Autonomous Learning Needs a Second Environmental Feedback Loop

Hazem Toutounji and Frank Pasemann

Abstract Deriving a successful neural control of behavior of autonomous and embodied systems poses a great challenge. The difficulty lies in finding suitable learning mechanisms, and in specifying under what conditions learning becomes necessary. Here, we provide a solution to the second issue in the form of an additional feedback loop that augments the sensorimotor loop in which autonomous systems live. The second feedback loop provides proprioceptive signals, allowing the assessment of behavior through self-monitoring, and accordingly, the control of learning. We show how the behaviors can be defined with the aid of this framework, and we show that, in combination with simple stochastic plasticity mechanisms, behaviors are successfully learned.

Keywords Neuromodulation · Learning · Plasticity · Sensorimotor loop · Auton-omous systems

1 Introduction

Only autonomous systems can learn autonomously. We use *animats* [1–3] as paradigmatic examples of autonomous systems. They are represented by simulated or physical robots. The animat approach is focusing on emergent behaviors and self-organizing processes which generate the life-sustaining interactions of an animat with its dynamically changing environment. It places emphasis on key features of autonomy to which learning is one of the basic properties. In addition, it takes into account the embodied and situated nature of relevant cognitive processes [4].

H. Toutounji (✉) · F. Pasemann
AG Neurocybernetics, Institute of Cognitive Science, University of Osnabrück,
Albrechtstr 28, 49069 Osnabrück, Germany
e-mail: htoutounji@uos.de
url: http://ikw.uni-osnabrueck.de/∼neurokybernetik/

F. Pasemann
e-mail: frank.pasemann@uos.de

K. Madani et al. (eds.), *Computational Intelligence*,
Studies in Computational Intelligence 613,
DOI 10.1007/978-3-319-23392-5_25

An animat is equipped with sensors to perceive the properties of its environment, with proprioceptors to perceive its body's internal (metabolic, physiological) states, with actuators to act in its environment, as well as with a behavioral control that relates its sensory signals and internal states to its actions such that it is able to satisfy its needs for survival.

Survival of a system depends upon some essential internal variables that are monitored and maintained within a given viability zone, i.e. on homeostatic properties [5]. The assumption here is that the primary role of autonomous learning is to establish and to enhance the homeostatic properties of the body. In other words, there will be a close interplay between learning mechanisms and proprioception. In the context of embodied cognition and neuronal plasticity, homeostasis has been examined by e.g., [6–8]. Regulating homeostatic properties is often applied for exploring the system's behavior space, and usually results in a behavior that is not goal-directed [9]. Here, however, goal-directed behavior is considered to be the essential starting point for any kind of learning.

With respect to autonomous learning one is then left with three basic questions: What to learn? When to learn? How to learn? The last question refers to internal mechanisms, such as synaptic plasticity rules [10, 11] and regulatory mechanisms of neuronal excitability [12], which will change dynamical properties of the neural control. But by now, there is no definite general answer or optimal method to generate such a faculty in the neural control of animats. Known learning rules like backpropagation [13] and variants of Hebbian rules [10] refer to specific network structures like feedforward networks or Hopfield networks [14], and to specific problems like pattern recognition or reconstruction. Thus, and since these methods are inadequate for learning a life-sustaining behavior in animats, in this paper, two simple stochastic plasticity mechanisms are deployed for testing the proposed framework.

On the other hand, the first of our questions seems easy to answer: A life-sustaining behavior has to be learned. But again, since environmental conditions and situations are changing frequently, the second of our questions can be rephrased as follows: What signals drive internal mechanisms and corresponding interactions towards a life-sustaining behavior?

A possible answer is to suggest a second environmental feedback loop. This idea can be traced back as far as the work of H. S. Jennings and his studies of lower-order animals [15], and was reformulated by W. R. Ashby in the early days of cybernetics [5]. The second environmental feedback loop is associated with our second question, namely, when an autonomous system has to learn a new behavior. This is assumed to be the case, for instance, when, during the interaction with the environment, there is a situation where "it hurts", or a situation which produces pleasant or unpleasant "feelings". These metabolic or physiological states stimulate the signals from the proprioceptors. For instance, those signals may be generated if joint angles of a legged animat exceed their limits, a motor gets hot, the system bumps into an obstacle, or the "low" state of an energy neuron signals "hunger". In all such cases, proprioceptors mediate corresponding internal, non-neural processes.

To systematically examine these problems, we implemented similar scenarios where proprioceptors are combined with artificial neuromodulators to form

modulator subnetworks. These networks monitor the behavior of the animat, and stimulate the artificial *neuromodulator cells* in response to undesired or beneficial behavior. Stimulated neuromodulator cells then produce neuromodulators to trigger or inhibit plastic changes in the control subnetworks of the animat.

The paper is organized as follows. Section 2 describes the modulator network model with a simple random plasticity method, and an alternative Gaussian walk plasticity method. Section 3 introduces the two simulated robots that are used to test the method, followed by a description of the experiments by which we test the neuromodulation framework in Sect. 4. Finally, the results are presented, and the finding are discussed in Sect. 5, followed by final conclusions on the advantages and limitations of this learning approach.

2 Methods

2.1 Modulated Neural Networks (MNN)

A MNN can be any kind of standard artificial neural networks extended by a *neuromodulator layer*. Some related approaches, though more specialized, are e.g., GasNets [16], Artificial Endocrine Systems [17], and Artificial Hormone Systems [18].

