computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

ONTOGENETIC TEACHING OF MOBILE AUTONOMOUS ROBOTS WITH DYNAMIC NEUROCONTROLLERS

Helmut A. Mayer [1)]

[1)] Department of Scientific Computing, University of Salzburg, Jakob-Haringer-Strasse 2, A-5020 Salzburg, AUSTRIA, helmut@cosy.sbg.ac.at

Abstract: *After a brief survey of work dealing with dynamic neurocontrollers changing their internal structure during the "lifetime" of a mobile autonomous robot, we present experiments employing a standard sensor–motor neurocontroller with self–adapting weights. The change of behavior of the robot is linked to inputs from the environment that cause the emission of artificial neuromodulators (ANMs) in the robot's neurocontroller. In its simplest form an outside teacher (human or machine) constantly evaluates the robot's actions by transmitting positive or negative feedback signals to the robot initiating the internal changes. The focus of investigations is put on the mechanisms of the interaction of teaching input and structural changes. A well–known concept for this interaction is Hebbian learning, which is regulated by ANMs in the presented approach. In extension to related work in evolutionary robotics (ER), we analyze important details of robotic (ontogenetic) learning by experiments measuring the ability of robots to learn simple tasks in a simulated environment without employing evolution. Specifically, we are interested in the comparison of Hebb learning variants, and the crucial question of the correct interpretation of reward or punishment signals by the robot.*

Keywords: – *Mobile Autonomous Robots, Dynamic Neurocontrollers, Artificial Neuromodulators, Reinforcement Learning.*

1. INTRODUCTION

Today, a variety of control techniques are implemented for mobile autonomous robots, e.g., fuzzy control, machine learning systems, or artificial neural networks. Some of these systems have reached an impressive level of performance, however, most of these systems are not adaptive in the sense that they cannot change their behavior by feedback from the environment. Although, a human observer of these systems might get the impression that the robots can adapt to different situations, they can only adapt to situations the control program is aware of in advance.

State of the art is that control programs for the robots are no longer constructed by humans but by computers employing evolutionary methods methods [1]. Still, in order to evolve working "robot brain" all possible scenarios have to be presented to the system in advance. Hence, the control system will deal sufficiently with most situations it has been trained with, but it is mostly not able to deal with new, unknown situations, and it also cannot adapt itself to these during the "lifetime" of the robot. When evolving robotic neurocontrollers, learning is taking place in a generational time frame (*phylogenetic* learning).

Obviously, the main problem of (most) current control systems is that they cannot "reprogram" themselves during their exploration of the environment. This static behavior of an artificial structure is the most fundamental difference to *Biological Neural Networks* (BNNs) exhibiting highly dynamic properties not only throughout their lifetime, but also within very short time spans of activity [2].

Dynamic changes in a neurocontroller may be induced employing *Reinforcement Learning* (RL) techniques [3] enabling *ontogenetic* learning, i.e., the robot's brain (consequently, its behavior) is shaped during exploration of the environment. A popular RL method applicable to neurocontrollers is *Temporal Difference* (TD) learning [4]. With this method the neurocontroller is not generating motor signals driven by sensor input, but evaluates potential (motor) actions. Actions are presented as an additional input, and a single output neuron predicts the value of a potential action. Learning is driven by the difference of predictions in consecutive time steps (temporal difference) and scalar feedback signals (reward or punishment) from the environment or a teacher.

Technically, learning in TD neurocontrollers is implemented by the common *Back–propagation* method. TD learning is purely ontogenetic and does not alter the structure of the neurocontroller. A biologically more plausible method to achieve a combination of phylogenetic and ontogenetic learning (as seen in nature) are evolved network structures, whose parameters are altered by *Artificial Neuromodulators* (ANMs) [5,6]. The ANMs influence learning by defining the type of *Hebb* learning based on the combination of modulators received by each neuron [5], or by specifically changing neurons' activation functions [6].

