Design and Implementation of an Adaptive Neuro-controller for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots

Francisco García-Córdova[1], Antonio Guerrero-González[1], and Fulgencio Marín-García[2]

[1] Department of System Engineering and Automation,
[2] Department of Electrical Engineering
Polytechnic University of Cartagena (UPCT),
Campus Muralla del Mar, 30202, Cartagena, Murcia, Spain.
{francisco.garcia, antonio.guerrero, pentxo.marin}@upct.es

Abstract. A kinematic adaptive neuro-controller for trajectory tracking of nonholonomic mobile robots is proposed. The kinematic adaptive neuro-controller is a real-time, unsupervised neural network that learns to control a nonholonomic mobile robot in a nonstationary environment, which is termed Self-Organization Direction Mapping Network (SODMN), and combines associative learning and Vector Associative Map (VAM) learning to generate transformations between spatial and velocity coordinates. The transformations are learned in an unsupervised training phase, during which the robot moves as a result of randomly selected wheel velocities. The robot learns the relationship between these velocities and the resulting incremental movements. The neural network requires no knowledge of the geometry of the robot or of the quality, number, or configuration of the robot's sensors. The efficacy of the proposed neural architecture is tested experimentally by a differentially driven mobile robot.

1 Introduction

Several heuristic approaches based on neural networks (NNs) have been proposed for identification and adaptive control of nonlinear dynamic systems. More recently, the efforts have been directed toward the development of control schemes which, besides providing improved performance, can be proved to be stable [1].

In the trajectory-tracking problem, the wheeled mobile robot (WMR) is to follow a prespecified trajectory. Using the kinematic model of WMRs, the trajectory-tracking problem was solved in [2]. Both the local and global tracking problem with exponential convergence have been solved theoretically using time-varying state feedback based on the backstepping technique in [3]. The kinematic model of mobile robots can be described in polar coordinates [4] and stabilization achieved using the backstepping technique. Dynamic feedback linearization has been used for trajectory tracking and posture stabilization of mobile robot systems in chained form [5].

The approximate linearization of robot equations about the desired reference trajectory has been used for developing trajectory tracking techniques [2], [6] as well as non-linear approaches like the local continuous nonlinear feedback action and the feedback linearization [7], [6]. The backstepping paradigm has been used as well and a number of recursive techniques for trajectory tracking have been proposed [1], [8]. Recent advances tend to focus on the robustness of trajectory tracking techniques with respect to input disturbances and/or parameters variations [9]. In this framework, sliding mode control has been extensively used. Tracking problems for mobile robots not satisfying the nonholonomic constraints have been addressed by a slow manifold method, by using continuous-time variable structure [10] and by a discrete-time sliding mode control technique [11].

Neural networks based controllers have been deeply investigated (see, e.g. [12], [13], [1]), since can be used to learn nonlinear functions, representing direct dynamics, inverse dynamics or any other mapping in the process. In the field of WMRs control, NN are embedded in the closed-loop control system of a WMR and the corresponding NNs-based controllers have shown to effectively deal with unmodelled bounded disturbances and/or unstructured unmodelled dynamics in the mobile robot model [13], [1]. In particular in [1] and [14] a neural network based controller has been developed by combining the feedback velocity control technique and torque controller, using a multilayer feedforward neural network. In this approach the universal approximation property of neural network is used to learn the dynamics of the mobile robot. The resulting control structure and the neural network learning algorithm are very complicated and they require high computational efforts. In the above quoted contributions only numerical simulations studies have been developed. In [13] a simple NN based controller has been proposed by integrating the kinematic control techniques with the universal approximation capabilities of neural networks. In this approach the neural network capabilities to learn unmodelled robot characteristics in a real experimental setup are used for improving the performance of the kinematic control techniques.

The study of autonomous behavior has become an active research area in the field of robotics. Even the simplest organisms are capable of behavioral feats unimaginable for the most sophisticated machines. When an animal has to operate in an unknown environment it must somehow learn to predict the consequences of its own actions. By learning the causality of environmental events, it becomes possible for an animal to predict future and new events [15]. Somehow this learning is possible for organisms in spite of what seem like insurmountable difficulties from a standard engineering viewpoint: noisy sensors, unknown kinematics and dynamics, nonstationary statistics, and so on [16].

