Biol. Cybern. 62, 549–555 (1990)

Biological Cybernetics

A Network Model for the Control of the Movement of a Redundant Manipulator

M. Brüwer and H. Cruse

Fachbereich Biologie der Universität, Postfach 8640, D-4800 Bielefeld 1, Federal Republic of Germany

Abstract. In an earlier investigation (Cruse and Brüwer 1987) an algorithmic model was proposed which describes targeting movements of a human arm when restricted to a horizontal plane. As three joints at shoulder, elbow and wrist are allowed to move, the system is redundant. Two models are discussed here which replace this algorithmic model by a network model. Both networks solve the static problem, i.e. they provide the joint angles which the arm has to adopt in order to reach a given point in the workspace. In the first model the position of this point is given in the form of $x-y$ coordinates, the second model obtains this information by means of a retina-like input layer. The second model is expanded by a simple procedure to describe movements from a start to an end point. The results qualitatively correspond to those obtained from human subjects. The advantages of the network models in comparison to the algorithmic model are discussed.

Introduction

The control of the movement of a multijointed manipulator in general includes strongly non-linear operations (Benati et al. 1980). The control algorithm has to cope with additional problems when the manipulator has more joints than necessary for a given task, i.e. when it has extra degrees of freedom. The human arm provides such a redundant manipulator.

To obtain the simplest version of a redundant arm, in earlier experimental investigations (Cruse 1986, Cruse and Brüwer 1987) the number of degrees of freedom of the human arm was reduced to three, namely shoulder, elbow and wrist joint, while allowing the arm to move in a two-dimensional, horizontal plane. The axes of rotation of the three joints were perpendicular to the plane. The position of the endeffector, the tip of a pointer attached to the palm, is determined by two cartesian coordinates in the horizontal plane. Therefore, two joints were sufficient to move the endeffector in the workspace. The existence of the third joint produced an additional degree of freedom and therefore made the system redundant (Fig. 1a). Thus a given point in the two dimensional workspace can be reached by a number of different combinations of joint angles of the manipulator.

The question arises of how the control system selects one of this infinite number of possible positions when trying to reach a given point. To solve this static problem, the following hypothesis was proposed (Cruse 1986). To each joint a cost function is attached which defines a cost value for each joint angle. The cost functions show a minimum at about the middle of the angle range of the joint and the cost values increase to either of the extreme angles. The total cost of a manipulator position is described as the sum of the actual cost values of all joints. When reaching to a given point in the workspace, according to this hypothesis, that manipulator position is selected out of the geometrically possible positions which shows the minimum total cost value. In this way the number of degrees of freedom of the system is reduced and thus the redundancy problem can be solved.

To solve the kinema problem, i.e. the question of how the joints are controlled during the movement from a start to an end or target point, a hypothesis was formulated which represents a compromise between the minimum cost strategy described above and two other strategies (Cruse and Brüwer 1987; Cruse 1989) which will be explained later in more detail. This hypothesis was formulated as an algorithmic model which at least qualitatively behaves like a human subject. This algorithmic model has however some disadvantages (see Discussion) which might be overcome when replacing the algorithmic model by an neuronal network model (Cruse 1989). As a first step towards this aim two network models will be presented in this paper which solve the static problem. For the second model in addition a way will be shown as to how the kinematic experiments can be simulated.

Two Network Models Solving the Static Problem

The first model consists of three layers of neurons (Fig. 1b), the input layer, an intermediate "hidden" layer and the output layer. The network models were simulated on a HP 9000/320 using the software of McClelland and Rumelhart (1988). The input layer consists of two neurons the excitation of which corresponds to the value of the cartesian workspace coordinates x and y (see Fig. 1a). The intermediate layer consists of 20 hidden units. The output layer contains three units which give the angle values of the three joints. Only feedforward connections and only connections between neurons of directly neighbouring layers are allowed. The excitation of each neuron was between 0 and 1 with a resting value of 0.5. To train the network we followed the error back propagation rule with a learning rate of 0.05 and a momentum of 0.9 (McClelland and Rumelhart 1983). For one learning cycle (epoch) 32 input patterns were used which correspond to the $x-y$ coordinates of the 32 points in the workspace as shown in Fig. 1a. The corresponding training patterns for the output, i.e. the angle tripels, were taken from the experimental results of one subject which made targeting movements to each of the 32 points (the results were from the same subject as were the data shown in Figs. 2 to 8 in Cruse and Brüwer, 1987).