As a consequence, very complex interactions can be observed in ANM neurocontrollers that make interpretations of the internal mechanisms nearly impossible. Hence, in this work we are concerned with neurocontrollers with pre–defined, simple modulator diffusion models and single learning rules for the whole network. The biologically plausible learning mechanism is based on the *Hebbian* learn rule promoting self–organization of the neurocontroller.

Especially, in feed–forward networks Hebb learning has the properties of implicit calculation of the *Principal Components* of the input data [7]. The *Principal Component Analysis* (PCA) is a statistical method linearly transforming a sample of points in an *n*–dimensional space such that the variance of the components in the new coordinate system are extremal. Components (or features) with low variance contribute little to the information content of the sample, hence they may be neglected in order to compress the input data. PCA networks can be used in signal classification, feature extraction, and data compression [7]. In the context of this work the feature extraction property could be useful in order to detect structures in the sensor signals of the robot, which might convey relevant information at the current time.

Before giving a description of our own work on dynamic neurocontrollers we present a brief review of pioneering scientific contributions having paved the way towards on–line teaching of robots by changing the internal states of its neurocontroller.

2. RELATED WORK

Nolfi and colleagues (1994) present a genotypic encoding of a neurocontroller allowing the phenotype to be influenced by the environment. The main idea is to let axons grow during ontogeny. The growing of axons is determined by a *Threshold Expression Gene* (not to be mistaken as a neuron's bias). If, in a certain number of subsequent activation states, the neuron's activation exceeds the threshold an axon grows under control of the *Branching Angle Gene* and the *Segment Length Gene*. These latter two genes encode the angle of axon branches, and the length of each branch (the number of branching events was fixed to five). If the growing axon hits another neuron, a connection between the two neurons is formed (the location of each neuron is also evolved in a two–dimensional grid). Moreover, neuron type (sensor, motor, hidden) and its weights are also evolved [8]. With this setting the development of the neurocontroller is not only dependent on the genotype, but also on the environment, as changing stimuli (sensor inputs) induce varying activations in different neurons. A simple task has been simulated for the evolution of neurocontrollers: the robot was placed in an arena where it should move to a specified area. In even generations a light source illuminated the target area, whereas in odd generations the light was switched off. After 500 generations genotypes have evolved which could solve the task with both lighting conditions. As could be expected, a robot evolving in a light environment could solve the task faster and more reliable than a robot evolving in a dark environment. The authors demonstrated some interesting properties of the evolved neurocontrollers. a) when developing the exact same genotype (clones) in different environments, different neurocontrollers emerged. b) moving a genotype having been evolved in an environment of type X and let it develop in an environment Y leads to a decrease in performance compared to development in X. c) a larger performance decrement can be observed, if transferring a phenotype from environment X to Y [8].

Vaario and colleagues (1997) modelled neural growth processes based on *Diffusion Limited Aggregation* (DLA). Here ontogenetic learning is introduced by (artificial) chemical concentrations governing the growth of neural connections which fits as nicely into the framework of Neural Darwinism [9] as the previously described work [8]. In a quite complex manner the chemical concentrations are also influenced by local reinforcement learning (whose parameters are evolved). Again, a beneficial coupling of phylogenetic evolution (of basic neuron properties) and ontogenetic learning could be observed for a looping maze task with simulated robots [10].

Floreano and Mondada (1998) presented very interesting experiments comparing evolved conventional neurocontrollers (with weights being fixed during operation of the robot) and plastic neurocontrollers. The plasticity has been introduced by evolving specific types of Hebbian learning (from a set of four types) for each neuron in the controller. The learning rule has been periodically applied every 300 ms during robot operation. With the latter approach neurocontrollers solving the task (looping in a maze) could be evolved with a considerably smaller number of generations than with conventional neurocontrollers. Moreover, the combination of *Phylogenetic* evolution and *Ontogenetic* learning achieved by the plastic neurocontrollers led to more sophisticated behaviors of the robots [11].

Ishiguro et al. (1999) presented a *Dynamically Rearranging ANN* employed as a robot's neurocontroller [5]. Based on studies on the dynamically rearranging *Stomatogastric Nervous System* in lobsters, where neuromodulators regulate the participation of identical neurons in different subsystems (for specific tasks) the authors introduce two main new features to a standard feed–forward ANN architecture.

