

## 12. CONTROL OF A MANIPULATOR BY A HIERARCHICAL NETWORK

### 12.1 The Robot for the New Task

In Chapter 11 we have shown how an artificial neural network can learn to position a robot arm with its gripper at a desired target. This fundamental capability is the basis of our approach to a new and more complex task in robotics: the control of coordinated grasping movements for simple objects (Martinetz and Schulten 1990).

As we reach for a can of beer or soda, we hardly ever think about the vast complexity involved in grasping an object or, correspondingly, what must be regarded in detail while driving an artificial manipulator. For one thing, the movement to be performed depends on the location of the object in the work space. The shape of the object plays a crucial role in planning a grasping movement. It must be estimated where the center of mass lies because, if there is too large a distance between the center of mass and the chosen grasping position, then undesirable torques can arise which make it difficult to hold the object steadily. The object has to be sufficiently narrow at some place to account for the limited opening of the gripper. The orientation of the object must be considered since grasping an upright object like a beverage requires a different gripper position than that which is effective for the same object on its side. Even the contour of the object must be considered so that the gripper does not slip, and so on.

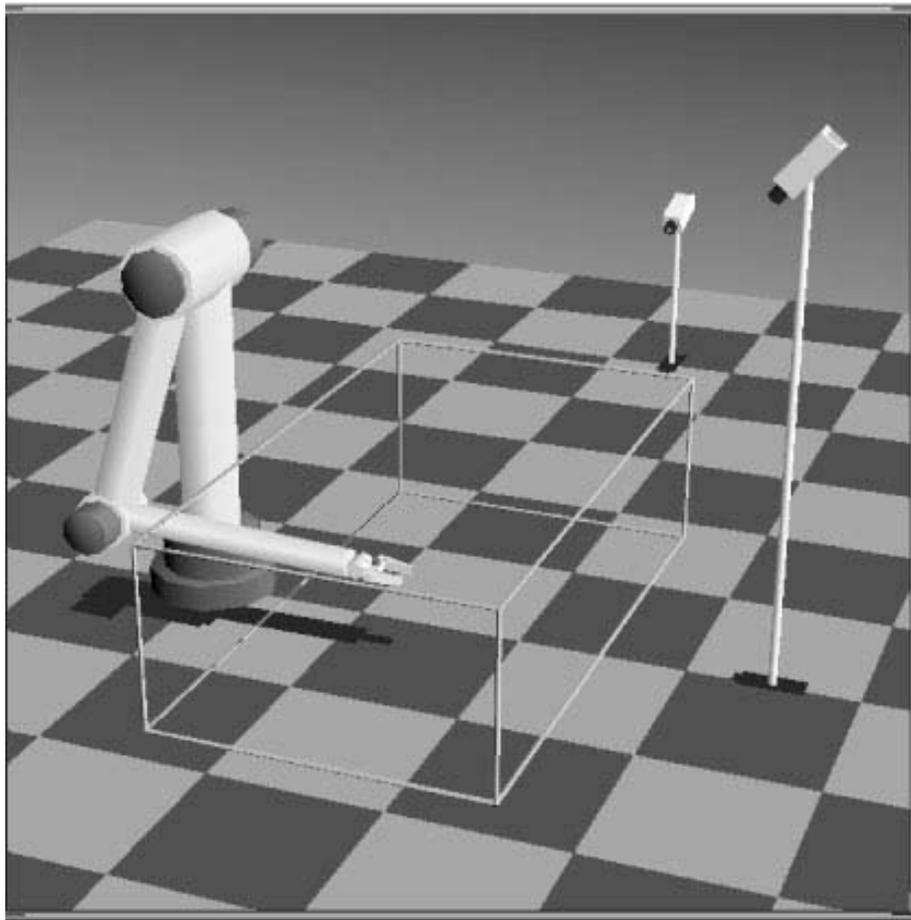
These complications make us realize the difficulty in developing an overall grasping strategy, and so it should not come as a surprise that this problem forms an area of current research. So far there have been only partial solutions. One promising approach, for instance, uses potential fields of artificial charge distributions which guide the arm and the gripper along the desired trajectory towards the final position (Hwang 1988, Ritter 1990).

We restrict the discussion in this chapter to the complications surrounding

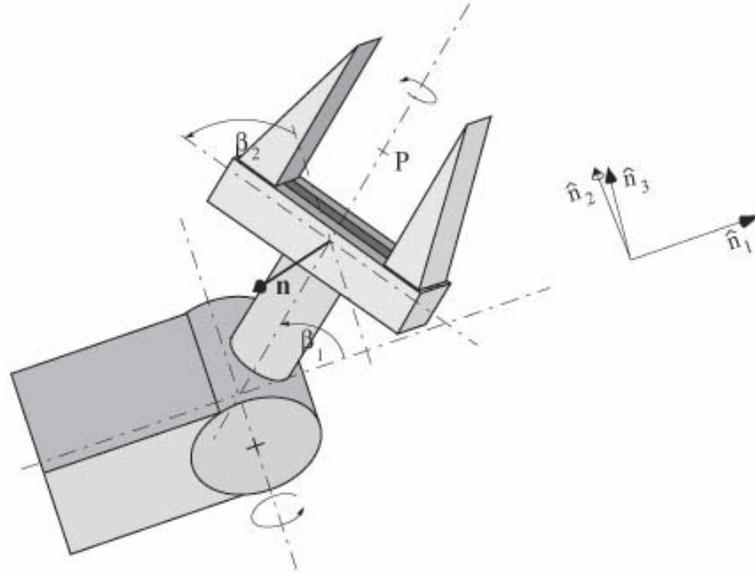
the orientation and location of an object to be fetched. We consider a pliers-type end effector of the form displayed in Fig. 12.2, and we choose a relatively simple shape for the object to be grasped, namely, a cylinder. By choosing a rotationally symmetric object the problems relating to the selection of a proper grasping point are simplified, since in this case reaching for the center of the cylinder from any direction is appropriate. Aside from the position in the work space, the orientation of the cylinder is the only other factor which determines the final position of the arm and the gripper.

In Fig. 12.1 we see the model of the robot that we used in our computer simulations. Neither the position of the cameras nor the work space has changed relative to the simulation model of the previous chapter. The geometry of the present robot model, however, imitates the shape of the human arm. This configuration allows the robot always to approach presented objects from the front. By “from the front” we mean that it is possible to bring the gripper into a position between the object and the base of the robot, and from this position the gripper proceeds to grasp the object. This anthropomorphically designed form of the robot arm will later allow the robot to use a grasping strategy which corresponds to a motion that is typical for humans. When grasping an object by hand, humans usually reach from the front. Only in exceptional cases does one opt for a different strategy as might happen, for example, when impeded by an obstacle or when picking up an object from its far side. The robot arm has three degrees of freedom by which each arm configuration is uniquely determined (there is no redundancy). As before, the arm can rotate around its vertical axis ( $\theta_1$ ) and around the axes of the middle ( $\theta_2$ ) and the outer ( $\theta_3$ ) joints, which are parallel to each other and parallel to the horizontal plane. For the orientation of the gripper, two degrees of freedom are available. The first degree of freedom is given by the axis at the “wrist,” which is parallel to the middle and outer joint of the arm. The second degree of freedom allows rotation around the gripper’s symmetry axis.

Figure 12.2 shows a sketch of the gripper with its two joint angles  $\beta_1$  and  $\beta_2$  and the normal vector  $\mathbf{n}$  which describes the orientation of the gripper. This vector is perpendicular to the symmetry axis and perpendicular to the flat side of the gripper. The point  $P$  in Fig. 12.2 denotes the place of the gripper which must be guided towards the center of the cylinder to be grasped. Simultaneously, it is necessary to orient the normal vector  $\mathbf{n}$  parallel to the axis of the cylinder. Therefore, this normal vector must be able to take on any orientation. This is ensured by the two joint angles  $\beta_1$  and  $\beta_2$ . The



**Abb. 12.1:** Model of the robot used in the simulation, the two cameras, and the work space. The robot itself is now in its geometry similar to the form of the human arm. This construction makes it always possible to drive towards objects from the front. The robot's arm has three degrees of freedom: rotation around the vertical axis plus the middle and outer joint, whose axes are parallel to each other and perpendicular to the vertical direction as in previous models. The orientation of the gripper can be modified by two degrees of freedom. The axis of the first joint is parallel to the joint axes of the middle and the outer joint of the arm. The second joint can rotate the gripper around its own symmetry axis.



