

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/2066/168467>

Please be advised that this information was generated on 2021-10-27 and may be subject to change.

Population coding in a neural net for trajectory formation ¹

R. Glasius

A. Komoda

S. Gielen

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

Abstract

In this study we investigate the time-evolution of the activity in a topographically ordered neural network with external input for two types of neurons: one network with binary-valued neurons with a stochastic behavior and one with deterministic neurons with a continuous output. We will demonstrate that for a particular range of lateral interaction strengths, changes in external input give rise to gradual changes in the position of clustered neural activity.

The theoretical results will be illustrated by computer simulations in which we have simulated a neural network model for trajectory planning for a multi-joint manipulator. The model gives a collision free trajectory by combining the sensory information about the position of target and obstacles. The position of the manipulator is uniquely related to the clustered activity of the population of neurons, the population vector. The movement of the manipulator from any initial position to the target position is the result of the intrinsic dynamics of the network.

¹This work has been accepted by 'Network: Computation in Neural Systems' (August 1994).

1 Introduction

Both in neurobiological and artificial (robot) systems, trajectory planning and formation is one of the major objects of study because it is one of the essential aspects in motor control, which is not yet fully understood. The early models for trajectory planning in robot control were based on algorithms which were valid only in some very specific (unnatural) conditions or which required too much time for real-time implementation.

The first models for trajectory planning dealt with static environments only [1, 4, 13, 15, 17, 25] and used global methods, which can be generally viewed as constructing a graph which is used in a search process for an accessible path in a cluttered environment. These models could only be used for static environments because a moving object or introduction of a new object requires that the whole graph has to be constructed anew. Moreover, when multiple obstacles are present, the complexity of the graph increases exponentially which makes it impossible to plan trajectories in real-time.

An other approach, which was proposed later, was the potential field method [14, 16] which improves upon the graph-search in having only linear complexity and non-discrete paths. Unfortunately, the potential field methods suffer from several problems, one being that of undesired local minima, which cause the system to become trapped by some configurations of obstacles [14, 16]. As a consequence, these models can not guarantee a solution to the trajectory-planning problem.

A more successful model is the wave propagation or distance transform model [11]. This approach does not suffer from local minima. If a solution exists the model will find it. However, the path consists of a series of discrete steps on a lattice which is the result of a supervised incremental algorithm to find the next node for the trajectory in the lattice by selecting the neighboring node with the largest activity. In addition to the incremental algorithm additional modules have to be inserted in order to calculate a smooth path along the discrete via-points and in order to reach positions which are not lattice nodes.

Other models have tried to improve upon existing models for robot control related to trajectory formation. For example, Ritter et al. [20, 21] introduced a neural network which was able to learn the transformation from cartesian space coordinates to joint coordinates. In another neural network approach van der Smagt [24, 23] used a backpropagation conjugate gradient neural network to control a robot manipulator which received input about targets from a hand held camera. The neural net learned the transformation from target position in camera domain to a motor command corresponding to the joint angles in order to move the manipulator in a few straight steps to the target. Both these networks do not specify the trajectory and do not take into account obstacles between the robot and the target position. Therefore, the aim of this study is to develop a model for trajectory planning, which does not have the disadvantages of the previous models for trajectory planning and which is complementary to the inverse kinematics models by Ritter et al. [20] and van der Smagt [24].

In this study we will provide a neural network model, which generates a movement trajectory towards a target position by the intrinsic dynamics of the network, i.e. without a supervisor. The trajectory is a smooth path in configuration space. The model also performs in a cluttered environment in which objects are moving in time.

We will give a theoretical analysis of the dynamics of a topographically ordered

neural network, which receives external (for example sensory) input signals, and we will analyze how the evolution of the neural activity can be interpreted in terms of the macroscopic variables. This will be done for two types of networks. The first network is a stochastic system. The activity of neurons is represented as a two-valued random variable and the time evolution of the network is given by the Glauber dynamics [8]. The second network is a deterministic system with activity of neurons given by a continuous real-valued variable. The dynamics of the network is given by a set of non-linear differential equations. In both cases the architecture of the network is the same. The results of the theoretical analyses will be illustrated by computer simulations of trajectory planning and trajectory formation in a topographically ordered map representing a cluttered environment. The system can also be used to control the movement of a multi-jointed manipulator in the presence of obstacles. By obstacles we will understand the regions of space which are occupied by objects or by constraints imposed on joints of the manipulator.

The paper is organized as follows: In chapter 2 we will define the model of discrete and continuous neurons. We will assume that the activity of each neuron contributes to the position coding of an object or manipulator. The specific position is not coded by an individual neuron but rather by the population of neurons, which we call a cluster of active neurons. The weighted vector sum of the positions coded by these neurons, the population vector, gives the position of the manipulator in space. In section 3 we will describe computer simulations to illustrate the performance of the networks and to confirm our analytical results obtained in 2.

2 The models

Both models consist of N neurons arranged in a two-dimensional regular lattice. However, both models can be easily extended to higher dimensions and may have other types of lattices. The neurons have two types of lateral interactions: Short range excitatory and long range inhibitory connections. We assume that the neural activity in the lattice gives a discrete topographically ordered representation of the configuration space \mathcal{C} of a two-joint manipulator. We will call it the neuronal space \mathcal{N} . The activity of each neuron i in \mathcal{N} represents events in a particular subset of \mathcal{C} , called its receptive field (rf_i). By the configuration space of the manipulator we understand the set of all its possible configurations. The dimension of this space is equal to the minimal number of parameters which are necessary to define any of these configurations uniquely. In this particular case the configuration space is a two-dimensional cartesian space with the two joint-angles of the manipulator along the coordinate axes.

We also assume that external input will provide information about the position of the targets and obstacles in the external world.

The activities of the neurons describe the state of the system. The set \mathcal{P} of all possible states of the network will be called the phase space of the system.

2.1 Discrete neurons

In this model we will assume that the N neurons in a topographically ordered map can be in one of the two states

$$\sigma_i \in \{-1, 1\} \quad i = 1, \dots, N \quad (1)$$

Since each neuron can be either in the state $+1$ or -1 , the system has 2^N possible configurations. Each of these states represents a corner of the hyper-cube in \mathbb{R}^N , so the phase space $\mathcal{P} = \{-1, 1\}^N$.