For each neuron an NM Diffusion Area is evolved indicating the activation interval where emission of a single NM type (out of two) takes place. Moreover, each neuron's reaction (receptor) to the diffused NMs is defined by an evolved *NM Interpretation Table*. As there is no diffusion radius implemented, each neuron receives all ANMs potentially emitted by all other neurons (the number of received ANMs is not taken into account). Consequently, each neuron may receive four different combinations of the two ANMs in the system. Each combination defines how the weights of the connections leading to this neuron are changed based on Hebbian Learning as given by

$$w_{i,j}^{t+1} = w_{i,j}^{t} + \eta\, R_{i,j}\left(NM_1, NM_2\right) a_i\, a_j \qquad (1)$$

where wi,jt is the weight from neuron j to neuron i at time step t, a is the neuron's activation, η the learning rate, and $R_{i,j}(NM_1,NM_2)$ the evolved artificial receptor. In [5] the possible values of $R_{i,j}$ are 1.0 (Hebbian, two entries in the NM interpretation table), -1.0 (Anti–Hebbian), and 0.0 (no learning). Experiments with evolved conventional ANNs (without ANMs) and the rearranging ANNs on a simple robot task (push a peg to a light source) showed that the conventional neurocontroller evolved in a simulator could not solve the task equally well on the real robot, while the ANN with ANMs could. Adding noise (motor output and peg movement) in the simulation also revealed that the ANM controller performs more robustly. Even more impressing was a video demonstration at the conference [1] where the ANM robot could solve the given task confronted with a peg (an additional weight has been eccentrically put on the peg) it has never seen before.

Smith and Philippides (2000) suggest a Dynamic Artificial Neural Network (DANN) based on the process of *Nitric Oxide* (NO) diffusion in real nervous systems. NO is a neurotransmitter passing freely through most matter in the brain. After elaborating on a physical NO diffusion model, they present an abstraction of the diffusion process in an ANN architecture (*GasNet*) [6]. The discrete time step DANN is built from units connected by binary links (with weights ±1). All but the motor neurons may receive input signals. The output o_i^n of a neuron i at time n is given by

$$o_i^n = \tanh \left\lfloor k_i^n \left(\sum_{j \in C_i} w_{ji} o_j^{n-1} + s_i^n \right) + b_i \right\rfloor \qquad (2)$$

where C_i is the set of neurons connected to neuron i with weights w_{ji}, s_i^n is an external input (sensor) signal, b_i is the bias, and k_i^n is a real number (of a set of predefined numbers) depending on gas concentration. Hence, the activation function is modulated by artificial neurotransmitters.

The DANN is evolved in a 2D Euclidean plane using a variable sized genotype resulting in ANNs with different numbers of neurons. The structure of the net is evolved based on position of the neurons and special *Link Points* giving an arbitrarily recurrent network. Two different gases may be emitted by a neuron. Gas emission is triggered by either an electrical (signal) or a gas threshold of each neuron. For each neuron a maximal diffusion radius r is evolved. Within this diffusion radius the gas concentration $C(d,t)$ is given by $C(d,t) \approx e^{(-d/r)^2}$. The time dependence is introduced by the time intervals neurons emit gas. The gas concentration in turn determines the actual value of k_i^n modulating the shape of the hyperbolic tangent activation function.

For a target discrimination task the connection between a pixel of the camera image and a specific neuron (input s_i^n) has also been evolved. In an arena two white paper targets, a rectangle and a triangle, have been fixed on a wall. Lighting conditions changed permanently (by randomly turning on and off spotlights) during the robot's exploration of the arena.

It has been found that neurocontrollers using GasNets could be evolved in a shorter time, and it seemed that a GASNet architecture solving the task could be easier evolved than a more conventional neurocontroller. The authors argue that the artificial gas introduces some kind of short–term memory, as

[1] IEEE Systems, Man, and Cybernetics, 1999, Tokyo.

gas concentration is influenced by not only the last but a number of time steps. Moreover, the artificial neurotransmitter may act as a low pass filter, as the changing lighting conditions (opposed to the constantly bright targets) during target discrimination did not affect the result [6].

