Proceedings of the Int. Conf. on Parallel
Processing in Neural Systems and Computers
(ICNC), Düsseldorf, (Elsevier, 1990) (in press)

Learning of Visuomotor-Coordination of a Robot Arm with Redundant Degrees of Freedom

Thomas Martinetz, Helge Ritter and Klaus Schulten
Department of Physics and Beckman Institute
for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 North Mathews Ave., Urbana, IL 61801, USA

Abstract: An improved version of an earlier introduced algorithm for learning of visuomotor-coordination has been successfully applied to a simulated robot arm system with five degrees of freedom, two of which are redundant. The learning algorithm was not affected by this redundancy, and with the improved version the robot arm system is able to reduce its positioning error to about 5% after only 200 learning steps and finally to about 0.3% of its linear dimensions after 6000 learning steps. Learning proceeds without the need of an external teacher by a sequence of trial movements using input signals from a pair of cameras. The topology conserving map used for the representation of the input-output transformation leads to an automatic resolution of the redundancy of the arm as it tries to minimize the variation of the joint angles over the work space. The improved learning algorithm incorporates some immediate feedback, so that now the robot arm not only is able to adapt to slowly occuring miscalibrations, but also can compensate for sudden changes in its geometry, such as picking up e.g. a tool.

1. Introduction

Compared to present-day robots, biological motor control systems excel with an enormous degree of flexibility. This flexibility is to a large extent due to two major factors. The first is the use highly developed sensors, most notably vision, capable to monitor a multitude of different aspects of ongoing movements simultaneously. The second factor is the presence of a high amount of redundancy in the muskulo-skeletal system, offering for most movement goals a wide range of alternative realizations. Both strengths are enhanced further by the high degree of adaptability that neural systems exhibit for both, sensory perception and selection and control of movements.

There have been many attempts to capture these remarkable features in biologically inspired neural network models, see e.g. [5-11]. Our own work has previously considered the problem of adaptive visuomotor coordination for a robot, using a non-redundant (simulated) three-link robot arm ([9,10,13]). In this contribution, we want to focus on the additional issue of controlling a redundant arm and to resolve its redundancy by the use of a topology conserving map imposing a "smoothness constraint" on the arm configurations. In addition, we present an improved version of the learning rule used in [9,10,13], which incorporates a more accurate visual feedback.

For a redundant robot arm, specification of a target location does not yet uniquely fix the configuration of the arm, but instead allows a wide range of possibilities still compatible with the given end effector location. The general problem of redundancy resolution is then to choose one of the infinity of possible configurations such that some overall cost-function or some performance measure is extremalized. Frequently, however, the precise form of the performance measure is of secondary interest, and serves merely as a tool to impose a smoothness constraint upon the system, i.e. to ensure that the selected configurations for two nearby end effector locations differ from each other as little as possible.

However, this is precisely the kind of task solved "naturally" by topology conserving maps. If such map is used to represent the mapping from task space to joint coordinates, neighboring task coordinates will activate neighboring nodes in the network for output. Due to the natural tendency of the map to smooth out any unnecessary variations in the outputs of neighboring nodes, any redundant degrees of freedom will be used by the map to make the variation between joint configurations for neighboring target points as small as possible. This method differs from previous approaches, like the pseudo-inverse technique, in that the optimization of smoothness comes "for free" by the natural learning dynamics of the network and does not require any auxiliary computations beyond the usual adaptation steps.

In the following Section 2, we will describe the model and the adaptation equations, and in Section 3 we will present the results of a simulation for a 5-degree-of-freedom manipulator. As we will not consider manipulator orientation, this amounts to the presence of two redundant degrees of freedom.

2. The Model

In Fig.1 we see the robot arm system consisting of a robot arm of five degrees of freedom and a pair of cameras providing the spatial information about the location of the object the robot arm shall reach for. For each trial movement, the target location is chosen randomly within the work space. Each target within the three-dimensional work space corresponds to a pair of two-dimensional vectors, namely the locations of the images of the object on the two camera "retinas". As described in more detail in [9,10,13], it is possible to combine both locations to a four-dimensional vector $\mathbf{u}$ which then carries the visual information necessary for the network to extract the spatial position of the object. These four-dimensional vectors form the input signals for a vector-quantization network of Kohonen-type [2-4]. This network is able to adaptively discretize the relevant three-dimensional submanifold within the four-dimensional input space. The highly nonlinear transformation $\vec{\theta}(\mathbf{u})$ from camera input to joint angles $\vec{\theta}$ is then linearized in the vicinity of each discretization point. Below we describe how the local linear transformations are learnt from trial movements and simultaneously with the adaptive discretization.

