A Cuneate-Based Network and Its Application as a Spatio-Temporal Filter in Mobile Robotics

Eduardo Sánchez, Manuel Mucientes, and Senén Barro

Grupo de Sistemas Intelixentes (GSI)
Departamento de Electrónica e Computación, Facultade de Física,
Universidade de Santiago de Compostela,
15706 Santiago de Compostela, Spain
{teddy, manuel, senen}@dec.usc.es
http://www-gsi.dec.usc.es

Abstract. This paper focuses on a cuneate-based network (CBN), a connectionist model of the cuneate nucleus that shows spatial and temporal filtering mechanisms. The circuitry underlying these mechanisms were analyzed in a previous study by means of a realistic computational model [9, 10] of the cuneate. In that study we have used experimental data (intracellular and extracellular recordings) obtained in cat in *vivo* [2, 3] to guide and test the model. The CBN is a high-level description of the realistic model that allows to focus on the functional features and hide biological details. To demonstrate the CBN capabilities we have applied it to solve a filtering problem in mobile robotics.

1 Introduction

The cuneate nucleus is a part of the somato-sensory system and constitutes, in conjunction with the gracile nucleus, the dorsal column nuclei (DCN). Its afferent inputs, called Primary Afferent Fibers (PAF), are originated in both cutaneous and proprioceptive receptors of the upper body. The input signals are processed by a circuitry composed mainly by two different types of cells: projection neurons, also called cuneothalamic or relay neurons, and local neurons, also called interneurons.

Intracellular recordings obtained under cutaneous and lemniscal stimulation show that the afferent fibers can establish excitatory and inhibitory synaptic connections with the cuneothalamic neurons [2]. In addition, distinct types of recurrent collaterals with the capability of either exciting or inhibiting both cuneothalamic neurons and interneurons were also discovered [3]. With these data we can generate hypothesis about which circuits are implicated and also elaborate computational models to study their processing capabilities [9, 10]. The Cuneate-Based Network (CBN) is a connectionist model that describes the local circuitry of the cuneate, that means the circuitry without considering cortico-cuneate inputs. Our studies show that such circuit can detect dynamic patterns [10]. The CBN will be introduced in section 2.

J. Mira and A. Prieto (Eds.): IWANN 2001, LNCS 2085, pp. 418-425, 2001.

To test the CBN capabilities we have applied it in a filtering problem in mobile robotics. The CBN performs a spatio-temporal filtering which goal is to improve the perceived trajectory made by a mobile obstacle. Section 3 explains that problem and how the network is integrated in a system that performs the task of collision avoidance of mobile obstacles [8]. The results, obtained in an experiment with a real robot, are shown in section 4.

2 Cuneate-Based Network

The Cuneate-Based Network focuses on the functional features of a realistic model previously developed. In that model we have studied: (1) the spatial filtering capabilities of the center-surround receptive fields and the recurrent lateral connections, and (2) the temporal filtering capabilities provided by presumed autoinhibitory connections. Figure 1 shows how the CBN architecture integrates these different connectivity in a single circuitry.

The output $y_j(t)$ of j-th neuron is calculated after applying a threshold function Ψ to the total input. This value is the result of adding, at a given time t, the contribution of: (1) afferent input $v_j(t)$, inhibitory lateral connections $u_j(t-\triangle t)$, and the output from the corresponding inhibitory neuron $o_j(t-\triangle t)$. u_j and o_j

