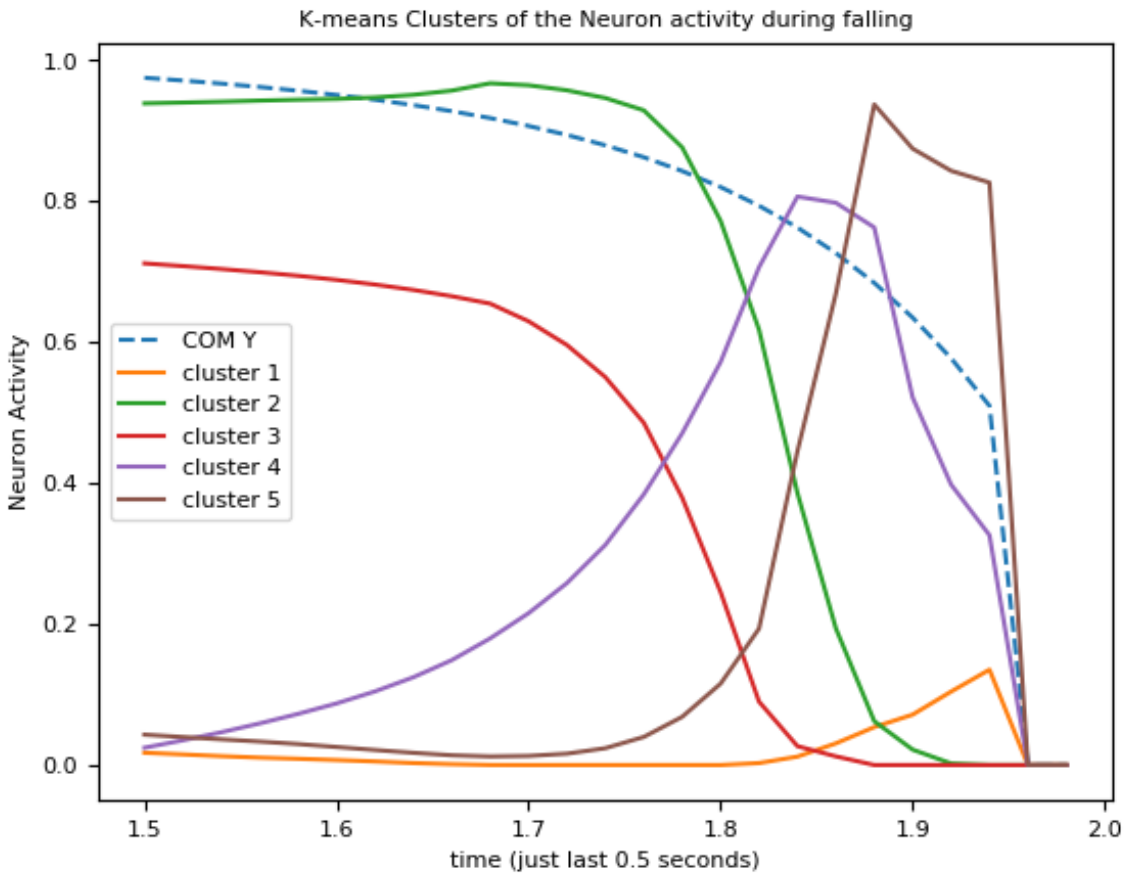

Figure 8: Neural response means obtained by K-means clustering applied on the neural response vectors corresponding to the final phase of squat-to-stand movement under the changed environment (perturbation constant reduced to 290 from 300). The blue dashed curve superimposed on the plot indicates the COM vertical position of the robot during the movement. (Best seen in colours)

that relationships may be emerge in many different ways in different environments which requires deep analysis.

CHAPTER IV

EMERGENT HIGH-LEVEL REPRESENTATIONS

In this analysis, we aim to check whether in the learned neural network a representation has emerged to capture COM dynamics. 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). COM is the most compact representation of the robot and also, COM has no direct relationship with the reward function of the RL task. Therefore, it is interesting to see if neural network has some symbol like representations related with the 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. During the calculation of the correlations, absolute values are used to reveal the relationship between neuron output and COM Y values as negative correlation also implies the strong relationship. 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.

