

## PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/23466>

Please be advised that this information was generated on 2022-08-24 and may be subject to change.

# A biologically inspired neural net for trajectory formation and obstacle avoidance

Roy Glasius, Andrzej Komoda, Stan C.A.M. Gielen

Department of Medical Physics and Biophysics, University of Nijmegen, Geert Grooteplein Noord 21, 6525 EZ Nijmegen, The Netherlands

Received: 22 August 1994 / Accepted in revised form: 6 February 1996

**Abstract.** In this paper we present a biologically inspired two-layered neural network for trajectory formation and obstacle avoidance. The two topographically ordered neural maps consist of analog neurons having continuous dynamics. The first layer, the sensory map, receives sensory information and builds up an activity pattern which contains the optimal solution (i.e. shortest path without collisions) for any given set of current position, target positions and obstacle positions. Targets and obstacles are allowed to move, in which case the activity pattern in the sensory map will change accordingly. The time evolution of the neural activity in the second layer, the motor map, results in a moving cluster of activity, which can be interpreted as a population vector. Through the feedforward connections between the two layers, input of the sensory map directs the movement of the cluster along the optimal path from the current position of the cluster to the target position. The smooth trajectory is the result of the intrinsic dynamics of the network only. No supervisor is required. The output of the motor map can be used for direct control of an autonomous system in a cluttered environment or for control of the actuators of a biological limb or robot manipulator. The system is able to reach a target even in the presence of an external perturbation. Computer simulations of a point robot and a multi-joint manipulator illustrate the theory.

approach is useful in practice only in a static environment, since the time required to construct a graph and perform the planning task becomes excessively long for a large number of obstacles.

As an alternative the idea of potential fields was proposed in which the target acts as an attractor in the midst of obstacles represented by repelling potentials (Kathib 1986; Krogh and Thorpe 1986). Unfortunately, the potential field methods suffer from several problems, one being that of undesired local minima. Although several suggestions have been made to circumvent these problems (Newman and Hogan 1987; Warren 1989; Barraquand and Latombe 1990; Conolly et al. 1991; Barraquand et al. 1992), no algorithm was presented which guarantees a solution. Glasius et al. (1994) proposed a neural network with continuous time dynamics that also suffers from local minima. However, it improved upon previous models in the quality of the path which is smooth and continuous and which results simply from the intrinsic dynamics of the network only. The basic elements of this model are similar to the neural fields model of Amari (1977; see also Kopecz and Schöner 1995).

In another approach a neural network type was proposed (Lozano-Perez 1983; Schwartz and Sharir 1983; Dorst et al. 1991) in which neurons with lateral interactions are positioned on the nodes of a grid. Due to lateral interactions activity from the target spreads through the network (except toward neurons representing obstacles). The optimal trajectory to the target is found by the sequence of neurons which is obtained by stepping along the gradient ascent to the peak of activity in the network, i.e., the target. Although this procedure is shown to give the optimal path, it is not an unsupervised method, since it requires an external observer to compare the activities of neighboring neurons in order to decide which neuron should be the next in the path towards the target. Recently, variations of this distance-transform or wave-propagation method have been proposed (Prassler 1989; Glasius et al. 1995) using topographically ordered maps.

Although supervised methods can be useful for applications, they cannot be used for biologically plausible models for trajectory planning and formation. However, unsupervised methods have been reported for topographically or-

---

## 1 Introduction

One of the main themes of research on robot control deals with the issue of trajectory planning and formation. The aim of this research is to develop algorithms that allow autonomous systems or manipulators to move in a cluttered environment, which may change as a function of time, to one of multiple targets along a path without collision.

The first steps to deal with this problem were based on global methods which can be considered as a search process for a path in a graph which represents the accessible paths along objects (Barraquand and Latombe 1989). This



dered maps and hierarchically organized topographically ordered neural networks are frequently found in the nervous system. With the local learning rule, these properties give the models by Glasius et al. (1994, 1995) and Prassler (1989) a biologically plausible basis. In this paper we present a neural network which is an amalgamation of our two previous models. Unlike the previous models it does not suffer from local minima and does not require an external observer. Our model guarantees a trajectory from any initial position to a target (if one exists) given an arbitrary environment with obstacles, which are allowed to change position. The trajectory is the result of internal dynamics of two layers of neurons with feedforward connections and is the shortest path in terms of the metric of the representation (Latombe 1991). The biological features that are contained in our model (i.e., layered topographic maps, large numbers of continuous valued neurons, analog dynamics and the use of neighborhood functions) result in a robust model which generates smooth and continuous paths.

This paper is organized in the following way. First we will describe the model and its dynamics. The theoretical results will be supported by numerical computer simulations, which demonstrate the performance of the model for non-trivial problems of trajectory planning. In the Discussion we will compare the results of the model with those of previous models. We will also discuss to what extent the model can be used as a biologically plausible model which captures the results on trajectory planning in the neurophysiological literature and which can be used to propose some new hypotheses.

## 2 The model

The model which we will present can be used for path planning and trajectory formation in a finite ( $D$ -) dimensional state space of any system. The state space  $\mathcal{C}$  can be a Cartesian work space or the configuration space of a multi-joint manipulator. Let us regard a topographically ordered neural map in which the activity of each neuron  $i$  is related to a subset in  $\mathcal{C}$ , called the receptive field  $rf_i$  of neuron  $i$ . As in visual neurophysiology, where the receptive field refers to that part of visual space where visual stimuli will affect the response of a neuron, we will use the term 'receptive field' as that part in stimulus space which affects the responses of the (artificial) neurons in this study. Related to the topographical ordering, neighboring neurons have neighboring receptive fields in  $\mathcal{C}$ , which in general will have some overlap. For the performance of the type of network presented in this paper the precise shape of the receptive field is not important (Amari 1977) and in our simulations we will use different shapes of receptive field. Each receptive field  $rf_i$  can be represented by a  $D$ -dimensional vector  $\theta_i \in \mathcal{C}$  which points to the center of the receptive field. Hence, to each neuron  $i$  corresponds a receptive field  $rf_i$  and a vector  $\theta_i$ . If the receptive fields cover the complete state space

$$\mathcal{C} = \bigcup_{i=1}^N rf_i \quad (1)$$

then the grid, consisting of all  $\theta_i$ , is a discrete representation of the state space  $\mathcal{C}$ .

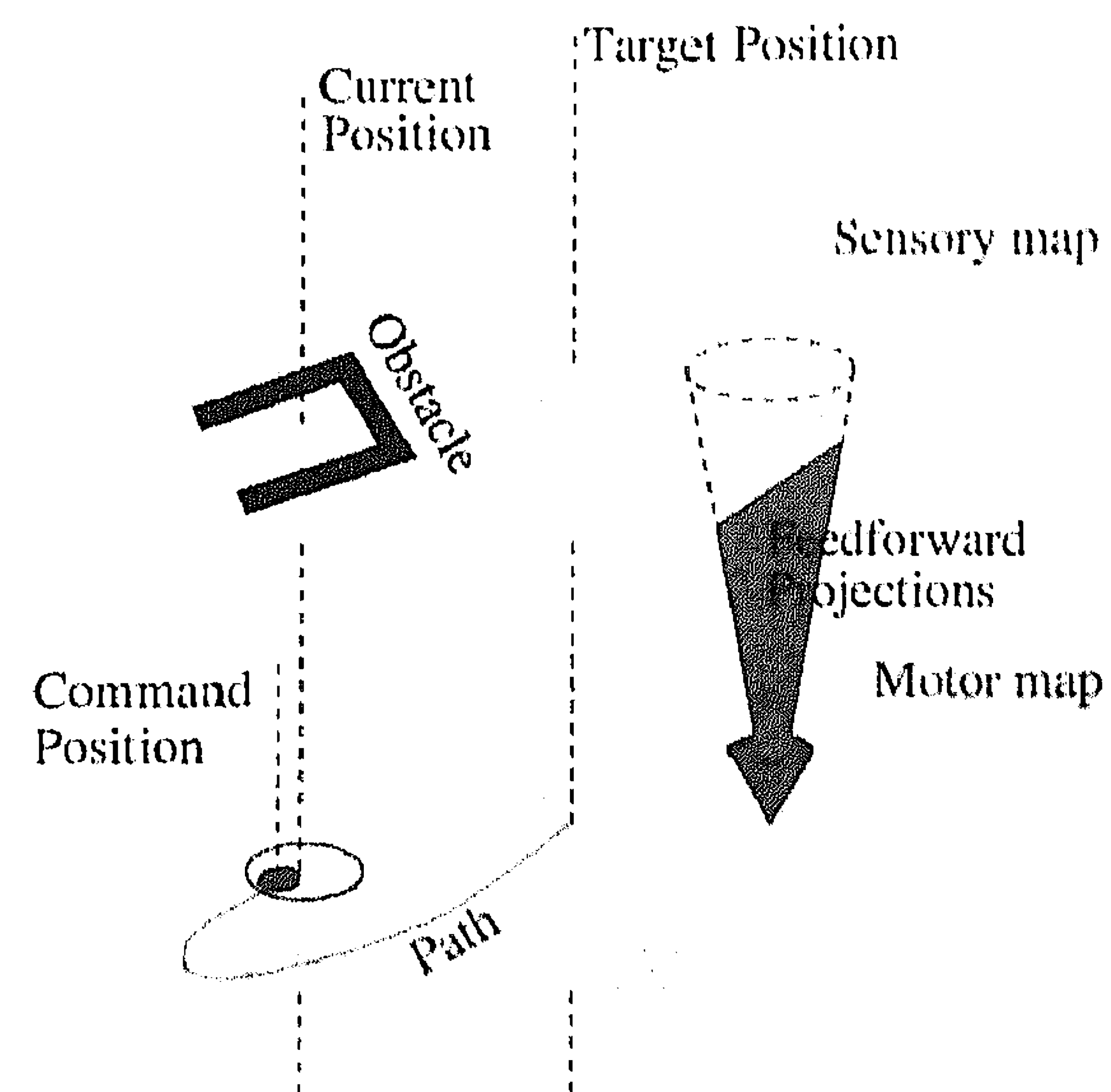


Fig. 1. A schematic representation of the two-dimensional sensory map and motor map as a two-layer feedforward neural network