Though not using neurocontrollers, but rather a behavior–oriented approach [12], the plasticity of the neural substrate has also been identified as the key to success of a robot in an unknown environment in [13]. Adaptive behavior of the robot is achieved by explicitly defined basic behaviors modulated by motivational quantities (e.g., energy level of the robot's battery). This approach has shown impressive potential in experiments, where robot's "agreed" on a self–defined language [14], however one should not forget that the categorization of feedback from the environment (state of the robot) and the linkage to a specific motivation are predefined by humans in above systems. Thus, the robot is confined to an environment which is encompassed by its basic behaviors, whereas the proposed method of ontogenetic teaching could adapt a robot to an arbitrary environment (provided the teacher gives meaningful feedback).

3. ONTOGENETIC LEARNING

In the neurocontrollers incorporating ANMs referred to above the parameters for the dynamic changes in the robot's neurocontroller have been evolved. Hence, the final neurocontroller intrinsically deployed the correct types and doses of ANMs so as to achieve the desired behavior of the robot. If we want to teach the robot during its lifetime (on–line), we have to know which ANMs cause the robot to change or enforce its behavior. More specifically, the reaction of a neuron receiving a modulator must be correctly implemented. E.g., in BNNs *Dopamine* acts as a "reward" hormone, which is emitted as a consequence to positive feedback [2]. Though, we can easily define such a reward ANM in the artificial brain, it is not clear which reaction (in our system Hebb learning variants) has to be chosen in order to link the rewarded behavior with the future behavior of the robot.

We want to emphasize that the neurocontroller we are going to present is enabling ontogenetic learning by feedback signals from the environment (mediated by ANMs). Though, being a classical reinforcement learning approach, the neurocontroller's architecture is different from RL methods, as it does not evaluate policies (potential actions), but represents the basic architecture of neurocontrollers employed in ER approaches. The robot's sensor signals at the input layer of the network generate motor signals at the output layer. In addition to pure phylogenetic learning achieved by evolving the structure of the robotic brain, ER researchers also suggested evolution of learning rules enabling lifetime learning [11]. The latter system is learning constantly, while in our approach learning is triggered by pre–defined events or an outside teacher, i.e., there may be only short time periods, where learning is activated or deactivated. Evidently, this should assist the robot in finding interesting subspaces of the input signal space, where it can extract the most useful information to learn the given task. The basic architecture of the neurocontroller employed in the following experiments is shown in Fig.1.

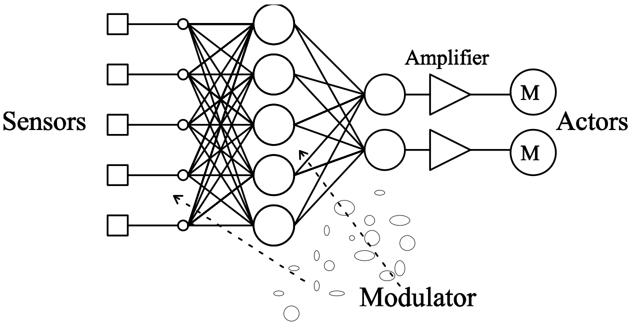

Fig.1 – Basic architecture of a dynamic neurocontroller with artificial neuromodulators.

The plasticity of the robotic brain (induced by the ANMs) also allows for adaptions even when the environment or the physical appearance of the robot (e.g., sensor loss) changes after it has successfully learned a task. In order to investigate the prerequisites for successful ontogenetic learning employing a dynamic neurocontroller, we set up a simple task: the robot should learn to avoid the walls of a rectangular arena (wall avoidance).

In the wall avoidance task the environmental feedback is given by a bumper sensor, which is activated, when the robot touches the wall of the arena. The sensor signal is fed into an input neuron, which emits an ANM signalling "pain" inside the robotic brain. As a consequence, wall avoidance should be learned by giving negative feedback for short periods of time (wall contact).