We focused on the vertical (Y) axis COM dynamics (as in in squat-to-stand movements physical work is mainly done along the vertical axis) 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 (Figure 9 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. This strong correlation between COM Y and neuron activities showed that a symbol-like structure can emerge within a neural network without direct relationship with the reward function. 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 (Figure 9 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 the position encoding. COM velocity is represented to a lesser degree compared to COM vertical position. It is not possible to give the exact reason of this trend but it may be speculated that static aspects (i.e. joint configuration) of the task addressed could be more critical to task execution compared to the dynamic aspects, thus require more neurons to represent fatefully. Constraining the squat-to-stand task completion to a shorted time may change this balance. However, this would require longer training for RL to find a stable stand up policy, which is left for another study.

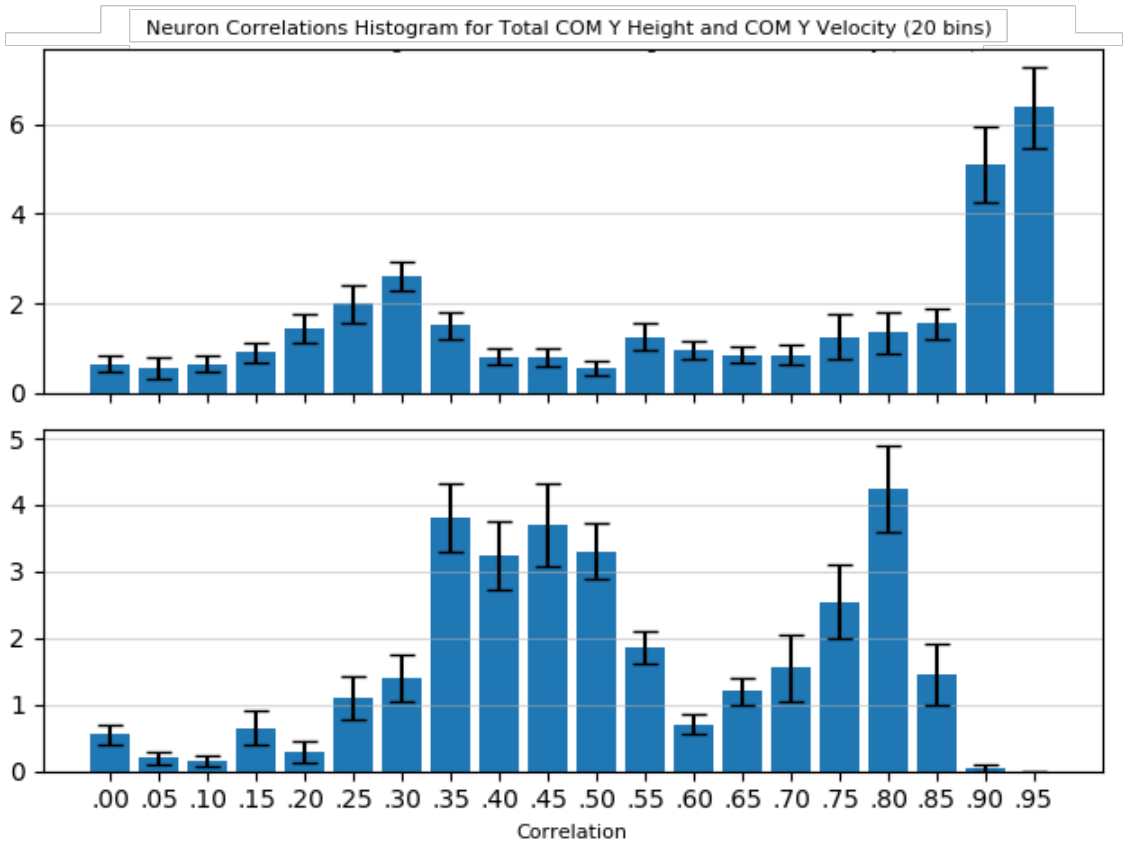

Figure 9: The mean correlations of the robot's vertical COM position (upper panel) and velocity (lower panel) with the neural activities of the policy network are shown as histograms (y-axis indicates the number of neurons). Training and testing were repeated 20 times. The resulting standard errors also are shown superimposed on the mean bars.

CHAPTER V

CONCLUSION

5.1 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 (Figure 4), (2) certain neuron populations in the policy network, which learned to implement a stand-up controller, form functional units that can be used to represent symbol-like constructs during learning (e.g., 'stability' - Figure 5), (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!' – Figure 8). Thus, this 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.

5.2 Future Work

Given the conclusion above, we believe that 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 should be addressed in multi-task learning scenario where the knowledge transfer can be explicitly measured. And other direction to pursue is to consider the effects of task parameters, such as allowed time and maximum torque limits on the representations formed. Finally, it would be very 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.