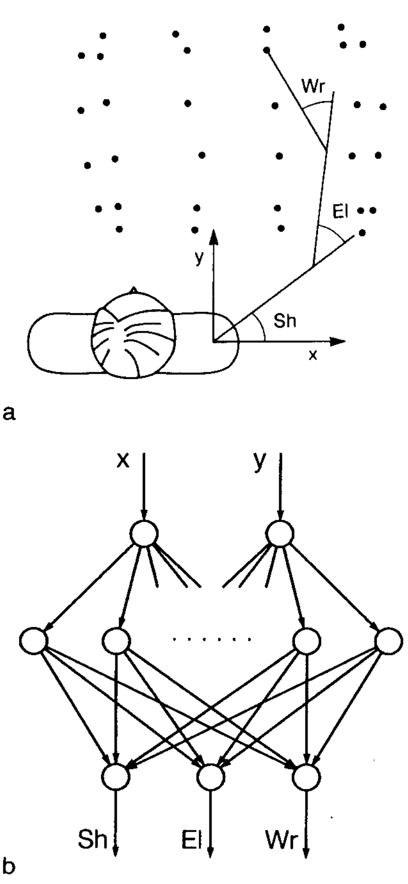

Fig. 1. a Top view of the experimental arrangement. The arm is moved in a horizontal plane with the coordinates x and y. The origin is at the shoulder joint. The three joint angles at shoulder (Sh), elbow (El) and wrist (Wr) determine the position of the tip of the arm. The dots show the position of the 32 target points. **b** A simple network consisting of two input units which obtain the coordinate values x and y of the target point, 20 hidden units and three output units which give the angle values of the three joints at shoulder, elbow and wrist

Figure 2a shows the progress of learning the network made during the presentation of 100000 cycles. The mean deviation between angles values calculated by the network and those assumed by the subject was then 1.4 degrees per joint. In Fig. 2b the $x-y$ coordinates of these points are compared with the positions the network actually pointed to. As the network calculates angles and not $x-y$ coordinates, the latter were determined from the angle values by a separate calculation using the corresponding trigonometric formulas. The mean Euclidean distance was 1.1 (± 0.8) cm for the 32 trained points (mean $\pm$S.D.). Using less than 20 hidden units led to slower learning and to larger mistakes.

In order to test the capacity of the network to respond to untrained target points, the response of the network to 117 new target points was calculated. These untrained points lie on a 9 × 13 grid in the same area as the 32 trained points. As was shown in Fig. 2b for the 32 target points, in Fig. 2c the deviation for the 117 untrained points is shown. The mean Euclidean distance is 0.9 (± 0.5) cm for the untrained points.

In this first model the input values are specified in the form of the $x-y$ coordinates of the target points. However, the position of the target points might also be measured by a "retina-like" sensory system, i.e. a sensory system which locates a point in the workspace not by two coordinates values – for example the cartesian $x-y$ coordinates – but by means of its position within a topologically ordered map. To simulate such a system the following simple network was investigated. The network consists of only two layers, which means that this second model does not contain a hidden layer. As in the first model the output layer consists of three neurons the excitation of which represents the value of the three joint angles. The input layer consists of a two-dimensional array of 10 × 10 neurons which can be considered as a form of a primitive "retina". A target point in the workspace is