Topographically ordered neural maps can be obtained in several ways [for example by using Kohonen's learning rule for topographically ordered maps (Kohonen 1982; Ritter et al. 1989) or by a vector-quantization learning rule (Ritter et al. 1989; Fritzke 1991; Martinetz 1993)]. Such topographically ordered maps are used in both layers of our neural network model. The first will be denoted as the 'sensory map', because it is assumed to represent sensory information about the environment of the system. The other map will be referred to as the 'motor map' because its activity codes action commands. To distinguish the variables and parameters of the two maps, the variables and parameters in the sensory map will have indices  $i$  and  $j$  and those in the motor map indices  $k$  and  $l$ .

The sensory map projects to the motor map with feedforward connections which will be defined later (see 8). For a two-dimensional system the architecture of the neural network containing the topographical maps and the connections are illustrated schematically in Fig. 1.

The activities of the neurons in both maps are characterized by real-valued variables

$$\sigma_i, \sigma_k \in [0, 1]$$

where  $\sigma_i$  denotes the activity of neuron  $i$  in the sensory map and  $\sigma_k$  that of neuron  $k$  in the motor map.

The input to a neuron, the local field, is defined as the weighted sum of the activities of all neurons in the network which project to that neuron minus a threshold. The local field for neurons is denoted in the sensory map by  $u_i$  and for neurons in the motor map by  $u_k$ . The activity of a neuron is given by the value of a sigmoid function on the local field. Our choice for the sigmoid function is

$$g(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \beta x & \text{if } x \in [0, 1] \\ 1 & \text{if } x \geq 1 \end{cases} \quad (2)$$

with  $\beta \in [0, 1]$ . However, it could be any bounded function between 0 and 1 which is monotonically increasing. Simulations demonstrated that the results of this study did not depend on the type of sigmoidal function.

Consequently, the neuronal activities are

$$\begin{aligned} \sigma_i(t) &= g(u_i(t)) && \text{for neurons in the sensory map} \\ \sigma_k(t) &= g(u_k(t)) && \text{for neurons in the motor map} \end{aligned}$$



## 2.1 The sensory map

We assume lateral excitatory short-range connections  $B_{ij}$  from neuron  $j$  to neuron  $i$  in the sensory map:

$$B_{ij} = f(|\theta_i - \theta_j|) \quad (3)$$

where  $f(x)$  is a monotonically decreasing neighborhood function and  $|\theta_i - \theta_j|$  is the Euclidean distance in state space between  $\theta_i$  and  $\theta_j$ . For example, a commonly used (Linsker 1986a,b,c) function is

$$f(x) = \begin{cases} e^{-\gamma x^2} & \text{if } |x| \in [0, r] \\ 0 & \text{if } |x| > r \end{cases} \quad (4)$$

with  $\gamma$  and  $r$  real positive numbers.

External input is assumed to provide information about target and obstacle positions. The sensory neuron the receptive field of which contains the target is called the target neuron. The state of the target neuron  $t$  is supposed to be equal to the value  $\sigma_t \approx 1$ , i.e., maximally activated. This can be achieved by a large input from external sensory neurons through excitatory connections. For simplicity we will assume that the size of a target position is smaller than the size of a receptive field such that there is only one target neuron at each moment in time. The set of positions or the states in  $\mathcal{C}$ , which cannot be reached because of obstacles, will be represented by  $\{S\}$ .  $\{S\}$  is defined both by the shape of the obstacles and by the shape of the robot. The set of neurons  $i \in \{S\}$  is supposed to be deactivated by a large inhibitory input from external sensory inputs, such that these neurons have a low activity, i.e.,  $\sigma_{i \in \{S\}} = 0$ . Because target and obstacles may move as a function of time, the neurons, which represent target and obstacles may change continuously.

In conclusion, the external sensory input to neuron  $i$  in the sensory map is given by

$$I_i = \begin{cases} v & \text{if target} \in \text{rf}_i \\ -v & \text{if obstacle} \in \text{rf}_i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where  $v$  is a positive number, large enough to dominate over all other inputs. The threshold of the neurons in the sensory map is set to zero.

The total input to a neuron in the sensory map is given by

$$u_i(t) = \sum_j^N B_{ij} \sigma_j(t) + I_i$$

The dynamics for the neurons in the sensory map can be formulated by the change in the neural input  $u_i$  per unit of time

$$\frac{du_i(t)}{dt} = \sum_j^N B_{ij} \sigma_j(t) + I_i - u_i(t) \quad (6)$$

The function

$$L(\sigma_i) = -\frac{1}{2} \sum_{i,j} B_{ij} \sigma_i \sigma_j - \sum_i I_i \sigma_i + \sum_i G(\sigma_i)$$

with  $G(\sigma_i) = \int_0^{\sigma_i} g^{-1}(x) dx$ , is a global Lyapunov function (Hopfield 1984; Grossberg 1988; Glasius et al. 1995) for the

sensory map. The existence of a Lyapunov function guarantees that the dynamics of the sensory layer always converges to a fixed point.

Due to the excitatory local interactions, the target neuron, kept at a high activity, feeds its nearest neighbors through the short-range connections, such that the activity spreads through the network. As shown in a previous paper (Glasius et al. 1995), the activation landscape which arises in the sensory neural map, is shaped like a hill with its top at the target neuron and with valleys at the locations of the obstacle neurons. As we have proved (Glasius et al. 1995), this activation landscape is unique and the condition for equilibrium  $[du_i(t)/dt] = 0$  guarantees that no plateaus can exist. Therefore, for each given initial position, target position and the set of obstacles, the sensory map contains the optimal solution (Glasius et al. 1995) constructed by successive gradient ascents to the top.

This neural activity field is passed to the motor map neurons, and, in addition, neurons in the motor map receive an external input which codes the current state of the system. The motor map has to construct a motor command which will alter the current position in a direction along the path towards the target.

In the case of moving objects, the set of obstacle neurons and the set of target neurons may change continuously in time. The activation landscape in the sensory map changes accordingly and if the obstacles move slowly relative to the time constant of the system dynamics, then the sensory map will always contain adequate information for the path planning problem.

## 2.2 The motor map

The motor map receives inputs through feedforward connections from the sensory map. In addition, the motor map receives input which provides information about the current state of the system. This input could originate from sensors in the actuators of the limb or manipulator. With both input signals the motor map is able to produce motor commands which will continuously change the current position of the actuator such that it moves from the initial position to the target position.

The motor map has both short-range and long-range lateral interactions. The short-range excitatory interactions are the same as those in the sensory system:

$$B_{kl} = f(|\theta_k - \theta_l|)$$

The lateral long-range interactions are inhibitory with strength

$$J_{ij} = \begin{cases} J_0/N & \text{if } i \neq j, \text{ with } J_0 \in \mathbb{R}^+ \\ 0 & \text{otherwise} \end{cases}$$

where  $N$  represents the number of neurons in the motor map. A motor neuron  $k$  has a constant threshold of

$$J_0\mu - \frac{\sum_l B_{kl}}{2} - \alpha \quad (7)$$

where  $\mu \in \mathbb{R}$  and  $\alpha \in \mathbb{R}^+$  are constants. From (7) one might conclude that each neuron  $k$  may have a different threshold. However, when the neurons are distributed homogeneously



in state space, all neurons have the same number of neighbors. This results in all neurons in the motor map having the same threshold. For neurons at the boundary of the map the threshold, which depends on the number of neighboring neurons [summation over index  $l$  in (7)] has been adjusted to correct for the smaller number of neighboring neurons.

Each state  $\theta \in \mathcal{C}$  of the actuator is represented in the sensory map and in the motor map. The connections from the sensory neuron  $i$  to the motor neuron  $k$  are given by

$$\hat{B}_{ki} = \frac{\alpha}{\sum_j B_{kj}} \hat{f}(|\theta_k - \theta_i|) \quad (8)$$

where  $\alpha$  is the same constant as in (7).  $\hat{f}(x)$  has the same shape as  $f(x)$  but differs in the steepness  $\hat{\gamma} \neq \gamma$  and the radius  $\hat{r} \neq r$ .

