Controlling the Learning of Behaviors in the Sensorimotor Loop with Neuromodulators in Self-Monitoring Neural Networks*

Christian W. Rempis, Hazem Toutounji, Frank Pasemann[1]

Abstract— **Using neuronal plasticity in the sensorimotor loop of embodied controllers to autonomously learn behaviors remains a great challenge. The difficulty lies not only in the development of sophisticated plasticity mechanisms, but also in controlling when, where and how to learn, in order to achieve the *correct* behavior. Borrowing from biology, we develop a general framework that deploys intrinsic biofeedback-like signals to assess the behavior through self-monitoring and to control learning accordingly. The framework augments a plastic control network (CSN) with a static *neuromodulator* (NM) subnetwork (MSN), containing cells capable of producing NM as feedback signal in response to observed network activations. A desired behavior can then be defined solely by specifying the conditions for NM release. As a response to NM exposure, the CSN changes until a new stable configuration without NM release is reached, i.e. the desired behavior is performed. We demonstrate this approach with experiments in which simple behaviors are learned from scratch. We show how the behaviors can be defined with MSNs and that – in combination with a simple stochastic plasticity method – behaviors are successfully learned.**

I. INTRODUCTION

Learning robot behaviors from scratch, i.e. autonomously searching for corresponding neural networks in the sensorimotor loop of a robot, is still a challenging task in the neurorobotics community. In most cases, neither the weight matrix, nor the network topology of the recurrent neural networks are known in advance and only a few algorithms are applicable to search for such controllers. The major approach still is evolutionary robotics (ER) [5][4], where the neuro-controllers are searched with evolutionary algorithms (EA). While successfully solving many behavior experiments, this approach has a number of drawbacks compared to an autonomous learning directly on the robot: Evolution experiments require a high number of repeated experiments to rate the performance of a behavior. For practical reasons this usually has to be done in simulation and requires a subsequent transfer of the solutions to the physical machine. In many cases this is a challenge due to the differences between the simulation and the physical machine, often called the Reality Gap [7]. Another problem is that a static, evolved neuro-controller cannot adapt to new challenges in the environment at lifetime, so that the controllers have to be very robust and be tested in all expectable situations during evolution. In most cases this is vastly impossible.

To overcome these problems, current research is focusing on learning in the sensorimotor loop, i.e. neural plasticity methods that are capable of adapting the neuro-controllers during runtime to allow the robot to learning by trial and error on a single machine. Such controllers could explore the capabilities of the body and its sensors without supervision, and develop their own behaviors during runtime. Also, unforeseen situations or changes in the body (e.g. misalignments of the sensors) can lead to an adaption of the controller, to improve the behavior or to temporarily alter it. This could also be done directly on a physical target robot so that its simulation, the laborious setup of evolution experiments and the subsequent transfer can be avoided.

Our research goals here are twofold: 1) The identification of suitable *plasticity methods* and 2) the investigation of mechanisms that *intrinsically* deploy *Instrumental Conditioning* for the control and guidance of the learning process, i.e. learning is controlled by biofeedback-like signals generated by a network itself while monitoring its own activity.

In this contribution, we focus on the second part, because we believe that the control of the learning process itself (*when* to learn, *where* and *how*) is an integral part of any *directed* learning system, i.e. a system with a given *target* behavior. Therefore, to develop and examine novel plasticity methods in the sensorimotor loop, we first develop a number of experiments in which the learning can be guided successfully with feedback signals. The target behavior, hereby, has to be described in a form that guides the learning process towards the desired behavior, for instance in ER with a fitness function [4]. If autonomous learning should be realized directly in a neural network without external algorithms (such as EA), then the feedback also has to be generated in – and be represented as part of – the network.