**Abb. 12.2:** A sketch of the gripper with its two joint angles  $\beta_1$  and  $\beta_2$  and the normal vector  $\mathbf{n}$ , which describes the orientation of the gripper. Also shown is a coordinate system which is attached to the last segment. The components  $n_1$ ,  $n_2$ , and  $n_3$  of the normal vector  $\mathbf{n}$  relative to this coordinate system are given by Eq. (164). The point  $P$  denotes the center and, thereby, the location of the gripper which must be guided towards the center of the cylinder to be grasped.

relation between the three components of the normal vector  $\mathbf{n}$  and the joint angles is given by

$$\begin{aligned} n_1 &= -\sin \beta_1 \cos \beta_2 \\ n_2 &= -\sin \beta_2 \\ n_3 &= \cos \beta_1 \cos \beta_2 \end{aligned} \quad (12.1)$$

where  $n_1$ ,  $n_2$ , and  $n_3$  are the projections of  $\mathbf{n}$  on the coordinate system shown in Fig. 12.2, spanned by the unit vectors  $\hat{n}_1$ ,  $\hat{n}_2$ , and  $\hat{n}_3$ . In Fig. 12.2  $\hat{n}_1$  runs parallel to the longitudinal axis of the outer segment,  $\hat{n}_2$  is perpendicular to  $\hat{n}_1$  and lies parallel to the  $x$ - $y$  plane, and  $\hat{n}_3$  is perpendicular to both.

That we are really able to orient  $\mathbf{n}$  in any direction can be recognized particularly well when we perform the transformation  $\beta_1 \rightarrow -\bar{\beta}_1$  and  $\beta_2 \rightarrow \bar{\beta}_2 - 90^\circ$ .

Then we obtain

$$\begin{aligned} n_1 &= \sin \bar{\beta}_1 \sin \bar{\beta}_2 \\ n_2 &= \cos \bar{\beta}_2 \\ n_3 &= \cos \bar{\beta}_1 \sin \bar{\beta}_2, \end{aligned} \tag{12.2}$$

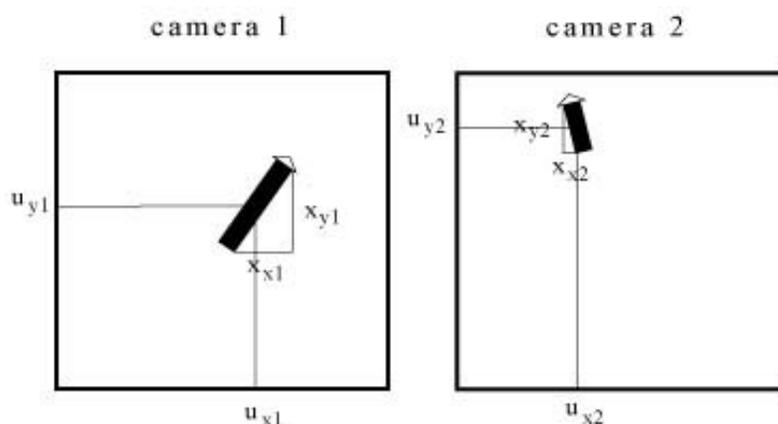
which corresponds to the polar representation of a three-dimensional vector  $\mathbf{n}$  with  $\bar{\beta}_1$  as the polar angle and  $\bar{\beta}_2$  as the azimuthal angle.

## 12.2 View through Cameras

The task of the neural network is again to transform input signals, delivered by the cameras, into suitable joint angles for the arm and the gripper. In the previous chapter we assumed that an image processing system extracts the required image coordinates of the target location from the camera images. Again we do not want to be concerned with details of the image processing, and we take for granted that we have a suitable image processing system which extracts the required input data from the camera images. To be able to grasp a cylinder, the neural network needs, in addition to the information about the location of the cylinder in the work space, information about the cylinder's orientation in space. At each position in the work space, the cylinder has two additional orientational degrees of freedom which determine the angles of the configuration that is required for arm and gripper while grasping.

In what form do the cameras deliver the necessary information about the orientation of the cylinder? In Fig. 12.3 we see the bars that result from the projection of a cylinder onto the image planes of the two cameras. The bar locations in the two image planes implicitly contain the information about the location of the cylinder in space. The orientations of the bars in the image planes provide the information about the spatial orientation of the cylinder axis.

As can be seen in Fig. 12.3 we describe the location of the center of each bar by its two-dimensional coordinates in the respective image plane of each camera and combine the two coordinate pairs to a four-dimensional vector  $\mathbf{u}$ . To describe the orientation of each bar we use its projection onto the x- and y-axes of each camera's image plane. To be able to determine this projection uniquely with respect to its sign, it is necessary to assign a direction to the bar in the image plane. In Fig. 12.3 a particular end of the bar is marked



**Abb. 12.3:** The cylinder as seen from cameras 1 and 2. One end of each bar is marked by an arrow head (see text). The position of the centers of both bars described by  $(u_{x1}, u_{y1}, u_{x2}, u_{y2})$  determines the spatial position of the cylinder. The coordinate set  $(x_{x1}, x_{y1}, x_{x2}, x_{y2})$  in a normalized form contains the information about the orientation of the cylinder that is needed by the neural net.

by an arrowhead. It does not matter which end is picked, but it must be certain that in both image planes the special end is the same actual end of the cylinder. This demands that the image processing system is able to identify the corresponding ends of both bars. This could be achieved, for example, through comparison of textures. In the case of a soda can, the image processing system must be able to find out which end of both bars corresponds, *e.g.*, to the top.

Thus, there are always two equal possibilities for selecting the special end of the bars. In our simulation, which will be described in greater detail, we choose always that end of the bar that lies “higher” in the image plane of camera 2, *i.e.*, in the coordinate system of the camera’s image plane, the one located in the more positive region of the  $y$ -axis. The image processing system must then identify the corresponding end of the bar in the image plane of camera 1, and also mark it there.

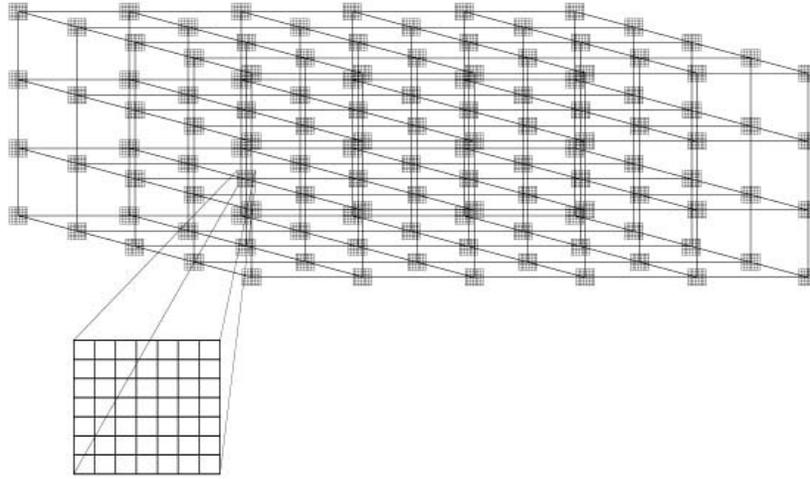
The projections yield a pair of two-dimensional vectors denoted by  $(x_{x1}, x_{y1})$  and  $(x_{x2}, x_{y2})$  which uniquely describe both bars, including the selected direction. Combining this pair of vectors, we obtain a four-dimensional vector

that contains, in addition to the orientation, information about the length of both bars. Because the information about the length of both bars which is correlated to the length of the cylinder is not relevant for the task, we normalize  $(x_{x1}, x_{y1}, x_{x2}, x_{y2})$  and obtain the four-dimensional input signal  $\mathbf{x}$  that now exclusively contains the required information on the cylinder's orientation. The two four-dimensional vectors  $\mathbf{u}$  and  $\mathbf{x}$  together comprise the complete information that is needed by the neural net to direct the grasping movement.