5.3 Discussion

In this thesis, reinforcement learning methods were used and their utility confirmed. However, high level representations during the development and after the development still have some missing points that should be investigated as to why and then filled in. Many analysis methods including unsupervised learning, correlations analysis, and behaviour mapping techniques have been used to reveal the representation formation.

Clustering neuron responses show that neuron activities have a pattern, rather than ordinary values just supporting the task execution. Some groups of neurons have a tight relationship with the simulation properties that are not part of the reward signal, on the other hand some other neurons have a minor effect on simulation properties. This indicates that in a continuous domain neurons learn some discrete representations.

Bibliography

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

[2] M. A. Tosches, "Developmental and genetic mechanisms of neural circuit evolution," *Developmental biology*, vol. 431, no. 1, pp. 16–25, 2017.

[3] B. Kuipers, E. A. Feigenbaum, P. E. Hart, and N. J. Nilsson, "Shakey: from conception to history," *Ai Magazine*, vol. 38, no. 1, pp. 88–103, 2017.

[4] A. Newel and H. A. Simon, "Completer science asemp rical inquiry: Symbols and search," *Communications*, 1976.

[5] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

[6] K. Plunkett, C. Sinha, M. F. Møller, and O. Strandsby, "Symbol grounding or the emergence of symbols? vocabulary growth in children and a connectionist net," *Connection Science*, vol. 4, no. 3-4, pp. 293–312, 1992.

[7] T. Taniguchi, E. Ugur, M. Hoffmann, L. Jamone, T. Nagai, B. Rosman, T. Matsuka, N. Iwahashi, E. Oztop, J. Piater, *et al.*, "Symbol emergence in cognitive developmental systems: a survey," *IEEE transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 494–516, 2018.

[8] J. Piaget, "Play, dreams and imitation ii childhood," *Trans, by . Gattegno and FH Hodgson. New York: WW Norton*, 1962.

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

[10] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[11] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[12] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, pp. 1057–1063, 2000.

[15] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, 2015.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[18] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[19] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete event dynamic systems*, vol. 13, no. 1-2, pp. 41–77, 2003.

[20] A. M. Saxe, A. C. Earle, and B. S. Rosman, "Hierarchy through composition with multitask lmdps," 2017.

[21] R. Sun, "Symbol grounding: a new look at an old idea," *Philosophical Psychology*, vol. 13, no. 2, pp. 149–172, 2000.

[22] E. Ugur and J. Piater, "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, 2015.

[23] E. Ugur and J. Piater, "Refining discovered symbols with multi-step interaction experience," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 1007–1012, IEEE, 2015.

[24] G. D. Konidaris, L. P. Kaelbling, and T. Lozano-Perez, "Constructing symbolic representations for high-level planning," 2014.

[25] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez, "Symbol acquisition for probabilistic high-level planning," AAAI Press/International Joint Conferences on Artificial Intelligence, 2015.

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

[27] M. Stolle and D. Precup, "Learning options in reinforcement learning," in *International Symposium on abstraction, reformulation, and approximation*, pp. 212–223, Springer, 2002.

[28] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[29] C. Daniel, H. Van Hoof, J. Peters, and G. Neumann, "Probabilistic inference for determining options in reinforcement learning," *Machine Learning*, vol. 104, no. 2-3, pp. 337–357, 2016.

[30] P. Ranchod, B. Rosman, and G. Konidaris, "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, 2015.

[31] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Advances in neural information processing systems*, pp. 3675–3683, 2016.

[32] J. Babič, E. Oztop, and M. Kawato, "Human motor adaptation in whole body motion," *Scientific reports*, vol. 6, p. 32868, 2016.

[33] T. Kunavar, J. Camernik, M. Kawato, E. Oztop, and J. Babič, "Failure as a reinforcement in motor learning.," Tech. Rep. 1, Mechanism of Brain and Mind: 19th Winter Workshop, Rusutsu, Japan, Jan. 2020.

[34] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, pp. 1928–1937, 2016.

[35] D. Pelleg, A. W. Moore, *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters.," in *Icml*, vol. 1, pp. 727–734, 2000.

VITA

Özgür Baran Öztürkçü graduated from Yildiz Technical University Electronics and Communication Engineering Department in 2000. He worked for several companies from 2000 till now as software engineer, project manager, analyst and software manager. Some of the companies he worked for, Toyota, Alcatel-Lucent, Turkcell and Nokia. After the long professional career, he decided to learn artificial intelligence and started master's degree at Ozyegin University.