Our variant of a neuromodulator layer provides *neuromodulator cells* (NMCs) that maintain spatial distributions of neuromodulator (NM) concentrations as part of the network. NM produced by a NMC usually diffuses into the surrounding tissue and influences nearby network structures. Due to this spatial nature of NMs, a MNN must provide a spatial representation, i.e. neurons and other network elements (e.g. NMCs) must have a location in space. Each NMC represents a single source for a specific NM type and maintains its own concentration level and distribution within the network. The NM concentration $c(t, x, y)$ at each point in the network at time t is the sum of all locally maintained concentration levels $c_i(t, x, y)$ at that position.

$$c(t, x, y) = \sum_{i=1}^{n} c_i(t, x, y), \qquad x, y \in \mathbb{R} \tag{1}$$

NMCs are always in one of two modes: In *production mode* the cell may increase its modulator concentration, in *reduction mode* it may decrease it. To enter the *production mode*, a NMC must be stimulated for some time, whereas it falls back into *reduction mode* when it *is not* stimulated for a while. The actual model that determines when and how the stimulation happens can be chosen freely for each NMC. The same holds for the production, distribution, diffusion and decay of NMs. Usually, the concentration of the NM and its area of influence increase and decrease depending on the current stimulation and mode. But the characteristics of the diffusion area and gradient are specifics of the chosen models and depend on the MNN variant that is used for an experiment.

The effect of NM exposure on network elements can be various, such as affecting the synaptic plasticity or the function of neurons. Therefore, the actual choice of these effects strongly depends on the experiments and the planned interaction between NMs and network components.

2.2 Linearly-Modulated Neural Networks (LMNN)

The specific variant of the MNN used for the first presented experiments is based on the standard discrete-time neuron model given by

$$o_i(t+1) = \tau_i(\theta_i + \sum_{j=1}^{n} w_{ij}\, o_j(t)) \quad \text{with} \quad i, j = 1, \ldots, n, \tag{2}$$

where $o_i(t)$ is the output of the neuron i at a discrete time step t, w_{ij} is the weight of the synapse from neuron j to neuron i, θ_i is a bias term of neuron i and τ_i a transfer function, for instance $tanh$.

In LMNNs, the stimulation of NMCs follows a simple linear model. The mechanism by which the presented framework guides plasticity is demonstrated schematically in Fig. 1. Each neuromodulator cell (NMC) is attached to a carrier neuron within a modulatory subnetwork (MSN), and is stimulated when the output of this neuron is within a specified range $[S^{min}, S^{max}]$. At each time step t in which the NMC is

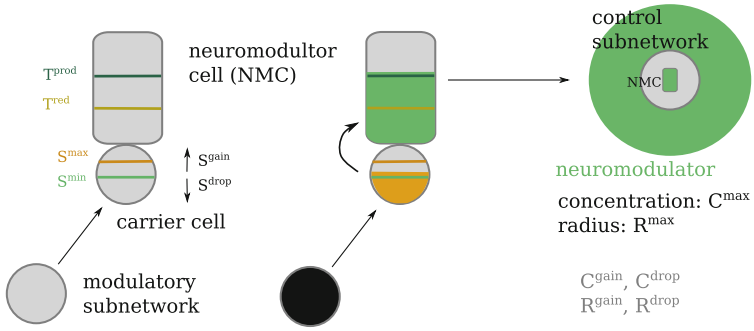

Fig. 1 Schematic representation of Linearly-Modulated Neural Networks. Each neuromodulator cell (NMC) is attached to a carrier neuron within a modulatory subnetwork, and is stimulated when the output of this neuron is within a specified range. At each time step in which the NMC is stimulated, its stimulation level increases, and it decreases If not stimulated. If the stimulation level exceeds a given threshold, the NMC enters the *production mode*. If the level decreases below a second threshold, the NMC re-enters the *reduction mode*. When in production mode, the neuromodulator defuses in time to the surrounding area of a control subnetwork, and initiates plasticity in rates that depend on its concentration at the locale of the synapse

stimulated, its stimulation level s_i increases by a small amount given by parameter S^{gain}. If not stimulated, it decreases by S^{drop}:

$$s_i(t+1) = \begin{cases} min(1, s_i(t) + S_i^{gain}) & \text{if } S_i^{min} \leq o_i(t) \leq S_i^{max} \\ max(0, s_i(t) - S_i^{drop}) & \text{otherwise.} \end{cases} \quad (3)$$

If the stimulation level exceeds a given threshold T^{prod}, the NMC enters the *production mode*. If the level decreases below a second threshold T^{red}, the NMC re-enters the *reduction mode*.

In *production mode*, the modulator concentration c and the radius r of a *circular* diffusion area are increased from 0 to C^{max} and R^{max} respectively. During *reduction mode* both decrease again. The rate of change of the concentration is given by parameters C^{gain} and C^{drop}, that of the radius similarly by R^{gain} and R^{drop}. The following formula shows this for the concentration level c_i; the area radius r_i is defined analogously.

$$c_i(t+1) = \begin{cases} min(C_i^{max}, c_i(t) + C_i^{gain}) & \text{if in } \textit{production mode} \\ & \text{and } \textit{still} \text{ stimulated} \\ max(0, c_i(t) - C_i^{drop}) & \text{if in } \textit{reduction mode} \\ & \text{and } \textit{not} \text{ stimulated} \\ c_i(t) & \text{otherwise.} \end{cases} \quad (4)$$

Due to NM diffusion, learning is triggered in control subnetworks (CSN), according to a particular learning rule whose dynamics depends on the NM concentration. The diffusion mode of each NMC can be chosen, so that the NM concentration is either constant across the diffusion area, or decays according to a linear or nonlinear function of the distance to the NMC. The inhomogeneous distributions are interesting for scenarios with local learning. However, in the shown examples, we will restrict the experiments to a homogeneous, global modulation to demonstrate that successful controllers can develop even in this simple case.