### 12.3 Hierarchical Arrangement of Kohonen Networks

To represent the input signals describing the location and the orientation of the cylinder, we use a network that is composed of an hierarchical arrangement of many subordinated Kohonen nets. As depicted in Fig. 12.4, the network architecture is composed of a set of two-dimensional *sub-lattices* which are arranged in a three-dimensional *super-lattice*.

To represent the input signals responsible for the spatial position, we again choose, as in the last chapter, a three-dimensional Kohonen lattice, however, providing it with two-dimensional subnets at its nodes. Each node of the three-dimensional super-lattice, *i.e.*, each sub-lattice, specializes in a small subregion of the work space during the learning process. Within each of these subregions of the work space, a topology-conserving representation of the different orientations of the cylinder emerges through the corresponding subnet.



**Abb. 12.4:** Hierarchical arrangement of Kohonen nets. A two-dimensional sub-lattice is assigned to every node of the three-dimensional super-lattice. If  $s$  is the subnet of the super-lattice that is closest to the cylinder, then the element  $q$  within  $s$  that best describes the orientation of the cylinder is selected to determine the output.

The output values are attached to the elements of the two-dimensional sub-lattices. A hierarchically organized selection process chooses the element out of all sub-lattices whose output values will later determine the joint angles for the particular object position and orientation. The selected element should represent the spatial position of the cylinder better than all the other elements. In the selection process we can restrict ourselves to that sub-lattice of the three-dimensional super-lattice which is closest to the object position. The element of this subnet that is also closest to the input signal in the space of cylinder orientations will then finally determine the output values.

If we denote the position of every subnet  $\mathbf{r}$  in the four-dimensional space of camera coordinates  $U$  by  $\mathbf{w}_r$ , then we search for the particular subnet  $\mathbf{s}$  for which holds

$$\|\mathbf{u} - \mathbf{w}_s\| \leq \|\mathbf{u} - \mathbf{w}_r\|, \quad \text{for all } \mathbf{r}.$$

This is followed by the selection of the neural unit within the subnet  $\mathbf{s}$  which is finally to be activated. By  $\mathbf{z}_{sp}$ , we denote the position attached to every element  $\mathbf{p}$  of the sub-lattice  $\mathbf{s}$  in the space of the input signals  $\mathbf{x}$  which describe the cylinder's orientation. By taking also into account the information

about the cylinder's orientation, the element  $\mathbf{q}$  of  $\mathbf{s}$  which finally determines the output signals is defined by

$$\|\mathbf{x} - \mathbf{z}_{\mathbf{s}\mathbf{q}}\| \leq \|\mathbf{x} - \mathbf{z}_{\mathbf{s}\mathbf{p}}\|, \quad \text{for all } \mathbf{p}.$$

Because of the hierarchical organization of the Kohonen nets, the time needed to search for the neural unit responsible for the output can be kept short. In our case the relevant submanifold of the whole input signal space, *i.e.*,  $U \otimes X$ , is five-dimensional because of the five degrees of freedom of the cylinder. If one made an unstructured assignment of this space with equal discretization points, *e.g.*, by applying a five-dimensional Kohonen net, the time needed for the selection of the element  $\mathbf{q}$  would increase as  $N^5$  where  $N$  is the number of elements of the Kohonen net along a single dimension. Due to the hierarchical structure of the network and, consequently, due to the hierarchically organized selection method, the search time  $t_{search}$  in our case increases only as

$$t_{search} \sim N_S^3 + N_E^2. \quad (12.3)$$

Here  $N_S^3$  is the number of subnets and  $N_E^2$  is the number of elements per subnet. The selection of the responding neuron can consequently be managed much faster, assuming  $N_E \approx N_S \approx N$ . Although the control task has become much more complex, the search time  $t_{search}$  does not increase faster with the number of net nodes per dimension than in the case of a robot without a gripper. The search time still increases only as  $N^3$ .

After the selection of the subnet with  $\mathbf{w}_s$  closest to  $\mathbf{u}$  and the neural unit with  $\mathbf{z}_{\mathbf{s}\mathbf{q}}$  closest to  $\mathbf{x}$ , adaptation steps are performed on both levels of the hierarchy of the network. These steps cause (i) a shift of all subnets in the space of camera coordinates  $U$  and (ii) an adjustment of all elements of all subnets in the space of cylinder orientations  $X$ . The shift of the subnets is accomplished by the familiar adaptation step

$$\mathbf{w}_r^{\text{new}} = \mathbf{w}_r^{\text{old}} + \epsilon \cdot h_{\mathbf{r}\mathbf{s}}(\mathbf{u} - \mathbf{w}_r^{\text{old}}), \quad \text{for all } \mathbf{r}. \quad (12.4)$$

The adjustment of the elements of the selected subnet  $\mathbf{s}$  is also accomplished according to the Kohonen rule, which yields

$$\mathbf{z}_{\mathbf{s}\mathbf{p}}^{\text{new}} = \mathbf{z}_{\mathbf{s}\mathbf{p}}^{\text{old}} + \delta \cdot g_{\mathbf{p}\mathbf{q}}(\mathbf{x} - \mathbf{z}_{\mathbf{s}\mathbf{p}}^{\text{old}}), \quad \text{for all } \mathbf{p}. \quad (12.5)$$

Just as  $h_{\mathbf{r}\mathbf{s}}$  determined the neighborhood within the three-dimensional superlattice, now  $g_{\mathbf{p}\mathbf{q}}$  determines the neighborhood within each two-dimensional

subnet. By the adaptation step ((12.4) we obtain a topology-conserving distribution of subnets in the input signal space  $U$ . Therefore, it is guaranteed that subnets that are neighbors in the super-lattice must represent a similar distribution of input signals  $\mathbf{x}$ . Hence, it makes sense to extend the adaptation step for the neurons  $\mathbf{p}$  in the selected subnet  $\mathbf{s}$  onto all neighboring subnets  $\mathbf{r}$  in a way that decreases with the distance from  $\mathbf{s}$ . To do so we again use  $h_{\mathbf{rs}}$  as a measure for the neighborhood within the three-dimensional super-lattice. This motivates us to replace ((12.5) by the adaptation step

$$\mathbf{z}_{\mathbf{rp}}^{\text{new}} = \mathbf{z}_{\mathbf{rp}}^{\text{old}} + \delta \cdot h_{\mathbf{rs}} g_{\mathbf{pq}}(\mathbf{x} - \mathbf{z}_{\mathbf{sp}}^{\text{old}}), \quad \text{for all } \mathbf{r}, \mathbf{p} \quad (12.6)$$

for all neurons  $\mathbf{p}$  of all subnets  $\mathbf{r}$ .

A hierarchical arrangement of Kohonen nets is useful when the input channels can be combined to groups of different modality with different priority for the quality of their representation. In our case the input signals had the modalities “position” and “orientation.” In selecting the best, *i.e.*, closest to  $\mathbf{u}$  and  $\mathbf{x}$ , element of the network, the modality “position” had a higher priority because the gripper attached to the end effector has to be first placed at the object before an alignment of the gripper position makes any sense. By combining elements into groups with identical receptive fields in the space of location information, we obtain a whole set of elements that represent the location information of a cylinder equally well. Within this set, the element is selected which, in addition, provides the best information on the orientation of the cylinder.