Results of various experiments should assist to resolve a number of design questions, namely, the rates of emitted modulators, and the reaction to reception of a modulator (actually changing network parameters via Hebbian learning).

4. EXPERIMENTAL SETUP

All experiments are conducted in a Java simulator allowing real time and soft time simulation. The latter enables to perform experiments, where many hours of robot action have to be simulated, in a few seconds or minutes. The cylindrical robot shown in Fig.2 is equipped with four distance sensors (front, back, left, right), and a contact sensor (wall avoidance).

The software sensor simulates a nonlinear, noise–free (Gaussian noise can be set by the user), real device measuring the reflection of a physical signal emitted exactly in direction of the line from robot center to the sensor positioned at the perimeter of the robot.

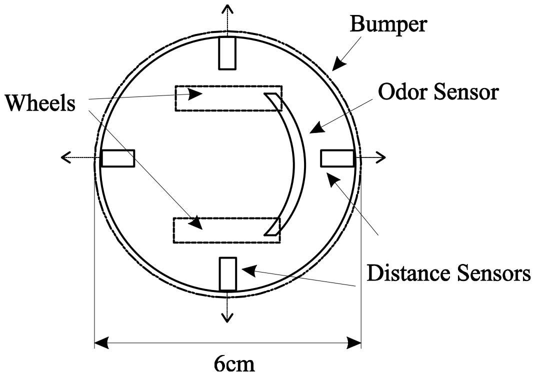

Fig. 2 – The cylindrical robot.

The neurocontroller is a standard *One–Hidden Layer* network (five hidden neurons) composed of neurons with logistic activation function. Each sensor is associated with an input neuron, whose activation determines the signals at the two output neurons (left and right motor). Each neuron is capable of receiving and reacting to the emitted ANM. In case of the wall avoidance experiment, a single type of ANM (pain) is diffused by the contact sensor neuron, when the robot touches the wall. If the ANM is emitted all neurons immediately are able to receive the modulator (in the next time step) by a given reception rate. The reception of a modulator triggers the unsupervised learning process. The dose (measured in mole) of the receipted modulator is directly proportional to the learning parameter η for basic Hebbian learning:

$$\Delta w_{j,i} = \eta \, a_i \, a_j \,, \tag{3}$$

where ai, aj are the pre–, and postsynaptic activations, respectively, of the neurons connected by the weighted link. Note that the weight change □wj,i only takes place, when an ANM is received by a neuron. By setting the emission rate larger than the (summed) reception rate of all neurons, it is possible to easily introduce a kind of short–term memory, as it takes a number of time steps (simulation cycles) to fully absorb the modulator.

4.1 Hebbian Learn Rules

Table 1 gives the definition of the four Hebbian learning rules that are used in the experiments.

Table 1. Variants of Hebbian learn rules.

Hebb (H)	$\Delta w_{j,i} = \eta \, a_i \, a_j$
Anti–Hebb (AH)	$\Delta w_{j,i} = -\eta \, a_i \, a_j$
Covariance Hebb (CH)	$\Delta w_{j,i} = \eta \left(a_i - \bar{a}_i\right)\left(a_j - \bar{a}_j\right)$
Covariance Anti-Hebb (CAH)	$\Delta w_{j,i} = -\eta \left(a_i - \bar{a}_i\right)\left(a_j - \bar{a}_j\right)$

The parameters a_i and a_j are the pre–, and postsynaptic mean activations, respectively, being defined as the running mean of the neuron's activation from *t*=0 ("birth") to the current time *t*.

4.2 The Wall Avoidance Experiment

In this experiment the robot is placed in a rectangular arena (1.05×0.70 m) and should learn to avoid wall contact. We performed experiments with 500 simulated robots initialized with different random weights and biases from the interval [-1.0,1.0]. The learning behavior is evaluated by a *Learn Ability* calculated in the following way:

1. Every robot is placed into each of the four corners (in a distance of ten cm to the walls) and then moves freely (without learning) for ten minutes. Throughout these 40 minutes we measure the time t_{pre} it is in contact with the wall.

