

Performance Evaluation and Analysis of SENMP in Robotics Experiments

Janne Haverinen

Department of Electrical and Information Engineering

University of Oulu

Oulu, Finland

e-mail: janne.haverinen@oulu.fi

Abstract—The idea of Stochastic Evolutionary Neuron Migration Process (SENMP) is to use artificial evolution process to arrange spatially interacting computational entities, *i.e.* artificial neurons, into a pattern in 2-space so that desired behavior or dynamics emerges within the pattern. In this paper, we analyze the role of space in regard to SENMP performance using the well known double pole balancing problem as a test case. We also study the effect of environmental change to the adaptation process during a robot navigation experiment. This analysis suggests that synaptic scaling like dynamics, resembling inverted Hebbian rule, can emerge in the stochastic pattern formation process between the laterally interacting computational entities.

Index Terms—SENMP, neural network, lateral interaction, dynamic system, pattern formation

I. INTRODUCTION

The Stochastic Evolutionary Neuron Migration Process (SENMP) [6] is a simple experimental tool to study if, and how, artificial neurons, or computational elements in general, using a lateral interaction scheme can be arranged into spatial patterns (see Fig. 1) that exhibit useful dynamics and behavior in regard to some utility measure. SENMP is purely a selectionist approach, relying exclusively on interactions between the evolved system and its environment. The migration of neurons is implemented in SENMP by using an Evolutionary Algorithm (EA) and by utilizing aspects of Simulated Annealing (SA) [9] in the design of the lateral interaction scheme.

The idea of using 2-D geometry to evolve neural networks is not new, and has been studied earlier by at least [3], [7], [8], [10], [11]. However, SENMP combines an EA and ideas from SA in a novel way which allows SENMP to utilize effectively both recombination and mutation. As the mutation rate is implicitly controlled by the diameter of the neuron distribution, it allows SENMP to stabilize without using mechanisms like mutation locking [8]. In addition, when the ‘temperature’ of the system is low the system follows approximately continuous time formalism which provides an opportunity to observe, in time, the behavior of the adapting system under environmental changes.

In this paper, we evaluate the SENMP performance with different parameter setups using the well known double pole balancing problem [6], [13] as a test case. SENMP is here also investigated for possible evidence of Hebbian-like rules

[1] when a robot adapts to a new environment. The hypothesis adopted was: if a Darwinian mechanism plays a role in the adaptation of a real nervous system, and Hebbian-like associative rules can be observed in those systems then, in theory, it could be possible to observe similar phenomena in an artificial counterpart. Furthermore, the exact mechanisms responsible for learning in biological nervous systems are still unknown, and theories have been proposed which suggest that adaptation in the brain could be a Darwinian process, *e.g.* [4].

II. SENMP

A. The Encoding Scheme

The definition of the neighborhood function used in SENMP is given in Eq. 1, which describes the efficacy of the connection between neurons j and k . In Eq. 1 θ_i and θ_j are the phase terms of the connected neurons, d_{kj} is the normalized Euclidean distance of neurons in 2-space, σ_j^2 represents the region of influence of the neuron j , ϕ_k is the feedback factor of neuron k , and $\Theta(\phi_k)$ is the feedback function of a neuron. The neighborhood function is responsible for defining the absolute physical limits for the efficacy of any connection departing from a neuron.

$$w_{kj} = \Theta(\phi_k) \sin(\theta_k - \theta_j) e^{-d_{kj}^2/2\sigma_j^2} \quad (1)$$

According Eq. 1, connection weights are implicitly determined by the spatial distribution of neurons and their phases. The definition of normalized distance d_{kj} between neurons j and k is given in Eq. 2 where $p_i = [x_i y_i]^T$ represents a position of the neuron i in 2-space.

$$d_{kj} = \frac{\|p_k - p_j\|}{\max_{l \neq m} (\|p_l - p_m\|)} \quad (2)$$

In order to break the symmetry of the neighborhood function and to establish both excitatory and inhibitory connections for the network, the variable called phase is introduced for each neurons’ state. The rationale behind phase, θ , is to provide a way of making neurons selectively more sensitive to some subset of neurons than to others and to create orthogonal groups of neurons, *i.e.* sets of neurons that do not interact with each other.