These connections provide part of the input signals to the motor map. The other part has an external origin. It provides an input with magnitude

$$K_k(\theta_A) \equiv -v + v \tilde{f}(|\theta_A - \theta_k|) \quad (9)$$

where  $\theta_A$  represents the 'current manipulator position' in  $\mathcal{C}$  and where  $\tilde{f}(x)$  is a similar function to  $f(x)$  with  $\tilde{\gamma} < \gamma$  and with  $\tilde{r} > r$ . The large inhibitory input  $-v$  suppresses all other inputs to neurons in the motor map with a receptive field at a distance larger than  $\tilde{r}$  from the current manipulator position. Therefore, the activity of these neurons is close to the minimal output value of the sigmoid function  $g$ .

With this particular combination of inputs to the motor map the activity in the motor map is grouped in a cluster with an almost constant size. The cluster of activities has to move in the map in a direction supplied by the inputs from the sensory map. In that case the neural activity in the motor map can be interpreted in terms of the position of the manipulator. By defining the following macroscopic quantities we raise our focus from the neuronal to the ensemble representation. We define the 'mean total activity'  $m$

$$m \equiv \frac{1}{N} \sum_k \sigma_k \quad (10)$$

Then we define  $\theta_B$  as the 'command manipulator position'

$$\theta_B(t) \equiv \frac{\sum_k \theta_k \sigma_k(t) \tau_k(t)}{\sum_k \sigma_k(t) \tau_k(t)} \quad (11)$$

with  $\tau_k(t) \equiv [1 + \text{sgn}(\sigma_k(t) - \xi)]$ . This is the weighted mean of all centers of receptive fields of motor neurons that have an activity larger than the threshold  $\xi$ . A neuron belongs to this set if  $\tau_k$  is nonzero.

The vector  $\theta_B(t)$  is the output of the system which is sent to the actuators as a command to realize this position in  $\mathcal{C}$ . However, it takes some time before the actuators receive the command signals and before the command moves the current position to that state. Therefore, the command manipulator position points to what will be the current position after a particular time delay. The command position always leads the current position until it finally reaches the target position.

Because the target position  $\theta_T(t)$  may move in time, more information can be provided using relative coordinates

with respect to the target position instead of absolute coordinates. Therefore we define the 'relative command manipulator position'  $\mathbf{X}$

$$\mathbf{X} \equiv \theta_B - \theta_T$$

Similarly, this vector can be defined as a weighted mean:

$$\mathbf{X}(t) \equiv \frac{\sum_k \mathbf{x}_k(t) \sigma_k(t) \tau_k(t)}{\sum_k \sigma_k(t) \tau_k(t)} \quad (12)$$

with  $\mathbf{x}_k(t) = \theta_k - \theta_T(t)$  as the position in  $\mathcal{C}$  of motor neuron  $k$  relative to the target position. The relative command manipulator position  $\mathbf{X}$  is the weighted mean of all relative vectors of motor neurons that have an activity larger than a certain activity threshold  $\xi$ . In this perspective we can regard this variable  $\mathbf{X}$  as a population vector (Caminiti et al. 1990; Georgopoulos et al. 1992; Kalaska et al. 1992).

The total input to a motor map neuron is given by

$$u_k(t) = \sum_l B_{kl} \sigma_l - J_0(m - \mu) - \alpha \left( 1 - \frac{\sum_i B_{ki} \sigma_i}{\sum_i B_{ki}} \right) + K_k(\theta_A) - \frac{\sum_l B_{kl}}{2}$$

As explained before (Glasius et al. 1994), each term in this expression has a different effect on the behavior of the network. The term  $J_0(m - \mu)$  tends to set the number of active units such that  $m = \mu$ . The local neighbor interactions  $\sum_l B_{kl} \sigma_l$  make it more advantageous for neighboring units to be active. Therefore, this term tends to impose clustering of the active units and causes the cluster to avoid the obstacle. Input from the sensory map causes the cluster to move to the maximum of that input which is the target position ( $\sigma_i \approx 1$  in the sensory map). The external input  $K_k(\theta_A)$  makes sure that the cluster, i.e., the command position, is near the current position (9).

The result of the combined interaction of the terms is a cluster of activities in the neural motor map, moving continuously and smoothly in the direction of the gradient of the local dragging field.

If we make the following substitutions

$$T_{kl} \equiv B_{kl} - \frac{J_0}{N} \quad (13)$$

$$I_k \equiv J_0 \mu - \alpha \left( 1 - \frac{\sum_i B_{ki} \sigma_i}{\sum_i B_{ki}} \right) + K_k(\theta_A) - \frac{\sum_l B_{kl}}{2}$$

then the equation for the input of the motor neurons is similar to the equation for the input of the sensory neurons.

$$u_k(t) = \sum_l^N T_{kl} \sigma_l(t) + I_k$$

Hence, the equations describing the dynamics for the neurons in the motor map are similar to that in the sensory map

$$\frac{du_k(t)}{dt} = \sum_j^N T_{kj} \sigma_j(t) + I_k - u_k(t) \quad (14)$$

resulting in a similar expression for the Lyapunov function,

$$L(\sigma) = -\frac{1}{2} \sum_{k,l} T_{kl} \sigma_k \sigma_l - \sum_k I_k \sigma_k + \sum_k G(\sigma_k)$$



with  $G(\sigma_k) = \int_0^{\sigma_k} g^{-1}(x) dx$ .

The system has the ability to react to external forces acting on the manipulator. If an external force suddenly changes the state of the system, the external input  $K_k(\theta_A)$  changes accordingly and causes an inhibition for all neurons except those in the neighborhood, characterized by  $\tilde{r}$  of  $\tilde{f}$  in (9), of the new current position. If the command position  $\mathbf{X}_B$  is outside this region then the cluster disappears due to the inhibition and reappears at the location with the highest local field (Glasius et al. 1995). Therefore, a change in the current position is followed by a change in the command position which can be interpreted as a reaction to the perturbation in order to adapt the path to the new situation.

If the external force is continuously present, the movement of the current position is due not only to the actuators but also to the external force. The cluster, however, is always trapped in the vicinity of the current position, i.e., the area defined by  $\tilde{r}$  of  $\tilde{f}$ . Inside this area the cluster is always located near the neurons with the highest local fields, i.e., in the direction along the optimal path to the target position. Hence if the force produced by the actuators is strong enough to dominate over the external force then the target will be reached.

The length  $\tilde{r}$  depends on the time delays in the motor system and on the velocity of the manipulator. It takes some time to send the motor command from the motor map to the actuators, to move the actuator and to send the new current position back to the motor map. Consequently the current position always lags behind the command position by a period of time  $\Delta t$ . Depending on the velocity of the cluster in the motor map, as a result of the system dynamics, the length  $\tilde{r}$  in state space can be calculated such that the command position is never further away than the cluster can move in a time  $\Delta t$ .

### 3 Computer simulations

#### 3.1 A point robot in a two-dimensional work space

In the first simulation we regard a point robot moving in a square subset of a two-dimensional Cartesian work space. The initial position of the point robot is in the cavity of a non-convex obstacle (see Fig. 2). The target position is placed at the other side of the obstacle.

The sensory map and the motor map are both two-dimensional neuronal maps, consisting of a square lattice of  $50 \times 50$  neurons. In both maps the receptive fields cover the complete work space according to (1) and both have the same topography as the work space.

As found in biology the lateral connection strength decreases with the distance between two neurons and the neurons do not project to themselves. The short-range lateral connections can be characterized by the neighborhood function

$$f(x) = \begin{cases} e^{-(x-1)^2} & \text{if } x \in \langle 0, 3 \rangle \\ 0 & \text{if } x = 0 \text{ and } x > 3 \end{cases}$$

Note that  $x$  is discrete, corresponding to the distances between the nodes on the lattice. Larger values than  $r = 3$  could have been chosen without loss of functionality, since