2. The robot is placed in the upper left corner with activated learning (modulators are diffused by the contact sensor neuron). From now on the robot has two hours to learn the task.

3. After the learning procedure the robot is tested in the same way as described in 1 measuring the wall contact time t_{post}.

The learn ability L_{WA} is defined as

$$L_{WA} = \frac{t_{pre} - t_{post}}{t_{pre} + t_{post}} \,. \tag{4}$$

We study the impact of different learning rules on the learning behavior of the robot using the mean learn ability *L* of the robots. Note that a number of robots avoid the wall without any learning, which we labelled *Genius*, as they perfectly master the task right from the time of "birth". Genius robots are not considered for calculation of the mean learn ability. The learn ability *L* is 1.0, if the robot has learned the task perfectly, e.g., never touches the wall after training. An L>0.0 indicates an improvement after learning, while an L<0.0 is the sign of a negative effect of training, i.e., the robot exhibits a worse behavior than before learning.

The learn ability is influenced by the dose d of modulator, which is emitted by the contact sensor neuron at wall contact. The emission rate is set to 12 mole per second. Each neuron in the network is able to receipt this modulator. The reception rate is set such that the complete amount of modulator diffused in one time step is consumed by all neurons at equal parts in the next time step. The consumed dose is directly mapped to the learn rate η of the neurons' pre–synaptic links, e.g., if a neuron consumes 1.0 mole of the modulator η=1.0. Note that in this setting the ANM concept only mediates start and stop of Hebbian learning with a specific learn rate. While this procedure has appealing biological analogies, it could be equally implemented in a simple algorithmic way. However, changes in the emission and/or reception rate would immediately introduce complex temporal interactions of feedback signals and weight changes.

5. RESULTS

Employing negative Hebb learning as the reaction to the received modulator in the wall avoidance experiments a number of robots is able to avoid the wall after a few collisions. Other robots (each "born" with a different random brain) take some minutes (real–time simulation) to learn the task, while a few never learn it, and sometimes always remain in contact with the wall. All robots learning the task develop an intuitively expected behavior of slowing down, when approaching a wall, and starting to turn away from the wall, then accelerating into "open terrain". A typical motion trail of a robot having quickly learned the task can be seen in Fig.Fig.3.

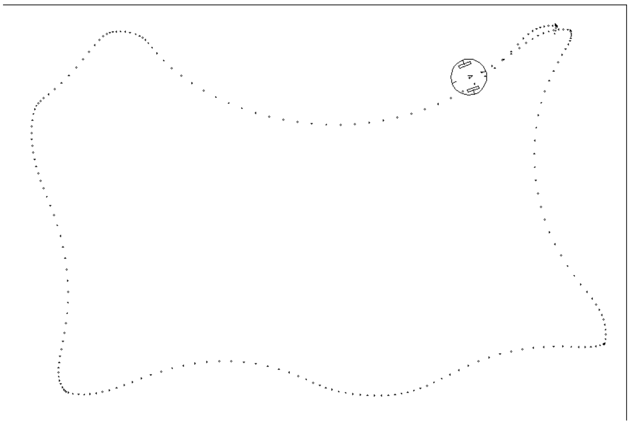

Fig.3 – A typical motion trail of a wall avoiding robot after ontogenetic learning.

In a number of experiments (Table 2) we expectedly saw that the type of learning reaction has a dramatic influence on the robot's learning ability. We also noticed that successful learning is greatly influenced by a detail neglected in most previous work on Hebb learning. Usually, the learn ability of the robot is considerably improved, when the bias values of a neuron are not subjected to Hebb learning, i.e., they remain constant. Consequently, we also present results comparing fixed with learned bias values in Table 2.

Table 2. Learn abilities L of 500 wall avoiding robots using different learn rules with (yes) or without (no) bias learning. The contact sensor neuron is synaptically connected to the hidden layer.

	AH, yes	AH, no	H, no	CAH, no	CH, no
L = 1	40	52	61	52	73
L > 0	27	161	55	180	138
L = 0	0	0	0	8	4
L < 0	235	83	205	62	97
Genius	198	204	179	198	188
L	-0.222	0.417	0.028	0.457	0.451
d	64880	12177	54260	15984	23219

As indicated above, training of initially random bias values results in a much worse learning ability L (averaged on all trained robots) than excluding the bias from training (genius robots never touch a wall, hence, they are never trained, and do note contribute to L).