The term $\sin(\theta_k - \theta_j)$ in Eq. 1 is used for breaking the symmetry of the neighborhood function and for providing

different connection types, i.e. excitatory and inhibitory, using only one variable for each neuron. Furthermore, the term has an important role in defining the network topology as it makes a neuron selectively more sensitive to some subset of neurons in the network than to others.

The rationale behind ϕ is to tune the feedback received from the surrounding neurons. More precisely, the outputs of the neurons propagating through the recurrent connections are multiplied by an asymmetric and bounded function $\Theta(\phi_k)$ in the target neuron k . Prior to SENMP all feedback factors are reset to zero and correspondingly $\Theta(\phi_k) = 0$. This means that initial neural networks are feedforward networks and recurrent connections evolve under the selection pressure during the migration process as the feedback factors of the neurons get non-zero values when subjected to the random noise. The definition of $\Theta(\phi_k)$ in Eq. 1 depends on whether the connection from neuron j to neuron k is a forward connection, i.e. a connection from the lower layer, or a recurrent connection, i.e. a connection from the upper or within the same layer: $\Theta(\phi_k) = \tanh(\phi_k)$ is used for recurrent connections enabling both positive and negative feedback, and $\Theta(\phi_k) = 1$ is used for forward connections.

B. The Evolutionary Algorithm

The stochastic motion of neurons establishing the spatial organization of the neural network ensemble (see Fig. 1) was realized by using an EA. In SENMP, the genotype G_i of the neural network i is a list $G_i = (\vec{g}_k : k = 0, 1, \dots, n - 1)$, where n is the number of neurons and the gene \vec{g}_k of the neuron k is $\vec{g}_k = (x_k, y_k, \theta_k, \phi_k, \tau_k, 2\sigma_k^2)^T$, where x_k , y_k , θ_k , ϕ_k , τ_k , and σ_k^2 are the coordinates in 2-space, phase, feedback factor, time constant, and the region of influence of the neuron k , respectively. Index k specifies whether \vec{g}_k represents an input neuron, a hidden neuron, or an output neuron: $k = [0, l - 1]$ for input, $k = [l, l + m - 1]$ for hidden, and $k = [l + m, n - 1]$ for output neurons, where l and m are the numbers of output, and hidden neurons, respectively. Correspondingly, the number of output neurons is $n - (m + l) > 0$.

SENMP is started by creating a list P of random genotypes $P = (G_n : n = 0, 1, \dots, N - 1)$, where N is the population size, i.e. the number of neural networks in the neural network ensemble. For each neuron, i.e. gene k , the phase θ_k gets a random value between $[-\pi, \pi]$ and the coordinates x_k and y_k , and τ_k get random values between $[-\lambda, \lambda]$, where λ is the maximum amplitude of the random noise ν introduced to the parameters \vec{g}_k of the neuron k by the mutation operator¹. The region of influence σ^2 is initialized with some suitable² real value and the feedback factor ϕ is initialized, in general, to zero for all neurons, meaning no recurrent connections prior to the adaptation process. After the initial neural network

¹ $\lambda = 0.01$ in the experiments presented in this paper

² $\sigma^2 = 0.025$ in the experiments presented in this paper. This value was chosen in order to constrain the region of influence of any neuron approximately to the normalized distance 0.5, in which the Gaussian neighborhood function is close to zero.

ensemble is created, a fitness function $f(G)$ is used to assign a fitness for each neural network G .

In SENMP, the recombination of genes is implemented using a factored sampling [2], [6], [12]. This means that the neural network ensemble is represented with a single network composed of a fixed number of neurons, and each neuron having N (i.e. population size) possible states.