Interestingly enough, one finds similar hierarchical structures in the visual cortex of higher animals (Hubel and Wiesel 1974; Blasdel and Salama 1986; Obermayer et al. 1990, 1991). Orientation-sensitive neurons are arranged according to a hierarchically composed topographic map in the visual cortex. Locations on the map represent “locations in the visual field” as well as the “orientation” of a bar appearing on the retina. The visual cortex can be parcelled into many small sections, each of which corresponds to one location in the visual field. Within each of these sections, each neuron is specialized for a different bar orientation, and the whole orientation spectrum of  $180^\circ$  is represented within each section. In this way, a connection between the sections on the visual cortex and the subnets presented in this chapter can be made. The arrangement of the sections on the visual cortex then corresponds to the arrangement of the subnets in the super-lattice.

## 12.4 The Output Values and the Positioning Process

The required joint angles are uniquely given by the spatial position and the orientation of the cylinder, *i.e.*, by the presentation of the target for  $P$  and the direction for  $\mathbf{n}$  (see Fig. 12.2). The overall five degrees of freedom of the robot's arm and gripper are required in order to handle the three degrees of freedom for the spatial position plus the two degrees of freedom of the cylinder's orientation. It is not possible to decouple the arm and gripper configurations which are required for the different positions and orientations of the cylinder. Similar orientations of the cylinder at different locations in the work space require not only different configurations of the arm but also different alignments of the gripper's joints. The situation is similar when the cylinder at the same location appears with different orientations of its axis. In this case not only must the orientation of the gripper be changed, as one might at first assume, but also the arm must compensate by small corrections of its joint angles for the small shift of the gripper's center  $P$  that was caused by a change of the direction of the normal vector  $\mathbf{n}$ . Mathematically put, this means that the arm's joint angles  $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$  and the angles of the gripper  $\vec{\beta} = (\beta_1, \beta_2)$  depend simultaneously on  $\mathbf{u}$  as well as on  $\mathbf{x}$ . If we combine all five joint angles to the vector  $\vec{\phi} = (\theta_1, \theta_2, \theta_3, \beta_1, \beta_2)$ , then it holds

$$\vec{\phi}(\mathbf{u}, \mathbf{x}) = \begin{pmatrix} \vec{\theta}(\mathbf{u}, \mathbf{x}) \\ \vec{\beta}(\mathbf{u}, \mathbf{x}) \end{pmatrix}. \quad (12.7)$$

After presentation of the object the neural unit  $\mathbf{q}$  within the subnet  $\mathbf{s}$  is selected which will be responsible for determining suitable output values for setting the joint angles  $\vec{\phi}$ . For this purpose each element  $\mathbf{p}$  of every subnet  $\mathbf{r}$ , in the following denoted by  $\mathbf{rp}$ , stores two terms, a term  $\vec{\phi}_{\mathbf{rp}}$  for gross-positioning and a tensor  $\mathbf{A}_{\mathbf{rp}}$  which serves to linearly interpolate between neighboring units  $\mathbf{rp}$ .  $\vec{\phi}_{\mathbf{rp}}$  in this case is a five-dimensional vector and  $\mathbf{A}_{\mathbf{rp}}$  is of dimensions  $5 \times 8$ . The representation of the transformation  $\vec{\phi}(\mathbf{u}, \mathbf{x})$  that has to be learned is achieved by covering the input signal space with locally valid linearizations of  $\vec{\phi}(\mathbf{u}, \mathbf{x})$ . The linearizations are done around the locations  $\tilde{\mathbf{w}}_{\mathbf{rp}} = (\tilde{\mathbf{w}}_{\mathbf{r}}, \mathbf{z}_{\mathbf{rp}})$ , where  $\tilde{\mathbf{w}}_{\mathbf{rp}}$  denotes the position in the whole input signal space (the product space  $U \otimes X$ ) connected with unit  $\mathbf{rp}$ . If we combine both input signals  $\mathbf{u}$  and  $\mathbf{x}$  into  $\tilde{\mathbf{u}} = (\mathbf{u}, \mathbf{x})$ , then the responding neural unit  $\mathbf{sq}$  generates as an output signal the joint angles

$$\vec{\phi}_i = \vec{\phi}_{\mathbf{sq}} + \mathbf{A}_{\mathbf{sq}}(\tilde{\mathbf{u}} - \tilde{\mathbf{w}}_{\mathbf{sq}}). \quad (12.8)$$

Equation (12.8) is of the same form as Eq. (12.2) for a robot arm without a gripper.

Equation (12.8) determines the first movement step which is followed by a corrective movement—just like in the case of the robot arm without a gripper. For this purpose it is necessary to determine the position of the arm and gripper from both camera perspectives after the first movement step. For one thing, one needs the image coordinates of the center  $P$  of the gripper in cameras 1 and 2. The pair of two-dimensional image coordinates is combined to a four-dimensional vector denoted by  $\mathbf{v}_i$ . Furthermore, one needs information about the orientation of the gripper that is provided by the direction of the normal vector  $\mathbf{n}$  (see Fig. 12.2). If we imagine the normal vector  $\mathbf{n}$  projected onto the camera image planes, then we obtain a two-dimensional vector in each camera. The orientations of both vectors describe the orientation of the gripper in camera coordinates, and they must be brought into alignment with the orientation of the two vectors  $(x_{x1}, x_{y1})$  and  $(x_{x2}, x_{y2})$  that describe the orientation of the cylinder. The pair of two-dimensional vectors that describe the orientation of  $\mathbf{n}$  after the first movement step is combined to a four-dimensional vector  $\mathbf{y}_i$ . The absolute value of the difference  $\|\mathbf{x} - \mathbf{y}_i\|$  of the vector  $\mathbf{y}_i$  and the input signal  $\mathbf{x}$  is to be minimized.

How can the projection of the normal vector  $\mathbf{n}$  on the image planes be determined from the images of the gripper? One possibility would be to attach a mark that can easily be identified, *e.g.*, a light, on each of the two flat sides of the gripper directly opposite to each other. The difference vector from the positions of the two lights as seen from the first camera yields the projection of the normal vector  $\mathbf{n}$  onto the image plane of camera 1. Analogously, we obtain the projection of  $\mathbf{n}$  onto the image plane of camera 2. Both difference vectors are then combined to a four-dimensional vector and are normalized to eliminate the irrelevant information about the distance of both lamps to each other. In that way, information about the direction of the virtual normal vector or equivalently, about the orientation of the gripper, can be obtained in a very simple way from the camera pictures of the gripper.

The information about the intermediate position of the arm  $\mathbf{v}_i$  and about the intermediate orientation of the gripper  $\mathbf{y}_i$  (resulting from the angles  $\vec{\phi}_i$  as given by (12.8) are again combined into a vector  $\tilde{\mathbf{v}}_i = (\mathbf{v}_i, \mathbf{y}_i)$ . The residual difference  $\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_i$  between the target coordinates  $\tilde{\mathbf{u}}$  and the intermediate coordinates  $\tilde{\mathbf{v}}_i$  determines a corrective step. This step uses the Jacobian matrix of the responding neural unit  $\mathbf{sq}$  and determines a correction for all

five joint angles,

$$\Delta\vec{\phi} = \mathbf{A}_{\text{sq}}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_i), \quad (12.9)$$

by which we obtain the final joint angle configuration  $\vec{\phi}_f = \vec{\phi}_i + \Delta\vec{\phi}$ . The corresponding position and orientation of the gripper is again observed through the cameras and denoted by  $\tilde{\mathbf{v}}_f = (\mathbf{v}_f, \mathbf{y}_f)$ .

The corrective step (12.9) can be performed several times in succession to further reduce the positioning and orientation error to values only limited by the imperfections of the devices used in practice, *e.g.*, by the camera resolution. The corrective step, being based on feedback control, will later allow us to use a grasping strategy that is similar to the grasping strategy that humans use. The feedback process allows to move toward an object cautiously, thereby avoiding a collision with the object. In the following considerations we will confine ourselves to a single corrective step (12.9).