The input to the neurons is partly provided through lateral connections and partly by external input which provides information about the position of objects in the external world. It is assumed that the external input clamps the activity of all neurons which correspond to obstacles, to the value -1 . The lateral symmetric connections consist of short range and long range interactions. The short range interactions B_{ij} between neurons i and j are defined by

$$B_{ij} = \begin{cases} 1 & \text{if } 0 < \|i - j\| < r \\ 0 & \text{otherwise} \end{cases}$$

where r is a positive number and where $\|i - j\|$ stands for the Euclidean distance between neuron i and j in the lattice. The long range inhibitory connections J_{ij} are given by

$$J_{ij} = \begin{cases} \frac{J_0}{N} & \text{if } i \neq j, J_0 > 0 \\ 0 & \text{otherwise} \end{cases}$$

The external input to the neuron i contains information about the target-position $\boldsymbol{\theta}_T$ and is given by

$$-\alpha \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_T\| \quad (2)$$

where the vector $\boldsymbol{\theta}_i$ represents the position of the receptive field rf_i of neuron i in the configuration space and where $\alpha > 0$ is a constant.

The input to each neuron contains also a bias term

$$-J_0 \mu \quad (3)$$

This term will appear to be relevant for the number of active neurons in the system.

We define an energy function of the system by the following Hamiltonian.

$$\begin{aligned} \mathcal{H} &= -\frac{1}{2} \sum_{ij} T_{ij} \sigma_i \sigma_j - \sum_i J_0 \mu \sigma_i + \sum_i \alpha \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_T\| \sigma_i \\ &= -\sum_i u_i \sigma_i \end{aligned} \quad (4)$$

where $T_{ij} = B_{ij} - J_{ij}$ and where u_i is the total input to neuron i which can be written as

$$\begin{aligned} u_i &= \sum_j B_{ij} \sigma_j - \sum_j \frac{J_0}{N} \sigma_j + J_0 \mu - \alpha \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_T\| \\ &= \sum_j B_{ij} \sigma_j - \sum_j \frac{J_0}{N} \sigma_j + J_0 \mu - \alpha x_i \end{aligned} \quad (5)$$

Here we use the vector \mathbf{x}_i of $\boldsymbol{\theta}_i$ relative to $\boldsymbol{\theta}_T$ defined by

$$\mathbf{x}_i(\boldsymbol{\theta}_T) \equiv \boldsymbol{\theta}_i - \boldsymbol{\theta}_T \quad \text{and} \quad x_i \equiv \|\mathbf{x}_i\| \quad (6)$$

Each component of the Hamiltonian has a different effect on the behavior of the network. The long range inhibitory interactions J_{ij} contribute to the stabilization of the total amount of neural activity in the network near the value μ . The local neighbor interactions B_{ij} make it more advantageous for neighboring units to be in the same state. Therefore, this term tends to impose clustering of the units having the same activity. Finally the external input $-\alpha x_i$ tends to shift the activity towards the target position $x_i = 0$. The relative magnitudes of these three terms determine the properties of the system.

The dynamics of the system is sequential and stochastic: at each time step only one neuron, chosen at random among the N neurons, is updated. The transition probability w_i , that neuron i changes its activity from σ_i to $-\sigma_i$ while the others remain momentarily fixed, is

$$w_i(\boldsymbol{\sigma}) = \frac{1}{2} [1 - \tanh(\beta u_i \sigma_i)] \quad (7)$$

where $\beta = \frac{1}{T}$ is the inverse of a noise-parameter T . In the zero noise limit ($T = 0$), the probabilities w_i become 0 or 1 and the updates become deterministic

$$\sigma_i(t+1) = \text{sign}(u_i(t)) \quad (8)$$

However, even for the case $T = 0$ the time evolution of the system is still stochastic due to the random choice of a neuron at each time step.

To give these microscopic states a physical interpretation we will define some macroscopic variables. The first macroscopic variable is called the magnetization m

$$m \equiv \frac{1}{N} \sum_i \sigma_i \quad (9)$$

and is related to the total activity in the whole population of neurons. As a result the total input can be rewritten as

$$u_i = \sum_j B_{ij} \sigma_j - J_0(m - \mu) - \alpha x_i \quad (10)$$

The second macroscopic variable of interest is a vector which codes the actual position of the manipulator relative to the target position in configuration space.

$$\mathbf{X}_B \equiv \boldsymbol{\theta}_B - \boldsymbol{\theta}_T \quad (11)$$

where $\boldsymbol{\theta}_B$ is the actual position of the manipulator in configuration space

$$\boldsymbol{\theta}_B \equiv \frac{\sum_i \boldsymbol{\theta}_i (\sigma_i(t) + 1)}{\sum_i (\sigma_i(t) + 1)} \quad (12)$$

$\boldsymbol{\theta}_B$ corresponds to the center of gravity of the activities in the neural map. The position vector \mathbf{X}_B which is a weighted vector sum of the contributions of all neurons, provides information about the position in configuration space. The k -th component of the vector \mathbf{X}_B is defined by

$$X_B^k(t) \equiv \frac{\sum_i x_i^k (\sigma_i(t) + 1)}{\sum_i (\sigma_i(t) + 1)} \quad (13)$$

The length of the vector \mathbf{X}_B is the distance in configuration space between the actual configuration and target configuration.

2.2 Continuous neurons

In this model the N neurons in the topographical map may assume a real output value in the interval $[-1, 1]$

$$s_i \in [-1, 1] \quad i = 1, \dots, N. \quad (14)$$

Hence in this model $\mathcal{P} = [-1, 1]^N$, where $[-1, 1]^N$ stands for a hyper-cube in R^N . The input to the neurons is similar to the case of the discrete model. The total input of neuron i can be written by

$$u_i = \sum_j B_{ij} s_j - J_0 (m - \mu) - \alpha x_i \equiv \sum_j T_{ij} s_j (t) + I_i \quad (15)$$

with

$$I_i \equiv J_0 \mu - \alpha x_i \quad (16)$$

The activity of neuron i is given by the sigmoid function g over the total input:

$$s_i = g(u_i) \quad (17)$$

The system dynamics, in this case, is parallel and deterministic

$$\frac{du_i(t)}{dt} = \sum_j T_{ij} s_j (t) + I_i - u_i(t) \quad (18)$$

It has been shown before [3, 7, 10] that the function

$$L(\mathbf{s}) = -\frac{1}{2} \sum_{i,j} T_{ij} s_i s_j - \sum_i I_i s_i + \sum_i G(s_i) \quad (19)$$

with $G(s_i) = \int_0^{s_i} g^{-1}(x) dx$, is a global Liapunov function for (18) with the sigmoid function $g(x) = \tanh(\beta x)$ and $\beta > 0$. The vector \mathbf{s} refers to the state of the neural network with components s_i . The existence of a Liapunov function guarantees that the dynamics of the network always converges to a fixed point which is an equilibrium state of the system. By direct computation,

$$\frac{dL(\mathbf{s}(t))}{dt} = \sum_i^N \frac{\partial L(\mathbf{s}(t))}{\partial s_i} \frac{ds_i(t)}{dt} = - \sum_i^N \left(\frac{du_i(t)}{dt} \right)^2 g'(u_i) \leq 0 \quad (20)$$

$g'(\cdot)$ refers to the first derivative of the function $g(\cdot)$ with respect to its argument. The global stability of the system defined by (18) and (19) is the same as that studied by Cohen [3], Grossberg [9] and Hopfield [10].