After recombination, i.e. resampling, the mutation operator is applied for all n neurons and their N states in such a way that for all k and i the new mutated state (gene) \vec{g}'_{k_i} is $\vec{g}'_{k_i} = \vec{g}_{k_i} + \vec{\nu}$, where $k = 0, 1, \dots, n - 1$ and $i = 0, 1, \dots, N - 1$, and where $\vec{\nu} = (\nu_x, \nu_y, n_\theta(\nu_\theta), n(\nu_\phi), n(\nu_\tau), \nu_{\sigma^2})^T$, where $\nu_x, \nu_y, \nu_\theta, \nu_\phi$ and ν_τ get random values from the interval $[-\lambda, \lambda]$, ν_{σ^2} gets a random value from the interval $[-0.1\lambda, 0.1\lambda]$, $n_\theta(\nu) = 2\pi\nu/d$, and $n(\nu) = \nu/d$. After the mutation, τ_k and σ_k^2 of the new gene \vec{g}'_k are constrained to be positive real numbers by replacing them with their corresponding absolute values. The reason why ν_{σ^2} is constrained to the interval $[-0.1\lambda, 0.1\lambda]$ is to emphasize the role of local interaction in the model, but at the same time to allow the region of influence of a neuron to vary in some limited degree in the long run.

The purpose of the normalization functions $n_\theta(\nu)$ and $n(\nu)$ is to control the short term variances of the parameters for whose noise terms the functions are applied³. Normalization decreases the effect of mutation into these variables as the diameter of the neuron distribution increases. In other words, the diameter of the neuron distribution can be seen as the temperature of the system. If normalization is not used for the parameters θ , ϕ , and τ then $n_\theta(\nu) = n(\nu) = \nu$.

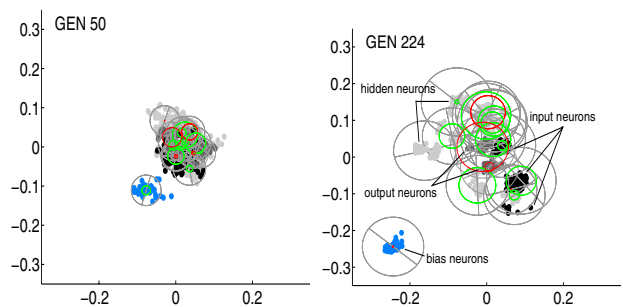


Fig. 1. The evolution of the neuron distribution during a non-Markovian double pole balancing experiment (DPNV): generations 50, and 224. The average σ ($= \sqrt{\sigma^2}$) for each neuron cluster is shown with a circle at the center of gravity of the corresponding cluster of neurons. The mean feedback factor (scaled with σ) is shown with a smaller circle at the center of a neuron cluster. The line drawn through the center of a neuron cluster represents the average θ , i.e. phase of the neurons within the cluster.

III. THE NETWORK MODEL

The neuron model used in the experiments for hidden and output units is given in Equation 3 where τ_k , v_k , φ , I_k , and w_{kj} are the time constant, the activation potential, the activation function, the current sensory input of the neuron

³for convenience this is called normalization of that particular variable in the remaining text.

k , and the connection weight between the neurons j and k , respectively.

$$\tau_k \dot{v}_k = -v_k + \sum_{j \neq k}^n w_{kj} \varphi(v_j) + I_k \quad (3)$$

In the experiments, the input units were linear with inputs normalized into the range $[-1, 1]$, representing the positions of the cart and the poles, for DPNV experiments, and into the range $[0, 1]$, representing the sonar readings, for the navigation experiments. The network model for DPNV experiments were 3-10-2. The activation functions for hidden and output units were hyperbolic tangent functions. At each time step, the most active output unit determined the sign of the constant force (10N) applied to the cart [6]. The network model for the navigation experiments was 11-30-30-4 for the adaptation experiments 1-17 and 11-20-20-4 for the experiments 18-23. The activation functions for hidden and output units were threshold functions. Two output units were used to control each of the two wheels of the robot: the angular velocity of a wheel k was $\omega_k = (y_{k1} - y_{k2})\omega_{max}$, where y_{k1} , y_{k2} , and ω_{max} are the output values of the two neurons controlling the wheel and the maximum angular velocity of the wheel, respectively [5].

IV. SPATIAL DISTRIBUTION OF NEURONS

The spatial distribution of neurons in 2-space has an important role for the function of the neural network in SENMP, as it defines the limits for the connection weights between neurons. However, the spatial distribution is only one of the factors that contribute to the overall behavior of the neural network. Other contributing factors are the phase θ , region of influence σ^2 , feedback factor ϕ , and time constant τ of a neuron.