## 12.5 The Learning Method for the Output Values

The use of a vector  $\vec{\phi}_{\text{rp}}$  and of a Jacobian matrix  $\mathbf{A}_{\text{rp}}$  to represent the transformation  $\vec{\phi}(\mathbf{u}, \mathbf{x})$  is analogous to the learning algorithm for the robot arm without a gripper in the previous chapter. This is also true for the positioning process with its two phases, gross positioning according to (12.8) and corrective step according to (12.9) (step 5 and step 6 in Section 11.2). Therefore, we can adopt the algorithm presented in Section 11.3 that we employed for the learning of the end effector positioning to improve the output values  $\vec{\phi}_{\text{rp}}$  and  $\mathbf{A}_{\text{rp}}$ . Only the transformation of the learning success of one neural unit onto its neighbors will be of a different form because of the hierarchical architecture of the network.

The corrective movement (12.9) again serves to iteratively determine the Jacobian matrices from small changes in the joint angles in conjunction with the corresponding small changes in the camera coordinates. This makes it possible to determine an improved estimate  $\mathbf{A}^*$  for  $\mathbf{A}_{\text{sq}}$ . Combined with the step that determines an improved estimate  $\vec{\phi}^*$ , we obtain

$$\begin{aligned} \vec{\phi}^* &= \vec{\phi}_{\text{sq}} + \delta_1 \cdot \mathbf{A}_{\text{sq}}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_i) \\ \mathbf{A}^* &= \mathbf{A}_{\text{sq}} + \delta_2 \cdot \mathbf{A}_{\text{sq}}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_f)\Delta\tilde{\mathbf{v}}^T, \end{aligned} \quad (12.10)$$

where  $\Delta\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_f - \tilde{\mathbf{v}}_i$ . For the adaptation step widths  $\delta_1$  and  $\delta_2$  we choose, as in Section 11.2, the optimal values  $\delta_1 = 1$  and  $\delta_2 = 1/\|\Delta\tilde{\mathbf{v}}\|^2$ .

The new estimates  $\vec{\phi}^*$ ,  $\mathbf{A}^*$  obtained by means of Eq. (12.10) are used to improve the output values of the neural unit  $\mathbf{s}_q$  and its neighbors. But in the present case of a robot arm with gripper we have two hierarchies of neighborhoods, one within the subnet that is described by the neighborhood function  $g'_{pq}$ , and the other between the subnets described by the neighborhood function  $h'_{rs}$ . Not only the neighboring neurons  $\mathbf{p}$  within the subnet  $\mathbf{s}$  participate in the learning of the activated neuron  $\mathbf{s}_q$ , but also the subnets  $\mathbf{r}$  which are neighboring in the super-lattice participate according to their distance to the subnet  $\mathbf{s}$ . This leads to the adaptation step

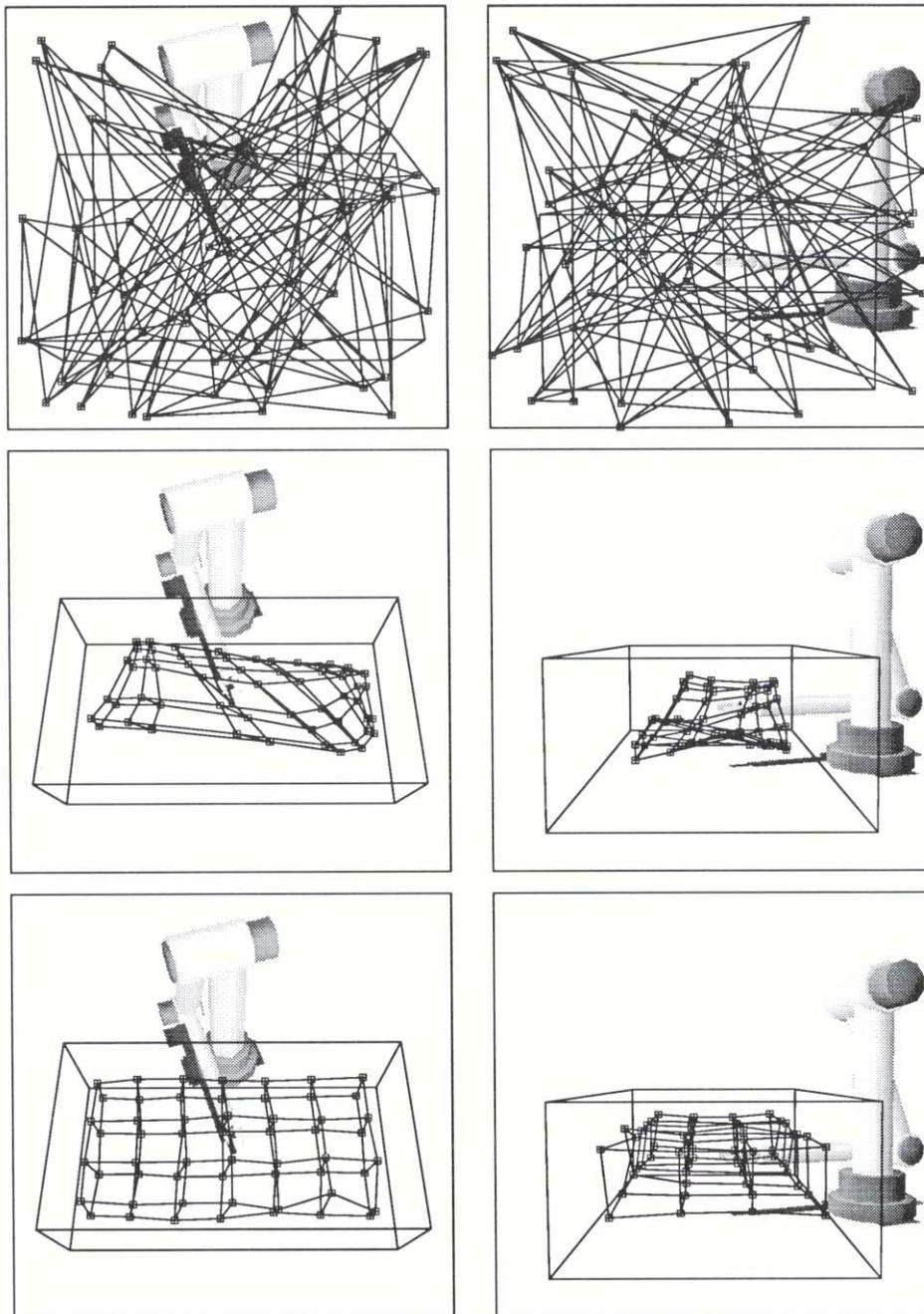
$$\begin{aligned}\vec{\phi}_{\mathbf{rp}}^{\text{new}} &= \vec{\phi}_{\mathbf{rp}}^{\text{old}} + \epsilon' h'_{\mathbf{rs}} g'_{\mathbf{pq}} (\vec{\phi}^* - \vec{\phi}_{\mathbf{rp}}^{\text{old}}) \\ \mathbf{A}_{\mathbf{rp}}^{\text{new}} &= \mathbf{A}_{\mathbf{rp}}^{\text{old}} + \epsilon' h'_{\mathbf{rs}} g'_{\mathbf{pq}} (\mathbf{A}^* - \mathbf{A}_{\mathbf{rp}}^{\text{old}})\end{aligned}\quad (12.11)$$

for the neural units of all subnets.

The learning steps (12.11) for the output values are of the same form as the learning step (12.6) for the position  $\mathbf{z}_{\mathbf{rp}}$  attached to each element  $\mathbf{rp}$ . Both affect neural units that are neighbors in the subnet and also subnets that are neighbors in the super-lattice. Instead of the input signals  $\mathbf{x}$ , the improved estimates  $\vec{\phi}^*$  and  $\mathbf{A}^*$  occur in Eq. (12.11).

## 12.6 Simulation Results

In this section we describe the results of a simulation of the learning algorithm. For the computer simulation we have employed a super-lattice consisting of  $4 \times 7 \times 2$  subnets. Each of the subnets contains  $3 \times 3$  neural units. The parameters that describe the work space and the position of the cameras are the same as in the previous chapter.