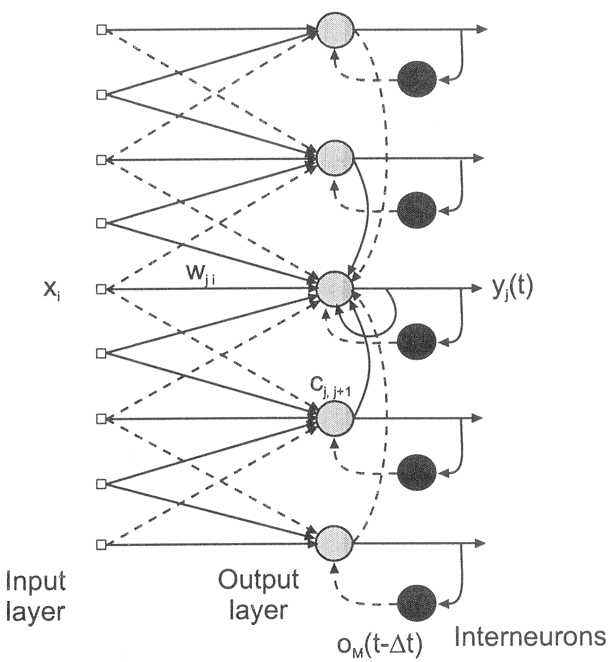


Fig. 1. CBN architecture.

are weighted by constant parameters α and β, respectively, that determine the strength of both types of connections:

$$y_j(t) = \Psi(z_j(t)) = \Psi(v_j(t) + \alpha u_j(t - \Delta t) + \beta o_j(t - \Delta t)) \; j = 1, 2, ..., M \tag{1}$$

where Ψ is defined as follows,

$$\Psi(z_j(t)) = \begin{cases} z_j(t) \text{ si } z_j \geq U \\ 0 \quad\;\; \text{si } z_j < U \end{cases} \tag{2}$$

The contribution of afferent input to the j-th neuron is computed considering that middle inputs are excitatory and lateral inputs are inhibitory.

$$v_j = \sum_{i=n_j-C}^{n_j+C} w_{ji} x_i, \; j = 1, 2, ..., M \tag{3}$$

where v_j denotes the total afferent input to the j-th neuron, 2C+1 denotes the width of the receptive field, w_{ji} denotes the strength of the connection between the i-th input and the j-th output neuron (such that w_{ji}=0, if $i \leq 0$ or $i > N$), and x_i is the value of i-th input.

The lateral inhibition mechanism can be implemented with the popular Mexican-hat function. The contribution of these connections is the following:

$$u_j = \sum_{k=-K}^{K} c_{j,j+k} y_{j+k}, \; j = 1, 2, ..., M \tag{4}$$

with K denoting the semi-width of the Mexican-hat and $c_{j,j+k}$ the value of the lateral connections between j-th and i-th neuron. Following the Mexican-hat distribution, $c_{j,j+k}$ is positive, i.e excitatory connection, for first-order neighbours, and negative, i.e inhibitory connection, otherwise (again, $c_{ji} = 0$, if $i \leq 0$ or $i > M$).

The temporal filtering mechanism relies in the inhibition of neurons of the output layer if the inputs persist in time. To implement this mechanism we have used the results from Koch et. al [7], that explain how the time constant of a biological neuron can vary as a function of the input activity. In our approach we have computed the difference $\triangle Y$ between the current and the last input vector:

$$\triangle Y = y_j^{output}(t) - y_j^{output}(t-1) \tag{5}$$

The time constant τ will be a function of the variable $\triangle Y$:

$$\tau(\triangle Y) = \begin{cases} \tau + \triangle\tau \quad\;\;\; \text{if } \triangle Y \simeq 0 \\ \tau - K \triangle \tau \text{ if } \triangle Y \not\simeq 0 \end{cases} \tag{6}$$

The time constant τ can take any value from a range between τ_{min}=1 and some maximum τ_{max}. The time constant defines the temporal window to integrate the inputs, which in turn, determines the total input sum to the interneuron $q_j(t)$:

$$q_j(t) = \sum_t^{t-\tau(\triangle Y)} y_j^{output}(t), \, j = 1, 2, ..., M \tag{7}$$

Finally, the output of interneurons will depend on the result of a threshold function similar to the one that was introduced for the output layer.

3 The problem of collision avoidance of mobile obstacles

As it was explained before, we want to demonstrate the functional capabilities of the CBN by applying it on a real problem. We found a suitable one in the domain of mobile robotics. The task consists on filtering sonar data to improve the trajectory estimation of a mobile obstacle detected in the environment. This task is one of the subproblems encountered in the design of behaviours for collision avoidance of mobile obstacles [8]. This task can be decomposed in three different phases (figure 2): the perception of mobile objects, the appropriate decision-making to avoid them, and the execution of a certain motor action. Because the robot has to react dynamically to changes in the environment, the main challenge of this task is to achieve real-time performance.