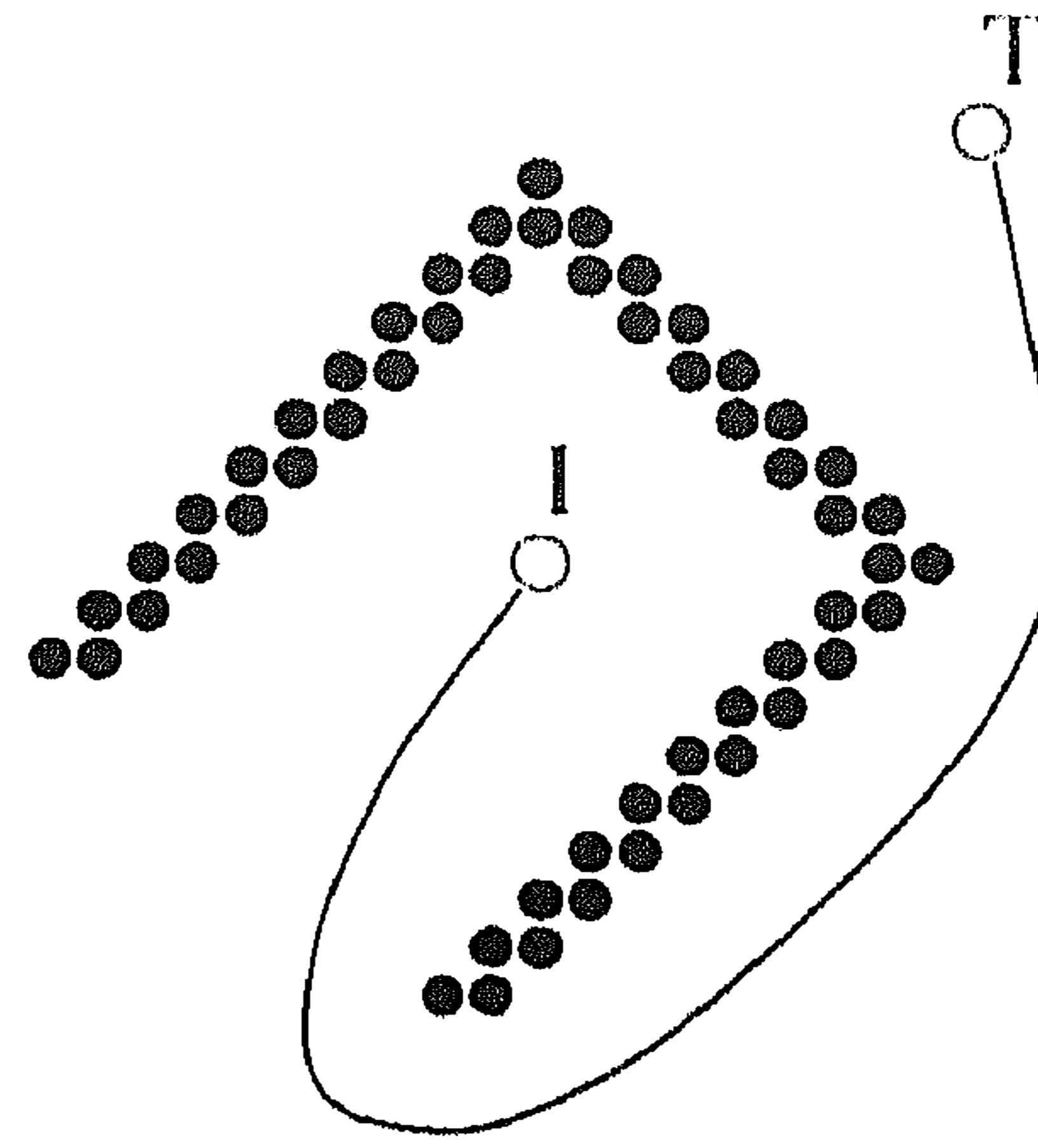


Fig. 2. Representation of work space. Black dots are part of a non-convex obstacle. The open circle marked with an *I* is the initial position. The open circle marked with a *T* is the target position. The line illustrates the smooth and continuous path, generated by the model

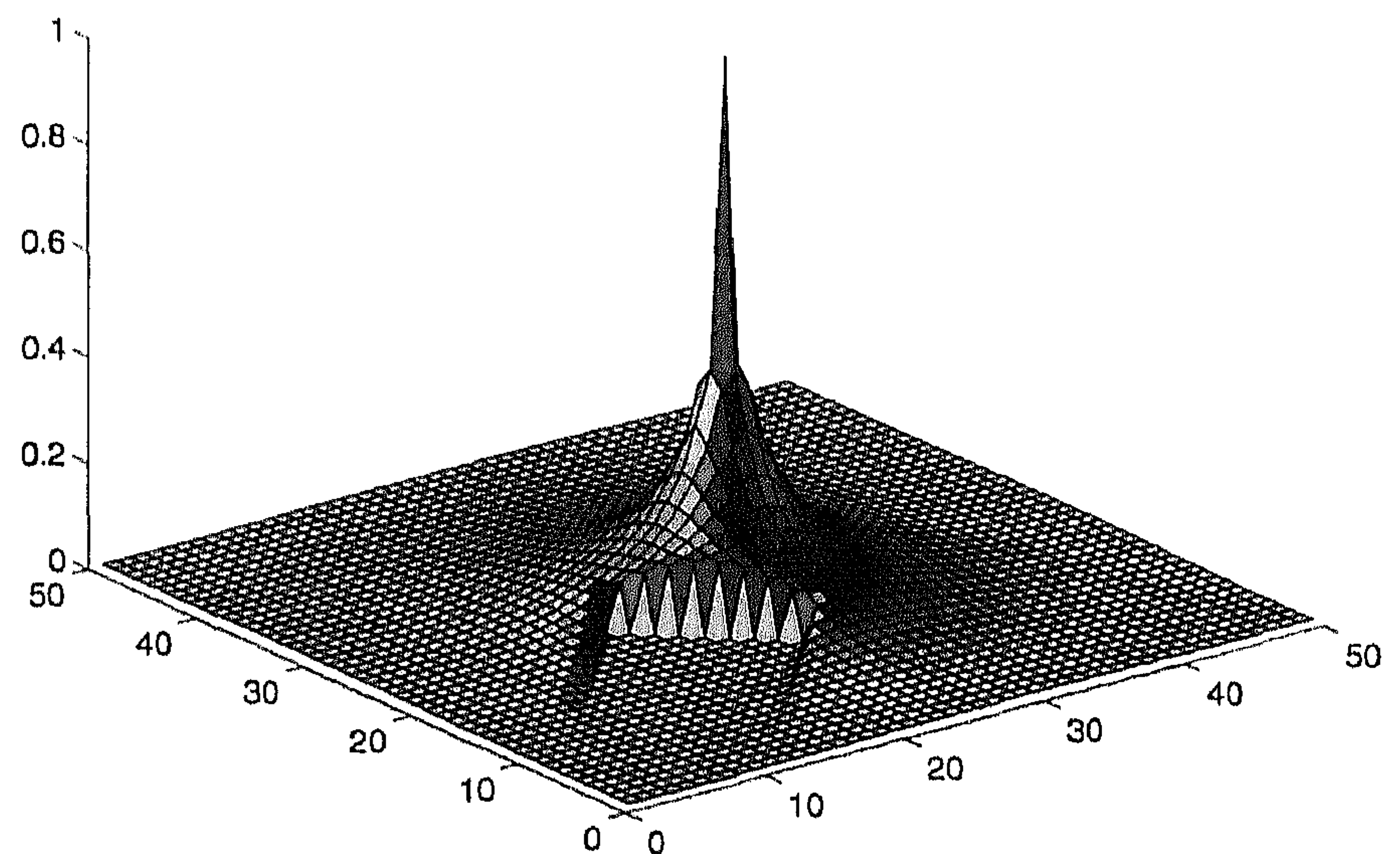


Fig. 3. Stable neural activity in the sensory map as a function of the position of the neurons in the  $50 \times 50$  neural grid (see Sect. 3.1). The map is isomorphic to the work space in Fig. 2

the exponential function becomes small for  $r = 3$ . To limit the computation time we chose the small range of  $r = 3$ .

Having defined the neighborhood in the maps, we assume that sensory neurons project to motor map neurons with receptive fields in the same neighborhood. Again values larger than  $\hat{r} = 3$  can be chosen but will only increase the computation time.

$$\hat{f}(x) = \begin{cases} 1 & \text{if } x = 0 \\ e^{-(x-1)^2} & \text{if } x \in \langle 0, 3 \rangle \\ 0 & \text{if } x > 3 \end{cases}$$