The population vector \mathbf{X}_B which gives the relative actual position, is now defined as

$$X_B^k(t) \equiv \frac{\sum_i x_i^k (1 + \text{sign}(s_i(t))) s_i(t)}{\sum_i (1 + \text{sign}(s_i(t))) s_i(t)} \quad (21)$$

Since the population vector is a function of time, it can be seen as a neuronal representation for the position in time relative to the target position. Each pattern of neural activity corresponds to a position in space via (21) and the time evolution of \mathbf{X}_B is described by the time evolution of the neural activity.

Due to external input, which provides information about the target position, the activity of neurons begins to change according to the dynamics of the network described by equation (18). This evolution can be seen in the phase space \mathcal{P} as a motion of a point along the curve on which the Liapunov function decreases. This down-hill motion ends when the network reaches an equilibrium state. The interpretation of the population vector transforms this curve from the phase space into a curve in the configuration space.

The equation (19) represents a free energy F of the discrete system (4) in the mean field approximation [19]:

$$\begin{aligned}
 F_{\text{MFA}} = & - \frac{1}{2} \sum_{i=1, j=1}^N T_{ij} s_i s_j - \sum_{i=1}^N I_i s_i \\
 & + \frac{1}{2\beta} \sum_{i=1}^N ((1 + s_i) \ln(1 + s_i) + (1 - s_i) \ln(1 - s_i))
 \end{aligned} \tag{22}$$

and is an upper bound of the exact free energy of the system. The last term in (22) is the entropy of the system and is exactly equal to the last term in the Liapunov function (19). To obtain the equilibrium free energy and the best upper bound on F one has to minimize the right hand side of (22) with respect to all u_i . This minimization procedure can be seen as a dynamical process given by (18).

3 The simulations

To illustrate the properties of the two models we simulated a two-dimensional neural network with $N = 2500$ neurons, arranged in a 50×50 grid. The neurons are placed on the nodes of a square lattice. However, the model can be easily extended to higher dimensions and may have other types of lattices. The neural map is a discrete representation of the two-dimensional configuration space. The configuration space variables $\theta_1, \theta_2 \in [0, 100]$. Hence each neuron has a square shaped receptive field with sides of $\gamma = 2$.

By choosing $r = 1.5$ each neuron on the lattice is connected with excitatory connections of strength 1 to its $D = 8$ neighbors. The neurons also interact with each other through long range inhibitory connections of strength J_0 .

Each initial state of the simulations consists of a small connected group of neurons with activity 1. The center of mass of their activity corresponds to the initial position of the manipulator in configuration space. All other neurons have an initial value of activity -1.

Information about the position of the target and obstacles are supplied to the network by external inputs. The input to each neuron i due to the target is αx_i . The obstacles are represented in the neuronal map by clamping the activity of the neurons which correspond to the obstacle positions, to the value -1.

3.1 Discrete neurons

The dynamics of this model is the stochastic sequential heat bath: at each time step a neuron i is selected at random, and subsequently u_i was calculated using equation (10). Finally, this neuron was updated according to equation (7).

The total input to the network is given by equation (10). Denoting the actual number of neurons in state "1" by n^+ and the desired number of neurons in state "1" by \tilde{n}^+ , the magnetization m and the desired magnetization μ can be expressed respectively as

$$m \equiv \frac{2n^+}{N} - 1 \quad \text{and} \quad \mu \equiv \frac{2\tilde{n}^+}{N} - 1 \quad (23)$$

Then the total input for neuron i can be rewritten to

$$u_i = \sum_j B_{ij}\sigma_j - \frac{2J_0}{N}\Delta n - \alpha x_i \quad (24)$$

with $\Delta n \equiv (n^+ - \tilde{n}^+)$ is the difference between the actual and the desired number of neurons in state "1".

Equation (24) for the total field of neuron i contains three terms.

The 'short range interaction' term $\sum_j B_{ij}\sigma_j$ causes clustering of active neurons. If all D neighbors of a neuron are active, then the 'short range interaction' contribution to that neuron is D . If all D neighbors are non-active the contribution is $-D$. The neurons at the boundaries, lying between positive- and negative-regions, receive a contribution to their local field by a value between $-D$ and D . The short range interaction contribution to the local field of the neurons located on a line passing through the center of the cluster, is shown in figure 1.1.

The 'magnetization' term $-J_0(m - \mu)$, which has been rewritten to $-\frac{2J_0}{N}\Delta n$, facilitates the stabilization of the number of neurons in state "1". For each neuron in state "1" which changes to state "-1", the local field of all neurons is raised by an amount of $\frac{2J_0}{N}$.

The 'dragging' term $-\alpha x_i$ is the driving force of the cluster towards the target. The larger the distance between the receptive field of a neuron and the target, the more the neuron will be inhibited. Therefore, neurons with a small relative distance towards the target have higher local fields.

To see the interplay of the three terms we have to examine the total local field (24). Each of the curves in figure 1.2 shows the contributions of the three terms to the local field. The absolute value of the slope of the curves is equal to α . The vertical position of the curve on the diagram depends on the 'magnetization' term: The more positive this term is, the lower the position of the curve.

The vertical distance between various curves corresponds to different numbers of active neurons, each giving rise to a difference in the local field by $\frac{2J_0}{N}$. Each curve has two maxima, one at the target position and one at the cluster position. It depends upon the relative distance between cluster and target which maximum is larger.

Because the behavior of the system depends on the **relative** magnitudes of the three parameters B_{ij} , J_0 and α and since we have already fixed $B_{ij} = 1$, the only free independent parameters are J_0 and α .

We want to chose values for J_0 and α such that the number of active neurons, during the time-evolution of the system, is close to \tilde{n}^+ and that they form a connected cluster. This allows us to interpret the activity of the population of neurons in terms of macroscopic variables defined in equation (13).