2.3 Plasticity via Modulated Random Search

The synapses of the network react to NM exposure with plastic changes. To demonstrate the viability of using neuromodulation to control the learning process, we choose one of the most simple plasticity methods available: *Random weight changes*. We chose this stochastic plasticity method because it is vastly unbiased and is capable of finding all kinds of network topologies and weight distributions within a given network substrate. Furthermore, the method does not require any heuristics for the choice of the network topology, except that solutions are possible with the given structure.

Table 1 Parameters of a *modulated random search* synapse

Parameter	Description
$Type$	The NM type the synapse is sensitive to
W	Weight change probability
D	Disable/enable probability
W^{min}, W^{max}	Min. and max. weight of the synapse
M	Max. NM sensitivity limit of the synapse

The parameters governing the modulated random search are summarized in Table 1. For a synapse i, the probability of a weight change p_i^w at time t is the product of an intrinsic weight change probability W_i and the current NM concentration $c(t, x, y)$ at the position (x_i, y_i) of the synapse. Hereby, each synapse may limit its sensitivity to NM to a maximal concentration level M_i to prevent too rapid changes when large amounts of overlapping NMs are present.

$$p_i^w(t) = min(M_i,\ c(t, x_i, y_i))\, W_i, \qquad 0 < W_i \lll 1 \tag{5}$$

Stochastic weight changes may occur at any time step, therefore W_i must be very small. If a weight change is triggered, a new weight w_i is randomly chosen from the interval $[W_i^{min}, W_i^{max}]$, given as parameters of the synapse.

In addition to weight changes, synapses can also *disable* and *re-enable* themselves following a similar stochastic process. The probability p_i^d for a transition between the two states during each time step is the product of the modulator concentration $c(t, x, y)$ and the disable probability D_i.

$$p_i^d(t) = min(M_i,\ c(t, x_i, y_i))\, D_i, \qquad 0 \leq D_i < W_i \tag{6}$$

If a transition is triggered, an enabled synapse becomes disabled and vice versa. A disabled synapse is treated as a synapse with weight $w_i = 0$, but its actual weight is preserved until it is enabled again. This mechanism allows for a simple topology search within a given neural substrate.

2.4 Plasticity via Modulated Gaussian Walk

An alternative to using random search as a learning mechanism, we propose a learning mechanism that depends on small changes of synaptic efficacies when neuromodulation is released. We term this learning mechanism the *Modulated Gaussian Walk* (MGW), where, similarly to MRS, the probability of a weight change is the product of an intrinsic weight change probability and the neuromodulator concentration. However, unlike the MRS, no maximal concentration sensitivity is present.

Instead of randomly assigning a value to the synaptic weight in the interval $[W_i^{min}, W_i^{max}]$, the amount of weight change is drawn from a normal distribution with zero-mean and σ^2-variance. As such, the weight changes according to

$$w(t+1) = w(t) + \Delta w \text{ where } \Delta w \sim \mathcal{N}(0, \sigma^2). \tag{7}$$

To assure that the weight remains within its bound (since the term Δw can be infinitely large), sampling the normal distribution is repeated until the resulting weight is within the range.

A mechanism for disabling synapses in also implemented within the MGW learning rule. However, we do not elaborate on this feature here, since later experiments do not make use of it.

3 Robots

Later experiments on linearly-modulated neural networks use robot systems typical for classical neurorobotics problems: a *simple pendulum* (Fig. 2c) and a *differential drive robot* (Fig. 2f). In all cases, motor neurons with an activation range $(-1, +1)$

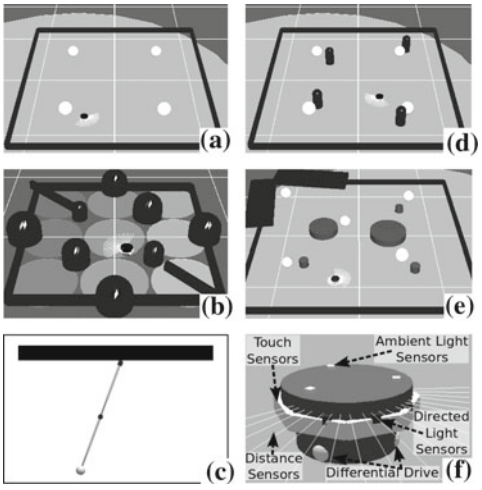

Fig. 2 **a, b, d, e** Environments for learning behavior of a differential drive robot. **a, b, d** The *white spheres* denote possible light source positions. Each light source is bright enough to cover the whole environment. **a** Light-tropism to one of four fixed light sources (`E1`). **b** Obstacle-avoidance with exploration (`E2`). **c** A simple pendulum simulator for learning oscillation to a target angle (`E5`). **d** Light-tropism to one of four fixed light sources, and avoiding nearby obstacles (`E3`). **e** Light-tropism to one of five randomly shifted light sources, and avoiding nearby obstacles, large obstacles, and a narrow corner (`E4`). **f** The differential drive robot with wheels and sensors shown

control the desired velocity of the motors. Negative activations are interpreted as backwards rotation.