In order to analyze the role of spatial distribution in the overall behavior of the neural network and adaptability, DPNV (Double Pole - No Velocity) experiments [6], [13] were conducted. The DPNV problem is non-Markovian as the network does not receive the velocity information of the cart and poles, making the task more difficult as the network has to estimate the internal state in lieu of velocity information, which requires recurrent connections.

Each experiment was initialized with a random neuron distribution whose initial radius offset r_o varied between $[0, 0.5]$ with step a size of 0.1. Examples of initial neuron distributions for $r_o = 0.0$ and $r_o = 0.2$ are shown in Figure 2.

For each radius offset r_o , 20 DPNV experiments were conducted. An experiment was considered successful if an acceptable solution, which had to maintain both poles balanced for 30 minutes of simulated time, was found within 1,000 generations. The fitness of the neural network was measured as the number of time steps that both poles remained balanced. A pole was considered balanced between -36 and 36 degrees from vertical.

The results of the experiments are summarized in Table I, which shows that the performance of SENMP degrades as the initial radius of the neuron distribution increases. However, it is not clear that the degradation of the performance is due to the decreased variability of the spatial distribution because, the diameter of the neuron distribution affects also the variability of the other neuron parameters via normalization.

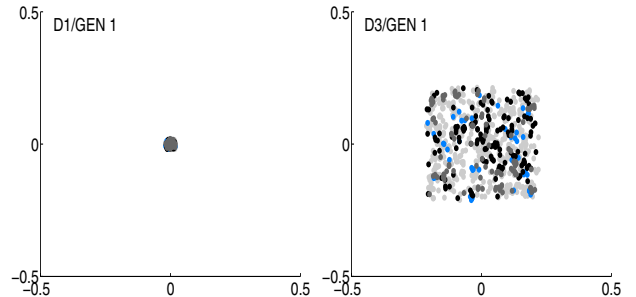


Fig. 2. Examples of initial neuron distributions for $r_o = 0.0$ (D1) and $r_o = 0.2$ (D3)

For reference, DPNV experiments were also conducted by starting the process with $r_o = 0$, which is the typical value of r_o in SENMP. However, in normalization functions $r_o = 0.3$ was used, which means that with regard to neuron parameters θ and ϕ (θ and ϕ are the only neuron parameters which are being normalized in the DPNV experiments presented here) the diameter of the initial neuron distribution is approximately 0.85, but with regard to spatial distribution of neurons, the evolution starts from a dense cloud whose diameter is approximately 2λ (corresponds D1 in Figure 2). By comparing the results shown in Table I and the results gained from the reference experiments it is possible, to some extent, to see if the spatial distribution of neurons alone contributes to the adaptation process. If it does, the variability, *i.e.* the small initial diameter of the neuron distribution at the early stages of SENMP should have a favorable effect on adaptability.

Figure 3 shows the results of two sets, F1 and F2, of 20 DPNV experiments in which the variability of the initial neuron distribution is altered while the variability of the normalization functions are kept approximately equal in both sets of experiments. The results clearly show that the variability of the spatial distribution of neurons is a critical factor in SENMP. In F1, in which the variability of the neuron distribution is constrained by applying $r_o = 0.3$, only 14 out of 20 experiments end successfully. Instead in F2, in which the variability of the neuron distribution is not constrained ($r_o = 0$), 19 out of 20 experiments were successful.

V. NORMALIZATION OF NEURON PARAMETERS

Normalization of neuron parameters including phase θ , feedback factor ϕ , and time constant τ provides a way of controlling the short term variances of these parameters as a function of the diameter of the evolving neuron distribution.