In this paper, a real-time, unsupervised neural network controller that can learn to guide a mobile robot towards a target located at an arbitrary location in a 2-D workspace. The robot's movements are controlled by selecting the angular velocity of each of two driving wheels. The neuro-controller we propose is based on existing neural networks of biological sensory-motor control. The neuro-controller requires no information about the robot's structure, and it is resistant to a variety of disturbances. The kinematic adaptive neuro-controller is a Self-Organization Direction Mapping Network (SODMN), uses an associative learning to generate transformations between spatial and velocity coordinates. The efficacy of the proposed neural controller is tested experimentally by a differentially driven mobile robot.

This paper is organized as follows. We first describe the neural control system to mobile robot tracking and approach behaviors in Section II. Experimental results with the proposed scheme for control of a mobile platform are addressed in Section III. Finally, in Section IV, conclusions based on experimental results are given.

2 Architecture of the Neural Control System

Figure 1 illustrates our proposed neural architecture. The trajectory tracking control without obstacles is implemented by the SODMN. The SODMN learns to control the robot through a sequence of spontaneously generated random movements. The random movements enable the neural network to learn the relationship between angular velocities applied at the wheels and the incremental displacement that ensues during a fixed time step. The proposed SODMN combines associative learning and Vector Associative Map (VAM [17]) learning to generate transformations between spatial and velocity coordinates. The nature of the proposed kinematic adaptive neuro-controller is that continuously calculates a vectorial difference between desired and actual velocities, the robot can move to arbitrary distances and angles even though during the initial training phase it has only sampled a small range of displacements. Furthermore, the online error-correcting properties of the proposed architecture endow the controller with many useful properties, such as the ability to reach targets in spite of drastic changes of robot's parameters or other perturbations.

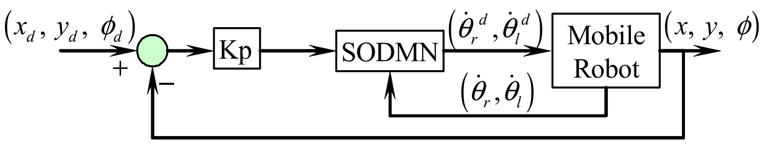


Fig. 1. Neural architecture for reactive and adaptive navigation of a mobile robot.

2.1 Self-Organization Direction Mapping Network (SODMN)

At a given set of angular velocities the differential relationship between mobile robot motions in spatial coordinates and angular velocities of wheels is expressed like a linear mapping. This mapping varies with the velocities of wheels. The transformation of spatial directions to wheels' angular velocities is shown in Fig. 2. The spatial error is computed to get a spatial direction vector (DVs). The DVs is transformed by the *direction mapping network* elements V_{ik} to corresponding motor direction vector (DVm). On the other hand, a set of tonically active inhibitory cells which receive broad-based inputs that determine the context of a motor action was implemented as a context field. The context field selects the V_{ik} elements based on the wheels' angular velocities configuration.

A speed-control GO signal acts as a nonspecific multiplicative gate and control the movement's overall speed. The GO signal is a input from a decision center in the brain, and starts at zero before movement and then grows smoothly to a positive value as the movement develops. During the learning, sensed angular velocities of wheels are fed into the DVm and the GO signal is inactive.

Activities of cells of the DVs are represented in the neural network by quantities $(S_1, S_2, ..., S_m)$, while activities of the cells of the motor direction vector (DVm) are represented by quantities $(R_1, R_2, ..., R_n)$. The direction mapping is formed with a field of cells with activities V_{ik}. Each V_{ik} cell receives the complete set of spatial inputs S_j, $j = 1, ..., m$, but connects to only one R_i cell (see Figure 2). The mechanism that is used to ensure weights converge to the correct linear mapping is similar to the VAM learning construction [17]. The direction mapping cells ($\boldsymbol{V} \in \mathbb{R}^{n \times k}$) compute a difference of activity between the spatial and motor direction vectors via feedback from DVm. During learning, this difference drives the adjustment of the weights. During performance, the difference drives DVm activity to the value encoded in the learned mapping.