The pendulum is equipped with an *angular sensor* for the current angle of the pendulum. The differential drive robot is equipped with *distance sensors* (DS) at the front, eight *touch sensors* (TS), three *ambient light sensors* (ALS) to measure brightness at three equally distributed positions on the robot, and three *directed light sensors* (DLS) in the front of the robot to sense the direction towards light sources (with a maximal viewing angle of ± 90 degrees). For simplicity, light can penetrate obstacles freely. All experiments have been simulated with the NERD Toolkit [19] and can be replicated with material from our supplementary page.

4 Experiments

4.1 Experiments with MRS

To demonstrate the method, five experiments with different complexities have been performed under modulated random search. The experiments are typical for early evolutionary robotics experiments and are still used in many learning scenarios. In all experiments, a robot has to learn a simple task from scratch, starting with a plain, specifically designed LMNN. The predefined MSN of the network produces global neuromodulators for undesired behaviors, while the given CSN defines the topology in which solutions can develop. Neuromodulation is global since all synapses of the CSN are sensitive to NM concentration, and they start out disabled, so that the network connectivity develops together with the synaptic weights, while all synapses of the MSN are insensitive to neuromodulation and are therefore static. As such, each experiment can be defined by a robot, a task, an environment and a control and modulatory subnetworks (MSN and CSN, respectively).

Tasks and Environments. The first experiment (`E1`) is a positive light-tropism task (Fig. 2a). Four light sources are distributed in some distance from the corners of a quadratic arena. At any time, only one light source is switched on. Each light source is bright enough to cover the entire arena. When the robot arrives at that light source, it is switched off and a randomly chosen source is switched on.

The second experiment (`E2`) focuses on an obstacle-avoidance task (Fig. 2b), where the robot has to navigate in a quadratic environment riddled with round objects and narrow corners. The robot also needs to explore its whole environment. Thus, the arena also comprises a number of light sources each emitting a different, homogeneous light that allow the robot to recognize different locations and hence to monitor its own exploration behavior.

As a combination of the previous experiments, `E3` extends the first experiment with four small obstacles placed with a small asymmetric shift near the four light sources (Fig. 2d). Here, the robot has to approach the lights and simultaneously avoid the obstacles next to the light sources.

Table 2 Experimental setups for global neuromodulation

Exp.	τ_{exp}	τ_{temp}	Sensors	NMC modules
`E1`	120	0.5	2 DLS	Light
`E2`	240	5	3 DS	Obst, Drive, Explore
`E3`	720	0.75	2 DS, 2 DLS	Light, Obst
`E4`	720	0.75	2 DS, 2 DLS	Light, Obst
`E5`	240	5	1 AS	2 × TurningAngle

τ_{exp} is the experiment time in simulated minutes, τ_{temp} is the duration in minutes without neuromodulation production to consider a behavior a successful temporary solution

A more difficult variant is experiment `E4`. While the task remains the same, there are now larger obstacles in the middle of the arena and one of the corners is more narrow (Fig. 2e). Furthermore, a fifth light source was added in the center of the arena. All lights are now also randomly moved away from their initial positions every time they get switched on. In contrast to `E3` the robot now gets confronted with many more different light-obstacle combinations, which makes the task quite difficult.

The pendulum experiment (`E5`, Fig. 2c) requires the controller to learn to swing with a specific amplitude between the two target angles $\pm 65°$ with a tolerance of $\pm 5°$. The difficulty is that the motors are too weak to get to the target angles without swinging the pendulum up first.

Control Sub-Networks (CSN). Each CSN includes the necessary sensory and motor neurons, a number of intermediate processing neurons and a bias neuron. The latter allows the bias of neurons to be changed using the same technique as used for other synapses. The network substrates vary over the different experiments, ranging from trivial feedforward networks over a layered network with 4 hidden neurons, to fully connected, recurrent networks with 2, 4 and 6 intermediate neurons. The network configurations for the experiments are summarized in Table 2.

Modulatory Sub-Networks (MSN). Each MSN uses *experiment-specific* network structures to detect undesired behavior based on (sensor) activations to produce neuromodulators when needed. As a reaction to the neuromodulators, synapses of the CSN randomly change and explore different topologies and weight distributions. This has an effect on the behavior and, accordingly, on the NM production in the MSN. Similar to the work by Ashby [5], the system is destabilized when an undesired behavior is detected, leading to continuous changes until the system stabilizes again in a new, valid configuration. In this spirit, six different NMCs are used in the experiments (see Table 2).

The `Obst` cell reacts on the activation of any of the eight force sensors to detect undesired contact with objects. The stimulation is quite rapid so that obstacle contact immediately leads to neuromodulation production to alter the behavior.

The `Drive` cell gets stimulated when the two motor signals are too low, the robot is moving backwards, or the difference of the motors becomes too large, i.e. the robot is moving in narrow circles. Because the desired behavior also may include

moving backwards and especially moving in circles, the stimulation is less rapid and tolerates such movements as long as they do not dominate the behavior.

The `Explore` cell is stimulated when the robot is not entering the detectable locations frequently (the task `E2`). Its associated modulating network classifies the signal of one of the ambient light sensors into the nine detectable locations and integrates these signals to determine the duration of each location not being visited. `Explore` is stimulated if some locations have not been visited for a long time. If a location is entered that has not been visited for a long time, then all integrator neurons for all locations are inhibited, so this potential behavior improvement already leads to a fast decrease in neuromodulator concentration to allow the new configuration to be tested.