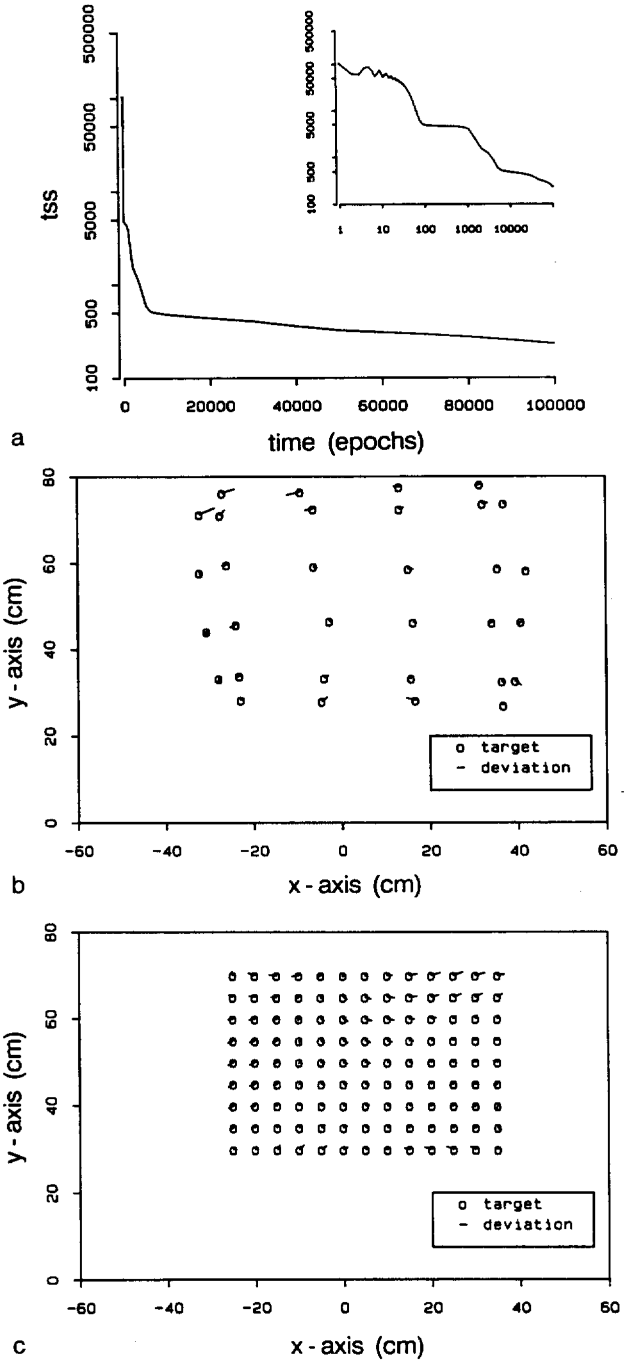

Fig. 2. **a** Learning curve for the network shown in Fig. 1b. Abscissa is the number of learning cycles (epochs). In one epoch each of the 32 target points was presented once. Ordinate is the total sum of squared errors (tss) in one epoch. This error is the deviation of the angle values produced by the network from the angle values obtained during experiments with the human subject. The inset figure shows the same learning curve with a logarithmic abscissa. **b** The mean deviation from each of the 32 target points after 100000 epochs. The circles show the position of the target points in workspace coordinates. The length and direction of the attached bar shows the mean deviation. **c** The deviation for 117 untrained target points shown in the same way

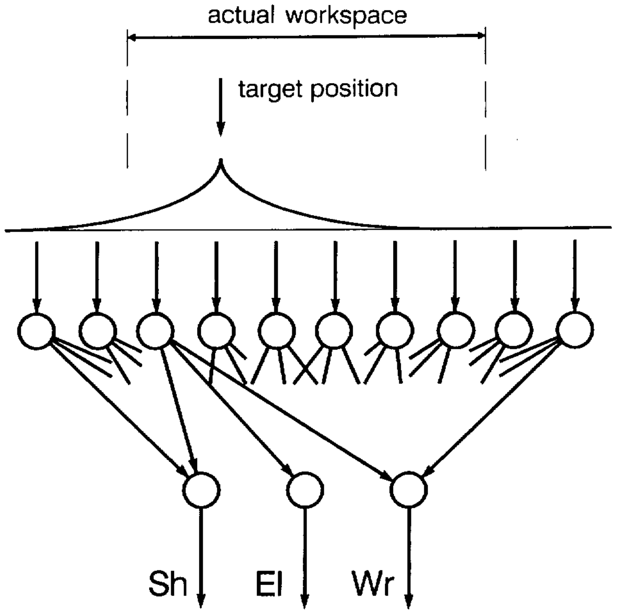

Fig. 3. A network using a two dimensional, "retina-like" input layer. The input consists of a 10 × 10 units input layer. Only a cross section along the x-axis of the workspace is shown. Around the target position an excitation function with exponential decay is calculated. These values serve as input to the 10 × 10 layer. All 10 × 10 units are connected to the three output units which give the angle values of the three joints at shoulder (Sh), elbow (El), and wrist (Wr)