A context field cell pauses when it recognizes a particular velocity state (i.e., a velocity configuration) on its inputs, and thereby disinhibits its target cells. The target cells (direction mapping cells) are completely shut off when their context cells are active. This is shown in Fig. 2. Each context field cell projects to a set of direction mapping cells, one for each velocity vector component. Each velocity vector component has a set of direction mapping cells associated with it, one for each context. A cell is "off" for a compact region of the velocity space. It is assumed for simplicity that only one context field cell turns "off" at a time. In Figure 2, inactive cells are shown as white disks. The center context field cell is "off" when the angular velocities are in the center region of the velocity space, in this two degree-of-freedom example. The "off" context cell enables a subset of direction mapping cells through the inhibition variable c_k, while "on" context cells disable to the other subsets.

The DVs cell activities, $\boldsymbol{S} \in \mathbb{R}^m$, are driven by the desired spatial direction, $\boldsymbol{xd} \in \mathbb{R}^m$, computed from the difference of the visual target position and the mobile-robot current position

$$\frac{d}{dt} S_j = \delta(xd_j - S_j). \tag{1}$$

where δ is a gain that controls the integration speed rate.

Direction mapping cells with activity V_{ik} compute the difference of the weighted DVs input and the DVm activity. The V_{ik} cell activities are described as

$$\frac{d}{dt} V_{ik} = \alpha(-V_{ik} + c_k(\sum_j z_{jik} S_j - R_i)), \tag{2}$$

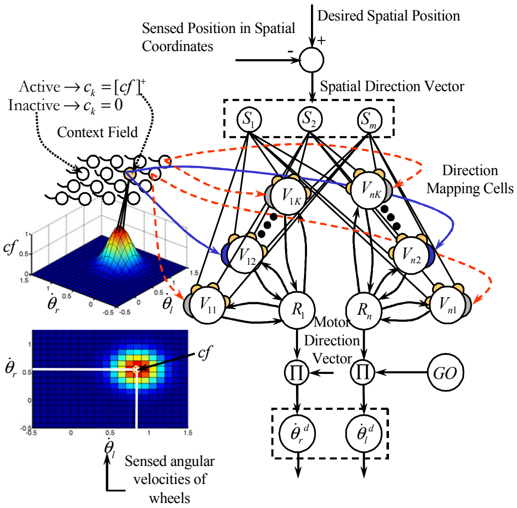


Fig. 2. Self-organization direction mapping network for the trajectory tracking of a mobile robot.

where α is a time constant, the coefficients c_k $(k = 1, ..., K)$ represent inhibition from the context field and z_{jik} represents the element of the inverse mapping which multiplies the j^{th} spatial components to contribute to the i^{th} velocity component. Here, the spatial representation contains m-components and n is the number of independent wheels. The k^{th} context field contacts the set $\{V_{ik}, i = 1, ..., n\}$ of direction mapping cells (see Fig. 2). When the context cell is active (modeled as $c_k = 0$), the entire input current to the soma is shunted away such that there remains only activity in the axon hillock, which decays to zero. When the k^{th} context cell shuts off, $c_k = 1$, the V_{ik} receive normal input. The number of cells of the context field is K and it is calculated as

$$K = \left(\prod_{j=1}^{n} D_j \right), \tag{3}$$

where D_j is the number of intervals of the discrete velocity range. Consequently, the total number of the weights, z_T, at the neural network is given by

$$z_T = n.m.K, \tag{4}$$

where m is the number of spatial dimensions.

The motor direction cell activities, $\boldsymbol{R} \in \mathbb{R}^n$, are driven by the V_{ik} during performance and by wheels' rotation velocities $\dot{\theta}_i$ during learning

$$\frac{d}{dt} R_i = \delta \left[(1 - e) \left(\sum_k V_{ik} - R_i \right) + e(\dot{\theta}_i - R_i) \right]. \tag{5}$$

In the learning phase, the endogenous random generator (ERG) circuit is activated, $e = 1$ and the R_i cells are driven to wheels' sensed velocities $\dot{\theta}_i$. During the performance, the ERG circuit is inactive, $e = 0$, and the input is the sum of the V_{ik}, only one of which will be actively processing input.

The Motor Present Direction Vector (PDVm) cell activities ($\dot{\theta}_{id}$) are given by

$$\dot{\theta}_{id} = \alpha(R_i g + e\dot{\theta}_{i_{ERG}}), \tag{6}$$

where g is the sigmoidal GO signal and is described by

$$\begin{aligned} \frac{d}{dt}g^{(1)} &= \varepsilon\left[-g^{(1)} + \left(Cs - g^{(1)}\right)g^{(0)}\right], \\ \frac{d}{dt}g^{(2)} &= \varepsilon\left[-g^{(2)} + \left(Cs - g^{(2)}\right)g^{(1)}\right], \\ g &= g^{(0)}\frac{g^{(2)}}{Cs}. \end{aligned} \tag{7}$$

where $g^{(0)}$ is the step input from a forebrain decision center; ε is a slow integration rate; Cs is the value at which the GO cells are saturated. In the model, wheels' velocities commands are represented by $\dot{\theta}_{id}$, and are given by ERG in the learning phase.