For this purpose we extend a standard network model by an *artificial neuromodulator layer* [2] (see Sect. II-A) that serves as a universal way to communicate intrinsic states locally between populations of (not necessarily connected) neurons. These neuromodulators (NMs) can be interpreted as chemicals (e.g. dopamine, serotonine, nitric oxide [3][14]) that are produced by special cells in response to certain network activities and that diffuse locally into the tissue surrounding that cell. The possible effects of NMs are various [19], e.g. they may influence plasticity, represent positive or negative reward [3], signal internal states [9], reflect motivations [17] or modulate the excitability of neurons [6][11]. Although this approach is inspired by biological systems [3], we do not intend to model an actual biological brain chemistry. The NMs, in our case, primarily signal when a behavior is *undesired* and has to be changed. As a paradigm, a *predefined* subnetwork monitors the signals

*This work was partially funded by DFG-grant PA 480/7-1

[1]The authors are with the Institute of Cognitive Science, AG Neurocybernetics, Osnabrueck University, 49076 Osnabrueck, Germany. christian.rempis@uni-osnabrueck.de

of the robot and produces NMs as a response to undesired activities, e.g. based on *comparator neurons* [12] that may monitor the difference between a desired and a received sensory signal.

In contrast to most comparable approaches, these modulator subnetworks are not learned or evolved, because we do not want to search for networks *capable of learning*, but instead we want to investigate plasticity mechanisms within systems that we consider *already capable of learning* a desired behavior. Therefore, the modulator subnetworks are specified *manually* for each experiment by a supervisor, analog to the definition of fitness functions in ER. Hereby, the desired behavior has to be expressed in terms of NM production, which will be shown in section III-C.

Once a *modulator sub-network* (MSN) is defined, the actual learning takes place in a second subnetwork, a substrate of neurons and synapses capable of plastic changes that connect the motors with the sensors of the robot. This learning subnetwork is called the *control sub-network* (CSN). The given substrate for this network, which defines its (maximal) topology, can be varied between runs of the experiment to foster learning of different controllers of various complexities.

A full learning network (i.e. the MSN and CSN together) then is run on a (simulated) robot until the desired behavior is learned. The learning, hereby, is understood as the *autonomous* exploration of the behavior space by the learning *CSN* based on plastic changes triggered and guided by the *MSN*. This leads to an unsupervised trial and error exploration of behaviors that will stabilize in a working network configuration after a sufficient runtime. Because, once a valid behavior has been discovered, no further neuromodulation is produced and the network becomes static.

In the following, we introduce our variant of a *modulated neural network* (Sect. II) and show in a number of experiments, how a desired behavior can be outlined in terms of NM based feedback signals directly in the networks (Sect. III-C). We further demonstrate in Sect. III with, four examples, that this approach – combined with a simple stochastic plasticity method – is already capable of autonomously learning controllers for many classical ER experiments from scratch, e.g. for obstacle avoidance, light tropisms and even combined tasks.

II. METHODS

A. Modulated Neural Networks (MNN)

An MNN can be any kind of standard artificial neural networks extended by a *neuromodulator layer* [2]. Some related approaches, though more specialized, are e.g. GasNets [6], Artificial Endocrine Systems [17] and Artificial Hormone Systems [9].

Our variant of a NM layer provides *neuromodulator cells* (NMCs) that maintain spatial distributions of 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 was *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.

B. 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

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

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 $tanh$.

The stimulation of NMCs follows a simple linear model. Each NMC is attached to a neuron 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 stimulated, its stimulation level s_i increases by a small amount given by parameter S^{gain}. If not stimulated, it decreases by S^{drop}. 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*. This allows the definition of various delays and hysteresis effects between the two modes, which is an important prerequisite for stable learning (see Sect. III).

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

TABLE I: Parameters of a NMC in a *LMNN*

Parameter	Description
$Type$	The NM type produced by this NMC
S^{min}, S^{max}	Neuron activation range that stimulates the NMC
S^{gain}, S^{drop}	Rate of stimulation gain and drop
T^{prod}, T^{red}	Thresholds for reduction and production mode
C^{max}	Maximal concentration produced by this NMC
C^{gain}, C^{drop}	Rates of concentration gain and drop
R^{max}	Maximal radius of the diffusion area
R^{gain}, R^{drop}	Rates of diffusion area gain and drop

In *production mode* the modulator concentration c and the radius r of, here, 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}. Equation 4 shows this for the concentration level c_i; the area radius r_i is defined analogously.

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

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.

C. 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.

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.