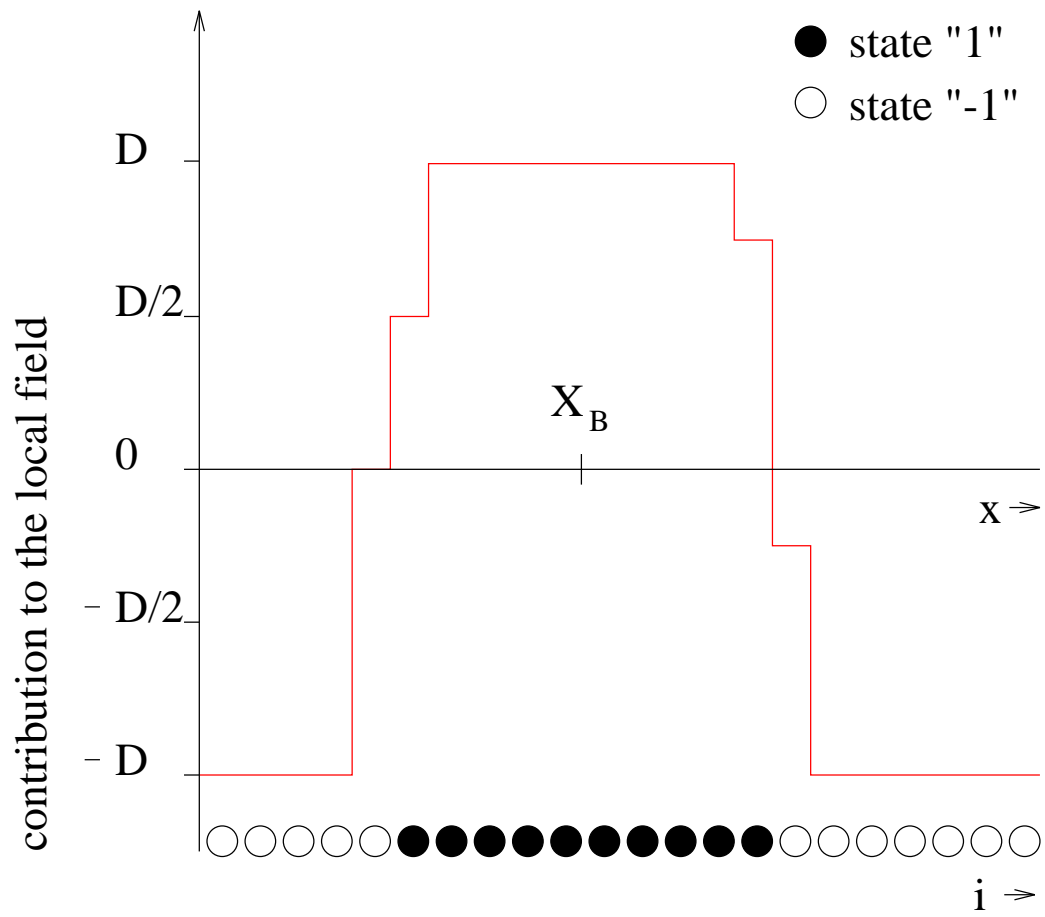


Figure 1: The 'short range interaction' contribution to the local field given for the neurons i along a one-dimensional cross section of the neuronal map.

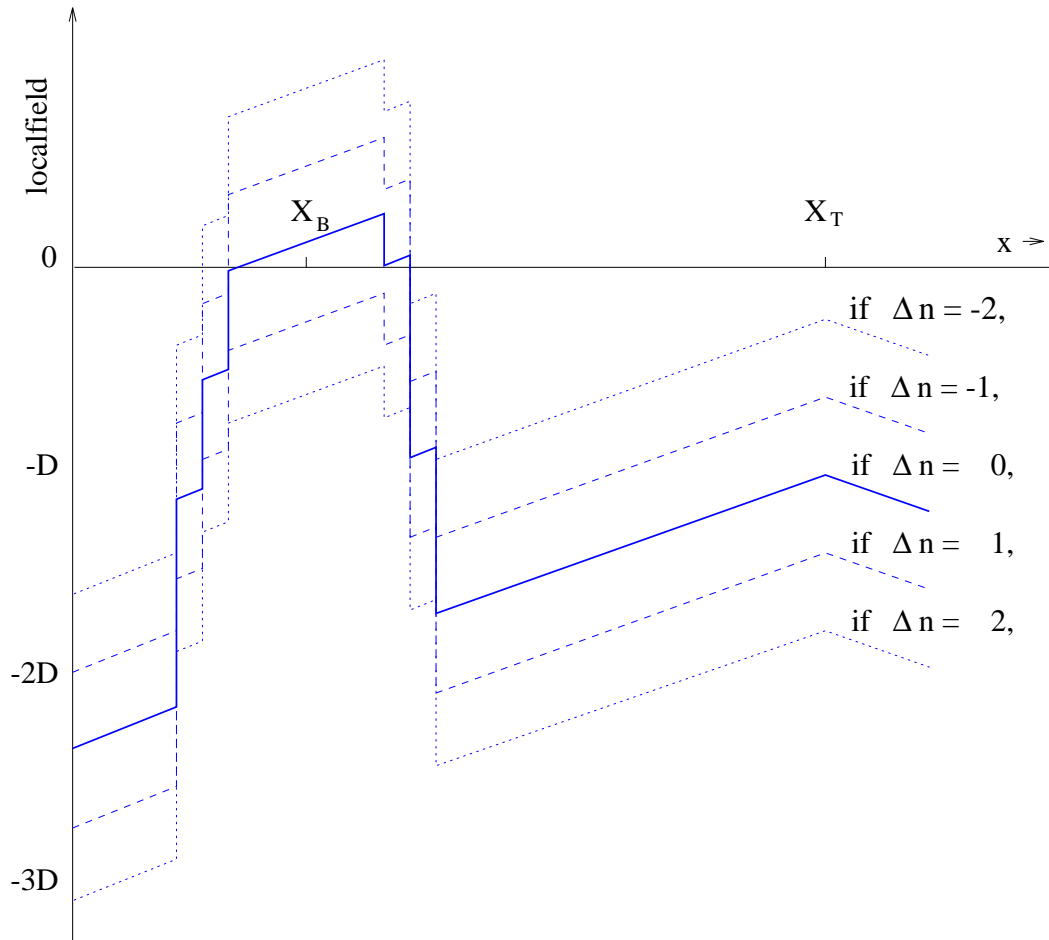


Figure 2: The total local field, the result of the weighted contribution of the three input terms, as a function of the relative distance x along the line between actual position and target position for different numbers of active neurons.