Learning phase The learning is obtained by decreasing weights in proportion to the product of the presynaptic and postsynaptic activities. The network can also be redesigned to have only positive activations and weights by using the appropriate push-pull mechanism as in Gaudiano and Grossberg [17]. The training is done by generating random movements, and by using the resulting angular velocities and observed spatial velocities of the mobile robot as training vectors to the direction mapping network. The weights of network are obtained as

$$\frac{d}{dt}z_{jik} = -\eta e V_{ik} S_j. \tag{8}$$

During learning in a particular context, k^{th}, with e=1, we can note that: $S_j \to xd_j$; $R_i \to \dot{\theta}_i$; $\dot{\theta}_i = \dot{\theta}_{id} \to \dot{\theta}_{i_{ERG}}$; and $V_{ik} \to \left(\sum_j z_{jik} S_j - R_i\right)$.

Therefore, the learning rule can be obtained by using the gradient-descent algorithm and the equation (8) is modified in discrete form as:

$$z_{jik}(t+1) = z_{jik}(t) + \eta\left(\dot{\theta}_i - \sum_j z_{jik} xd_j\right) xd_j \tag{9}$$

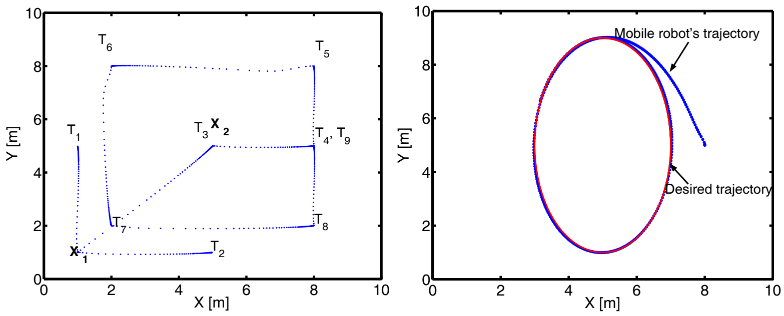
where η is learning rate and is a positive constant gain.

3 Experimental results

The proposed control algorithm is implemented on a mobile robot from the UPCT named "CHAMAN". The platform has two driving wheels (in the rear) mounted on the same axis and two passive supporting wheels (in front) of free orientation. The two driving wheels are independently driven by two DC-motors to achieve the motion and orientation. The wheels have a radius $r = 18$ cm and are mounted on an axle of length $2R = 22$ cm. The aluminum chassis of the robot measures 102.25×68×44 cm ($L \times W \times h$) and contains, transmission elements, 12-VDC battery, two CCD cameras, and 12 ultrasound sensors. Each motor is equipped with incremental encoder counting 600 pulses/turn and a gear which reduces the speed to 1.25 m/s.

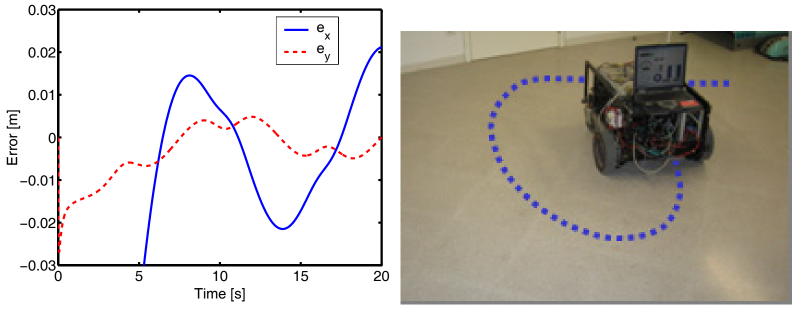
High-level control algorithms (SODMN) are written in VC++ and run with a sampling time of 10 ms on a remote server (a Pentium IV processor). The PC communicates through a serial port with the microcontroller on the robot. Wheel PWM duty cycle commands are sent to the robot and encoder measures are received for odometric computation. The lower level control layer is in charge of the execution of the high-level velocity commands. It consists of a Texas Instruments TMS320C6701 Digital Signal Processor (DSP). The microcontroller performs three basis tasks: 1) to communicate with the higher-level controller through RS 232; 2) reading encoder counts interrupt driven; and 3) generation of PWM duty cycle.