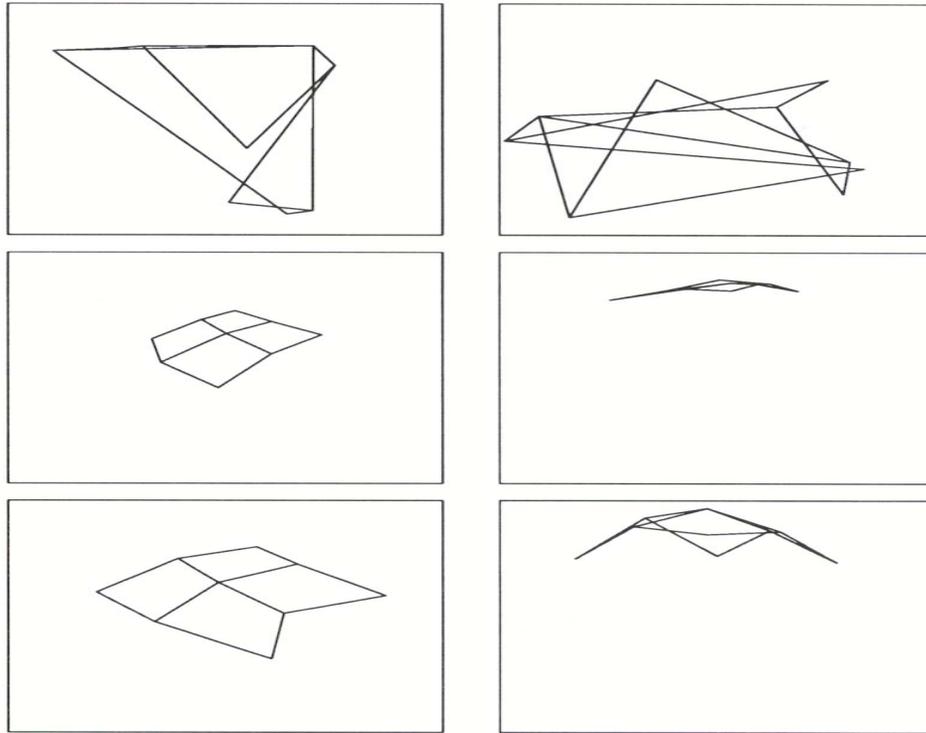
**Abb. 12.5:** The three-dimensional super-lattice, initially (top), after 1000 (middle), and after 10,000 learning steps (bottom). The left column shows the focal plane of camera 1, and the right column shows the focal plane of camera 2. At each node of the super-lattice the corresponding subnet is schematically depicted.

In total, the robot arm performed 10,000 learning steps. In Fig. 12.5 we present the development of the three-dimensional super-lattice. We again show the focal planes of cameras 1 and 2. In the left column of Fig. 12.5 we see the projection of the centers  $\mathbf{w}_r$  of the receptive fields of the subnets on the focal plane of camera 1. In the same way the projections of  $\mathbf{w}_r$  on the focal plane of camera 2 are depicted in the right column. In each projection, subnets which are adjacent in the super-lattice are connected by straight lines. The state of the super-lattice is shown at the beginning, after 1000, and after 10,000 learning steps. The development of the super-lattice corresponds, except for a lower number of lattice points, to the development of the three-dimensional Kohonen lattice in the previous chapter.

We illustrate the development of all the two-dimensional subnets by displaying the development of one of them. The input signals of each subnet consist of four-dimensional vectors  $\mathbf{x} = (x_{x1}, x_{y1}, x_{x2}, x_{y2})$ , the components of which were restricted to the interval  $[-1, 1]$  through the imposed normalization. The input signals  $\mathbf{x}$  are represented by the four-dimensional vectors  $\mathbf{z}_{rp}$  that are assigned to the elements of each subnet. The first two components of  $\mathbf{z}_{rp}$  represent the occurring bar orientations  $(x_{x1}, x_{y1})$  as seen by camera 1, and the last two components represent the bar orientations  $(x_{x2}, x_{y2})$  seen by camera 2. In Fig. 12.6 we show the state of the subnet in the beginning, after 1000, and after 10,000 learning steps. In the left column the first two components and in the right column the last two components of  $\mathbf{z}_{rp}$  are depicted. Each of the squares in Fig. 12.6 represents the region  $-1 < x_x < 1$ ,  $-1 < x_y < 1$ .

Since the edges of the presented cylinders always lie on the surface of a sphere (end-points of normalized vector  $\mathbf{x}$ ), it follows that the two-dimensional submanifold that is to be represented by a subnet is also spherical. Each of the two-dimensional subnets tries to adhere to the surface of the sphere. In the right column of Fig. 12.6 we see that the subnet shown lies completely in the upper half of the box. This results from our assignment of the “special” end of the bar in camera 2 to the end with the larger  $y$ -value. Hence,  $x_{y2}$  is always positive and at the end of the learning, only the upper half of the sphere is covered by the subnet. The vector that describes the bar as seen by camera 1 may have almost any direction in the focal plane. Thus, in the left column of Fig. 12.6 the net at the end of the learning phase is more evenly spread out.

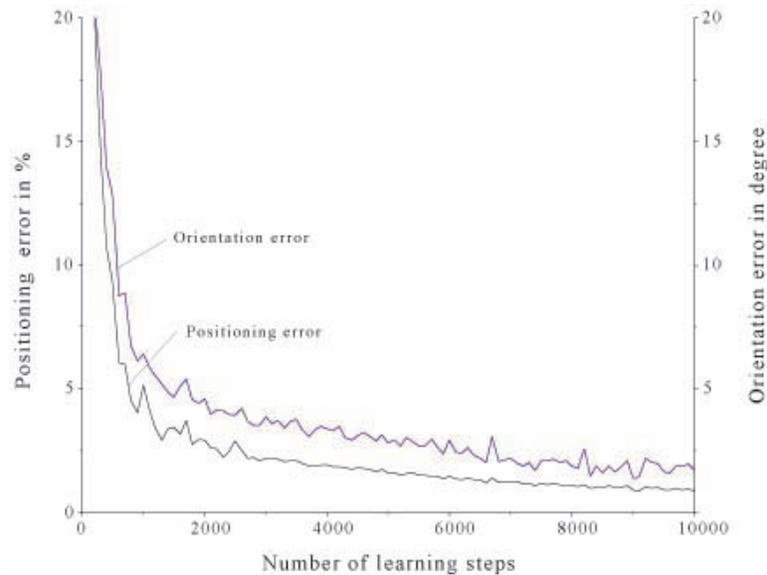
In the simulation just described, we assumed for all the parameters  $\epsilon$ ,  $\delta$ ,  $\epsilon'$ ,  $\sigma$ ,  $\sigma'$ ,  $\rho$ , and  $\rho'$  the same time dependence  $x_i(x_f/x_i)^{t/t_{max}}$  with  $t_{max} =$



**Abb. 12.6:** The state of a sample subnet at the beginning (top), after 1000 (middle), and after 10,000 learning steps (bottom).

10,000. The parameters  $\rho$  and  $\rho'$  denote the width of the Gaussians  $g_{\mathbf{p}\mathbf{q}}$  and  $g'_{\mathbf{p}\mathbf{q}}$  introduced in 12.6) and (12.11). For the initial and final values of the parameters we chose  $\epsilon_i = 1$ ,  $\epsilon_f = 0.01$ ,  $\delta_i = 1$ ,  $\delta_f = 0.01$ ,  $\epsilon'_i = 0.8$ ,  $\epsilon'_f = 0.4$ ,  $\sigma_i = 1.5$ ,  $\sigma_f = 0.3$ ,  $\sigma'_i = 1$ ,  $\sigma'_f = 0.3$ ,  $\rho_i = 1$ ,  $\rho_f = 0.1$ ,  $\rho'_i = 1$ ,  $\rho'_f = 0.3$ .