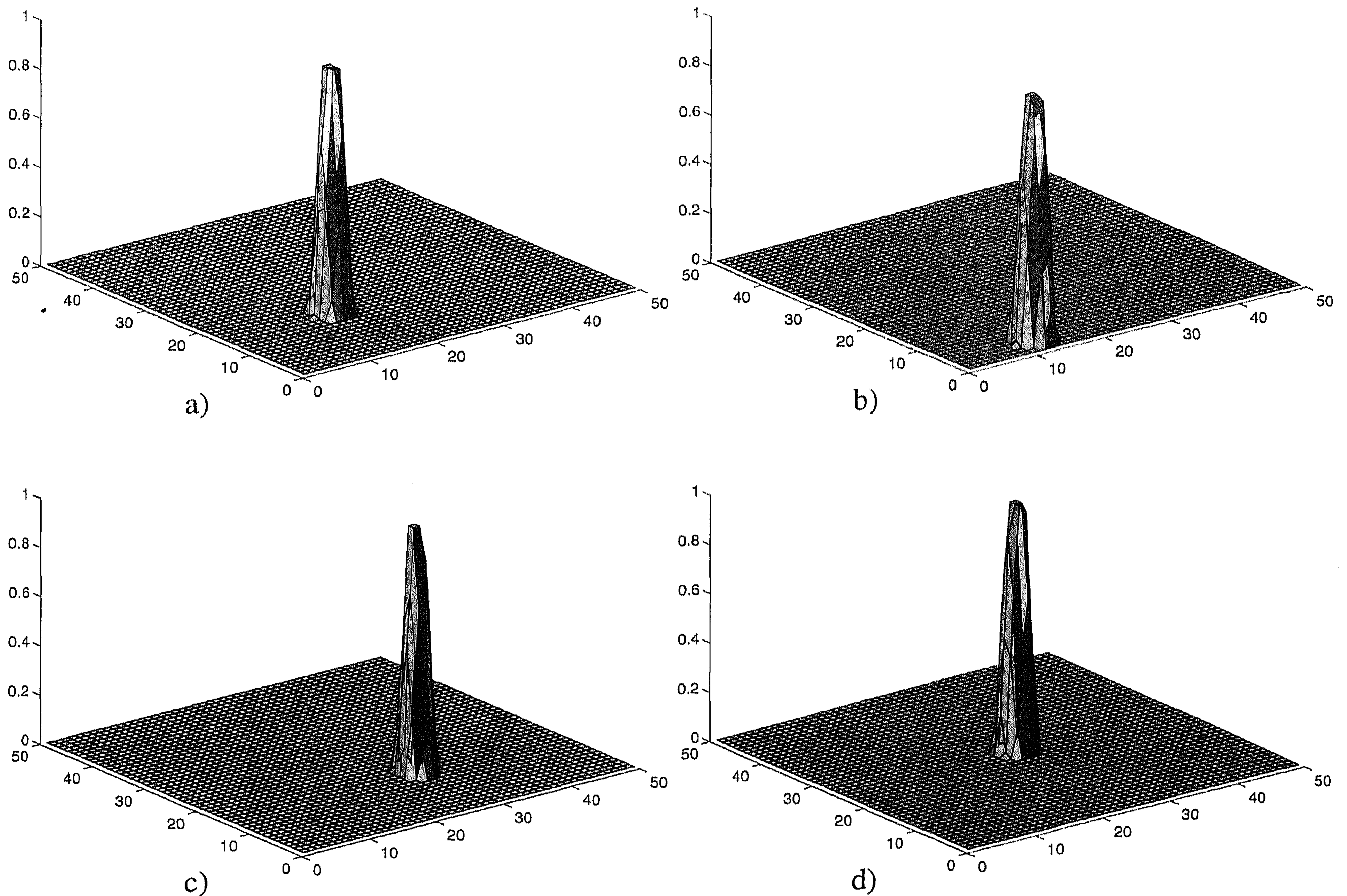
For the projections of sensory inputs from external origin to the motor map (9) we chose the neighborhood function

$$\tilde{f}(x) = \begin{cases} e^{-0.01x^2} & \text{if } x \in [0, 6] \\ 0 & \text{if } x > 6 \end{cases}$$

The functionality of the model is insensitive to the choice of  $\tilde{r}$ . The value  $\tilde{r} = 6$  adheres to the constraint that  $\tilde{r} > r$  and corresponds to a particular  $\Delta t$ . The external inputs to the sensory map are described in (5).

Larger values for  $r$ ,  $\hat{r}$  and  $\tilde{r}$  imply that each neuron has a larger number of connections. In addition to the larger amount of computation time in the simulations, these larger values make the system less sensitive to broken connections or noisy neurons.





**Fig. 4.** Activation landscape in the motor map at four points in time (a, b, c and d, respectively) during the movement of the cluster to the target position (see Sect. 3.1). The neuronal map is isomorphic to the work space in Fig. 2

Because the input to target and obstacle neurons must be large such that target neurons in the sensory map will have an activity  $\sigma_i = 1$  and obstacle neurons in the motor map will have an activity  $\sigma_k = 0$ , we chose  $v = 1000$  [see (5), (9)]. If we choose  $\beta$  in the transfer function (2) high, then a non-target sensory neuron with low input can have a large output. Moreover, it can have an output  $\sigma_k = 1$  which, in the following moment, will be passed to its neighbors. The result is that all neurons will end up in maximal activated state. If we choose  $\beta$  small then the activity of a neuron will be low, even when its input is high (e.g. when the neuron is a neighbor of the target neuron). In that case the hill in the activity landscape will decrease rapidly to activity values close to zero. From trial and error we found that  $\beta = 0.3$  is, in this case, just small enough to prevent the system from blowing up, and large enough to have a broad hill on the activity landscape. In the simulations we used a discrete-time version of the dynamics (6). At each time step  $dt$  all neurons change their state according to

$$\begin{cases} u_i(t+dt) = u_i(t) + dt \left( \sum_j^N B_{ij} \sigma_j(t) + I_i - u_i(t) \right) \\ \text{sensory neuron} \\ u_k(t+dt) = u_k(t) + dt \left( \sum_j^N T_{kl} \sigma_l(t) + I_k - u_k(t) \right) \\ \text{motor neuron} \end{cases}$$

In the sensory map a peak of activity around the target neuron grows in width. The target neuron, activated by external input to its maximum, activates its neighbors through the short-range connections. The neighbors of the target neuron then activate their neighbors, and so on. In the vicinity of an obstacle neuron, the activity of which is kept at the minimal value, the spread of activities stops in the direction of the obstacle neuron. The resulting neural activity values in the sensory map after some time are shown in Fig. 3. If neither target nor obstacle moves, then the resulting hill is stable.

The neural activity in the motor map at four points in time is shown in Fig. 4.

At all times there is a cluster localized somewhere in the map. Initially the cluster is located at the start position, i.e., in the cavity of a non-convex obstacle. Then, as a result of the system dynamics only, the cluster moves away from the target, rounds the obstacle and eventually moves to the target and reaches it. The path is smooth and continuous, as illustrated in Fig. 2.



### 3.2 Simple connections for real-time robot trajectory formation

Consider a  $D$ -link robot manipulator. In this case the state space is a  $D$ -dimensional configuration space and can be described by the  $D$  joint angles of the links. The structure of both topographical maps is a  $D$ -dimensional cubic lattice of neurons. The neurons have a  $D$ -dimensional square-shaped receptive field that does not overlap with the receptive field of other neurons. The receptive fields are contiguous and the map is periodic such that they cover the complete configuration space. Therefore, the centers of the receptive fields are also ordered according to a  $D$ -dimensional lattice.

The neighborhood functions in the lateral short-range connections of the two topological maps are

$$f(x) = \begin{cases} 1 & \text{if } x \in \langle 0, \sqrt{2} \rangle \\ 0 & \text{otherwise} \end{cases}$$

resulting in lateral short-range connections that are symmetric with a value 1. Every neuron  $i$  in the sensory map projects to only one neuron  $k$  in the motor map with the same receptive field

$$\hat{f}(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

The neighborhood function, characterizing the projections from external origin, has been chosen as:

$$\tilde{f}(x) = \begin{cases} 1 & \text{if } x \in [0, 6] \\ 0 & \text{otherwise} \end{cases}$$

The feedforward projections from the sensory map to the motor map consist of connections with strength  $\alpha$  between the corresponding neurons. Let  $\sigma_{i(k)}$  be the state of sensory neuron  $i$  which projects to neuron  $k$  in the motor map. Then the input from the first layer is