To estimate J_0 and α we will reconsider the energy function (4) in a new form

$$\mathcal{H} = -\frac{1}{2} \sum_{ij} B_{ij} \sigma_i \sigma_j - \frac{NJ_0}{2} (m^2 - 2m\mu) + \sum_i \alpha x_i \sigma_i \quad (25)$$

where we used (9). The first term in the energy function is at most of order ND and has its minimum when all neurons are in the same state. The last term is at most of order $\alpha N^{\frac{2}{3}}$ and forces all neurons to have activity "-1". The middle term has its minimum at $m = \mu$ and is then equal to $-\frac{1}{2} J_0 N \mu^2$. Since the number N of neurons in the network is large but finite, the contribution from the first and third term will cause that m is not precisely equal μ . To construct a system with magnetization m as close to the desired magnetization μ as possible, the middle term must be the dominating term. If, for example, we require that $(m - \mu) \propto N^{-1}$ then from (25) it follows that $J_0 \propto N$ and $J_0 \approx \alpha N^{2/3}$. This gives $\alpha \propto N^{-1/2}$.

The initial configuration of the system consists of a closed, circular cluster of activity, placed some distance away from the target position. The size of the initial cluster is always such that $m = \mu$. With these settings, the target neuron has a local field of $-D$.

If $\alpha \geq D$ then all neurons, at time zero, have negative local fields. Because changes in the state of neurons occur only if the local field and the state of the neuron have opposite signs, only active neurons can change their sign. If one active neuron changes its state, then $\Delta n = 1$ and the local fields of all neurons are incremented by an amount of $\frac{2J_0}{N}$. If still all neurons have negative fields, another neuron in the cluster changes state to "-1" and again all local fields are raised by $\frac{2J_0}{N}$. This continues until one of three possible situations occurs:

A) No active neurons are left to change their state to "-1" and all neurons still have negative local fields. This happens when $\alpha \geq D + J_0 (\mu + 1)$. This is a stable state of the system. All neurons which were initially in state "1", have changed their state to the rest value "-1" and the local field of all neurons is raised by $\frac{2\bar{n}^+ J_0}{N}$. However, the elevation was not large enough to raise one of the maxima above the zero level.

B) The local field of the neurons in the cluster exceeds that of the neurons in the target and is raised by successive steps to a positive value. This will happen when J_0 and α satisfy the following inequalities.

$$J_0 \leq \frac{D}{(m - \mu)} \quad \text{and} \quad \alpha \leq \frac{D}{(m - \mu) N^{\frac{2}{3}}} \quad (26)$$

Then neurons in the cluster, having all D neighbors active, have positive local field. Neurons outside the cluster, having negative valued neighbors only, have negative local fields. The instable neurons, having opposite state and local field, are those located near the border of the cluster. These neurons do not have all their neighbors in the same state.

As a result of the changing 'magnetization term' these border neurons are responsible for the movement of the cluster. The border neurons at the side of the cluster directed towards the target, have higher local fields relative to the border neurons at the opposite side of the cluster. In the case that Δn is negative, the system tends to change the state of a border neuron from "-1" to "1". This will happen with larger probability at the target side of the cluster. In the same way, if Δn is positive and

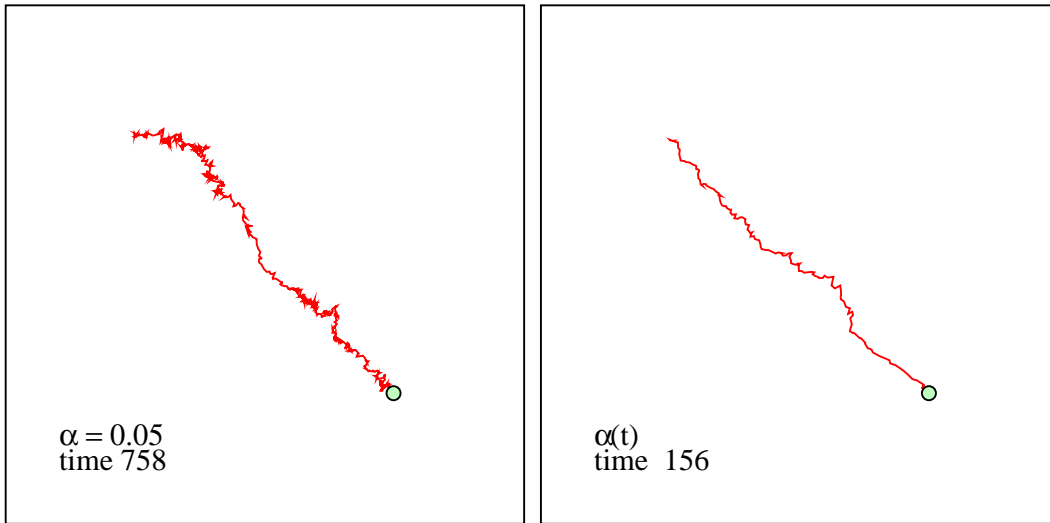


Figure 3: A) A trajectory of the center of the cluster with active neurons as the result of the dynamics of the neural network with α constant. The position of the target is shown by the circle. B) The simulation with optimal $\alpha(t)$. Both simulations were done with parameters $N = 2500$, $\tilde{n}^+ = 50$, $J_0 = 2500$ and $\beta = 2$. Note that the duration of the movement towards the target is longer in A than in B.

the system tends to turn the state of a neuron from "1" to "-1", this will happen with larger probability at the opposite side of the cluster. Both processes result in a shift of the cluster towards the target representation in the map.

C) The local field of the target neurons exceeds that of the neurons in the cluster and the local field of these neurons is raised by successive steps to a positive value. The cluster neurons having negative local fields change their state from "1" to "-1". The target neurons having positive local fields change their states from "-1" to "1". Hence the cluster disappears at its initial position and appears at the target position. The cluster seems to jump.

To increase the speed of movement of the cluster, we can increase α during the movement towards the target while J_0 is fixed. Because the distance of the cluster relative to the target representation decreases, the 'dragging' term for the cluster neurons becomes smaller. Consequently the influence of the 'magnetization' term becomes more noticeable which results into a smaller difference between m and μ . Regarding equation (26) this gives us the opportunity to increase α . A larger α results in a stronger 'dragging' field and a higher velocity of the movement of the cluster.

We performed simulations with constant α and time-varying $\alpha(t)$, with and without obstacles. Initially we chose $\alpha = 0.05$ and $\mu = -0.96$ which corresponds to a desired number of $\tilde{n}^+ = 50$ active neurons. We chose $J_0 = 2500$.

Figure 1.3A shows a typical result of a simulation with constant $\alpha = 0.05$. Figure 1.3B shows a result for the optimal value of $\alpha(t)$. Obstacles were not included.