The `Light` cell also uses an auxiliary network that interprets the ambient light sensors to detect whether the robot is getting closer to the light. If not, the NMC is stimulated. This achieved by utilizing neural differentiators of the ambient light sensors activity.

The `TurningAngle` cell gets persistently, but slowly stimulated over time. However, if the pendulum changes its swinging direction within the desired angle range, then the NMC stimulation decreases rapidly. The desired angular range can be adjusted independently for each of the two NMCs in the pendulum networks.

Table 2 shows which NMCs, with their corresponding auxiliary networks, are used in each experiment. Figure 3 shows the structure of both the CSN and the MSN for experiments `E2` and `E3`, giving also the neural structures for the six auxiliary sub-networks. The experiments here are restricted to a global modulator release with a uniform concentration levels. Table 3 summarizes the parameter choices for the NMCs used across the experiments.

Experiments Setup. Each experiment has been run with five different network substrates for the CSN: a layered network with 4 intermediate neurons (`L4`) and four fully, recurrently connected networks with 0, 2, 4 and 6 intermediate processing neurons (`N0`-`N6`). Due to the differing number of motors and sensors, the total number of synapses varies. An overview can be found in Table 4. All additional settings of the network, specifically the settings for the plastic synapses and the NMC settings, have been fixed at the values given in Table 3.

Each such learning scenario (experiment + network substrate) has been repeated 50 times with identical settings, each starting with a new CSN composed of disabled synapses with zero weights. Thus, the entire network topology and the synaptic weights had to be learned from scratch within the given network substrate.

4.2 Comparative Experiment with MRS and MGW

We compare the two plasticity mechanism on a task that combines light tropism and obstacle-avoidance with no exploration, i.e. the MSN contains NMCs `Obst`, `Drive`, and `Light`. The chosen CSN of this task is similar to the layered architecture `L4`, but

Table 3 Parameter values for NMCs and the modulated random search in the global modulation experiments

Param.	`Obst`	`Drive`	`Explore`	`Light`	`TurningAngle`	Param.	Synapses
S^{min}, S^{max}	0.9,1.0	0.9,1.0	0.4,1.0	0.9,1.0	0.5,1.0	W	0.0001
S^{gain}, S^{drop}	0.01,0.01	0.001,0.001	0.001,0.01	0.0002,0.0001	0.005,1	D	0.00002
T^{prod}, T^{red}	0.95,0.95	0.95,0.95	0.95,0.95	0.99,0.99	0.95,0.95	W^{min}	-1.5
C^{max}	2	1	1	1	1	W^{max}	1.5
C^{gain}, C^{drop}	0.1,0.1	0.001,0.01	0.001,0.01	0.01,0.1	0.001,1	M	1.0

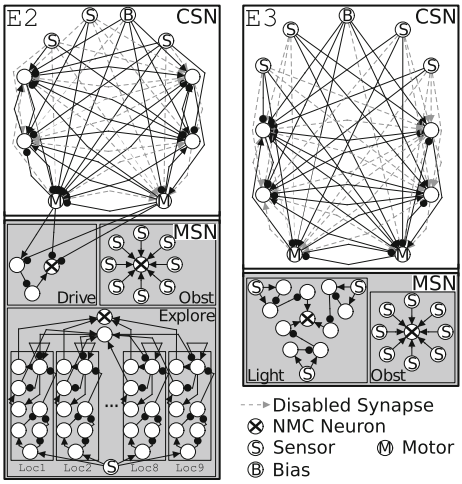

Fig. 3 Two exemplary control subnetworks that result from learning, with their associated modulator subnetworks

Table 4 Number of plastic synapses in each of the experiments. All configurations include a bias neuron. `L4` provides a layered network with 4 neurons, all others are fully connected

	Number of processing neurons				
	`N0`	`N2`	`N4`	`N6`	`L4`
`E1`	14	32	60	96	46
`E2`	10	28	54	88	42
`E3`	14	32	60	96	42
`E4`	14	32	60	96	42
`E5`	4	15	35	63	32

with few simplifications that decrease the number of plastic synapses considerably. First, the hidden layer is split into two pairs of neurons. One pair is connected to the two distance sensors only, while the other pair is connected to the two directed light sensors. This results in a modular structure that enforces a kind of specialization to each pair. The two modules are also fully-connected to each other, adding eight plastic synapses that are responsible for the fusion of behavior. Furthermore, a symmetry constraint is added to each module. This means that a change of some synapse at the left side of the module would be copied to corresponding synapse at the right side. This constraint is meant to reflect the symmetry in the body morphology of the robot, which would result in a symmetric behavior. No constraints are imposed on the connections between the two modules. As such, the number of plastic synapses in this CSN, including those coming from a bias neuron, are only 22.

The parameters of MRS are chosen as before but with the probability of enabling or disabling a synapse set to zero. The range of weights is restricted to ± 1.5 for both MRS and MGW. For the latter, the variance σ^2 is set to 0.2. Each learning rule was tested on 64 runs, with 8 hours simulation time.

5 Results and Discussion

5.1 Results on Modulated Random Search

For all experiments and with all but one of the different network substrates, solutions have been found within the given time windows. All behaviors discovered in this way have been sufficiently effective and comply with the desired and expected behaviors. However, as can be seen in Fig. 4, by far not all runs did finally end up with a proper behavior network during the limited learning time. Consistent with intuition, the easier the task is, the larger the percentage of successful learning trials.