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

TABLE II: 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}	Minimal and maximal weight of the synapse
M	Maximal NM sensitivity limit of the synapse

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.

$$p_i^d(t) = min(M_i,\ c(t,x_i,y_i))\, D_i, \qquad 0 \le D_i < W_i \quad (6)$$

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.

III. EXPERIMENTS

To demonstrate the method, four experiments with different complexities have been performed using the NERD Toolkit [13] software. The experiments are typical for early ER 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 NM for undesired behaviors, while the given CSN defines the topology in which solutions can develop. All synapses of the CSN start out disabled, so that the network connectivity develops together with the synaptic weights, while all synapses of the MSN are insensitive to NM and therefore static. As such, each experiment can be defined by a robot, a task, an environment and a LMNN with a MSN and a CSN.

A. Robot, Tasks and Environments

All experiments use a differential drive robot equipped with a number of motors and sensors (Fig. 1-a). The two motors are controlled by two motor neurons whose activation range [-1,1] corresponds to the desired velocity of each wheel. Negative activations are interpreted as backwards rotation. Three *distance sensors* (DS) at the front activate their corresponding sensory neurons according to the measured distance. Eight *touch sensors* (TS) detect force applied to them from outside. Three *ambient light sensors* (ALS) measure brightness at three equally distributed positions on the robot. Finally, three *directed light sensors* (DLS) in the front of the robot consider – in addition to the brightness

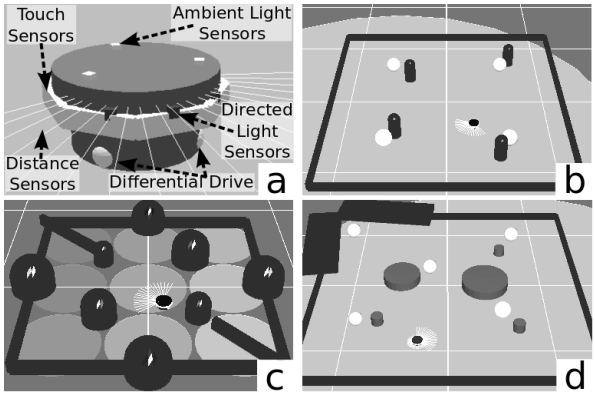

Fig. 1: The robot and three of the environments. The white spheres denote light source positions.

– also the angle towards the light source, with a maximal viewing angle of ± 90 degrees. For simplicity, light can penetrate obstacles freely.

The first experiment (`E1`) is a positive light tropism task (Fig. 1-b without obstacles). 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. 1-c), where the robot has to navigate in a quadratic environment riddled with round objects and sharp corners. 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. 1-c). Here, the robot has to approach the lights and simultaneously avoid the obstacles next to the light sources.

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. 1-d). 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.

B. Control Sub-Networks (CSN)

Each CSN includes sensory neurons for all distance and directed light sensors, two motor neurons, some intermediate 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 substrate always is a fully connected, recurrent network spanning over four intermediate neurons from a selection of *behavior-relevant* sensory neurons to the motors. The network configurations for the experiments are summarized in Table III.

TABLE III: LMNN configurations. See text for description.

	CSN		MSN
Exp.	**Sensors**	**Synapses**	**Used NMC Modules**
`E1`	2 DLS	48	Light
`E2`	3 DS	54	Obst, Drive, Explore
`E3`	2 DS, 2 DLS	60	Light, Obst
`E4`	2 DS, 2 DLS	60	Light, Obst

C. Modulatory Sub-Networks (MSN)

Each MSN uses *experiment-specific* network structures to detect undesired behavior based on (sensor) activations to produce NMs when needed. As a reaction to the NMs, 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 of Ashby [1], 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, four different NMCs are used in the experiments: Neuron `Obst` reacts to bumping into obstacles, `Drive` favors fast forward movements, `Explore` responds to non-exploratory behavior, and `Light` reacts when the robot is not actively approaching light.

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 NM production to alter the behavior.

`Drive` 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.

`Explore` is stimulated when the robot is not entering the detectable locations frequently. Its associated modulating network classifies the signal of one of the ambient light sensors (ALS) into the nine detectable locations and integrates these signals to determine the duration of each location not being visited. If enough locations have not been visited for a too long time, `Explore` is stimulated. 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 NM decrease to allow the new configuration to be tested.