The effect of normalization of neuron parameters was studied by conducting DPV (Double Pole with Velocity)

TABLE I

Radius offset r_o	Solved (out of 20)	Solved %
0.0	20	100
0.1	20	100
0.2	17	85
0.3	14	70
0.4	10	50
0.5	8	40

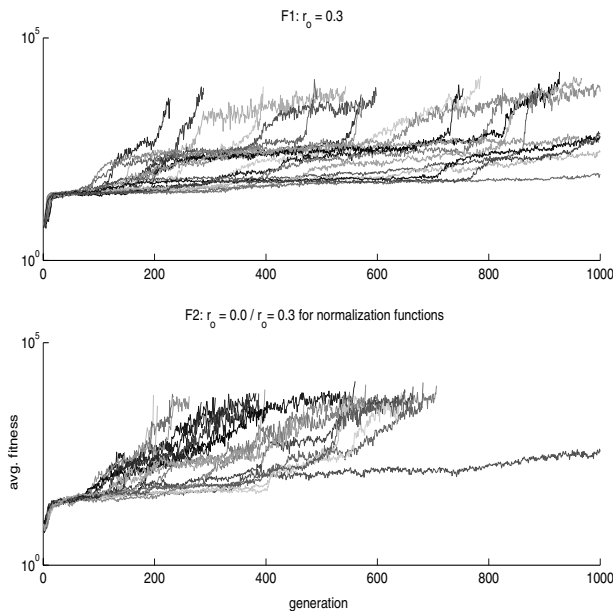


Fig. 3. The effect of changing the variability of the spatial distribution of neurons. F1: The evolution of the average fitness of 20 DPNV experiments initialized with neuron distributions whose radius offset was 0.3. F2: DPNV experiments initialized with neuron distributions whose radius offset was 0.0, corresponding to D1 in Figure 2. However, the radius offset was set to 0.3 for normalization functions in F2.

and DPNV experiments using different normalization setups. Eight sets of 20 experiments were made using different combinations of normalized parameters. Figure 4 shows the results of DPNV experiments. Each plot shows the evolution of the average fitness of 20 experiments for 1,000 generations. Table II summarizes the results for DPV and DPNV experiments. Table II shows that SENMP performs best, *i.e.* requires fewest evaluations in DPV experiments when only θ is normalized. However, in the DPNV experiments, the best performance is gained when both θ and ϕ are normalized. The results clearly illustrate the importance of recurrent connections in solving DPNV problem.

VI. REGION OF INFLUENCE

The lateral and local interaction is the fundamental property of SENMP. The region of influence, represented by σ^2 , is the key factor of the lateral interaction scheme. σ^2 is used to set physical constraints on the efficacy of a connection between any two neurons. When $\sigma^2 \rightarrow \infty$ for all neurons, the lateral interaction disappears and the phase terms θ of neurons define the connection weights alone. On the other hand, if $\sigma^2 \rightarrow 0$

TABLE II

Normalized	DPV solved %	DPV generations	DPNV solved %	DPNV generations
()	100	233	10	680
(θ)	100	80	95	534
($\theta; \phi$)	100	165	100	215
($\theta; \phi; \tau$)	95	290	100	226
(ϕ)	90	224	90	383
(τ)	100	181	70	542
($\phi; \tau$)	85	329	85	251
($\theta; \tau$)	100	148	95	370

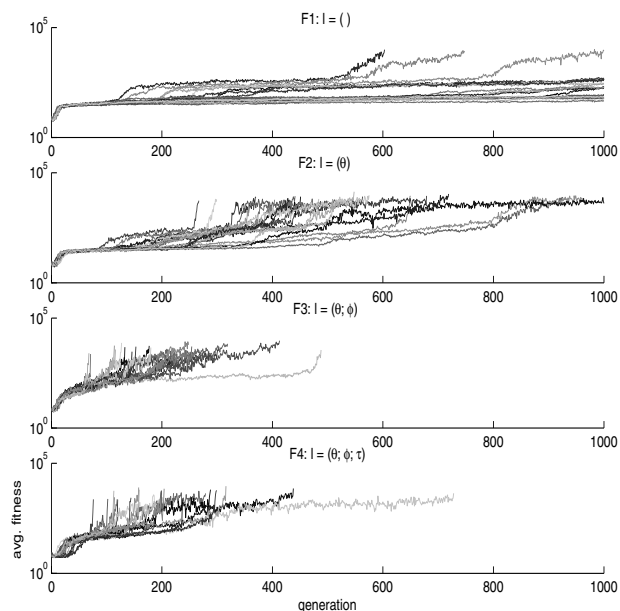


Fig. 4. The effect of normalization of neuron parameters in DPNV experiments. The list l of neuron parameters that are normalized in each set of 20 DPNV experiments are F1: $l = ()$, F2: $l = (\theta)$ F3: $l = (\theta, \phi)$, F4: $l = (\theta, \phi, \tau)$. Based on these plots, the best performance is achieved by using normalization for θ and ϕ .

for all neurons, all connection weights approach zero.