To each discretization point $\mathbf{s}$ corresponds a formal neuron of the Kohonen-network; the neuron associates a location $\mathbf{w_s}$ within the four-dimensional input space with a five-dimensional vector $\vec{\theta}_\mathbf{s}$ of joint angles and a 5×4 matrix $\mathbf{A_s}$. Angles $\vec{\theta}_\mathbf{s}$ and matrix $\mathbf{A_s}$ are used to position the end effector at the given target, if the corresponding input was closer to the discretization point $\mathbf{w_s}$ than to any other $\mathbf{w_r}$, $\mathbf{r} \neq \mathbf{s}$. The discretisation occurs in a topologically ordered manner, which means that neighboring neurons of the Kohonen-network are associated with neighboring subsets of the input space.

The positioning of the end effector consists of two phases, namely a gross-positioning and a subsequent, usually small correction. For the gross-positioning the robot arm system uses the five-dimensional vector $\vec{\theta}_\mathbf{s}$ each element of which determines one joint angle. After this gross movement the position of the end effector in both camera "retinas" will be denoted by a four-dimensional vector $\mathbf{v}_i$, which is close to the four-dimensional retinal target location $\mathbf{u}$. Departing from the ansatz in [9,10,13], we now use the linear correction term

$$\Delta\vec{\theta} = \mathbf{A_s}(\mathbf{u} - \mathbf{v}_i) \tag{1}$$

for the final corrective joint movement. Here $\mathbf{A_s}$ is the Jacobian of the transformation $\vec{\theta}(\mathbf{u})$ at the discretisation point $\mathbf{s}$ and $\Delta\vec{\theta}$ is usually sufficient to correct the error of the gross movement very precisely. The final position of the end effector is seen by the cameras at $\mathbf{v}_f$.

For the learning step of $\mathbf{A_s}$ we use a linear error correction rule of Widrow-Hoff-type [14] which minimizes the quadratic error

$$E = \frac{1}{2}(\Delta\vec{\theta} - \mathbf{A_s}\Delta\mathbf{v})^2 \tag{2}$$

with $\Delta\mathbf{v} = \mathbf{v}_f - \mathbf{v}_i$ by using steepest descent. Together with (1) and by choosing the optimal step size we obtain as an improved estimate $\mathbf{A}^*$ for $\mathbf{A_s}$

$$\mathbf{A}^* = \mathbf{A_s} + ||\Delta\mathbf{v}||^{-2} \cdot \mathbf{A_s}(\mathbf{u} - \mathbf{v}_f)\Delta\mathbf{v}^T. \tag{3}$$

Because of the topologically correct assignment of the neurons to the inputs neighboring neurons of $\mathbf{s}$ have to learn similar output values $\vec{\theta}_\mathbf{r}$ and $\mathbf{A_r}$. Therefore, $\mathbf{A}^*$ is used to improve the Jacobians $\mathbf{A_r}$ for all neurons $\mathbf{r}$ in a whole neighborhood of $\mathbf{s}$. This neighborhood is defined by a function $h_{\mathbf{rs}}$ which is unity at $\mathbf{r} = \mathbf{s}$ and decays to zero as $\mathbf{r}$ moves away from $\mathbf{s}$. The resulting adjustments are

$$\mathbf{A}_\mathbf{r}^{new} = \mathbf{A}_\mathbf{r}^{old} + \epsilon h_{\mathbf{rs}}(\mathbf{A}^* - \mathbf{A}_\mathbf{r}^{old}), \tag{4}$$

and provide an enormous increase in speed and stability of the convergence of the learning algorithm ([13]).

With the improved Jacobian $\mathbf{A_s}$ we are now also able to improve $\vec{\theta}_\mathbf{s}$ because $\vec{\theta}_{correct} - \vec{\theta}_\mathbf{s} = \mathbf{A_s}(\mathbf{w_s} - \mathbf{v}_i)$ so that we obtain as an improved estimate θ^* for $\vec{\theta}_\mathbf{s}$

$$\vec{\theta}^* = \vec{\theta}_\mathbf{s} + \mathbf{A}_\mathbf{s}^{new}(\mathbf{w_s} - \mathbf{u}). \tag{5}$$

Like the Jacobians, the joint angles $\vec{\theta}_\mathbf{r}$ of neighboring units share their learning steps through

$$\vec{\theta}_\mathbf{r}^{new} = \vec{\theta}_\mathbf{r}^{old} + \epsilon h_{\mathbf{rs}}(\vec{\theta}^* - \vec{\theta}_\mathbf{r}^{old}). \tag{6}$$

3. Simulation Results

In the following we describe the results of a simulation using the same parameter values for ϵ and $h_{\mathbf{rs}}$ and the same network topology (i.e. a three-dimensional $7 \times 12 \times 4$ lattice of 336 neurons) as in ([9,10]). However, in contrast to this work we now employed the improved adaptation rules (1), (3), (5) and the robot arm with five degrees of freedom shown in Fig.1. The initial output values for each neuron were chosen at random, but subject to the constraint that the resulting shape of the robot arm was convex in all cases.