The `Light` cell also uses an auxiliary network that interprets the ALSs to detect whether the robot is getting closer to the light. If not, the `Light` cell is stimulated.

Table III shows which NMCs, with their corresponding auxiliary networks, are used in each experiment. Figure 2 shows the structure of both the CSN and the MSN for

TABLE IV: Parameter values for NMCs in the experiments

Parameter	Obst	Drive	Explore	Light
S^{min}, S^{max}	0.9,1.0	0.9,1.0	0.4,1.0	0.9,1.0
S^{gain}, S^{drop}	0.01,0.01	0.001,0.001	0.001,0.01	0.0002,0.0001
T^{prod}, T^{red}	0.95,0.95	0.95,0.95	0.95,0.95	0.99,0.99
C^{max}	2	1	1	1
C^{gain}, C^{drop}	0.1,0.1	0.001,0.01	0.001,0.01	0.01,0.1

experiments E2 and E3, giving also the neural structures for the four auxiliary sub-networks.

While the MNN framework allows for concentration gradients and localized release of modulators, we restrict the experiments here to global modulators and a uniform concentration across the whole CSN. This also entails that only one type of NM is used. In addition, the diffusion area increases and decreases instantaneously. Table IV summarizes the parameters choice for the four NMCs used across the experiments.

IV. RESULTS

In order to test the viability of the MNN framework, we ran multiple simulations of each of the four experiments. The plasticity parameters $(W, D, W^{min}, W^{max}, M)$ have been fixed to be $(0.0001, 0.00002, -1.5, 1.5, 1.0)$ for all plastic synapses of the CSNs. As a first step, we restrict our results to qualitative observations. Due to space limitations, a more thorough examination of the learning performance will be given in future work.

For experiments E1 and E2, many stable solutions have been found often within minutes. In contrast, runs of E3 required between 2 and 8 hours to get to *stable* solutions, although, during that time, many *partial* solutions have been found, only to be destroyed after a while due to still undesired behavior in specific situations. For E4, no stable solutions have been found yet in runs of up to 10 hours. However, as in E3, many temporary (partial) solutions did appear and persisted for a while.

Videos of the behavior learning can be found in the supplementary material. For illustration purposes, we show networks of E2 and E3 after successful learning in Fig. 2.

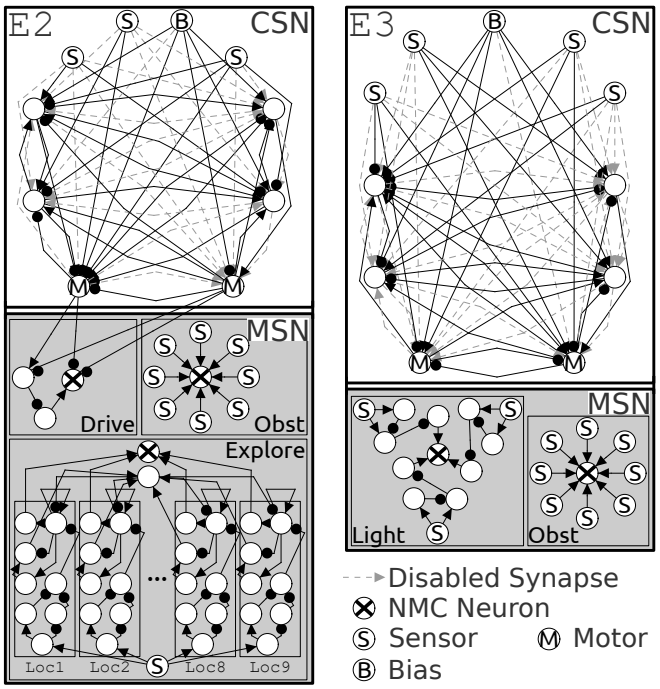

Fig. 2: Successful networks for E2 (left) and E3 (right), showing their learned CSNs(white) and fixed MSNs (gray).

