Self-Organizing Neurocontrol

T. Fomin, Cs. Szepesvári, and A. Lörincz

***Abstract*—Self-organizing neural network solutions to control problems are described. Competitive networks create spatial filters and geometry connections in a self-organizing fashion. The goal position, the obstacles and the object under control all create neural activities through the filters. Spreading activation that discriminates between the controlled object, the goal position and the obstacles is utilized on the internal representation. Local self-training method and Hebbian learning develops the self-organizing control connections. The algorithm provides maneouvring capability in unseen scenes.**

I. Introduction

In this paper we are introducing the method of self-generated examples for self-organized learning of motion control. An original method e.g. applies Kohonen discretization of the external world and sets up a linear approximator for the control around each neuron ([1]). Feedback from an external observer is needed to produce error signal for learning. The error is then propagated backwards. This counterpropagation approach is enlightening: it creates an internal representation of the world. The drawbacks of the model are: (i) it may be applied if the topology of both the work place and the control space fit the prewired topology of the Kohonen network, (ii) it does not lead to self-organized path planning and is thus unprotected against obstacles. Our first solution ([2]) that overcomes problem (i) applies a combination of *Competitive* learning for spatial filter (SF) formation and develops geometry connections between neurons via *Hebbian* (H) learning. The competitive part may than be weakened to soft competition with the help of Kohonen like neighbour training through the self-developed connections. The second solution applies Hebbian and anti-Hebbian (HAH) learning [3]. HAH approach provides neighbour training through its soft competition property in an inherent fashion. The network is similar to that of [4]. Figure 1 shows a typical spatial filter formed by HAH learning [3].

The authors are with the Department of Photophysics, Institute of Isotopes The Hungarian Academy of Sciences Budapest, Hungary 1525. Cs. Szepesvári is with the Department of Mathematics, Attila József University of Szeged, Szeged, Hungary 6720

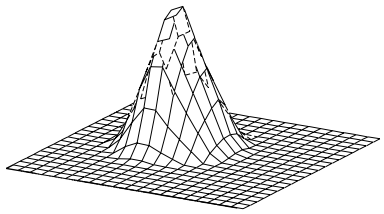

Figure 1: Typical spatial filter

The second part of the problem is to connect the internal representation to the control field in a self-organizing fashion. One is seeking for a method that is capable of avoiding obstacles if those are in the way. It has been shown by Lei [5] that a version of spreading activation (SA) neural methods may be used for path planning in the presence of obstacles. It has also been shown, that the linear version of that model finds the shortest path between any start and goal positions and for an arbitrary set of obstacles [6]. A two-layered prewired version of the Lei model was suggested for neurocontrol [7]. The self-developed internal representation of the geometry of the external world is a suitable candidate to perform SA with: In order to provide an error, or a reinforcement signal we do not need an external agent if the system has goals and is equipped with activation spreading on the internal representation. Changes in the activity of the neuron(s) activated by the goal can provide the necessary information. In other words, when controlling the motion of our hand, for example, the position of that hand is not given to us: we judge our performance from changes in the internal representation. The solution assumes

a module that can distinguish between the object under control, obstacles and the goal to be reached, e.g. the self-organizing system of [8].

II. Setting up neurocontrol

A preliminary algorithm may be designed as follows: The object under control is presented to the network. Active neurons become the sources of SA and they will spread activation in the procedure. Obstacles (recognized by a larger framework) are presented to the network. Neurons that develop activities have receptive fields that overlap with the obstacle. These receptive fields should be avoided. The goal object is presented to the network and creates an activity pattern. Spreading activation from the object under control to the goal can provide a self-reinforcement signal for a given move. We shall discuss two possibilities: For neurocontrol, version I there is a positive reinforcement if the flow to these neurons, i.e. if spreaded activity gradient at these neurons increases. For neurocontrol, version II reinforcement is positive if the spreaded activity at the "goal neurons" increases after the move was made. For version II two examples are shown in Figs. 2 and 3.

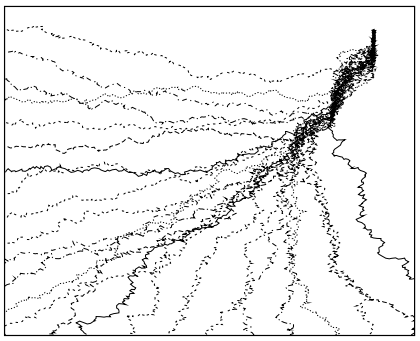

Figure 2: Untrained motion in free space (Neurocontrol, version II)

In both cases a move was accepted if the reinforcement signal was positive and was rejected otherwise. Fig. 2 is the case of free space. Fig. 3 shows motion in the presence of an obstacle when neurons activated by the obstacle are blocked and cannot spread activities. One would like to train connections of units that could learn from these trial-and-error examples. Since spreading activation is along the geometrical connections of the internal representation it has the information about better routes (strong flow) and worse routes (weak flow). That is we have to introduce units that monitor the flow itself.