projected onto this layer in a kind of blurred projection. The target point produces an excitation which is highest at the position of the target point and decays exponentially with the Euclidean distance in all directions (Fig. 3). Thus, the input layer obtains a volcano-like excitation function defined on the $x-y$ coordinates of the workspace decreasing with a decay constant of 10 cm. The individual unit of the input layer measures only that value which appears at its coordinate position. In order to avoid cutting parts of this excitation function when the target point is near the margin of the workspace, the application of these "receptive fields" required the introduction of units in the input layer which have no direct corresponding points in the workspace but lie outside the actual workspace. They only serve as acceptors for the outside parts of the excitation functions. Two rows of neurons were used for this purpose. Thus, only the central 6 × 6 neurons correspond to that range of the workspace within which the 32 target points lie. This means that the projection of the workspace onto the input layer has a rather coarse spatial resolution as the distance between two neurons projected onto the workspace was about 12 cm in the direction of the x-axis and 8 cm in the direction of the y-axis. However, because of the spread excitation function the receptive

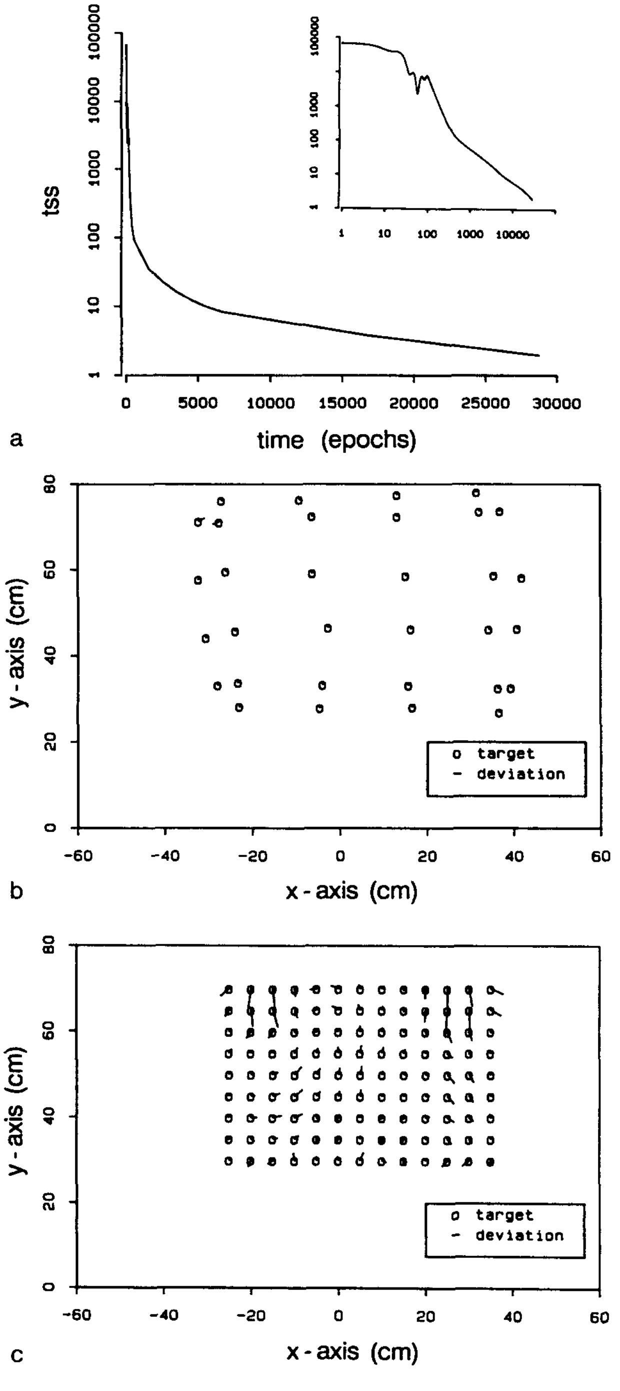

Fig. 4. a Learning curve for the network shown in Fig. 3. Abscissa is the number of learning cycles (epochs). In one epoch each of the 32 target points was presented once. Ordinate is the total sum of squared errors (tss) in one learning cycle. This error is the deviation of the angle values produced by the network from the angle values obtained during experiments with the human subject. The inset figure shows the same learning curve with a logarithmic abscissa. **b** The mean deviation from each of the 32 target points after 29000 epochs. The circles show the position of the target points in workspace coordinates. The length and direction of the attached bar shows the mean deviation. **c** The deviation for 117 untrained target points shown in the same way

fields of the neighbouring units overlap and each input neuron obtained stimuli from a range of an area of about 23 x 19 cm in the workspace.