$$-\alpha (1 - \sigma_{i(k)}) \quad (15)$$

Finally, the input to a sensory neuron is

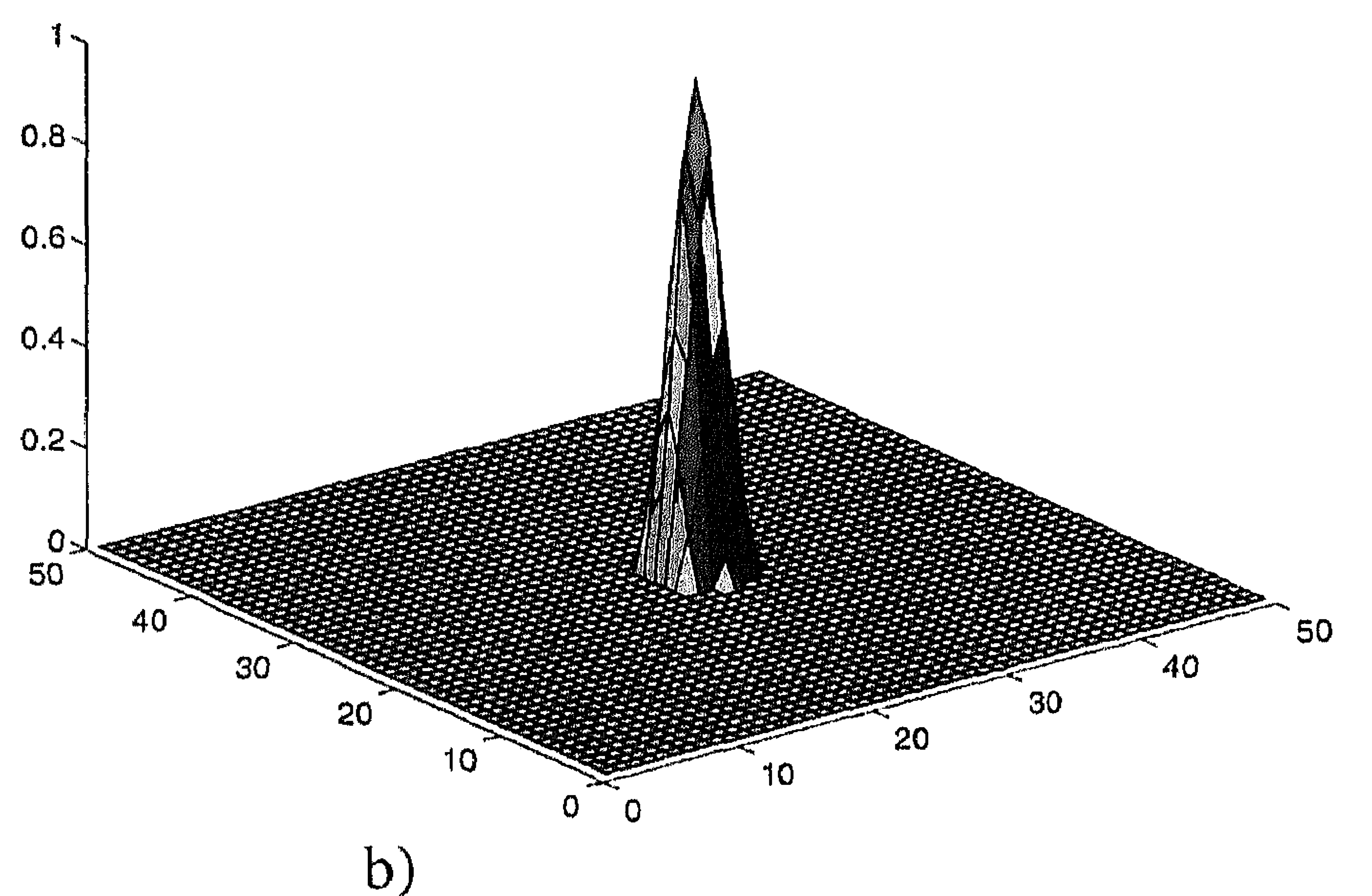
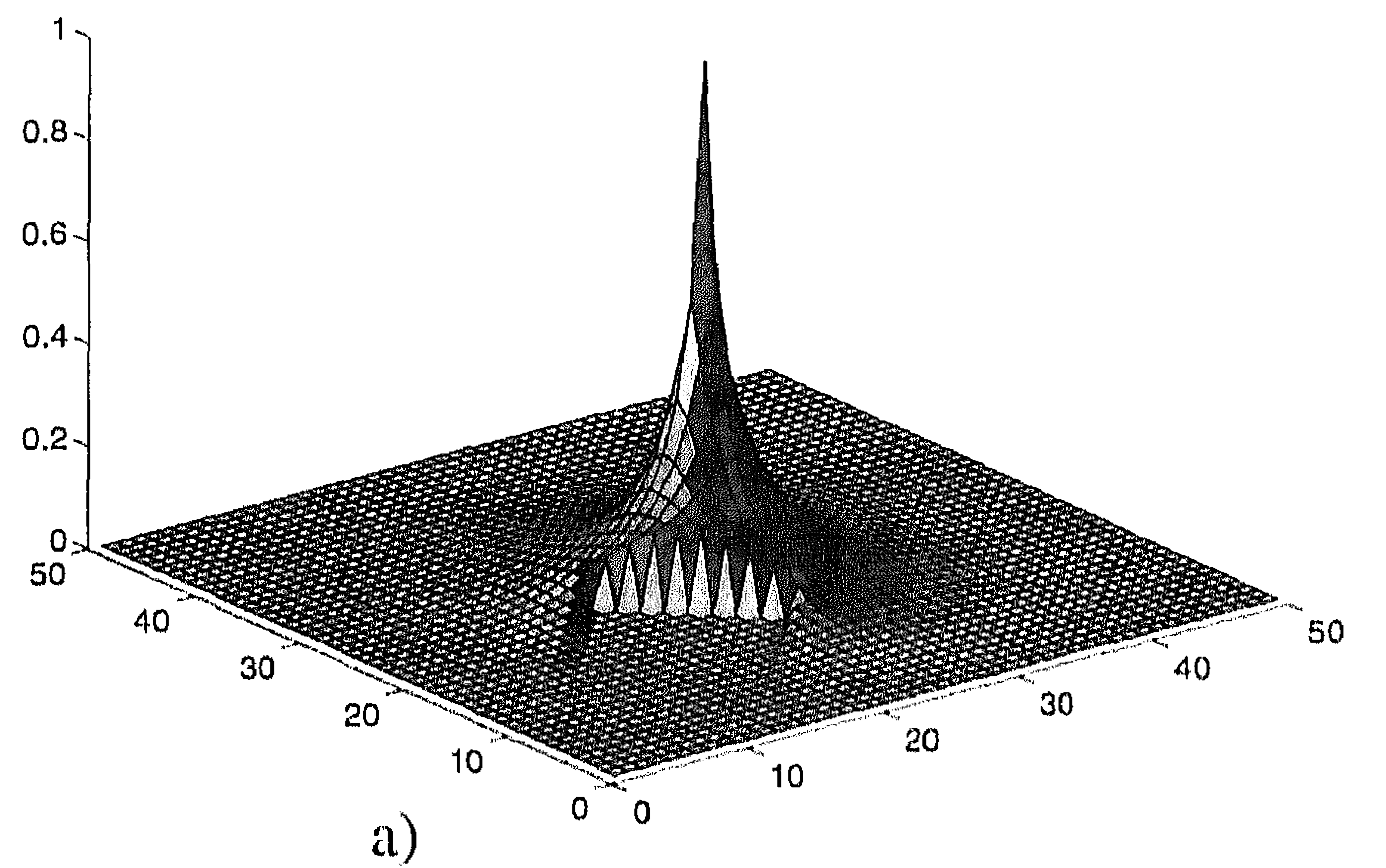
$$u_i(t) = \sum_j^N B_{ij} \sigma_j(t) + I_i$$

and to a motor neuron

$$u_k(t) = \sum_l B_{kl} \sigma_l - J_0 (m - \mu) - \alpha (1 - \sigma_{i(k)}) + K_k (\theta_A) - \frac{\sum_l B_{kl}}{2}$$

The long-range connections in the motor map provide a global field. Each neuron temporarily receives the same amount of input through the lateral connections. The number of connections per neuron in the motor map can be decreased dramatically by using an extra external integrating neuron. The  $N$  long-range connections to the other neurons can be replaced by one connection of the original strength  $J_0/N$  to the external neuron and one connection from this neuron back with weight 1.

In the simulation we chose  $\alpha = 1$  and  $D = 2$ . The results, however, do not critically depend on the value of these parameters. With a square lattice the last term is now



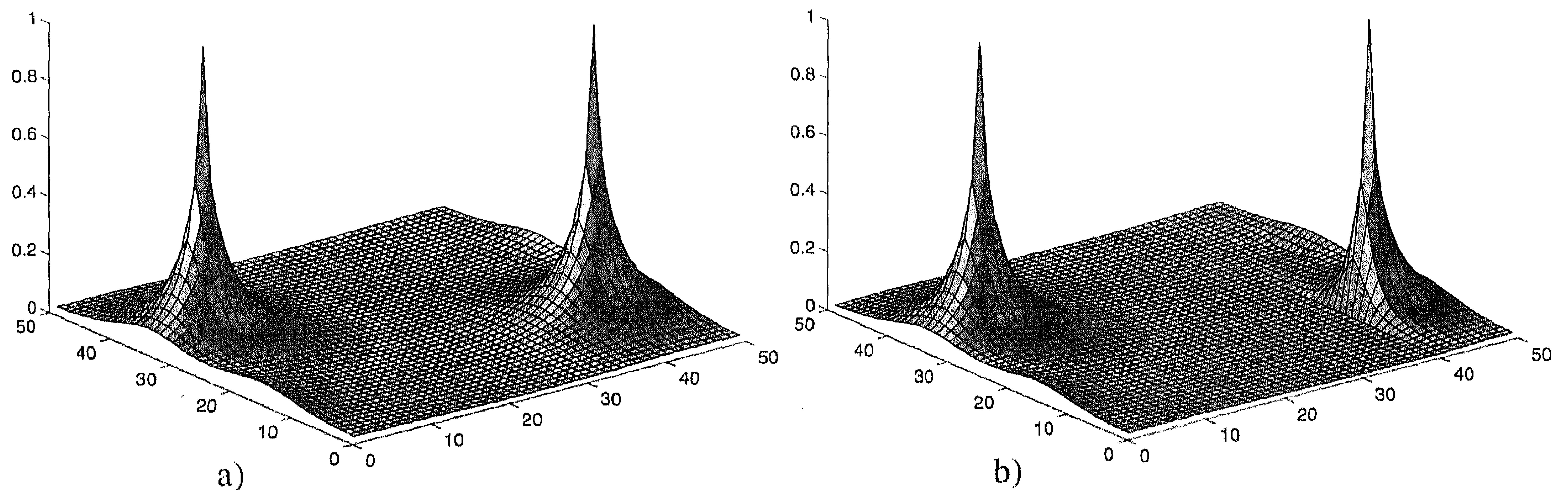
**Fig. 5a,b.** Stable neural activity as a function of the position of the neurons in the  $50 \times 50$  neural grid (see Sect. 3.2). Both neural maps are isomorphic to the work space in Fig. 2. **a** The sensory map. The width of the activation hill is smaller than the width of the activation hill in Fig. 3. **b** The motor map. The cluster is at the target position. The cluster has the same mass of positive activity on an area larger than that in Fig. 4d

$(\sum_l B_{kl}) / 2 = 4$ . The other parameters as well as the initial configuration, the target configuration and the obstacle configuration are chosen to be the same as in the former simulation (see Fig. 2).