In both cases the cluster moved in small discrete steps as a result of the dynamics of the neurons. The successive positions of the center of mass of the cluster present the trajectory in the configuration space. Because of the stochastic character of the model each simulation resulted into a different path.

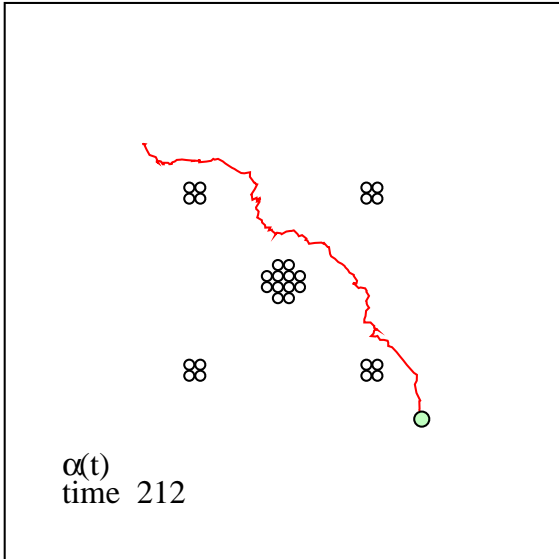


Figure 4: A) A trajectory of the center of the cluster with active neurons in the presence of obstacles: $\alpha = 0.05$. The position of an obstacle is shown by the circles. B) The simulation with optimal $\alpha(t)$. Both simulations were done with parameters $N = 2500$, $\tilde{n}^+ = 50$, $J_0 = 2500$ and $\beta = 2$. Note that the duration of the movement towards the target is longer in A than in B.

In the case that $\alpha = 1.0$ was chosen too large, the cluster disappeared at the initial position and appeared, due to the ‘magnetization’ term, at the position with the highest local field. The cluster jumped to the target.

Figure 1.4 shows the typical trajectory of the cluster in the presence of obstacles for optimal $\alpha(t)$. In the presence of noise ($T > 0$) and because of the random choice of a neuron at each time step, with $\alpha < \alpha(t)$ the jump of the cluster to the target position may still occur. The probability of a jump however, decreased with decreasing noise. For $\beta = 5$ these jumps were rarely seen.

The motion of the cluster can be seen as the Brownian motion in the presence of an external deterministic field αx_i . For the large parameter $\alpha(t)$ the deterministic dragging part becomes more important relative to the stochastic part. Hence the fluctuations decreased, the trajectory became more straight and the time to reach the target decreased.

3.2 Continuous neurons

In the simulations with continuous neurons we used a discrete-time version of the dynamics (18). At each time step Δt all neurons changed their state according to

$$u_i(t + \Delta t) = u_i(t) + \Delta t \left(\sum_j^N T_{ij} g(u_j(t)) + I_i - u_i(t) \right) \quad (27)$$

The sigmoid function $g(x) = \tanh(\beta x)$ where β was the gain parameter.

We chose $\beta = 0.1$, $J_0 = 2500$, $\alpha = 0.05$ and $\mu = 0.988$ which corresponded to a cluster with fifteen active neurons. Because of the deterministic nature of this

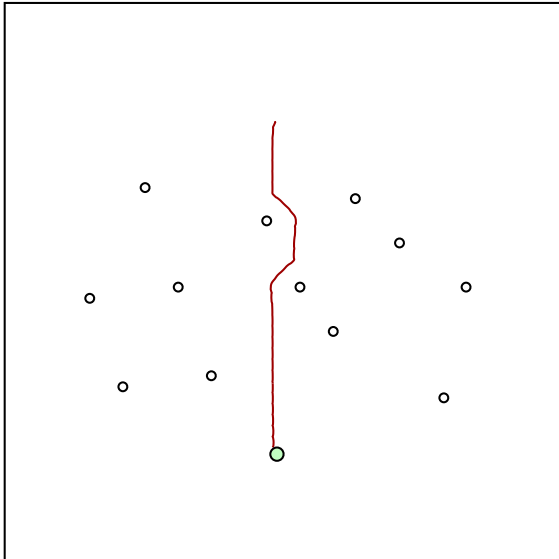


Figure 5: A) A continuous trajectory as a result of the dynamics of continuous neurons. The continuous line is the movement of the center of activities of the cluster in time. The gray circle corresponds to the target configuration. B) Trajectory in the presence of obstacles. The obstacle positions are represented by the open circles.

network, the paths which are generated by the network, are the same for all values of $\alpha < \alpha(t)$. The value of α only affects the speed of the movement of the cluster. Created trajectories were straight lines except for the case when the presence of obstacles prevented the straight line.

The results are illustrated in figure 1.5. Since the function (19) decreased on the trajectories given by (18) and was bounded from below, the system reached an equilibrium state in which the cluster reached the position of the target.

3.3 A Two-link Robot Manipulator

In this simulation we use the continuous model to control a two-link robot manipulator in the vicinity of an obstacle. The configuration space is a set of all possible configurations of the manipulator. All configurations with an overlap of manipulator and obstacle are called obstacle configurations. The neural map has the topographical order of the configuration space and is a discrete representation of it.

Figure 1.6(A) shows the configuration space with the positions of the obstacle neurons and the path as a result of the system dynamics. The open circles on the path correspond to the configurations of the manipulator in world space shown in figure 1.6(B).

Note that without the obstacle the path in configuration space, between the initial and target position, would be a straight line. In the presence of the obstacle the trajectory follows a curved path in order to avoid a collision with the obstacle.

Figure 1.7 shows the local field (left column) and the neural activity (right column) of the neurons in the network at three moments of time during the trajectory formation. The trajectory is given by the movement of the activity cluster in the neural map. The obstacles are represented by neurons with a negative local field.