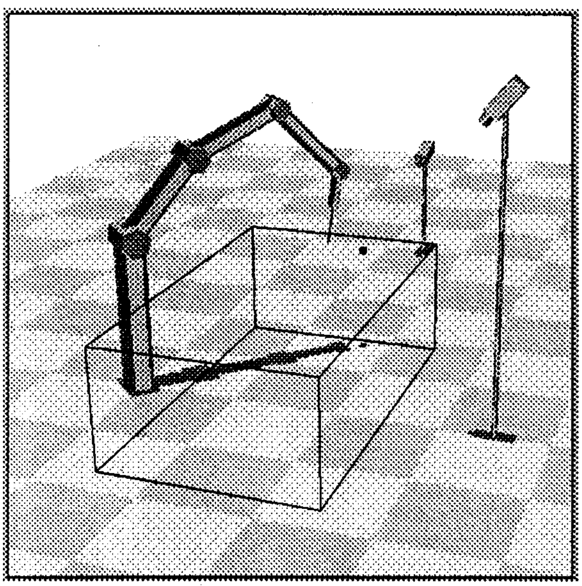

Fig.1: The robot arm and the two cameras.

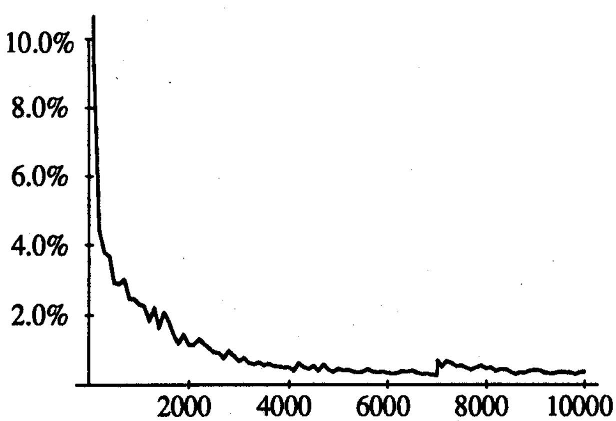

Fig.2: Positioning error versus number of trials.

Fig.2 shows the development of the remaining positioning error after each trial. The error decays very rapidly to below one percent of its initial value. Although the number of links is now larger (5 vs. 3), the decay of the error is faster than in [9,10,13], which clearly shows the superiority of the new learning rule over that employed in [9,10,13]. This stems mainly from basing the corrective fine movement on the actual distance $\mathbf{u} - \mathbf{v}_i$ remaining after the preceeding gross movement, instead of using the distance $\mathbf{u} - \mathbf{w_s}$ between the target and the closest discretization point. This provides the system with an accurate feedback about its residual error, and enables the fine movement to compensate even for sudden and unexpected changes in the geometry of the arm. To demonstrate this, we suddenly increased at trial 7000 the length of the last arm segment by about 5% of its linear dimension, a situation which could result from picking up e.g. a wrench. As we see from Fig.2, even this perturbation causes only a small increase in error, which rapidly returns to its baseline value.

Due to the redundancy of the arm, each target location is compatible with a continuous subset of different arm postures. Initially, as a result of the random initialization of the network, mutually close target locations may give rise to very different arm postures selected for reaching. As learning proceeds, those posture differences not contributing to end effector location get more and more wiped out and postures which only vary smoothly over the target space emerge. An example is provided by Fig.3, which shows successive postures traversed if the end effector follows a path along a diagonal of the workspace.