V. DISCUSSION

A. Runtime and Behavior Discovery

The simple random search already finds successful controllers in different scenarios, but its scalability is clearly limited. The larger the networks become, the more difficult the random discovery of working solutions gets. However, especially in the simple experiments, networks could be chosen much larger than we primarily expected. The search does not only succeed with small feed-forward networks, but also with larger, fully recurrent networks in many cases. And also the run-time is, in principle, comparable with the runtime of ER algorithms (from few minutes to few hours). One reason certainly is that in such experiments (E1, E2), a large fraction of the search space contains sufficient solutions. In contrast, the fraction of suitable solutions within the search spaces of E3 and E4 is much smaller, hence the runtime is longer and the probability of success within a given time drops. In these latter experiments, a second problem – particularly with the random search – is that two (or more) partial solutions have to be found independently of each other. A global random search tends to destroy already working functionality, even at the smallest deviation from the desired behavior, because NM is released globally, affecting also the working subnetworks. So, random solutions have to immediately be fully operational (all or nothing), because the chance for iterative improvements is small. To overcome this, more sophisticated plasticity methods will be needed, as we will address in a forthcoming publication. In future work, we will also test local NMCs with restricted diffusion areas to examine whether location dependent modulation helps to scale the learning approach by reducing destructive effects on unrelated network areas, fostering independent learning of sub-behaviors.

B. Comparison with Evolutionary Robotics

Despite the constraints, this random search approach works successfully in many complexity domains in which ER has been the major search algorithm in the past [10][4][8]. The major advantage of a local search is that, in contrast to ER, neither an external EA implementation, nor many repeatable experiments for the comparison of the generated networks are needed. Instead, the controllers may run directly on a single robot, as long as it is capable of running *safely* for a sufficient time, i.e. it must be robust enough to withstand behavior failures during trial and error learning. Another advantage is that such a robot can reconfigure its behavior on

demand in unforeseen situations, in which a static, evolved controller would just fail. One major limitation in comparison with ER is that behaviors are *not* getting optimized. The *first suitable* behavior is kept until it has been proven faulty. Therefore, this approach is primarily interesting when a robot has to *only* cope with its environment, without further demands on peculiarities and the efficiency of the behaviors.

C. Modulator Sub-Network (MSN) Definition

Defining MSNs is, like the definition of fitness functions in ER, a matter of experience and an iterative process. Similar to fitness functions, the definition of an MSN strongly affects the outcome. The design effort in both cases is comparable, but the paradigms are much different. In contrast to ER, robots in learning experiments always have to be able to *continue*, otherwise they cannot learn from experience. Common fitness measures, e.g. killing the robot, using limited power supplies or allowing other unrecoverable situations, are not suitable. Furthermore, a behavior has to be expressed in terms of NM release in response to *local sensory signals*. This may require additional network structures to process these signals, which necessitates some experience in neurodynamics. The actual configuration of the NMCs, however, does not require much fine tuning and is not more difficult than choosing suitable parameters of an EA.

Experience shows that a proper MSN has to be *complete*, i.e. *all* undesired behaviors have to lead to NM release. Otherwise, the system may get stuck with undesired behavior. The different aspects of a behavior should be reflected on different time scales. The stimulation delays play a key-role here. Obvious and drastic behavior failures should immediately lead to NM release. But many sensory signals are ambiguous and can indicate a failure, but are also temporary events of valid behaviors. Such signals should require a longer stimulation time of the NMCs so that violations of such assumptions are tolerated for a while.

In combination with random search, plastic changes should be relatively rare events, so that new configurations can be explored for a while before further changes alter the network. Runtime, however, slows down, so there is always a trade-off in performance and stability of solutions.

VI. CONCLUSIONS

In this work, we presented a framework to allow neurocontrollers for simple behaviors to be learned autonomously in the sensory-motor loop. This is achieved by defining a relation between the observed performance of the behavior and the release of neuromodulators (NM) in the CMN in order to induce learning. As a baseline for future work, we define learning here as changes in synaptic weights and network topology, triggered and guided by concentrations of NMs. In the first examples, we used one of the most simple plasticity methods, a pure random search [1]. We demonstrated that simple behaviors can be successfully learned from scratch in feasible time in networks that have comparable complexity to many early ER experiments [10][4][8] and even current experiments with other plasticity methods [11][14][15][16][18]. Other conducted experiments, like the pendulum problem, pole balancing and more complex locomotion behaviors, will be presented in forthcoming work.