The calculation of this excitation function could have been done by using an parallel network. For this purpose an additional preprocessing layer with a fine grid had to be used with feedforward connections to the input layer and fixed weighting factors (Reichardt and Mac Ginitie 1962). However, in order to use as few neurons as possible to save computing time, in this simulation the excitation values were calculated algorithmically and then given as input values onto the 10×10 input layer. Thus by the training procedure only the weighting factors from each of the 10×10 neurons to the 3 output neurons were changed. This was done as was described for the first model. Figure 4a and b shows the learning process and its result in the same way as was done in Fig. 2a, b for the earlier model (learning rate 0.01, momentum 0.98). Correspondingly, Fig. 4c presents the deviation for a set of 117 unlearned target points in $x-y$ coordinates. The mean Euclidean distance was 0.2 (± 0.4) cm for the 32 trained points (Fig. 4b) and 1.4 (± 1.1) cm for the untrained points (Fig. 4c). The higher deviations in the latter case were mainly due to the deviations in the upper right and left corners as shown in Fig. 4c. The mean deviation given in angular degrees was 0.1 deg per joint.

Kinematics

The task of the models described up to now was to provide the angle values for shoulder, elbow and wrist joint so that the arm points to a position arbitrarily given within the workspace, i.e. to solve the static problem. To model the movement from a given start to a given end or target point, the system needs at least the information on the position of these two points. Furthermore the question is open along which path the tip of the arm moves between start and end point. Most earlier authors proposed that in general the tip (endeffector) followed a straight line between both points. This was however shown not to be true in all cases (Atkeson and Hollerbach 1985; Cruse and Brüwer 1987; Flash 1987). If we for a moment leave aside the problem of how the mechanism producing the form of the path is constructed, for a given path a very simple solution for this kinematik problem would be the following. Using the second model, this unknown mechanism could be thought to move an imaginary target point along the path from the start to the end point. For each position of the imaginary target point both models described above could then provide the corresponding angle values.

However, it is not necessary to introduce such a mechanism which moves an imaginary point. Different