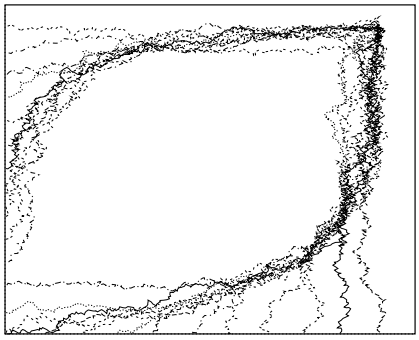

Figure 3: Untrained motion around an object (Neurocontrol version II)

III. Neurocontrol, version I.

The model for neurocontrol is shown in Fig. 4. The bottom of the figure depicts the detector array, arranged in a line. The middle of the figure shows the spatially tuned neurons (prewired in this example). Connections between these two layers represent the SF connections. Connections between the neurons of the middle layer are the intralayer

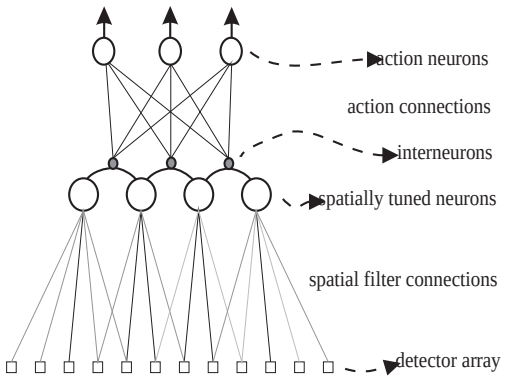

Figure 4: Hierarchy of self-organizing neurocontrol

geometry connections. There are two connections between every neuron (one is shown), identical in strength. We should, however, keep both since opposite motions may need different motor neurons. There are small grey dots on every geometry connection, these are the interneurons. Interneurons detect the flow in SA. Interneurons activate the action neurons through the action connections and according to their excitation level governed by SA. The effect of motor neurons is prewired: in our 2D example there are four motor neurons: 'up', 'down', 'left' and 'right' starting motion accordingly.

The first solution applies the linear version of Lei's

SA with prewired SF's and geometrical connections (GC). Motion control is preceded by a learning phase: the object of control is presented to the network. Active neurons become the sources of spreading activation. Activation spreads and relaxes according to the following equation:

$$\frac{du_i}{dt} = \sum_{j=1, j\neq i} T_{ij}(u_i - u_j) + I_i \qquad (1)$$

where u_i is the activity of the i^{th} neuron governed by input set $\{I_i, i = 1, 2, \ldots, n\}$ where n is the number of neurons. A move is made by giving random activation values to the interneurons. The *new position of the object is selected as the goal* [9]. Active neurons will be the sinks of spreading activation. Spreading activations starts and relaxes according to (6). Interneurons produce activities proportional to the SA flow. The largest activity source neuron, N_j is selected. The largest activity interneuron R_{jk} around N_j is selected. It detects SA from neuron N_j to neuron N_k. (Without selections learning slows down considerably.) Action connections of the winning interneuron are trained by modifying the weight vectors:

$$\mathbf{h}_{jk}^{\text{new}} = \frac{(\mathbf{h}_{jk}^{\text{old}} + \gamma\mathbf{m})}{(|\mathbf{h}_{jk}^{\text{old}} + \gamma\mathbf{m}|)} \qquad (2)$$

$\mathbf{h}_{jk}$ and $\mathbf{m}$ denote the action connections of the winning interneuron and the randomly selected motion combination, respectively. In the controlling phase the motion vector is composed of the vectorial sum of the motor connections filtered interneuron activities (Fig. 5).

IV. Neurocontrol, version II.

SF's (32 in number) were now, self-developed via HAH learing (see Fig. 1) Motion control is again preceded by a learning phase: after the small random move (a few pixel distance) one can detect zero$^{\text{th}}$ order SA, $r_{jk} = s_j w_{jk} g_k$ for all interneurons R_{jk}, with GC w_{jk}, object and goal spatial filtered input activities s_j, g_k, respectively. All interneuron to motor neuron connections (IMNC), $h_{jk,l}$ are trained with H learning in a distributed fashion:

$$\Delta h_{jk,l} = \varepsilon h_{jk,l} + (1 - \varepsilon) r_{jk} m_l \qquad (3)$$

where m_l is the actual action made by motor neuron l. Motion control is regulated as follows: (i) take the average $\overline{R}$ for r_{jk} values for all j and k indices during training, (ii) allow spreading:

$$r_{jk}(\tau) = s_j(\tau) v_{jk}(\tau) g_k(\tau) \qquad (4)$$

$$b_l(\tau + 1) = \sum_i b_i(\tau) v_{il}(\tau) \qquad (5)$$

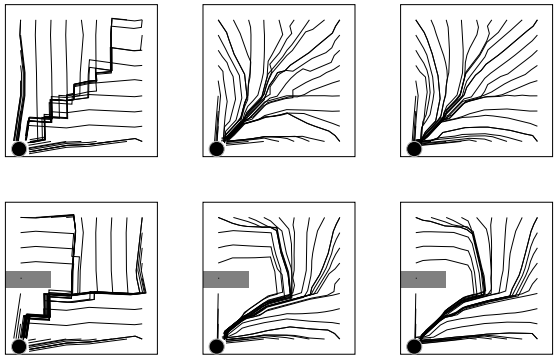

Figure 5: Motion trajectories, version I

Motion in free space and in the presence of an obstacle. Right row: motion with adding up all the interneural contributions, middle row: as the right row but only for the highest activity neuron, left row: as the middle row but only with the highest activity interneuron.

where both $s(\tau)$ and $g(\tau)$ are subject procedure Eq. 7 and v_{jk} is the regularized w_{jk}:

$$v_{jk}(t) = \xi(t) w_{jk} \qquad (6)$$

$$\Delta\xi(\tau) = \kappa\left(\overline{R} - \frac{1}{N}\sum_{j,k} r_{j,k}(\tau)\right), \qquad (7)$$

where N is the number of all j, k connections. The regularization allows spreading to remote goal positions. Motion is again accomplished by summing up and normalizing all IMNC filtered inputs to motor neurons in free space (Fig 6). Figure 7 shows

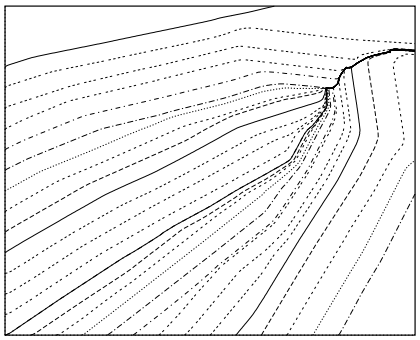

Figure 6: Motion in free space, version II.

motion of the trained network when an obstacle is present. Short range obstacle repulsion is introduced by adding an unspreaded negative term to the IMNC summation:

$$t_{jk} = -\zeta s_l(0) w_{ij} o_j(0) \qquad (8)$$

where $o_j(0)$ is the obstacle input activity.

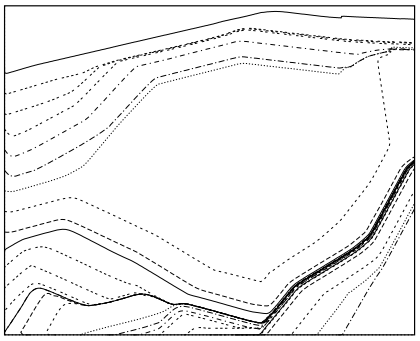

Figure 7: Motion in the presence of an obstacle, version II.

V. Conclusions

The suggested methods of self-organizing neurocontrol was shown viable. Learning rules were kept local in the procedures. The procedure is self-organizing at every step. The learning rule is a variant of H learning at each stage. The technique of self-generated examples – or exploration – was used in training. This approach is fast, robust and local: it learns motion and motion control together. It gives manoeuvring capability on unseen scenes.

References

[1] T. Martinetz H. Ritter and K. Schulten. 2:159–166, 1988.

[2] Cs. Szepesvári, L. Balázs, and A. Lörincz. Topology learning solved by extended objects: a neural network model. *Neural Computation*, 1993. in press.

[3] T. Fomin and A. Lörincz. Positioning and feature extraction with Hebbian and anti-Hebbian networks. *Neural Networks*, 1993. submitted.

[4] P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

[5] G. Lei. *Biological Cybernetics*, 64:61–67, 1990.

[6] D. Keymeulen and J. Decuyper. In *Toward a practice of autonomous systems*, pages 64–69. Cambridge: MIT Press, 1992.

[7] R. Glasius, A. Komoda, and S. Gielen. In *ICANN '93: Int. Conf. on Artificial Neural Networks*, pages 646–648. London, Springer-Verlag, 1993.

[8] Cs. Szepesvári and A. Lörincz. Behavior of adaptive self-organizing autonomous agent working with cues and competing concepts. *Adaptive Behavior*, 1994. in press.

[9] Cs. Szepesvári and A. Lörincz. Self-organizing neurocontrol. *Neural Networks*, 1993. submitted.