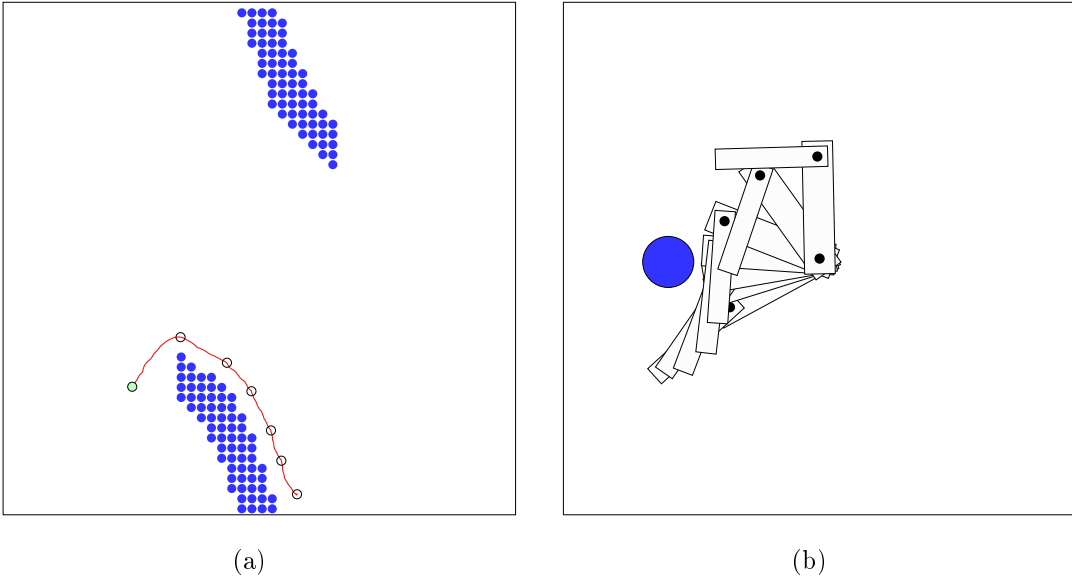
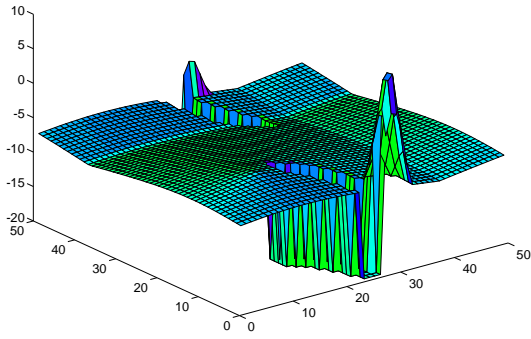


Figure 6: A) Configuration space of the manipulator. The dark dots represent the positions of the obstacle neurons. The smooth line is the path generated by the model. The open circles indicate intermediate positions of the cluster in configuration space corresponding to the manipulator configurations shown in figure: B) World space. Some intermediate manipulator configurations during the movement from the initial to the target configuration. The dark circle is the obstacle in world space.

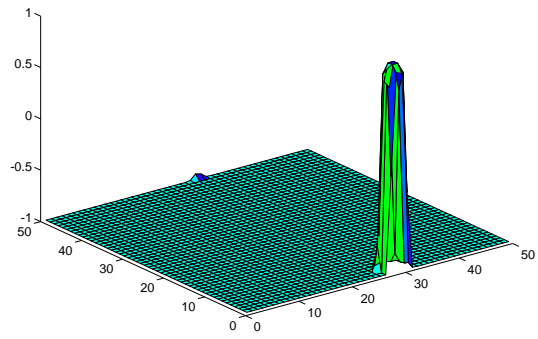
4 Discussion

In this paper we presented a neural network for trajectory planning and formation. The network consists of a large number of neurons connected by two types of interactions: short range excitatory and long range inhibitory connections. All neurons with positive activity formed a connected cluster due to the short range interactions. The total activity of these neurons was kept constant (neglecting small fluctuations in the case of discrete neurons) due to long range inhibitory connections. When the network received information about the position of the target, the initial activity of neurons started to change. During the time evolution the cluster moved to the target and the total activity of the network was constant. To give a macroscopic interpretation of the activity of the population of neurons we defined a vector-valued function. This function mapped the state of the network, the pattern of activity of neurons, into the position of the manipulator in configuration space. Encoding information about the position of the manipulator by the population of neurons gives the possibility to generate a smooth motion of the manipulator.

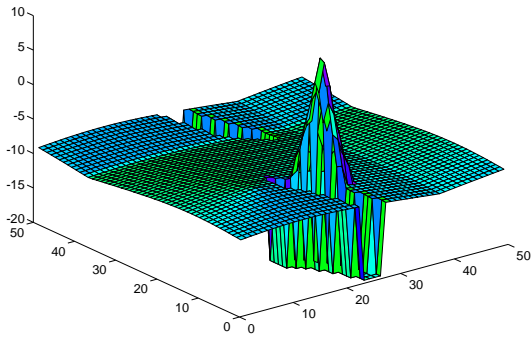
With a cluster consisting of many active neurons, the position of the manipulator corresponds to the position of the center of gravity of all active neurons. Therefore, in the model with discrete neurons, the trajectory of the manipulator through the configuration space has a resolution better than the resolution of the configuration space corresponding to the discrete grid of neurons. In the continuous neuron model and with continuous time, the path is also continuous. The network could also control the movement of the manipulator in the presence of obstacles. However, the performance of the network was limited only to obstacles of very regular and convex shape, because



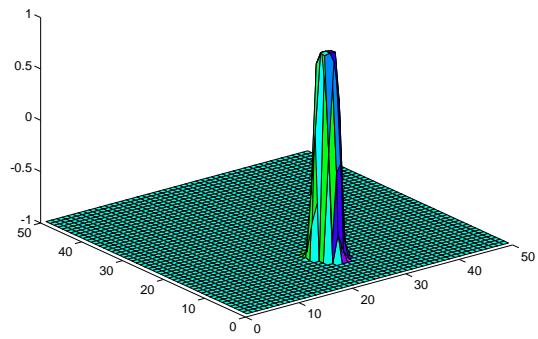
(a)



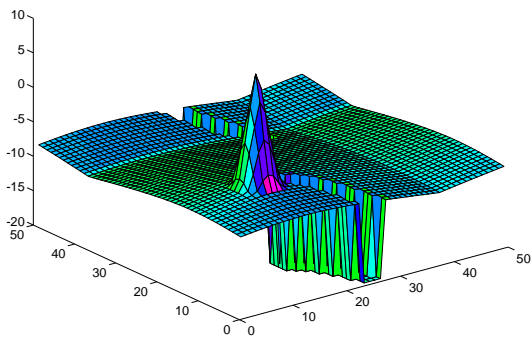
(b)



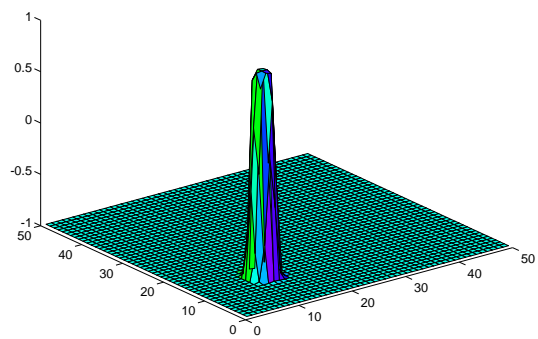
(c)



(d)



(e)



(f)

Figure 7: Local field (left column) and neural activity (right column) at three moments of time (at the beginning (A,B), at the end (E,F) and at an intermediate time (C,D)).

the dragging term $-\alpha x$ might drive the system into a local minimum for non-convex shaped objects. If obstacle and target move then the local minima change or may even disappear.