Figures 3, 4 and 5 show approach behaviors and the tracking of a trajectory by the mobile robot with respect to the reference trajectory. From these figures, it is clear that the tracking of the reference trajectory is very accurate. In the Figure 3, reaches to targets (such as $\times_1$ to T_1, $\times_2$ to T_2 and $\times_1$ to $\times_2$) and a sequence of movement (starting from $\times_2$ to T_4, T_4 to T_5, T_5-T_6, T_6 -T_7, T_7-T_8, T_8-T_4, T_4-$\times_2$) were carried out.



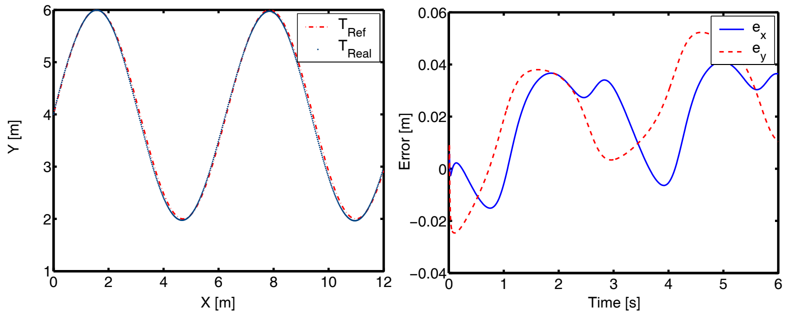
(a) (b)

Fig. 3. Adaptive control by the SODMN. a) Approach behaviors. The symbol X indicates the start of the mobile robot and T_i indicates the desired reach. b) Tracking control of a desired trajectory.



(a) (b)

Fig. 4. Tracking control of the SODMN. a) The trajectory error of the Figure 3b. b) Real-time tracking performance.



(a) (b)

Fig. 5. Tracking control of a desired trajectory. (a) Tracking trajectory of a sine. (b) Estimated tracking error.

4 Discussion

In present model, appropriate operations are learned in an unsupervised fashion through repeated action-perception cycles by recoding proprioceptive information related to the mobile-robot. The resulting solution has two interesting properties: (a) the required transformation is executed accurately over a large part of the reaching space, although few velocities are actually learned; and (b) properties of single neurons and populations closely resemble those of neurons and populations in parietal and cortical regions [18]. The activity of the population of motor cortical cells which encode movement direction appears to represent the instantaneous velocity of movement [19]. In addition, the preferred directions of individual cells shifts with the movement origin, indicating that the directional coding of motor cortex may be influenced by velocity configuration (in the model is the context field) [20], as is necessary for a Jacobian-based mapping. Correspondence between layers of the network and brain regions can be made tentatively base on anatomical and physiological arguments [18, 19]. The representation of DVs could be in posterior parietal cortex (PPC) [21]. Neurons in PPC exhibit activity patterns correlated with the spatial direction of movement [22]. A candidate region for participating in the direction mapping computation is the cerebellum [23]. Also, note that there are certain similarities between the nature of the context field cells in the mobile-robot movement model and the Purkinge cells of the adaptive timing model. Both types of cells are tonically active and allow a response by "pausing" this tonic activity. Thus, the possibility that a context field type of function is performed by cerebellar cortex. The proposed direction mapping model also posits a learning site separate from the context field computation, which might be a cerebellar function. In the model, motor commands were emitted by a layer containing R_i neurons, which contribute to the movement by a displacement along a direction in velocity space. The individual influence of a command neuron is proportional to its discharge level.

4.1 Conclusions

In this paper, a biologically inspired neural network for the spatial reaching tracking has been developed. This neural network is implemented as a kinematic adaptive neuro-controller. The SODMN uses a context field for learning the direction mapping between spatial and angular velocity coordinates. The transformations are learned during an unsupervised training phase, during which the mobile robot moves as result of randomly selected angular velocities of wheels. The performance of this neural network has been successfully demonstrated in experimental results with the trajectory tracking and reaching of a mobile robot. The efficacy of the proposed neural network for reaching and tracking behaviors was tested experimentally by a differentially driven mobile robot.