In Fig. 12.7 we illustrate the learning success by presenting the positioning and orientation error as a function of the number of learning steps. The corresponding errors were determined by performing a test after every 100 trial movements. For each test we suspended the learning and monitored the performance by presenting a randomly oriented cylinder at 1000 randomly chosen locations within the work space. The mean error at that stage of learning was computed by averaging over the errors of the 1000 test movements. Two quantities were monitored: (i) the positioning error, *i.e.*, the difference between the center of the gripper  $P$  and the center of the cylinder,



**Abb. 12.7:** The mean positioning (in % of the length of the workspace) and orientation error (in degree) as a function of the number of learning steps. After 10,000 learning steps the positioning error is 0.004, which corresponds to about 0.6% of the length work space. The slightly larger error compared to the robot arm without a gripper in the previous chapter is due to the smaller three-dimensional main lattice. The orientation error of the gripper decreased to the small value of  $1.7^\circ$  after 10,000 learning steps. This value is much smaller than necessary for successful performance of the task.

and (ii) the difference between the gripper orientation and the orientation of the cylinder, measured in degrees. The positioning error after 10,000 learning steps has decayed to 0.004, which corresponds to 0.6% of the length of the work space. The positioning error is slightly larger than in the previous chapter where only the end-effector positioning was learned. This results from a six-times smaller number of nodes of the three-dimensional super-lattice and, therefore, a much smaller resolution for the positioning task. Yet, a positioning error of 0.004 is still acceptable for the task. The error in orienting the gripper at the end of the learning phase measured  $1.7^\circ$ . The precision in orienting the gripper is much higher than is usually necessary for grasping tasks and comparable to human performance. It is remarkable that such a

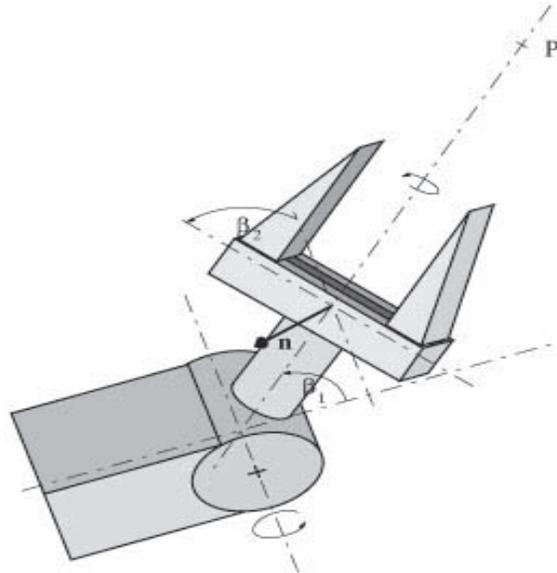
precision could be achieved through subnets consisting of only nine nodes.

## 12.7 A Simple Grasping Strategy

We have seen that the robot system can learn orientation and positioning of a gripper in relation to a cylindrical object. However, this is not enough in order to be able to grasp the object. In addition, an adequate strategy for approaching the object is necessary. The object has to be approached such that the robot's arm and gripper do not collide with it. For that reason we have chosen, as mentioned earlier in this chapter, a robot architecture which allows the robot to always approach the object from the front (see Fig. 12.1). That means the gripper can always be positioned between the base of the robot and the object. In the following we present a grasping strategy which takes advantage of such a robot architecture.

Humans usually carry out grasping movements by first coarsely positioning the hand in front of the object and then, controlled by a feedback loop, smoothly approach and finally grasp the object. We will choose a corresponding strategy for the robot arm. Until now the robot has tried to directly move the center of the gripper  $P$  to the center of the cylinder by means of movement step (12.8). However, such an approach would lead in most cases to a collision with the object. The learning algorithm has to acquire a trajectory of the manipulator which during the transition from the previous joint angles to the new ones avoids collisions. Which type of trajectory avoids collisions with the cylinder? If we assume that the gripper already has its proper orientation, then it is sufficient for the last portion of the trajectory to be arranged such that the continuation of the symmetry axis of the gripper always crosses the symmetry axis of the cylinder. To ensure this, we must modify our previous moving strategy.

For that purpose the point  $P$  which until now was placed in the center of the gripper and by design was moved to coincide with the center of the cylinder, is now slid out along the symmetry axis of the gripper to a position shown in Fig. 12.8, namely to a position in front of the gripper. The learning algorithm, by placing the point  $P$  at the center of the cylinder, positions the gripper in front of the cylinder rather than colliding with it. At the same time, during the first movement step, the gripper adjusts to its proper orientation. A feedback-guided movement follows that leads the center of the



**Abb. 12.8:** The sketch of the gripper shown in Fig. 12.2. We slide the point  $P$ , which is to be moved to the center of the cylinder by the first movement step, out along the symmetry axis of the gripper until it lies well in front of the gripper.

gripper smoothly and without collisions towards the center of the cylinder.<sup>1</sup> To accomplish the latter motion we employ a corrective movement described by (12.9). This corrective movement, rather than having to reduce the deviation between  $P$  and the center of the cylinder, now has to reduce the remaining discrepancy between the center of the gripper and the center of the cylinder. This latter discrepancy may be relatively large, depending on how far  $P$  lies in front of the gripper. Therefore, we now have to carry out the corrective movement not just once, but several times, until the residual positioning error drops below a desired minimal value. By  $\mathbf{v}'_i$  we denote the location of the center of the gripper after the first movement step (12.8) as seen by the cameras. This location along with the orientation  $\mathbf{y}_i$  of the

<sup>1</sup> The reader may note the similarity between the approach presented here and the observed strategy of saccadic eye movements which undershoot fovea targets as discussed in Section 9.5

gripper is denoted by  $\tilde{\mathbf{v}}'_i = (\mathbf{v}'_i, \mathbf{y}_i)$ . We then obtain

$$\Delta \vec{\phi} = \gamma \mathbf{A}_{\text{sq}}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}'_i) \quad (12.12)$$

for the corrective movement in approaching the object, where  $\gamma$  is the parameter which determines the step size.

With the first movement step, we achieve the crossing of the symmetry axis of the gripper with the symmetry axis of the cylinder. While the gripper approaches the cylinder by the corrective movements (12.12), these axes must remain intersected with one another. This would be guaranteed if  $\mathbf{A}_{\text{sq}}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}'_i)$  were exactly the required joint angle difference  $\vec{\phi}(\tilde{\mathbf{u}}) - \vec{\phi}(\tilde{\mathbf{v}}'_i)$ . Since this is not exactly the case, the choice of  $\gamma = 1$  can lead to a significant deviation of the resulting trajectory from the desired trajectory along the line connecting the gripper center and the center of the cylinder. By choosing  $\gamma \ll 1$  and, consequently, by adding a number of feedback loops (12.12), we force the deviation from the desired trajectory to remain small, enabling the approach towards the object to proceed smoothly and collisionless.

As an improved estimate for  $\vec{\phi}_{\text{sq}}$  we obtain, as before,

$$\vec{\phi}^* = \vec{\phi}_{\text{sq}} + \delta_1 \cdot \mathbf{A}_{\text{sq}}(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}_i). \quad (12.13)$$

Nonetheless, the equation that determines the improved estimate for  $\mathbf{A}_{\text{sq}}$  needs to be modified compared to (12.10) since we now employ several correction movements with step sizes  $\gamma < 1$ . As before we denote the position of  $P$  on the focal planes of the cameras after the first movement step by  $\mathbf{v}_i$ , and in combination with  $\mathbf{y}_i$  we define  $\tilde{\mathbf{v}}_i = (\mathbf{v}_i, \mathbf{y}_i)$ . If, as before, we denote the position of  $P$  and the gripper's orientation after the first corrective movement by  $\tilde{\mathbf{v}}_f$ , the expression