Therefore, The simple light-tropism task `E1` led to successful behaviors in almost all cases, despite its comparably short learning time of up to only two hours. Also, the final solutions have been found very fast (Fig. 5a-E1) without many intermediate temporal solutions (Fig. 5c-E1).

In contrast, the almost similarly short duration of the obstacle-avoidance task `E2` with four hours seems to be much too low to consistently find solutions, contrary to our expectation. Therefore, only about half of the experiments were successful. A reason for this may be the relatively slow detection of insufficient exploration behavior with the `Explore` NMC. This modulator has to react with a larger delay to give the networks a chance to actually do exploration.

So, behaviors violating the exploration condition – while still doing a fine obstacle-avoidance – are detected only after a significant delay. Also, such intermediate solutions get destroyed quite easily when a bad exploration behavior is detected, leading to the destruction – not to a refinement – of the temporary solution. This, obviously, is one of the major limitations of the stochastic search: due to the missing directedness of the learning, temporary solutions are usually not improved, but rather destroyed and replaced by very different networks.

The results for combining light tropism and obstacle avoidance (`E3`) reflect the increasing difficulty of the task. Even though the experiment was simulated 12 hours

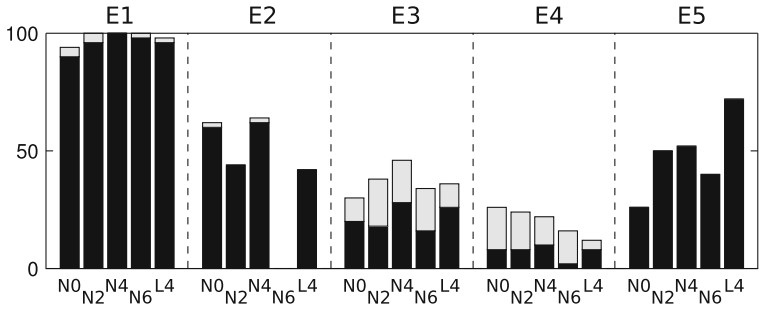

Fig. 4 Percentage of successful experiments with stable solutions. The *gray tips* indicate the number of temporary solutions with a continuous modulator-free behavior during at least 30 min, which would be interpreted as solutions in intermediate-term evaluations

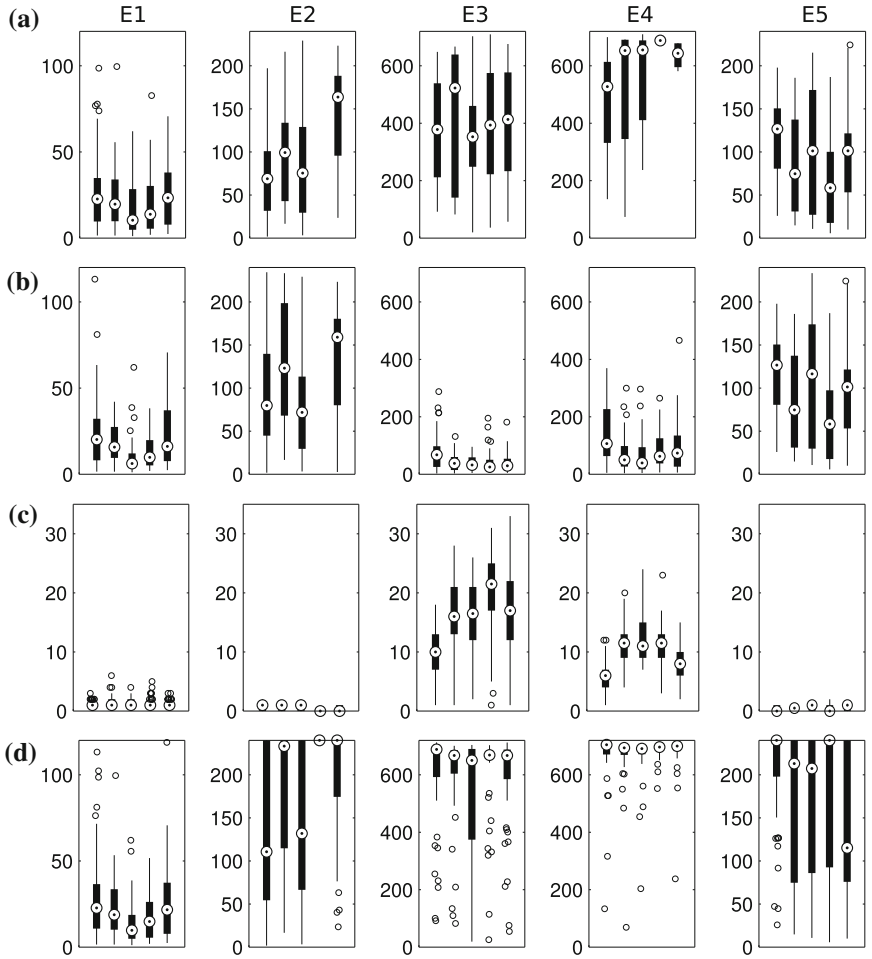

Fig. 5 **a** Time to final solution. **b** Time to first (temporary) solution. **c** Number of (temporary) solutions. **d** Minutes spent in learning mode

per try, only $\approx$ 20 % of the runs lead to a fully stable behavior. First temporary solutions have been found quite fast (Fig. 5b-E3), but most light tropism behaviors with only a partial obstacle-avoidance behavior are easily destroyed due to hitting one of the small obstacles close to the light sources. Because the light sources are approached with slightly different angles, at some point a situation is encountered where the obstacle-avoidance behavior briefly fails and the obstacle is hit. This leads to a strong production of NM and the behavior is usually destroyed. This alternation between many temporary solutions (Fig. 5c-E3) and the subsequent network destruction, and thus long phases with enabled plasticity (Fig. 5d-E3), describes the typical way how network configurations are explored with the stochastic search: only if *all*

requirements of the behavior are *fully* met with a single mutation burst, the behavior remains stable in the long run. This *all or nothing* approach is another limiting characteristics of the simple stochastic search.