APPENDIX

Supplementary material can be found at our homepage:

`nerd.x-bot.org/neuromodulator-benchmarks`

ACKNOWLEDGMENT

We thank Josef Behr and Florian Ziegler for testing and refining the simulation model and for their contributions to the NERD toolkit. Thanks to Tanya Beck for proofreading.

REFERENCES

[1] W. R. Ashby, "Design for a brain." 1952.
[2] C. L. Buckley, "A systemic analysis of the ideas immanent in neuromodulation," Ph.D. dissertation, University of Southampton, 2008.
[3] K. Doya, "Metalearning and neuromodulation," *Neural Networks*, vol. 15, no. 4-6, pp. 495–506, 2002.
[4] D. Floreano, P. Husbands, and S. Nolfi, "Evolutionary robotics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 1423–1451.
[5] I. Harvey, P. Husbands, D. Cliff, A. Thompson, and N. Jakobi, "Evolutionary robotics: the sussex approach," *Robotics and Autonomous Systems*, 1997.
[6] P. Husbands, "Evolving robot behaviours with diffusing gas networks," in *EvoRobots*, 1998, pp. 71–86.
[7] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," *Advances in artificial life*, pp. 704–720, 1995.
[8] L. A. Meeden and D. Kumar, "Trends in evolutionary robotics," in *Soft Computing for Intelligent Robotic Systems*, L. C. Jain and T. Fukuda, Eds. Physica-Verlag, New York, 1998, pp. 215–233.
[9] R. Moioli, P. Vargas, and P. Husbands, "A multiple hormone approach to the homeostatic control of conflicting behaviours in an autonomous mobile robot," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 47–54.
[10] F. Pasemann, U. Steinmetz, and U. Dieckman, "Evolving structure and function of neurocontrollers," in *Proceedings of the Congress on Evolutionary Computation*, P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, Eds., vol. 3. Mayflower Hotel, Washington D.C., USA: IEEE Press, 6-9 1999, pp. 1973–1978.
[11] L. Pitonakova, "Ultrastable neuroendocrine robot controller," *Adaptive Behavior*, 2012.
[12] W. T. Powers, *Behavior: The control of perception*. Aldine de Gruyter, New York, 1973.
[13] C. W. Rempis, V. Thomas, F. Bachmann, and F. Pasemann, "NERD - Neurodynamics and Evolutionary Robotics Development Kit," in *SIMPAR 2010*, ser. Lecture Notes in Artificial Intelligence, N. A. et al., Ed., vol. 6472. Springer, Heidelberg, 2010, pp. 121–132.
[14] T. Smith, P. Husbands, A. Philippides, and M. O'Shea, "Neuronal plasticity and temporal adaptivity: Gasnet robot control networks," *Adaptive Behavior*, vol. 10, no. 3-4, p. 161, 2002.
[15] O. Sporns and W. H. Alexander, "Neuromodulation in a learning robot: Interactions between neural plasticity and behavior," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 4. IEEE, 2003, pp. 2789–2794.
[16] K. O. Stanley and R. Miikkulainen, "Evolving adaptive neural networks with and without adaptive synapses," in *Genetic and Evolutionary Computation Conference Late Breaking Papers*, B. Rylander, Ed., Chicago, USA, 12–16 July 2003, pp. 275–282.
[17] J. Timmis, M. Neal, and J. Thorniley, "An adaptive neuro-endocrine system for robotic systems," in *Robotic Intelligence in Informationally Structured Space, 2009. RIISS'09. IEEE Workshop on*. IEEE, 2009, pp. 129–136.
[18] J. Urzelai and D. Floreano, "Evolution of adaptive synapses: Robots with fast adaptive behavior in new environments," *Evolutionary Computation*, vol. 9, no. 4, pp. 495–524, 2001.
[19] Q. Xu and L. Wang, "Recent advances in the artificial endocrine system," *Journal of Zhejiang University-Science C*, vol. 12, no. 3, pp. 171–183, 2011.