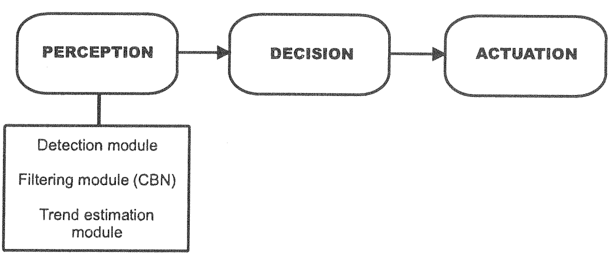


Fig. 2. Modules of the mobile obstacle avoidance task.

In the perceptual phase, the detection operation takes the raw data provided by a ring of sonar sensors and indicates a set of locations with high probability of being occupied by a mobile object. Unfortunately, this processing is not enough to estimate accurately both the direction and the speed of the obstacle. Some problems arose after the detection process:

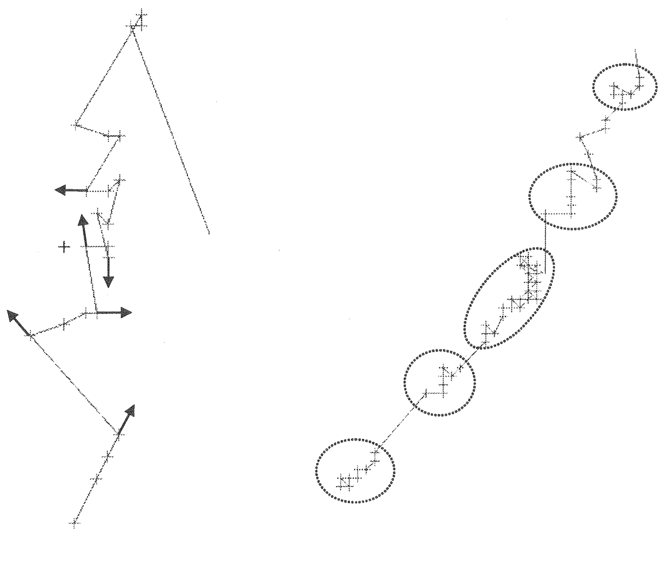
1. Estimation of motion direction. Because of the motion of both robot and obstacle, the sonar shows a irregular trajectory of mobile obstacle points. Even if the trajectory tendency is clearly linear, in a local region the points seem to follow a random pattern (figure 3).

2. Estimation of speed. If the mobile obstacle is detected by a number of different ultrasound sensors, the data entries associated to that obstacle show space discontinuities, also called "jumps". These "jumps" appear when the sonar that tracks the motion changed. These discontinuities, abrupt in most cases, can be understood as important changes in the mobile obstacle speed (figure 3).

To solve these problems we have introduced the CBN after the detection operation (see figure 2). The overall system can detect obstacles, perceive obstacles trends and act accordingly to avoid collisions. With these features, the system is robust and can operate in real time.

4 Results

Before integrating the CBN with the architecture shown in figure 2, we have tested its individual performance with off-line simulations. For testing we have



(A) **(B)**

Fig. 3. Issues in the detection of mobile obstacles. (A) Set of points associated with a mobile obstacle. Arrows indicating the perceived direction at any given time is shown. Although the global trend is linear, the local trend is irregular and shows continuous changes in direction. (B) An example with data groupings is shown. These groupings are originated when there is a change in the ultrasound sensor that is performing the detection. These groupings impede an accurate speed estimation.

used real data obtained from the sensors and preprocessed by the detection stage. The goal of the simulations was to test if the CBN would be able to remove noise and improve the estimation of the trajectory of the mobile obstacles. The CBN was implemented in NSL (*Neural Simulator Language*) and the simulations were run on the NSL environment for Unix platforms.