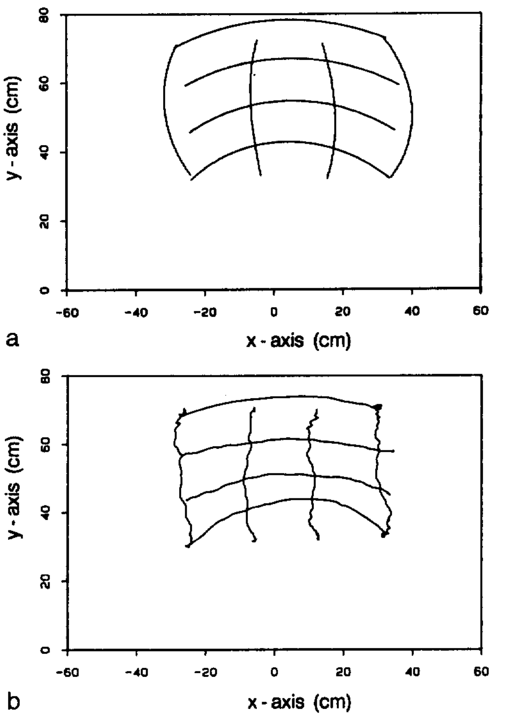

Fig. 5a and b. The path of the tip of the arm in workspace coordinates of eight movements. In four cases the line connecting start and end point are parallel to the x-axis, in the other four cases parallel to the y-axis. Direction of movement is from left to right or from top to bottom. **a** Movement calculated by the network shown in Fig. 3. **b** Movement of the human subject. The results are from that subject, the static data of which were used to train both networks

other solutions are feasible to produce a movement between start and end point without explicitely determining the path. A simple approach using the second model is as follows. To bring the arm to the starting position, the trained network is presented with the coordinates of the start point or, to be more exact, with the corresponding excitation function as was described above for the static situation. Then, to introduce the position of the end point, the excitation function belonging to the end point is also additively given to the input layer. However, at the very beginning the whole excitation function of the end point was multiplied with a scaling factor Fe = 0. This means that the occurrence of the end point does not affect the behaviour of the network. To produce a movement of the arm, this scaling factor Fe was increased while a corresponding scaling factor Fs attached to the excitation function of the start point was decreased following the condition Fs = 1 − Fe. The parameter time is introduced in such a way that the scaling factor Fe is increased in steps of 0.01 and altogether 100 steps were used for the movement from start to end point.

This simple procedure actually produced movements of the arm and some examples are shown in Fig. 5a. The results are presented in workspace coordinates and show the movement of the tip of the arm. As can be seen in Fig. 5a the path in general does not follow a straight line. Remarkably these results qualitatively show similar properties as do those of the human subjects. For comparison movements of a human subject are shown in Fig. 5b (see also Cruse and Brüwer 1987). The curvature of the path produced by the network is however somewhat stronger. This is true in particular of the top down paths in Fig. 5.

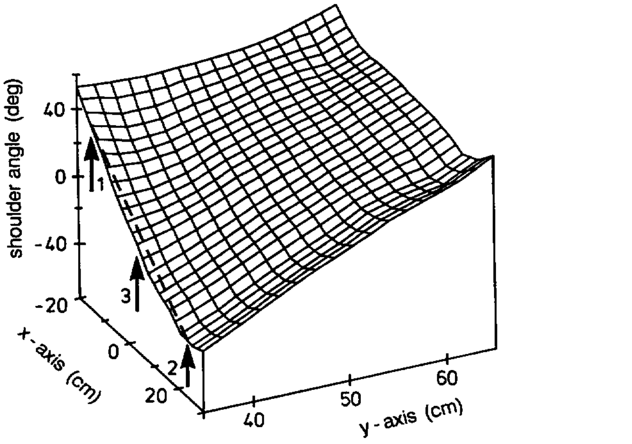

Fig. 6. The values of the shoulder angle (ordinate) calculated by the network shown in Fig. 3 for a grid of 25 × 17 target points plotted over their $x-y$ coordinate values of the workspace. For further explanation see text

Why do these curved paths appear? Figure 6 shows the angle values for the shoulder joint produced by the network over the workspace coordinates. As an example let us consider the movement from the start point to the end point which in Fig. 6 are marked by 1 and 2, respectively. This corresponds to a movement parallel to the x-axis and at small y values ($y = 35$ cm, see also Fig. 5). When during the movement from point 1 to point 2 both points are excited equally (Fs = Fe), this should result in a position at about the middle between the points 1 and 2, which is marked as point 3 in Fig. 6. However, as a comparison with the interrupted line connecting both values qualitatively shows, the actual angle value obtained in the static situation is smaller than the mean value of point 1 and point 2 which are given into the network. This means that the movement deviates to higher values of shoulder angle which produces curves paths in the way shown in Fig. 5a. A similar effect is found when considering the angles of elbow and wrist joint.