This becomes even more severe in the aggravated variant of this experiment (E4), in which large and more various obstacles enforce the robot to do significant detours against the desired direction towards the light. Here, a proper behavior requires a fine tuning of weights, which makes it much more difficult to accidentally stumble upon a working network. The percentage of final solutions, therefore, is even lower with only about 10 %. However, the number of long-term temporary solutions with a continuous runtime of more than 30 min exceeds the number of stable solutions by a factor of ≈ 2 (Fig. 4-E4). These behaviors would in many evaluations with a short test (e.g. evolutionary algorithms) already be considered solutions, but it shows that even slight weaknesses due to an unfortunate sequence of target light sources can lead to a destruction of such *almost stable* networks in the long run. As in E3, temporary solutions are found quite fast (Fig. 5b-E4), but are destroyed later, so that most of the time is spent trying new network configurations (Fig. 5d-E4).

The pendulum behavior again is an example of a simpler single-goal task. The number of successful runs is, with almost 50 %, quite high and the networks are also found fast within the first 2 hours (of a total of 4 hours). Due to the characteristics of the experiment, there are almost no temporary solutions: if a solution is found, then this solution tends to be stable in the long run, because there are no disturbances in the simple pendulum motion (compare Fig. 5a-E5, 5b-E5, and 5c-E5).

An interesting observation can be made concerning the network complexity. It was expected, that the performance of the experiments primarily depends on the size of the neural substrate, because with an increasing search space the probability of finding a stable solution should drop down significantly. However, at least for the network sizes used in these experiments, there is only a small influence of the network substrate on the performance (Fig. 4). Only in E2 the largest network showed a significant drop in the number of solutions compared to the other substrates in the same experiment. And in E5 it seems that the layered network has an advantage over the fully recurrent neural networks. This may indicate, that – as long as the topology can vary within the substrate – there are similar or equivalent network configurations contained in all substrates and that with an increasing number of synapses, the fraction between feasible and improper network configurations may remain in the same order of magnitude. In forthcoming experiments, larger networks have to be tested to find the actual limiting size for this simple class of robot experiments. In these experiments, anyway, the impact of the chosen experiment complexity has a much higher impact on the performance than the chosen network substrate, so the major effort in designing such experiments should probably be focused on defining a well suited experiment, not on choosing a particularly suited network substrate.

To examine the learning process in more detail, Fig. 6 shows the weight changes and the related neuromodulator concentrations for one of the learning runs in experiment E2. As expected, the weight changes in learning phases are random and undirected. However, from time to time, the system stabilizes in a network configuration, because no neuromodulator is produced as a response to the (partially) working

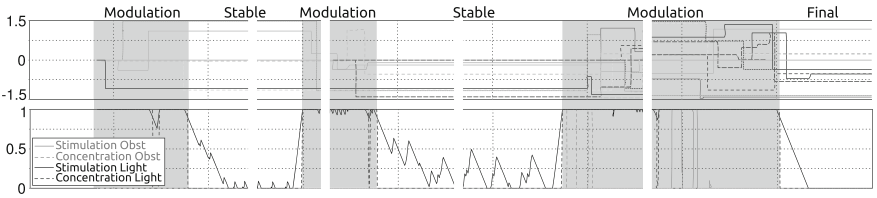

Fig. 6 Example run for the light tropism-behavior, showing the alternation between stable and plastic states during the behavior learning. The *upper graph* shows the individual weights over time, the *lower graph* the stimulation and concentration level of the two NMCs

behavior. It can also be seen in the lower part of Fig. 6 that even during these stable states, the stimulation of the NMCs is not just zero, but that their stimulation level remains active, though not high enough to enter their *production mode*. So, slight violations of the behavior restrictions still take place, but these violations are not strong enough to be interpreted as a failing behavior. But if the stimulation level exceeds the limit to *production mode*, then often one of the first random changes destabilizes the system so much, that other neuromodulators are triggered as side-effect. This leads to a strong relearning, usually destroying the previous temporary solution, until the modulation stops when a new potentially working configuration has been found.

5.2 Comparing MRS and MGW

As the previous section demonstrated, due to its uncontrolled random changes to network structure, MRS leads to the destruction of solutions. In comparison, limiting random changes to small values, as is the case in MGW, results in the preservation of found solutions. In the modular light-tropism/obstacle-avoidance experiment, outlined in Sect. 4.2, MRS has shown 34 temporary solutions that lasted longer than 5 min in simulation time, with an average of 5.7 min per solution. On the other hand, MGW found almost double the number of temporary solutions, with an average of 12.5 min per solution. Also, under MGW, the agent spends more of its time exploiting the found solutions. While temporary solutions that lasted longer than 5 min occupied more than 11.4 % of the experiment time of robots trained by MGW, only 8.2 % of the experiment time is covered by the temporary solutions found by MRS. This means that learning with Gaussian walk is more stable since the learning rule does not result in the sudden destruction of behavior when neuromodulation is released due to minor lapses in behavioral fitness. Further results suggest that MGW refine network structures that are on the verge of becoming a solution by inducing small changes to the networks' synaptic weights. This is demonstrated by the fact that only 40 % of controllers trained by MRS found a temporary solution at all, while MGW lead to 70 % of the runs leading to a temporary solution at some stage of learning.