We studied a network with discrete and graded neurons. In both cases the network could control the movement of the manipulator from any initial configuration to any final configuration. However, there are a few properties which are worth mentioning. In the case of discrete neurons the system is stochastic. The movement of the activity cluster is similar to the Brownian motion of a particle in the presence of a deterministic external field. In the case of graded neurons the system is deterministic. This feature gives the possibility to make the system much faster than the previous one by parallel hardware implementation. The size of the cluster can change from a few neurons to hundreds of neurons and the system can still perform correctly.

Just like the potential-field methods, our model can be trapped in local minima. However, it is preferable over potential-field methods for the following reason. If the potential-field method is implemented in hardware for fast real-time applications, the accuracy of the method depends on the resolution of the discrete grid which represents the configuration space. For real-time applications our neural network is also implemented on a grid, but the resolution is much higher than that of the discrete potential field because the relative actual position is the weighted average of the cluster of activity. Our method has continuous dynamics and generates a continuous trajectory, contrary to the discrete potential field.

The continuous model possesses some biological features. For example, it has been shown in neurophysiological experiments that information is not coded by a single neuron, but rather by a group of active neurons, which are frequently clustered together. An example may be the superior colliculus which plays an important role in the generation of saccadic eye movements [18]. The neurons in the deep layers of the superior colliculus have a movement field, which is defined as the set of vectorial eye movements for which the neuron will be active. The neurons are organized in a topographically ordered way in the sense that neighboring neurons have neighboring movement fields [22]. Recent experiments [18] have shown that at the beginning of a saccade a particular cluster of neurons is active. This cluster of active neurons shifts along the superior colliculus map until the eye is on target. For these observations a model has been proposed [5], which basically represents a shift of a cluster of active neurons in a topographical map. The present paper gives a more mathematically elaborated version of that model and in addition explains how the "shifting" cluster can avoid obstacles.

In the last few years the notion of "population vector" was used frequently for interpreting neuronal responses in motor cortex. The activity of neurons in motor cortex is related to arm movements [2, 6, 12]. Because it not known whether motor-cortical neurons are organized in a topographically ordered way, it is not possible yet to apply the model directly to the dynamics of the shifting population vector. However, the model helps to understand the characteristics of the population vector.

Summarizing, we have presented a neural network, which by its intrinsic dynamics can generate a continuous trajectory without collisions with obstacles. This is true even when the obstacles are moving.

Acknowledgments

We would like to thank Ton Coolen for suggesting this approach.

This work is partly supported by the Dutch Foundation for Neural Networks and by the Dutch Foundation for Scientific Research (NWO).

References

- [1] J. Barraquand and J.C. Latombe. Robot motion planning with many degrees of freedom and dynamic constraints. In *Proceedings of the 5th International Symposium on Robotics Research (Tokyo)*, pages 74–83, 1989.
- [2] R. Camaniti, P. B. Johnson, Y. Burnod, C. Galli, and Ferraina S. Shifts of preferred directions of premotor cortical cells with arm movements performed across the work-space. *Experimental Brain Research*, 83:228, 1990.
- [3] M. A. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:815, 1983.
- [4] J.L. Crowley. Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, RA-1:31, 1985.
- [5] J. Droulez and A. Berthoz. A neural network model of sensoritopic maps with predictive short term memory properties. *Proceedings of the National Academy of Sciences of the United States of America*, 88:9653, 1991.
- [6] A.P. Georgopoulos, J. Ashe, N. Smyrnis, and M. Taira. The motor cortex and the coding of force. *Science*, 256:1692–1695, 1992.
- [7] R. Glasius, A. Komoda, and C.C.A.M. Gielen. Neural network dynamics for trajectory formation and obstacle avoidance. *Neural Networks (in press)*, 1994.
- [8] R.J. Glauber. Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4:294–307, 1963.
- [9] S. Grossberg. Nonlinear neural networks: principles, mechanisms, and architectures. *Neural Networks*, 1:17, 1988.
- [10] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 81:3088, 1984.
- [11] R. A. Jarris. Collision-free trajectory planning using distance transforms. *Mech Eng Trans of the IE Aust.*, ME10:187, 1985.
- [12] J. F Kalaska, D. Crammond, D. A. D. Cohen, Prud’homme M., and M. L. Hyde. *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 Verlag, New York, 1992.
- [13] K. Kant and S.W. Zucker. Towards efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research*, 5:72–89, 1986.
- [14] O. Kathib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5:90–98, 1986.

- [15] B.H. Krogh. A generalized potential field approach to obstacle avoidance control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, page 948, Los Angeles, 1984. Computer Society Press of the IEEE.
- [16] B.H. Krogh and C.E. Thorpe. Integrated path planning and dynamic steering control for autonomous vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation (Washington DC)*, pages 1664–1669, Los Angeles, 1986. Computer Society Press of the IEEE.
- [17] T. Lozano-Perez. Spatial planning: a configuration space approach. *IEEE Transactions on Computers*, C-32:108–120, 1983.
- [18] D.P. Munoz, D. Pelisson, and D. Guitton. Movement of neural activity on the superior colliculus motormap during gaze shifts. *Science*, 251:1358–1360, 1991.
- [19] C. Peterson and J. R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.
- [20] H. J. Ritter, T. M. Martinetz, and K. J. Schulten. Topology-conserving maps for learning visuo-motor-coordination. *Neural Networks*, 2:159–168, 1989.
- [21] H. J. Ritter, T. M. Martinetz, and K. J. Schulten. *Neural computation and self-organizing maps. An introduction*. Addison-Wesley Publ. Comp., 1992.
- [22] D. L. Sparks. Neural cartography: Sensory and motor maps in the superior colliculus. *Brain Behavior and Evolution*, 31:49–56, 1988.
- [23] P. P. van der Smagt. Minimisation methods for training feed-forward networks. *Neural Networks*, 7:1–11, 1994.
- [24] P. P. van der Smagt and B. J. A. Krose. A real-time neural robot controller. In *Proceedings of the International Conference on Artificial Neural Networks (Espoo/Finland)*, pages 351–356, Amsterdam; North-Holland, 1991. Elsevier Science Publishers.
- [25] C.K. Yap. *Algorithmic motion planning*. Hillsdale, N.J., Erlbaum, 1986.