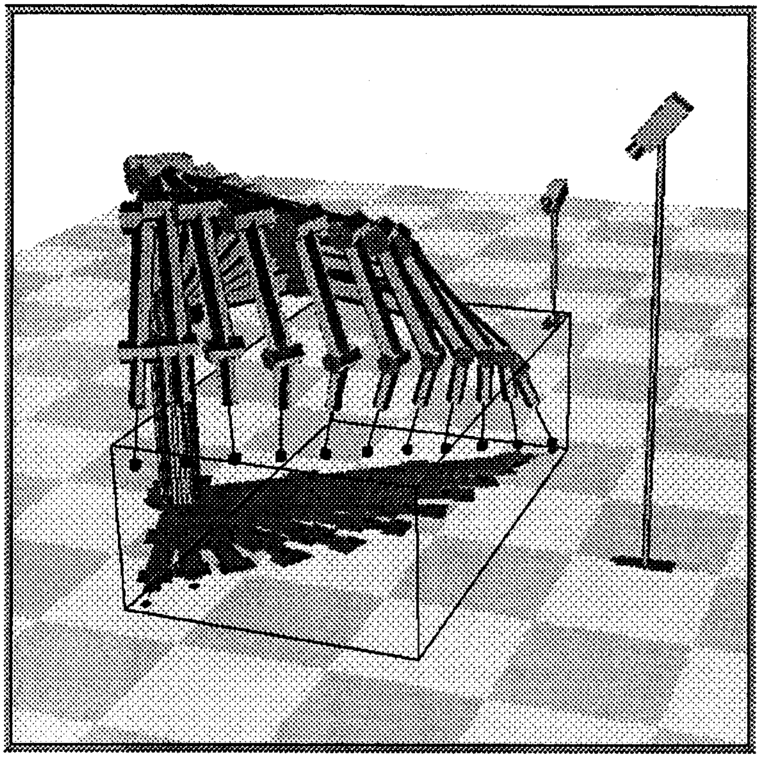

Fig.3: Stroboscopic rendering of successive robot arm postures on traversing a trajectory in the workspace after learning. Due to the redundancy of the arm, each trajectory point can be reached by an infinitude of different postures. Although the system has never obtained any explicit information how to resolve this redundancy, the topology conserving property of the map used for the learning algorithm has forced a solution leading to a particularly smooth movement.

Acknowledgement:

This work has been supported by the German Ministry of Science (BMFT) under contract ITR-8800-G9 and by the Volkswagen Foundation.

References:

[1] Eckmiller R., Beckmann J., Werntges H., Lades M. (1989) Neural Kinematics Net for a Redundant Robot ARM. IJCNN-89, Vol.II:333-340.

[2] Kohonen T. (1982a) Self-organized Formation of Topologically Correct Feature Maps. Biological Cybernetics 43:59-69

[3] Kohonen T. (1982b) Analysis of a Simple Self-organizing Process. Biological Cybernetics 44:135-140

[4] Kohonen T. (1982c) Clustering, Taxonomy and Topological Maps of Patterns. Proceedings of the 6th Int. Conf. on Pattern Recognition, Munich pp 114-128

[5] Kuperstein M. (1987) Adaptive Visual-Motor Coordination in Mulitjoint Robots using Parallel Architecture. Proc. IEEE Int. Conf. Automat. Robotics 1595-1602, Raleigh NC

[6] Kuperstein M. (1988) Neural Model of Adaptive Hand-Eye Coordination for Single Postures. Science 239:1308-1311

[7] Kuperstein M., Rubinstein J. (1989) Implementation of an Adaptive Neural Controller for Sensory-Motor Coordination. IEEE Control Systems Magazine, Vol.9, No.3, pp 25-30

[8] Lane, Handelman, Gelfand (1988) A Neural Network Computational map approach to reflexive motor control. Proceedings of the 1988 IEEE conference on intelligent control.

[9] Martinetz T., Ritter H., Schulten K. (1989) 3D-Neural Net for Learning Visuomotor-Coordination of a Robot arm. IJCNN-89, Conference Proceedings, Washington 1989, Vol.II:351-356.

[10] Martinetz T., Ritter H., Schulten K. (1989) Three-Dimensional Neural Net for Learning Visuomotor-Coordination of a Robot arm. IEEE Transactions on Neural Networks, submitted.

[11] Mel B.W. (1987) A Robot that Learns by Doing. AIP Proceedings 1987, Neural Information Processing System Conf., Denver, CO, 1987

[12] Miller W. T. (1989) Real-Time Application of Neural Networks for Sensor-Based Control of Robots with Vision. IEEE Transactions on Systems, Man, and Cybernetics, Vol.19, No.4, pp 825-831.

[13] Ritter H., Martinetz T., Schulten K. (1988) Topology-Conserving Maps for Learning Visuomotor-Coordination. Neural Networks 2, pp. 159-168.

[14] Widrow B., Hoff M.E. (1960) Adaptive switching circuits, WESCON Convention Record, part IV pp. 96-104