References

1. Fierro, R., Lewis, F.L.: Control of a nonholonomic mobile robot using neural networks. IEEE Trans. Neural Netw. **9** (1998) 589–600
2. Kanayama, Y., Kimura, Y., Miyazaki, F., Noquchi, T.: A stable tracking control method for an autonomous mobile robot. In: Proc. IEEE Int. Conf. Robotics and Automation. Volume 1., Cincinnati, OH (1990) 384–389
3. Ping, Z., Nijmeijer, H.: Tracking control of mobile robots: A case study in backstepping. Automatica **33** (1997) 1393–1399
4. Oriolo, G., Luca, A.D., Vendittelli, M.: WMR control via dynamic feedback linearization: Design, implementation and experimental validation. IEEE Trans. Control. Syst. Technol. **10** (2002) 835–852
5. Das, T., Kar, I.N.: Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. IEEE Transactions on Control SystemsTechnology **14** (2006) 501–510
6. De-Luca, A., Oriolo, G., Samson, C.: Feedback control of a nonholonomic car-like robot. In Laumond, J.P., ed.: Robot Motion Planning and Control. Springer, Berlin Heidelberg, New York (1998) 171–253
7. Novel, B.D., Bastin, G., Campion, G.: Control of nonholonomic wheeled mobile robots by state feedback linearization. Int. J. Rob. Res. **14** (1995) 543–559

8. Fukao, T., Nakagawa, H., Adachi, N.: Adaptive tracking control of a nonholonomic mobile robot. IEEE Trans. Robot. Autom. **16** (2000) 609–615
9. Corradini, M., Orlando, G.: Robust tracking control of mobile robots in the presence of uncertainties in the dynamical model. J. Robot. Syst. **18** (2001) 318–323
10. Dixon, W., Dawson, M., Zergeroglu, E.: Tracking and regulation control of a mobile robot system with kinematic disturbance: A variable structure like approach. ASME Trans. J. Dyn. Syst. Meas. Control **122** (2000) 616–623
11. Corradini, M., Leo, T., Orlando, G.: Experimental testing of a discrete-time sliding mode controller for trajectory tracking of a wheeled mobile robot in the presence of skidding effects. J. Robot. Syst. **19** (2002) 177–188
12. Chen, F., Liu, C.: Adaptively controlling nonlinear continuous-time systems using multilayer neural networks. IEEE Trans. Automat. Contr. **39** (1994) 1306–1310
13. Corradini, M., Ippoliti, G., Longhi, S.: Neural networks based control of mobile robots: Development and experimental validation. J. Robot. Syst. **20** (2003) 587–600
14. Lin, S., Goldenberg, A.: Neural-network control of mobile manipulators. IEEE Trans. Neural Netw. **12** (2001) 1121–1133
15. Grossberg, S., Levine, D.: Neural dynamics of attentionally moduled Pavlovian conditioning: Blocking, interstimulus interval, and secondary reinforcement. Applied Optics **26** (1987) 5015–5030
16. Chang, C., Gaudiano, P.: Application of biological learning theories to mobile robot avoidance and approach behaviors. J. Complex Systems **1** (1998) 79–114
17. Gaudiano, P., Grossberg, S.: Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. Neural Networks **4** (1991) 147–183
18. Baraduc, P., Guigon, E., Burnod, Y.: Recording arm position to learn visuomotor transformations. Cerebral Cortex **11** (2001) 906–917
19. Georgopoulos, A.P.: Neural coding of the direction of reaching and a comparison with saccadic eye movements. Cold Spring Harbor Symposia in Quantitative Biology **55** (1990) 849–859
20. Caminiti, R., Johnson, P., Urbano, A.: Making arm movements within different parts of space: Dynamic aspects in the primate motor cortex. Journal of Neuroscience **10** (1990) 2039–2058
21. Rondot, P., De-Recondo, J., Dumas, J.: Visuomotor ataxia. Brain **100** (1976) 355–376
22. Lacquaniti, F., Guigon, E., Bianchi, L., Ferraina, S., Caminiti, R.: Representing spatial information for limb movement: Role of area 5 in the monkey. Cerebral Cortex **5** (1995) 391–409
23. Fiala, J.C.: Neural Network Models of Motor Timing and Coordination. PhD thesis, Boston University (1996)