The purpose of this analysis is to study the effect of changing the initial value of σ^2 of neurons on the performance of SENMP. For analysis, eight sets of 20 DPV and DPNV experiments were conducted with different values of initial σ^2 for the neurons. The results are summarized in Table III. The table shows that lateral interaction is indeed an important feature in SENMP and that the performance of SENMP decreases as the initial σ^2 deviates from the value 0.025, which is the value used in the other SENMP experiments presented in this paper. With $\sigma^2 = 0.025$ a solution was found in all 20 DPV/DPNV experiments, and the average number of generations required to find a solution was the smallest: 118 for DPV, and 252 for DPNV.

The results presented in Table III give further evidence on the importance of the spatial distribution of neurons, which was studied in section IV. More precisely, the spatial distribution of neurons matters if, and only if, the lateral and local interaction has a meaningful role in SENMP, which is

TABLE III

Initial σ^2	DPV	DPV	DPNV	DPNV
	solved %	generations	solved %	generations
0.0050	95	322	85	609
0.0125	100	126	100	304
0.0250	100	118	100	252
0.0750	100	148	90	434
0.1250	100	238	25	900
0.1750	100	390	5	998
0.2250	85	566	0	1,000
0.2750	90	537	5	997

exactly what the results suggest. Hence, it can be concluded that the spatial distribution of neurons in 2-space has an important role in the operation of SENMP.

VII. SYNAPTIC PLASTICITY

A change in the environment and re-adaptation provides an opportunity for analyzing the adaptation process. This work concentrates on observing for possible evidence of Hebbian-like dynamics or synaptic scaling from the resulting adaptation process by considering the associative requirement suggested by [1]: “The strength of a neuronal connection is modified as a function of recent changes in the correlation between the activities of the two cells”. The reason for this choice is that it ensures the causality⁴ if the correlated activities of the neurons of the neural network ensemble are measured in two different environments, just before and immediately after the environmental change, while the ensemble is kept fixed during the measurements. This ensures that the possible changes in correlated activities between the neurons have a significant environmental origin.

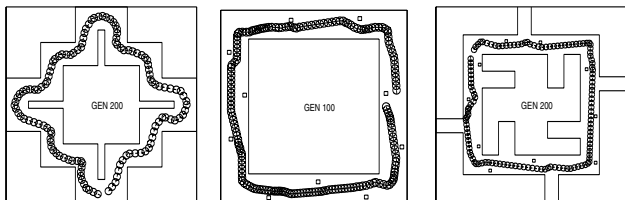


Fig. 5. Three different environments used in the adaptation experiments.

The adaptation experiments were initialized with a neural network ensemble⁵, which was evolved earlier for 200 generations in the environment shown in Figure 5 (left). The fitness function used to assign a score for each individual neural network within the neural network ensemble was $f(s, c) = s/(1 + p_c)$, where s , p_c , and c are the Euclidean distance traveled by the robot from the starting point, the collision

⁴The causality referred here is of course a virtual one, which is caused by the selectionist model. However, it is important to isolate the cause and the effect in order to show that a selectionist approach alone can create an illusion of causality.

⁵The term ‘neural network ensemble’ is used here to refer to a neural network population in order to emphasize the fact that the continuous behavior of a robot is a result of the evaluation sequence of the individual neural networks, and that each neural network controlling the robot inherits the neuronal and environmental state of the previous network.

penalty rate, and the number of collisions, respectively. A typical value of the constant p_c was [0.1, 0.3].