The stable state of neural activities in the sensory map is shown in Fig. 5a. The cluster in the motor map located at the target configuration is shown in Fig. 5b. The differences compared with the results of the first simulation, due to the simplified connections, are the width of the activity hill in the sensory map, which is smaller than that in Fig. 3, and the number of motor map neurons inside the cluster, which is larger than in Fig. 4d.

The speed of the cluster movement depends on the slope of the activation hill. Because the slope of the activation hill in the outer region is smaller than in the previous simulation, the speed in system-time units is smaller in this simulation than in the previous simulation. However, because of the simple connections the time to simulate this system on a sequential computer (DEC 3100) is about an hour while that of the first simulation is a few days.





**Fig. 6a,b.** Stable neural activity in the sensory map as a function of the position of the neurons in the  $50 \times 50$  neural grid. **a** Two target positions in configuration space correspond to the same manipulator endpoint coordinate in work space. **b** An obstacle in work space changes the flow of activities in the sensory map and the resulting landscape

A more localized cluster has the same total mass of positive activity on a smaller area, which means that the cluster is denser. The density of the cluster is important when it is in the vicinity of an obstacle. A dense cluster has higher neural activities, which results in a better preservation of the spherical shape of the cluster.

Nevertheless, the resulting paths in the two simulations are qualitatively equal and we want to stress that the model is quite insensitive to the choice of the neighborhood functions.

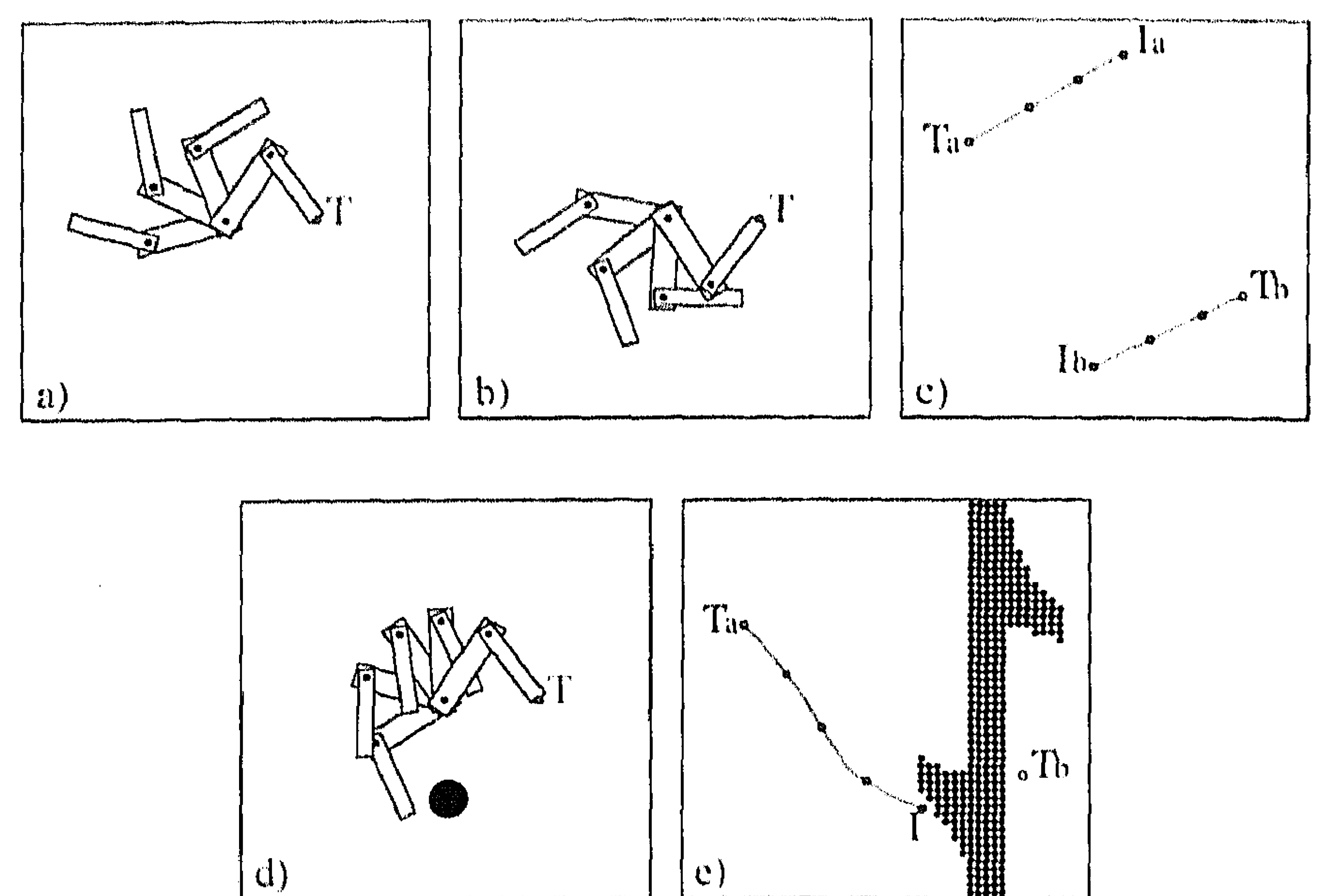
### 3.3 Choosing between multiple targets

In the following simulations we regard a two-link manipulator working in a two-dimensional work space. To reach any point in the work space with the manipulator end-effector two configurations are possible. Hence the system has to choose between two target manipulator configurations having the same end-effector work space coordinates. The simulations will be done under two conditions: with and without obstacles in the work space.

The configuration space of the manipulator can be described by the two joint angles of the manipulator. The neural maps, again, consist of a neural grid of  $50 \times 50$  neurons. Each neuron has a hyper-cube-shaped receptive field. Because the joint angles are periodic the neuronal maps have to be cyclic. Hence the shape of both maps is that of a torus. The other parameters are chosen the same as in the second simulation described in Sect. 3.2 of the point robot with simple connections.

The resulting neural activity on the sensory map is shown in Fig. 6a. Depending on the initial position of the manipulator in configuration space, steepest ascent will lead to one of the peaks. Both target configurations have a basin of attraction consisting of all initial configurations that will result in a stable end-configuration at that peak.

In the first simulation we chose two initial configurations, one leading to one peak and the other to the other peak. In Fig. 7a and b the manipulator is shown at four different time instances. Note that the end-effector target configuration in both pictures is the same. In Fig. 7c both paths are shown in configuration space.



**Fig. 7a-e.** Manipulator configurations at four time instances during the movement to the command manipulator endpoint position (black dot marked with a  $T$ ). **a, b** and **c** When no obstacle are present; **d, e** in the case of one round obstacle in work space. **a** Starting configuration and all intermediate configurations belong to one basin of attraction, leading to the first target configuration. **b** Starting configuration and all intermediate configurations belong to the other basin of attraction, leading to the second target configuration. **c** Paths in configuration space corresponding to the two manipulator movements resulting in the same manipulator end-point position. **d** An obstacle (filled black circle) blocks one path. **e** Representation of configuration space. Black dots are obstacle positions in configuration space corresponding to the obstacle in work space. The positions of the open circles correspond to the five manipulator configurations in **d**. Note that the starting configuration ( $I$ ) belongs to the first basin of attraction although the second target configuration is much closer in configuration space

In the second simulation, all manipulator configurations which are blocked by the obstacle are called obstacle configurations. The neurons with an obstacle configuration in their receptive fields are referred to as the obstacle neurons. The obstacle neurons block the activity flow from both target neurons. The resulting activity landscape in the sensory map is shown in Fig. 6b. Note that the basin of attraction belonging to the target configuration far from the obstacle is enlarged, while that of the near target configuration became smaller. We chose an initial configuration close to the target configuration near to the obstacles. The resulting manipulator movement is illustrated in Fig. 7d, the path in config-



uration space is shown in Fig. 7e. Note that in the former simulation this initial configuration would lead to the other target configuration.

#### 4 Discussion

In this paper we have proposed a model for trajectory formation and obstacle avoidance based on a two-layered neural network with continuously valued neurons. The evolution of the network is given by parallel continuous-time dynamics. The advantage of the two-layered network is that it allows trajectory formation in a cluttered environment without an explicit algorithm, how to reach the object in work or joint space. This was not possible in a single-layer network.

The first layer, the sensory map, consists of a large number of locally connected neurons. When the layer receives external input the activity pattern in the map starts to evolve towards a landscape with hills and valleys. The locations of the tops of the hills in the map topographically coincide with target locations in state space, the bottoms of the valleys coincide with obstacle positions. Each path is determined by the neural activity gradient on the activity landscape in the sensory map.

The sensory map is functionally equivalent to the wave-propagation or the distance-transform model (Jarris 1985; Dorst et al. 1991; Glasius et al. 1995) and, therefore, it is able to find the shortest possible path. However, the previously mentioned models need additional modules to find the next node on the lattice with the largest activity to generate a smooth path along the discrete via-points.

The second layer, the motor map, is similar to the first layer but has in addition long-range lateral connections and different inputs. Over time a cluster of activity shifts over the map directed by the input from the sensory map, until it reaches the target location. The movement of the cluster is a smooth path corresponding to a smooth motion in work space.