When employing AH learning, the weights are decreased in each learn step being triggered by wall contact. As a consequence, a trained robot has mostly (large) negative weights. If it approaches the wall of the arena, a strong signal is generated by one of the distance sensors, which leads to low activation (close to zero) of the hidden neurons. Then, the activity of the motor neurons is only determined by its bias values. If the bias values are fixed and different for the two motor neurons, the robot will turn, which is what it should learn to do near the wall. However, if the bias values are subjected to AH learning as well, they will mostly become negative resulting in zero activation of the motor neurons, actually moving the robot straight with full reverse speed.

The essence of these considerations is that in this case the learn ability of the robot is only dependent on its fixed bias values given at "birth". AH learning more and more reveals the basic "character" of the robot, but it does not change this character. Thus, learn ability is only determined by traits already existing at the time of the robot's "birth". The results in Table 3 confirm this observation, but they also show that CAH Learning does not depend on the initial bias values.

Table 3. Learn abilities L of 500 wall avoiding robots with bias values fixed to 0.0.

	AH	CAH
L = 1	48	72
L > 0	77	176
L = 0	0	3
L < 0	184	67
Genius	191	184
L	0.022	0.485
d	47554	15784

With bias values fixed to 0.0 AH learning achieves a much smaller learn ability than with fixed random values, as the key to successful learning in this setting is a difference in the bias values of the output neurons (enabling turning behavior).

CH learning does not only not exhibit this dependency, but also is successful regardless of the positive or negative variety. There are a number of possible explanations to this behavior. This Hebb variant allows weight changes in both directions even in the same learn step (simulation cycle). The mean activations represent a very basic form of memory, which makes learning dependent on time, or in other words on the robot's age. Learning is also dependent on the mobility of the robot. A robot mostly staying in a certain area of the arena, will process similar input signals most of the time leading to a convergence of the mean activations. If the same robot moves to another area, the difference of input signals to the mean activations commanding the actual weight change will be larger (stronger learning) than for a more mobile robot. Putting all together and considering that learning only takes place at certain points in time (wall contact) the complexity of this still simple Hebb variant becomes obvious.

Naturally, the dose of the emitted modulator contributes to the learning process of the robot. Thus, we measured how the learn ability of the robots is influenced by the modulator dose. Comparing AH and CAH learning (fixed random bias) in Fig.4 reveals interesting properties.

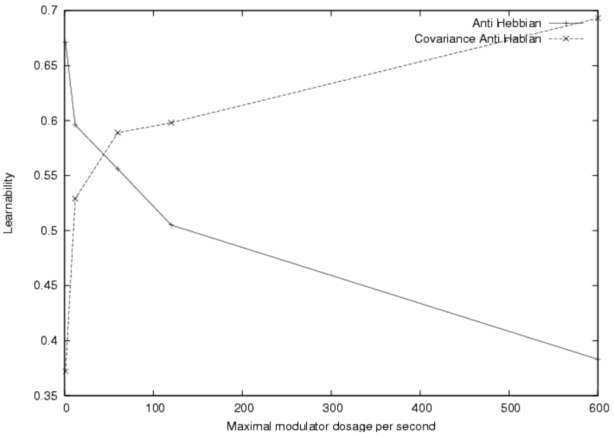

Fig.4 – The dependence of the learn ability on the modulator rate for wall avoidance.

The increase of the rate of modulator emission is balanced with a proportional increase of the reception rate. Hence, the complete dose of modulator emitted in a time step is consumed in the next triggering the given type of learning. In case of the CH rule the weight changes are in both directions. Increasingly strong learn signals lead to a complete perturbation of the network weights, i.e., a new random network. With the simple wall avoidance task it might only take a few wall contacts until a genius contributing to improved learn ability is found. High modulator doses in combination with AH learning make the robots more and more insensitive to the input signals, as even weak sensor signals lead to deactivation of the hidden layer. Hence, the robot no longer switches between a behavior close to the wall and a different one in "open terrain". If the robot moves in a rather straight manner, wall contacts are inevitable, and the strong learning signals further enforce the singular behavior.