Discussion

Models on the basis of neuronal networks for the control of a manipulator have already been presented by several authors. Kohonen (1982a) and Ritter and Schulten (1987) and Ritter et al. (1989) provide systems which are based on the Kohonen algorithm (Kohonen 1982b). Another model given by Josin (1988) uses the error back propagation procedure of McClelland and Rumelhart (1988). All these models concern the non redundant arm. Here two simple models are proposed which are able to solve the redundancy problem for the static situation. The first model obtains $x-y$ coordinate values as input as does the model of Josin (1988). The second model obtains the input information via a retina-like input layer which is similar to the approach of Kohonen (1982a, b) and Ritter and Schulten (1987).

The redundancy problem was solved by an earlier algorithmic model by the application of cost functions attached to each joint (see Introduction). Both models proposed here solve the problem on the basis of distributed networks. This occurs in such a way that during the training session the weighting factors in the network connections were learned. However, no obvious explicit representation of the cost functions was found in the values of the weighting factors. The network was trained using the data obtained from a human subject and thus used an external teacher. However, as the data of this subject could be well described by three cost functions, the network could have also be trained with practically the same result by implementing these cost functions into the training program.

The second network model was extended by a simple procedure to be able to describe movements of the redundant arm from a start to an end point within the workspace. The system showed curved paths qualitatively similar to those found in humans. In the algorithmic model (Cruse and Brüwer 1987) the curved paths were produced by superimposing to the minimum cost strategy a second strategy called mass-spring strategy. For this strategy the angles of the end point have to be calculated before the movement starts and each joint moves independently of the other joints to its final angle value. The extension proposed for the network model follows a similar logic as by means of the excitation function the final angles are also determined. However, it was not immediately clear to us that the system shows these similarities also during the movement.

The network in its actual form cannot deal with three properties found in the human experiments. First, one type of experiment performed by Cruse and Brüwer (1987) was to start the movement with the arm in an "uncomfortable" position, i.e. a position which does not obey the minimum cost principle. These experiments cannot be modelled by this network. For this purpose the network had to be enlarged by input units which monitor the actual value of the three joint angles. Second, our earlier algorithmic model in addition to the minimum cost strategy and the mass-spring strategy contains a third strategy which also is not yet implemented in the network model. This so-called pseudoinverse control has the effect that during the movement the incremental changes of the three angles dS, dE, and dW obey the rule that the sum $dS^2+dE^2+dW^2$ assumes a minimum. This strategy is also important when modelling movements starting with uncomfortable arm positions. Third, both the experimental investigations and the algorithmic model showed the following qualitative property which was not found in the network. The actual movements show some kind of hysteresis so that the exact form of the path between two points depends on the direction of the movement, i.e. which of the points is the start and which is the end point. In the algorithmic model this property arises from the implementation of the pseudoinverse control.

However, two disadvantages of the algorithmic model are not found in the network model. In the algorithmic model the necessary linearisation required an iterative method of calculation. Higher exactness then requires smaller iteration steps and thus larger computing time. In the parallel network computing time is extremely short when implemented on a real parallel system. Higher exactness is only a question of the resolution of the individual units or could be increased by increasing the number of units but does not necessarily influence the computing time. Another disadvantage of the algorithmic model is the appearence of so-called singularities. As described by Cruse (1989) a singularity occurs in the algorithmic model when the wrist angle obtains a value of zero. In this case it is not possible for the model to calculate the incremental angle values for the subsequent movement step. For the network model this problem does not exist because the angle values are not calculated but represented in the combination of weighting factors of the different synapses and thus correspond to a kind of distributed look-up table. This property is also of practical importance for the control of artificial manipulators as these also have to deal with the problem of singularities.

Finally a more technical problem concerning the second network model should be mentioned. As described above, the grid of the input layer is rather coarse. Nevertheless, because of the intensive overlapping of the receptive fields – which can be considered as a sort of spatial low pass filter (v. Seelen 1968) – the network is able to interpolate and to develop a continuous projection (Fig. 6) (see also Baldi and Heiligenberg 1988). In addition, during training because of this overlap a stimulus also influences neighbouring units which therefore has a similar effect as has the Kohonen algorithm (1982b).