The real experiments were performed with a *NOMAD 200* robot in the Department of Electronics and Computer Science, of our university. Initially we located the robot in one of the sides of the entrance. The robot begins its movement following a linear trajectory with the goal of reaching the other side avoiding mobile obstacles that would imply collision or impede the goal achievement. Figure 4 shows how a mobile obstacle is detected from the left side of the robot during three cycles. In the two following cycles the ultrasound sensors do not

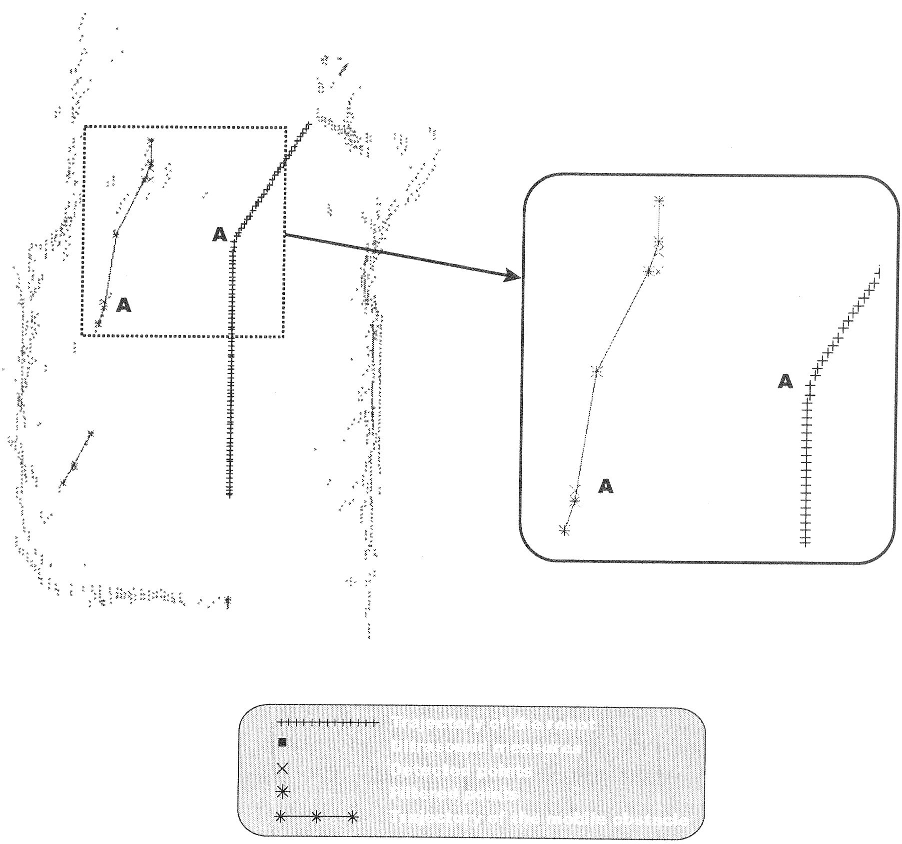


Fig. 4. Label A indicates the position of both the robot and the mobile obstacle at the time when robot begins to avoid the obstacle. The cross density in the robot trajectory indicates its speed at any given time. The less the speed, the more number of crosses, and vice versa.

provide new object information and the robot temporarily loose the trajectory. When the detection is retaken, the trend module perceives a collision situation. The object trend is, initially, *indifferent*, because no change in motion or angle is detected. As a result, the robot decides to implement the *observe* behavior and it will wait to detect either changes in the object behavior or a decrease of the collision time. In the next cycles, the mobile obstacle reduces its speed, from 41 to 34 cm/s, and so the parameter **collision time** increases. As a consequence, the control system selects the *cross first* behavior even though the mobile obstacle trend is still *indifferent*. The robot executes this behavior by turning right 30 degree and accelerating from 17.5 to 22.5 cm/s. Both actions are enough to avoid the possible collision and to continue the trajectory to achieve the initial goal.

5 Discussion

The three local mechanisms of the cuneate nucleus discussed in this paper (center-surround receptive fields, lateral recurrent connections and autoinhibitory connections) can be found in many other nucleus of the brain and have been used independently to develop other bio-inspired connectionist models. At this stage, the purpose of the CBN is not to provide a novelty general-purpose artificial neural network, but to integrate all mentioned mechanisms in a single architecture and analyze the functional capabilities of it.