The output activities of the neurons of the 200th generation were recorded and saved, and the prototype network was provided by averaging all the neuron parameters of each neuron cluster of the neural network ensemble: each neuron of the prototype network corresponds to a neuron cluster in the original neural network ensemble. The output activities for a prototype neuron were provided by linking together the recordings of the neurons belonging to the corresponding neuron cluster that formed the prototype neuron. The connections from the continuous value input neurons were excluded from this study because the hidden neurons and the output neurons, used in the adaptation experiments, were integrate-and-fire units, which makes the notion of correlated activity with the continuous output units unclear. Furthermore, only connections between two positively correlated neurons were considered because negative correlation coefficients are more difficult to justify when a causal relationship of firings of two neurons is considered. The standard definition of the correlation coefficient was used to measure the correlation of firings of two neurons, i.e. $r^2 = \frac{ss_{xy}^2}{ss_{xx}ss_{yy}}$, where ss_{xy} , ss_{xx} , and ss_{yy} are sums of squared values of a set of n data points (x_i, y_i) about their respective means

All 23 experiments were started by changing the environment of the robot and initializing SENMP with the neural network ensemble representing the 200th generation in the previous environment. After the initialization of SENMP, the robot was adapted for the new environment for the next 100 generations and the output activities of the neurons of the neural network ensemble for the first generation in the new environment were recorded and saved. Two different initial neural network ensembles were used for the experiments, 1-17, and 18-23 respectively.

For the first four experiments, the robot were transferred to the environment shown in Figure 5 (right). Other experiments were conducted in the environment presented in Figure 5 (middle). After 100 generations, i.e. re-adaptation, in the new environment, the structural changes, i.e. changes in connection weights, were analyzed for possible Hebbian-like dynamics.

Table IV shows the statistics for the 23 adaptation experiments. Column 1 in each table shows the index of the experiment. Column 2 (**pp**) shows the ratio of positive weight changes accompanied by a positive change in correlation coefficient. Column 3 (**pn**) shows the ratio of positive weight changes accompanied by a negative change in the correlation coefficient. Columns 4 and 5 (**nn** and **np**) give the same ratios for the negative weight changes. Columns 6 and 7 show the total number of positive, and negative weight changes, respectively. The absolute value of the weight change must exceed the given weight threshold, which is used to get a list of connection weights whose values show a significant increase/decrease during the process. The total number of connections considered in each experiment is the sum of the numbers in columns 6 and 7.

Table IV shows the statistics for the 23 experiments when the change in correlated activity is compared with the later observed LTP/LTD of the connection weights for 99 generations, *i.e.* 2 to 100. Regardless of the notable exceptions (experiments 1, 8, 14, 15, and 17), there is a clear tendency for inverted Hebbian synaptic plasticity that resembles synaptic stabilization: a positive change in correlation seems to statistically cause LTD of the connection weight more often than LTP, *i.e.* LTD in 60% of the cases, and LTP in 40% of the cases. The opposite is true for the negative change in correlation, *i.e.* LTP in 55% of the cases, and LTD in 46% of the cases. If the five exceptions are excluded the corresponding percentages are: LTD in 62%, and LTP in 32% of the cases for a positive change in correlation, and LTP in 57%, and LTD in 42% of the cases for a negative change in correlation.

One tailed t-tests were performed for the data in Table IV at significance level 0.001 to determine if a sample from the **pp** or **nn** has a mean of less than 0.5, and correspondingly if a sample from the **pn** or **np** has a mean of greater than 0.5. According to the t-tests, this was the case at probability > 99.999%, supporting the synaptic stabilization hypothesis.

TABLE IV

Index	pp	pn	nn	np	No. + ΔW	No. - ΔW
1	0.43	0.57	0.51	0.49	131	154
2	0.44	0.56	0.47	0.53	147	112
3	0.37	0.63	0.38	0.62	149	139
4	0.39	0.61	0.41	0.59	124	139
5	0.46	0.54	0.46	0.54	131	115
6	0.39	0.61	0.43	0.57	120	141
7	0.30	0.70	0.37	0.63	145	112
8	0.41	0.59	0.50	0.50	153	151
9	0.35	0.65	0.42	0.57	155	160
10	0.40	0.60	0.45	0.55	125	108
11	0.35	0.65	0.43	0.57	123	119
12	0.37	0.63	0.48	0.52	128	121
13	0.36	0.64	0.44	0.56	138	113
14	0.50	0.50	0.49	0.51	142	154
15	0.44	0.56	0.56	0.44	110	118
16	0.43	0.57	0.47	0.53	134	107
17	0.50	0.50	0.50	0.50	100	70
18	0.30	0.70	0.30	0.70	82	70
19	0.39	0.61	0.38	0.62	74	73
20	0.39	0.61	0.49	0.51	59	71
21	0.47	0.53	0.48	0.52	88	73
22	0.37	0.63	0.37	0.63	84	63
23	0.31	0.69	0.45	0.55	71	78
Mean	0.40	0.60	0.46	0.55	118	111