5. SUMMARY

The results show that ontogenetic learning of mobile autonomous robots with neurocontrollers regulated by external feedback mediated by ANMs is sufficient to teach robots simple tasks. However, we have found that the learning ability of the robots is dependent on parameters that are randomly assigned at the "birth" of the robot. The crucial question to be addressed in future research is, if there exists an unsupervised learning method allowing the robot to correctly interpret the feedback signals so as to learn the appropriate behavior. In conventional reinforcement learning the problem of interpretation is solved by assigning values to actions, while in this work we investigate the classical neural mapping of sensor to motor (action) signals. Assuming that Hebbian learning plays an important role in BNNs, the finding that a sensor–motor neurocontroller cannot be generally trained by unsupervised learning, would possibly imply that biological systems rely on action–value networks as suggested by various researchers.

7. REFERENCES

[1] Stefano Nolfi and Dario Floreano. *Evolutionary Robotics – The Biology, Intelligence, and Technology of Self–Organizing Machines*. MIT Press, 2000.

[2] G. M. Shepherd. *Neurobiology*. Oxford University Press, 3rd edition, 1994.

[3] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[4] William D. Smart and Leslie Pack Kaelbling. Effective Reinforcement Learning for Mobile Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002.

[5] Akio Ishiguro, Siji Tokura, Toshiyuki Kondo, Yoshiki Uchikawa, and Peter Eggenberger. Reduction of the Gap between Simulated and Real Environments in Evolutionary Robotics: A Dynamically–Rearranging Neural Network Approach. In *IEEE Systems, Man, and Cybernetics Conference*, pages III – 239–244. IEEE, October 1999.

[6] Tom M. C. Smith and A. Philippides. Nitric Oxide Signalling in Real and Artificial Neural Networks. *British Telecom Technology Journal*, 18(4): 140–149, October 2000.

[7] E. Oja, J. Karhunen, L. Wang, and R. Vigario. Principal and Independent Components in Neural Networks – Recent Developments. Technical report, Helsinki University of Technology, 1996.

[8] S. Nolfi, O. Miglino, and D. Parisi. Phenotypic Plasticity in Evolving Neural Networks. In *Proceedings of the First Conference From Perception to Action*. IEEE Computer Society Press, 1994.

[9] Gerald M. Edelman. *Neural Darwinism – The Theory of Neuronal Group Selection*. Basic Books, New York, 1987.

[10] Jari Vaario, Akira Onitsuka, and Katsuo Shimohara. Formation of Neural Structures. In *Proceedings of the Fourth European Conference on Articial Life*, pages 214–223. MIT Press, 1997.

[11] Dario Floreano and Francesco Mondada. Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, 11(1998): 1461–1478, 1998.

[12] L. Steels and R. Brooks, editors. *Building Situated Embodied Agents. The Alife route to AI.*, New Haven, 1995. Lawrence Erlbaum Ass.

[13] Luc Steels. The origins of intelligence. In *Proceedings of the Carlo Erba Foundation, Meeting on Artificial Life*, 1996.

[14] Luc Steels. The Spontaneous Self–organization of an Adaptive Language. In S. Muggleton, editor, *Machine Intelligence 15*, Oxford, 1996. Oxford University Press.

__Helmut A. Mayer__ received the master's degree in electrical engineering in 1987 from the Technical University of Vienna, Austria, and the doctoral degree in natural sciences in 1997 at the University of Salzburg, Austria, where he also gained the habilitation in informatics in 2004.

From 1988 to 1993 he worked as a software engineer for automation and control systems with B & R, Eggelsberg, Austria. He now is an associate professor at the Department of Scientific Computing at the University of Salzburg, where he has been working since 1993. His research interests include evolutionary computation, artificial neural networks, mobile autonomous robots, computational neuroscience, and real-world applications.