The motor map is functionally similar to the model suggested by Glasius et al. (1994) and by Droulez and Berthoz (1991). Compared with these models and with the potential field method (Kathib 1986; Krogh and Thorpe 1986) the present model does not suffer from undesired local minima, due, for example, to concave-shaped obstacles, and is able to react to unforeseen external forces.

Some points are worth noting about the functional properties of the network:

1) The model is capable of choosing one out of multiple targets. We can regard the set of initial positions which will lead to the same peak as the basin of attraction belonging to that peak.

2) In the case of a multi-link manipulator, more than one configuration may correspond to the same effector endpoint in work space. The model will make a smooth path from the present configuration to the closest target configuration. If an obstacle obstructs the path to the closest target in configuration space, then the basin of attraction belonging to the closest peak is reduced in size while that belonging to the other peaks has been enlarged. The model will choose the next nearest target configuration which is not obstructed.

3) The output of the motor map can be regarded as a motor command. In the case of a spring-like muscle system, each manipulator configuration can be accomplished by defining a particular set of spring constants (Bizzi et al. 1982; Hogan 1984) or resting lengths (Feldman 1986) of the spring-like muscles. If each motor neuron  $k$  sends a motor command proportional to  $[\theta_k(t)\sigma_k(t)\tau_k(t)]/[\sum_k \sigma_k(t)\tau_k(t)]$ , then the command manipulator position is a motor command consisting of votes of individual neurons to move the actuator to a certain position.

4) External forces acting on the subject, e.g., robot or manipulator, in work space may exist. Our model is capable of reacting to these forces and if the actuators are strong enough to overcome the external forces, and if the targets do not move too fast, a target configuration will be reached.

5) The inverse kinematics problem and the inverse dynamics problem, connected with the redundant manipulator, are problems distinct from the problem of trajectory formation discussed in this paper. The mapping from low-dimensional work space to high-dimensional configuration space could be done with an additional layer. Each target and obstacle position in work space correspond to a fixed subset of target and obstacle neurons in the neural map that depend on the shape of the manipulator only. If the actuators are strong enough, coriolis forces and inertia can not prevent the system from reaching a target.

6) As in other methods that use grids (Lozano-Perez 1983; Schwartz and Sharir 1983; Kathib 1986; Krogh and Thorpe 1986; Newman and Hogan 1987; Prassler 1989; Warren 1989; Barraquand and Latombe 1990; Conolly et al. 1991; Dorst et al. 1991; Barraquand et al. 1992; Glasius et al. 1994, 1995), the number of units grows exponentially with the number of degrees of freedom (DOF). Unlike the fully connected neural networks, the number of connections in our model can be of the order of the number of units. One can think of different architected or graded grids to circumvent the unit-consuming problem. In some 6DOF manipulators, the rotation of the gripper could be done without ever colliding into an obstacle. Hence the sensory map has to represent the targets only in this lower-dimensional map and rotating of the gripper can be done in a one-dimensional sensory sub-map. This structure needs a smaller number of neurons but possesses the same functionality. Because the motor map produces continuous paths, not depending on the number of neurons in it, the number of neurons could be reduced to a minimum just sufficient to contain the area defined by  $\tilde{r}$ .

7) Pilot computer experiments demonstrate that the network performance is not sensitive to the choice of the transfer functions  $g(x)$  and the neighborhood functions  $f(x)$ .

8) The model could be used with very simple connections which makes it possible to implement it in a fast analog electronic circuit. This can be used for autonomous robot trajectory formation in real time.

#### References

1. Amari S (1977) Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol Cybern* 27:77-87



2. Barraquand J, Latombe J (1989) Robot motion planning with many degrees of freedom and dynamic constraints. In: Proceedings of the 5th International Symposium on Robotics Research (Tokyo) pp 74–83
3. Barraquand J, Latombe JC (1990) A Monte-Carlo algorithm for path-planning with many degrees of freedom. In: Proceedings of the IEEE International Conference on Robotics and Automation (Cincinnati, Ohio). IEEE Computer Society Press, Los Alamitos, 1712–1717
4. Barraquand J, Langlois B, Latombe J (1992) Numerical potential field techniques for robot path planning. *IEEE Trans Syst Man Cybern* 22:224–241
5. Bizzi E, Accornero N, Chapelle N, Hogan N (1982) Arm trajectory formation in monkeys. *Exp Brain Res* 46:139–143
6. Caminiti R, Johnson PB, Burnod Y, Galli C, Ferraina S (1990) Shifts of preferred directions of premotor cortical cells with arm movements performed across the work-space. *Exp Brain Res* 83:228
7. Connolly C, Burns J, Weiss R (1991) Path planning using Laplace's equation. In: Proceedings of the IEEE International Conference on Robotics and Automation. IEEE Computer Society Press, Los Alamitos, pp 2102–2106
8. Dorst L, Mandhyan I, Trovato K (1991) The geometrical representation of path planning problems. *Robotics Autonomous Syst* 7:181
9. Droulez J, Berthoz A (1991) A neural network model of sensorimotor maps with predictive short term memory properties. *Proc Natl Acad Sci USA* 88:9653–9657
10. Feldman AG (1986) Once more on the equilibrium-point hypothesis ( $\lambda$  model) for motor control. *J Mot Behav* 18:17–54
11. Fritzke B (1991) Unsupervised clustering with growing cell structures. Proceedings of the International Joint Conference on Neural Networks (Seattle, Wash) vol II. Piscataway NJ, IEEE, pp 531–536
12. Georgopoulos AP, Ashe J, Smyrnis N, Taira M (1992) The motor cortex and the coding of force. *Science* 256:1692–1695
13. Glasius R, Komoda A, Gielen CCAM (1994) Population coding in a neural net for trajectory formation. *Network: Comput Neural Syst* 5:549–563
14. Glasius R, Komoda A, Gielen CCAM (1995) Neural network dynamics for trajectory formation and obstacle avoidance. *Neural Networks* 8:125–133
15. Grossberg S (1984) Nonlinear neural networks: principles mechanisms and architectures. *Neural Networks* 1:17
16. Hogan N (1984) An organizing principle for a class of voluntary movements. *J Neural Sci* 4:2745–2754
17. Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proc Natl Acad Sci USA* 81:3088
18. Jarris RA (1985) Collision-free trajectory planning using distance transforms. *Mech Eng Trans IE Aust ME10*:187
19. Kalaska JF, Cohen DAD, Prud'homme M, Hyde ML (1992) Comparison of cell discharge in motor, premotor and parietal cortex during reaching: load direction-related activity in primate motor cortex using a two dimensional reaching task. Springer Berlin Heidelberg New York
20. Kathib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robotics Res* 5:90–98
21. Kohonen T (1982) Self-organized formation of topologically correct feature maps. *Biol Cybern* 43:59–69
22. Kopecz K, Schöner G (1995) Saccadic motor planning by integrating visual information and pre-information on neural dynamic fields. *Biol Cybern* 73:49–60
23. Krogh BH, Thorpe CE (1986) Integrated path planning and dynamic steering control for autonomous vehicles. Proceedings of the IEEE International Conference on Robotics and Automation (Washington, DC). Computer Society Press of the IEEE, Los Angeles, pp 1664–1669
24. Latombe J (1991) Robot motion planning. Kluwer Academic, Boston
25. Linsker R (1986a) From basic network principles to neural architecture: emergence of spatial-opponents cells. *Proc Natl Acad Sci USA* 83:7508–7512
26. Linsker R (1986b) From basic network principles to neural architecture: emergence of orientation-selective cells. *Proc Natl Acad Sci USA* 83:8779–8783
27. Linsker R (1986c) From basic network principles to neural architecture: emergence of orientation columns. *Proc Natl Acad Sci USA* 83: 7508–7512
28. Lozano-Perez T (1983) Spatial planning: a configuration space approach. *IEEE Trans Comput* 32:108–120
29. Martinetz TM (1993) Competitive Hebbian learning rule forms perfectly topology preserving maps. In: Proceedings of the International Conference on Artificial Neural Networks (Amsterdam, The Netherlands). North-Holland/Springer-Verlag, Amsterdam/Berlin Heidelberg New York, pp 427–434
30. Newman W, Hogan N (1987) High speed robot control and obstacle avoidance using dynamic potential function. In: Proceedings of the IEEE International Conference on Robotics and Automation (Raleigh, NC). Computer Society Press of the IEEE, Los Angeles, pp 14–24
31. Prassler E (1989) Electrical networks and a connectionist approach to path-finding. In: Connectionism in perspective. North-Holland, Amsterdam, p 421
32. Ritter HJ, Martinetz TM, Schulten KJ (1989) Topology-conserving maps for learning visuo-motor-coordination. *Neural Networks* 2:159–168
33. Schwartz JT, Sharir M (1983) On the 'piano' mover's problem. II. General techniques for computing topological properties of real algebraic manifolds. *Adv Appl Math* 4:298–351
34. Warren C (1989) Global path planning using artificial potential fields. In: Proceedings of the IEEE International Conference on Robotics and Automation (Scottsdale, Ariz). Computer Society Press of the IEEE, Los Angeles, pp 316–321