VIII. CONCLUSION

Based on the observations, it is argued that Hebbian-like synaptic plasticity and synaptic scaling phenomena can emerge within a selectionist system, and that no explicit local mechanism for synaptic plasticity is a necessity. In SENMP, the necessary variability of synaptic strengths for the operation of the selectionist model is based on the variability of the neuron parameters including the spatial distribution of neurons. The location of a neuron is the primary source

of the synaptic variability in the experiments shown here, as the other parameters are not normalized, *i.e.* they are only subjected to a small component of random noise. If the locations of the neurons are considered to be constant, an artificial analogy can be drawn with real neurons in biological systems, where the variability is now shifted into the geometrically complex dendritic trees of neurons. If the variability in the dynamics of dendritic trees are subjected to selectionist pressure via some reward mechanism(s) which interacts with the electro-chemical dynamics of dendritic trees at some critical time interval, it might be possible to create a mechanism that would explain the observations made of synaptic plasticity *in vivo* without universal and local synaptic rules. However, it could be equally true that, by using a stochastic search, a selectionist system, *e.g.* SENMP, just tries to mimic the operation of synaptic rules that work in the real nervous system. Nevertheless, by investigating selectionist systems, like SENMP, and the way they behave under environmental changes, it might be possible to have experimental evidence and new insights into adaptation mechanisms present in biological nervous systems.

REFERENCES

- [1] E. Ahissar, M. Abeles, M. Ahissar, S. Haidarliu, and E. Vaadia. Hebbian-like functional plasticity in the auditory cortex of the behaving monkey. *Neuropharmacology*, 37(4-5):633–655, Apr-May 1998.
- [2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.
- [3] A. Cangelosi, S. Nolfi, and D. Parisi. Cell division and migration in a 'genotype' for neural networks. *Network: Computation in Neural Systems*, 5:497–515, 1994.
- [4] Gerald M. Edelman. *Neural Darwinism*. Basic Books, Inc., 1987.
- [5] J. Haverinen and J. Rönig. Adaptation through a stochastic evolutionary neuron migration process (SENMP). In *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1008–1013, EPFL, Lausanne, Switzerland, Sep 30 - Oct 4 2002.
- [6] J. Haverinen and J. Rönig. Dynamics from patterns: Creating neural controllers with SENMP. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2630–2635, Sendai International Center, Sendai, Japan, Sep 28 - Oct 2 2004.
- [7] Phil Husbands. Evolving robot behaviours with diffusing gas networks. In P Husbands and J Meyer, editors, *Proceedings of the First European Workshop on Evolutionary Robotics*, pages 123–136. Springer Verlag, 1998.
- [8] N. Jakobi and M. Quinn. Some problems (and a few solutions) for open-ended evolutionary robotics. In P Husbands and J-A Meyer, editors, *Proceedings of the First European Workshop on Evolutionary Robotics: EvoRobot98*, pages 108–122. Springer Verlag, 1998.
- [9] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [10] J. Kodjabachian and J.-A. Meyer. Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9(5):796, September 1998.
- [11] Stefano Nolfi and Domenico Parisi. Evolving artificial neural networks that develop in time. In *European Conference on Artificial Life*, pages 353–367, 1995.
- [12] B. D. Ripley. *Stochastic Simulation*. New York: Wiley, 1987.
- [13] Kenneth O. Stanley and Risto Miikkulainen. Efficient reinforcement learning through evolving neural network topologies. In Morgan Kaufmann, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, 2002.