Just after we finished this work, a paper appeared by Massone and Bizzi (1989) in which the control of a three joint arm moving in a horizontal plane is simulated by means of a three layered network. This model differs from our second model in the following respects. Whereas our model simulates the results of human experiments, the model of Massone and Bizzi adopts a control algorithm proposed by Mussa Ivaldi et al. (1988). It provides a very elegant method to learn and control movements of a redundant manipulator whereby the movements show a bell-shaped velocity profile. As presented, the model can perform movements in the horizontal plane in all directions but all of them start from the same arm position. In contrast, in our model each point of the workspace can be used as start or end point. However, no effort was made to produce a bell-shaped velocity profile in our model nor was the movement simulated at the level of individual muscles as was the case in the model of Massone and Bizzi (1989). Although our model uses only two layers – there is no hidden layer in the second model – it provides a finer spatial resolution which is not limited to a given grid of pixels. Another difference is that no explicit formulation of the path of the endeffector is used, whereas this path was part of the training procedure in the model of Massone and Bizzi. As was described above, our model in its present form is not able to simulate movements starting from an uncomfortable arm position. As the model of Massone and Bizzi contains feedback from the output values to an upper layer, it contains the basic architecture necessary to solve this problem. Therefore only minor changes of the latter model might be necessary to also fulfill the requirements which are needed to simulate movements starting from uncomfortable positions.

References

Atkeson CG, Hollerbach JM (1985) Kinematic features of unrestrained vertical arm movements. J Neurosci 5:2318–2330

Baldi P, Heiligenberg W (1988) How sensory maps could enhance resolution through ordered arrangements of broadly tuned receivers. Biol Cybern 59:313–318

Benati M, Gaglia S, Morasso P, Tagliasco V, Zaccaria R (1980) Anthropomorphic robotics. I. Representing mechanical complexity. Biol Cybern 38:125–140

Cruse H (1986) Constraints for joint angle control of the human arm. Biol Cybern 54:125–132

Cruse H (1989) The control of path and joint angles in a human arm. In: Personnaz L, Dreyfus G (eds) Neural networks from models to applications. 71–77. IDSET, Paris

Cruse H, Brüwer M (1987) The human arm as a redundant manipulator: the control of path and joint angles. Biol Cybern 57:137–144

Flash T (1987) The control of hand equilibrium trajectories in multi-joint arm movements. Biol Cybern 57:257–274

Josin G (1988) Neural-space generalization of a topological transformation. Biol Cybern 59:283–290

Kohonen T (1982a) Clustering, taxonomy, and topological maps of patterns. Proceedings of the Sixth International Conference on Pattern Recognition. IEEE Computer Society Press, Silver Springs, pp 114–128

Kohonen T (1982b) Self-organized formation of topologically correct feature maps. Biol Cybern 43:59–69

Massone L, Bizzi E (1989) A neural network for limb trajectory formation. Biol Cybern 61:417–425

McClelland JL, Rumelhart DE (1988) Explorations in parallel distributed processing. MIT Press, Cambridge, Mass

Mussa Ivaldi FA, Morasso P, Zaccaria R (1988) Kinematic networks – a distributed model for representing and regularizing motor redundancy. Biol Cybern 60:1–16

Reichardt W, Mac Ginitie G (1962) Zur Theorie der lateralen Inhibition. Kybernetik 1:155–165

Ritter HJ, Schulten KJ (1987) Extending Kohonen's self-organizing mapping algorithm to learn ballistic movements. In: Eckmiller R, von der Malsburg C (eds) Neural computers. Springer, Berlin Heidelberg New York, pp 393–406

Ritter HJ, Martinez TM, Schulten KJ (1989) Topology-conserving maps for learning visuo-motor coordination. Neural Net 2:159–168

Seelen W von (1968) Informationsverarbeitung in homogenen Netzen von Neuronenmodellen. Kybernetik 5:133–148

Received: October 26, 1989
Accepted in revised form: December 11, 1989

Prof. Dr. Holk Cruse
Fachbereich Biologie
Universität Bielefeld
Postfach 8640
D-4800 Bielefeld 1
Federal Republic of Germany