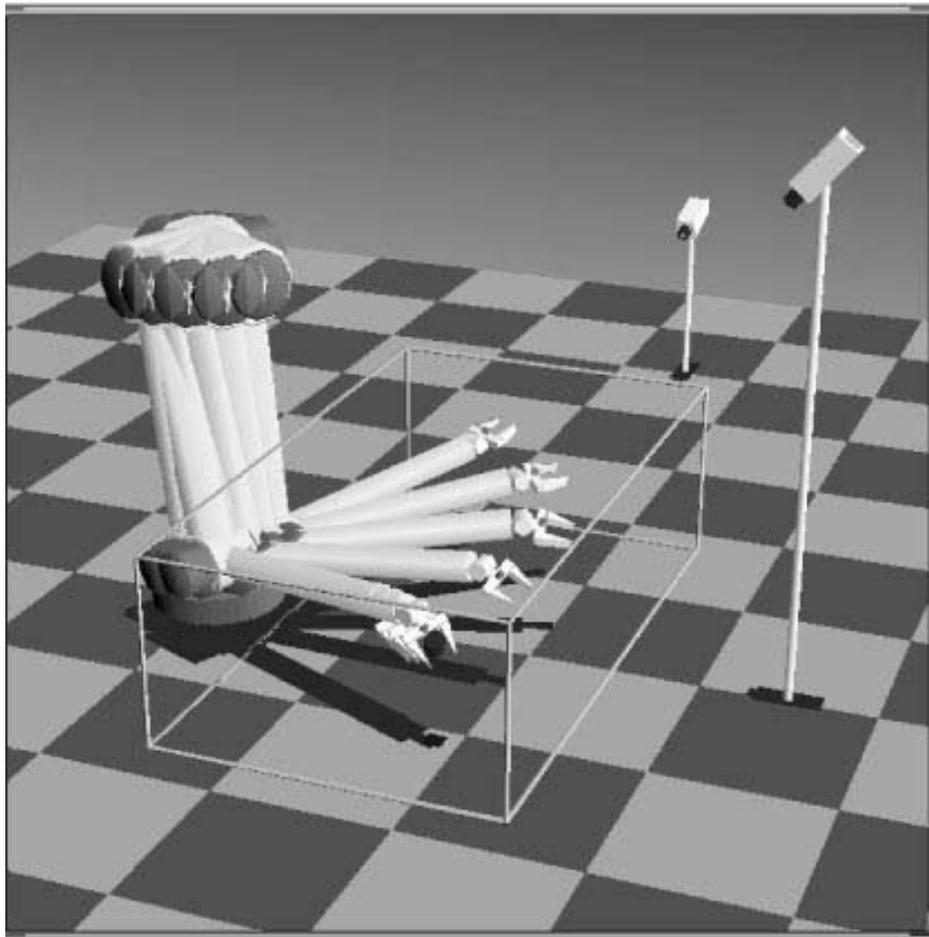
$$\mathbf{A}^* = \delta_1 \cdot [\Delta \vec{\phi} - \mathbf{A}_{\text{sq}}(\tilde{\mathbf{v}}_f - \tilde{\mathbf{v}}_i)] [\tilde{\mathbf{v}}_f - \tilde{\mathbf{v}}_i]^T \quad (12.14)$$

which corresponds to Eq. (11.11) yields in conjunction with (12.12)

$$\mathbf{A}^* = \delta_1 \cdot \mathbf{A}_{\text{sq}} [\gamma(\tilde{\mathbf{u}} - \tilde{\mathbf{v}}'_i) - (\tilde{\mathbf{v}}_f - \tilde{\mathbf{v}}_i)] [\tilde{\mathbf{v}}_f - \tilde{\mathbf{v}}_i]^T \quad (12.15)$$

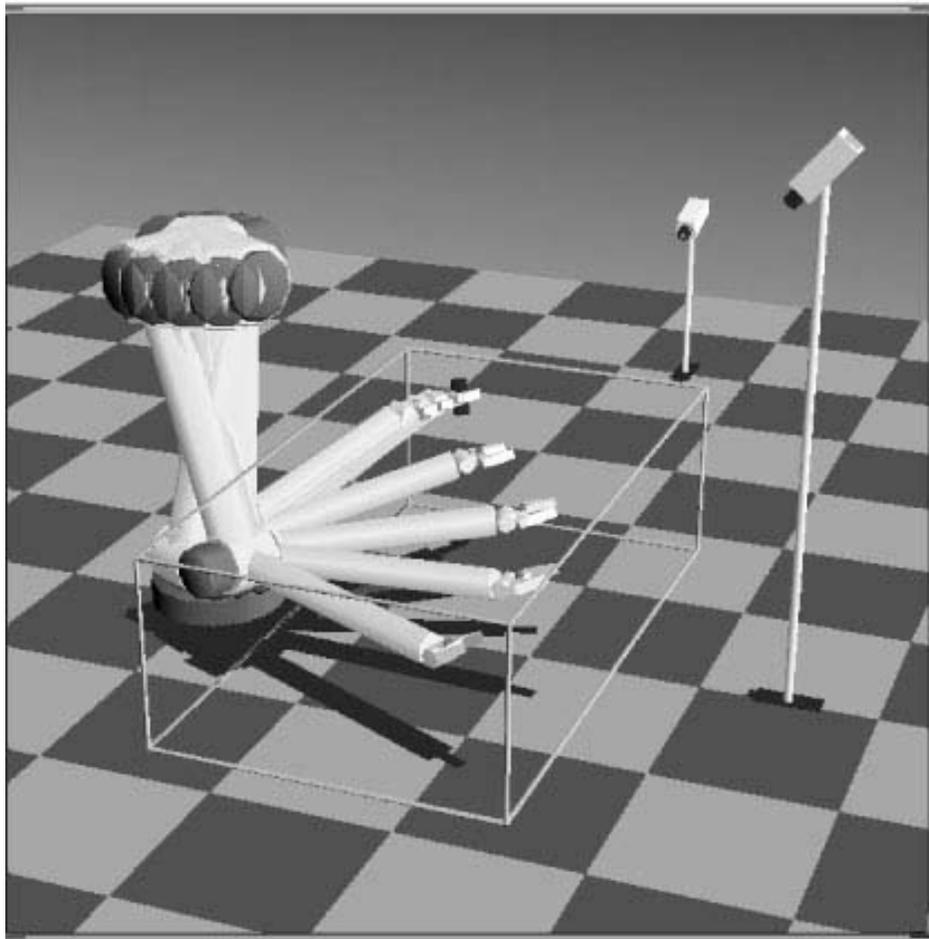
as an improved estimate for  $\mathbf{A}_{\text{sq}}$ . As in the earlier procedure,  $\mathbf{A}^*$  is computed using only values provided by the cameras. As adaptation steps for all  $\vec{\phi}_{\text{rp}}$  and  $\mathbf{A}_{\text{rp}}$  we employ again (12.11).

Without major revisions we have been able to realize the described grasping strategy by our algorithm; the Jacobian matrices  $\mathbf{A}_{\text{rp}}$  allow us to set up the



**Abb. 12.9:** “Stroboscopic” rendering of a grasping movement of the robot. The chosen movement strategy enables the robot to approach the cylinder without collisions.

feedback loop in a natural way. The robot arm is now able to approach the cylinders that are presented within the work space without inadvertently colliding with them. To demonstrate the action of the robot arm, we show “stroboscopic” renderings of two grasping movements in Figs. 12.9 and 12.10. In every approach of the cylinder three corrective fine movements (12.12) were carried out after the gross positioning. For the step size  $\gamma$  of the fine movements, we chose  $0.3 + 0.2 \cdot n$  with  $n = 1, 2, 3$  as the number of the



**Abb. 12.10:** “Stroboscopic” rendering of a second grasping movement. Again the robot carries out the grasping movement successfully.

currently performed correction step. One can see in Figs. 12.9 and 12.10 that in both cases the robot arm system accomplishes the grasping movement successfully.

We have seen that it is possible by the neural network algorithm introduced to solve not only the basic problem of end-effector positioning, but also to approach the more complex task of grasping simple objects. It turns out that less computational power and memory for controlling the robot arm are required if, as the complexity of the task rises, the network architecture

becomes increasingly structured. By employing a hierarchical arrangement of Kohonen networks, the input signals for controlling the grasping movements can be represented in a natural way. In Chapter 13 we will turn to questions which arise under the *dynamic* control of robot arms.