6 Conclusions

We demonstrated with five typical experiments from the field of robot learning and early evolutionary robotics, that a simple random search on a given network topology is sufficient to find many suitable solutions, as long as the network changes are started and stopped by a reasonable feedback signal. In our case, this feedback is realized with neuromodulators that are triggered as a reaction to the sensed behavior. Because of this, and the simplicity of the implementation, the learning should also work directly on physical robots without external supervision. The tasks show that the feasibility of the method strongly depends on the experiment complexity, not so much on the chosen network substrate. Also, temporary solutions appear and get relearned when the behavior proves ineffective in some situations. These aspects – already available in such a simple approach – are highly desired in the field of robot learning to allow adaptive, self-contained robots with life-long learning capabilities.

Simple random search, however, is not meant to be used as a competitive learning paradigm for real robots. Our results show that by simply replacing the fully-random search with a more confined random walk of synaptic weights lead to a huge increase in the number of solutions and of their stability. This points to the possible benefits of incorporating more directed learning rules and synaptic dynamics to the neuromodulation framework. Our intention of this study is to provide the mechanism that signals to an autonomous system the need to start learning, i.e. when to learn. The suggested learning mechanism itself, i.e. how to learn, needs to prove superior to the simple random search, as was demonstrated by the Gaussian walk, in order to justify its increased complexity.

Acknowledgments This research was partially funded by the German Research Foundation (DFG) priority program 1527. The contribution of Christian Rempis to this project is gratefully acknowledged. The authors thank Josef Behr, Andrea Suckro, and Florian Ziegler for testing and refining the simulation models in the NERD Toolkit, and particularly the latter for his role in the current study. Thanks to Kevin Koschmieder for implementing the modulated Gaussian walk.

References

1. Dean, J.: Animats and what they can tell us. Trends Cogn. Sci. **2**(2), 60–67 (1998)
2. Meyer, J.A.: The animat approach to cognitive science. In: Roitblat, H. Meyer, J.A. (eds.) Comparative Approaches to Cognitive Science, pp. 27–44. The MIT Press/Bradford Books (1995)
3. Meyer, J.A., Guillot, A.: Biologically inspired robots. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, pp. 1395–1422. Springer (2008)
4. Pfeifer, R., Bongard, J.: How the body shapes the way we think: a new view of intelligence. MIT press (2007)
5. Ross Ashby, W.: Design for a brain: the origin of adaptive behavior (2nd edn). Chapman and Hall, London UK (1960)
6. Di Paolo, E.A.: Organismically-inspired robotics: homeostatic adaptation and teleology beyond the closed sensorimotor loop. In: Murase, K., Asakura, T. (eds.) Dynamical Systems Approach

to Embodiment and Sociality, pp. 19–42. Advanced Knowledge International, Adelaide, Australia (2003)
7. Ziemke, Tom: The embodied self: theories, hunches and robot models. J. Conscious. Stud. **14**(7), 167–179 (2007)
8. Ikegami, T., Suzuki, K.: From a homeostatic to a homeodynamic self. BioSystems **91**(2), 388–400 (2008)
9. Der, R.: Artificial life from the principle of homeokinesis. In: Proceedings of the German Workshop on Artificial Life (2008)
10. Hebb. D.O.: The Organization of Behavior. Wiley, New York (1949)
11. Cooper, L.N., Intrator, N., Blais, B.S., Shouval, H.Z.: Theory of Cortical Plasticity. World Scientific (2004)
12. Turrigiano, G.G., Nelson, S.B.: Homeostatic plasticity in the developing nervous system. Nat. Rev. Neurosci. **5**(2), 97–107 (2004)
13. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(9), 533–536 (1986)
14. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA **79**(8), 2554–2558 (1982)
15. Jennings, H.S.: Contributions to the study of the behavior of lower organisms. Number 16, Carnegie institution of Washington (1904)
16. Smith, T., Husbands, P., Philippides, A., O'Shea, M.: Neuronal plasticity and temporal adaptivity: Gasnet robot control networks. Adapt. Behav. **10**(3–4), 161–183 (2002)
17. Timmis, J., Neal, M., Thorniley, J.: An adaptive neuro-endocrine system for robotic systems. In: Proceedings of the IEEE Workshop on Robotic Intelligence in Informationally Structured Space, RIISS'09, pp. 129–136 (2009)
18. Moioli, R.C., Vargas, P.A., Husbands, P.: A multiple hormone approach to the homeostatic control of conflicting behaviours in an autonomous mobile robot. In: Proceedings of IEEE Congress on Evolutionary Computation, CEC'09, pp. 47–54 (2009)
19. Rempis, C., Thomas, V., Bachmann, F., Pasemann, F.: NERD—Neurodynamics and Evolutionary Robotics Development Kit. In: Simulation, Modeling, and Programming for Autonomous Robots, pp. 121–132. Springer (2010)