The lateral inhibition, for example, is a well-known mechanisms to accomplish competitive computation. A reference model that captures this computational feature is the Didday model [4]. The goal here was to analyze competitive mechanism to understand the circuitry underlying the capturing process in frogs. The CBN architecture is though different in two aspects: (1) the number of required interneurons, and (2) the degree of inhibition exerted over each neuron in the output layer. The Didday model shows only one interneuron that, in turn, determines the same level of inhibition over the neurons in the output later. On the other side, the CBN shows an interneuron per neuron in the output layer and the degree of inhibition on each neuron is different.

Similarly, the center-surround receptor fields have also been a source of inspiration for other connectionist models. Buonomano and Merzenich [1] have developed a hierarchical network with this type of receptive field. This arrangement induces a response in the neurons of the output layer, whose temporal codification allows input pattern recognition. Other classical example can be found in the *Neocognitron* network [5]. This network was inspired by Hubel and Wiesel ideas about the hierarchical organization of the visual receptive fields.

Recently, autoinhibitory mechanisms have been used in attentional models in the visual system [6]. In this study, the autoinhibition is called *inhibition-of-return* and allows the system to change the focus of attention if new remarkable features appears in the environment. The functionality, very similar to the one presented in the CBN, consists on inhibiting, after some period of time, those winner neurons in the previous iteration with the aim to allow new incoming events to be detected.

From a functional point of view, the cuneate-based network performs a kind of spatial and temporal filtering, which was clearly showed by applying it on a collision avoidance of mobile obstacles problem. Center-surround receptive fields and lateral inhibition select salient maximal locations in local regions of the trajectory of the mobile obstacle. With this mechanism, CBN sends the most salient group of points of the mobile obstacle trajectory to subsequent processing modules. The temporal filtering removes the persistent objects, so it permits that other processing modules would consider those points relevant to the trajectory of the mobile obstacle. As a conclusion, the CBN has demonstrated the cuneate capabilities to perform an spatio-temporal filtering over incoming sensory information. As far as we know, this is the first connectionist model about the cuneate nucleus.

Acknowledgments

Authors would like to thank to **Laboratorios de Neurociencia y Computación neuronal (LANCON)**, and also, to acknowledge the support from grants TIC2000-0873 and PGIDT99PXI20601B.

References

1. Buonomano D. V. and Merzenich M. A neural network model of temporal code generation and position-invariant pattern recognition. Neural computation. Vol. 11 (1999) 103–116.
2. Canedo A., Aguilar J.: Spatial and cortical influences exerted on cuneothalamic and thalamocortical neurons of the cat. European Journal of Neuroscience. Vol. 12 **2** (2000) 2513–2533.
3. Canedo A., Aguilar J., Mariño J.: Lemniscal recurrent and transcortical influences on cuneate neurons. Neuroscience. Vol. 97 **2** (2000) 317–334.
4. Didday R. L. The simulation and modelling of distributed information processing in the frog visual system. Ph. D. Thesis. Stanford University. (1970).
5. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechnsim of pattern recognition unaffected by shift in position. Biol. Cybern. Vol. 55 (1980) 5–15.
6. Itti L. and Koch C. Computational modelling of visual attention. Nature Reviews Neuroscience. Vol. 2 (2001) 1–9.
7. Koch C., Rapp M., Segev I.: A Brief History of Time Constants. Cerebral cortex. Num. 6 (1996) 93–101.
8. Mucientes M., Iglesias R., Regueiro C. V., Bugarín A., Cariñena P., Barro S.: Use of Fuzzy Temporal Rules for avoidance of moving obstacles in mobile robotics. Proceedings of the 1999 Eusflat-Estylf Joint Conference. (1999) 167–170.
9. Sánchez E., Barro S., Mariño J, Canedo A., Vázquez P.: Modelling the circuitry of the cuneate nucleus. In Lecture Notes in Computer Science. Volume I. Springer Verlag. Mira J. and Sánchez Andrés J. V. (Eds). (1999) 73-85.
10. Sánchez E., Barro S., Mariño J., Canedo A.: A realistic computational model of the local circuitry of the cuneate nucleus. Also included in this volume. (2001).