STAGED LEARNING OF SACCADIC EYE MOVEMENTS WITH A ROBOT CAMERA HEAD

W. SCHENCK AND R. MÖLLER
Max Planck Institute for Psychological Research,
Cognitive Robotics Group,
Amalienstr. 33, 80799 München, Germany
E-mail: schenck@psy.mpg.de

In motor learning, two main problems arise: the missing teacher signal, and the necessity to explore high-dimensional sensorimotor spaces. Several solutions have been proposed, all of them limited in some respect. In the present work, an alternative learning mechanism is developed for the example of saccade control, implemented on a stereo vision robot camera head. This approach relies on two main principles: averaging over imperfect learning examples, and learning in multiple stages. In each stage, a saccade controller is trained with a set of imperfect learning examples. Afterwards, the output of this controller serves as starting point for the creation of a new training set with better quality. By the repetition of these steps, the controllers’ performance can be incrementally improved without the need to search from scratch for the rare learning examples with very good quality.

1. Introduction

Internal models relate sensory signals from different modalities as well as sensory and motor signals. They are used to model functions of biological motor control systems on a computational level [5]. Internal models are divided into two broad categories: forward models and inverse models. Forward models predict the outcome of motor actions, while inverse models generate the necessary motor commands to attain sensory goal states. Controllers are inverse models with a constant goal state.

In this framework, motor learning is interpreted as training of adaptive internal models. In the following, we will focus on the training of controllers. From the current sensory state given as input, the controller produces a motor command as output (see Fig. 1). The training of controllers requires learning examples which pair the current sensory state with an appropriate motor command transforming the current sensory state into the constant goal state. These examples are difficult to collect. First, no teacher signal is
available in motor space. While the deviation between the desired sensory state and the resulting sensory state after a movement is easily to obtain, in general the mapping of this error back to motor space is unknown. Thus, the sensory error cannot be exploited for learning. Alternatively, one could search systematically in sensorimotor space for good learning examples. However, in general this simple approach is computationally too expensive due to the high dimensionality of sensorimotor spaces.

To cope with both the missing teacher signal and the high dimensionality of sensorimotor spaces, several approaches to the learning of controllers (and inverse models) have been proposed: Feedback error learning [6], direct inverse modelling [4], distal supervised learning [4], and learning by interpolation [8]. Each approach has its own strengths and weaknesses which will be discussed later. In the present work, we suggest an alternative solution: staged learning in combination with learning by averaging.

We will illustrate the practical use of this approach in the domain of saccade control. Saccades are rapid eye movements (up to 900 deg/s in humans) for the fixation of interesting points in visual space. The fixed points are projected onto the fovea, the region with highest cone density on the retina. Only when both eyes fixate the same object, the images of this object on both retinas can be fused to a single percept by the human brain. Since retinal delay is in the order of 50 msec (longer than the duration of most saccades), visual feedback is not fast enough to control saccades [3]. Therefore, the movement parameters for the eye muscles have to be precisely determined before a saccade starts. As research on infants shows [1], proper saccades towards peripheral targets have to be learnt during infancy. Thus, it is an important question how this learning works. In our study, we show the applicability of staged learning for saccade-like fixation movements of a robot camera head. This camera head is supposed to fixate target objects on a horizontal surface (a table).

2. Learning of Controllers

2.1. Overview of Related Approaches

In feedback error learning [6], a feedback controller is suggested for producing an approximate teacher signal in motor space. For saccade control, this approach works well [2, 3]. On the downside, one has to presume the a priori existence of a simple feedback control scheme which limits the general applicability of feedback error learning.

In direct inverse modelling [4], random motor commands are produced.
The sensory effect of each movement is incorporated in the respective learning example as desired sensory state while the movement command itself is used as motor output and the initial sensory state as sensory input of the learning example. This learning scheme suffers heavily from the high dimensionality of sensorimotor spaces. Controller learning is virtually impossible since one must search for movements which result exactly in the fixed sensory goal state of the control task.

Distal supervised learning [4] combines a controller with a forward model. First, the forward model has to be trained to predict sensory states resulting from the application of motor commands. Afterwards for the training of the controller, the teacher signal in motor space is generated by backpropagating the sensory error through the forward model. On the first glance, this seems to be an elegant and universal solution, but unfortunately the acquisition of a sufficiently precise forward model may suffer from the high dimensionality of sensorimotor spaces as well.

Learning by interpolation [8] is a more technical solution. The error in the resulting sensory state is transformed into a change in the initial sensory state before the movement. As a prerequisite, one needs to know how the mapping between both sensory states looks like. This is very unlikely for biological organisms.

2.2. Learning by Averaging

In our approach, learning by averaging is the principle underlying each stage in the learning process. Let \( s_t \) denote the sensory state at time step \( t \), while \( s_D \) is the desired sensory state. In time step \( t \), a controller \( C \) produces the motor command \( m_t \). Due to this motor command, the world state changes, resulting in a new sensory state \( s_{t+1} \) (see Fig. 1). The difference between \( s_D \) and \( s_{t+1} \) is the sensory error \( \varepsilon_s \). We define a function \( Q_s(m) \) which quantifies the quality of motor command \( m \) applied in sensory state \( s \): the lower \( \varepsilon_s \), the higher \( Q_s(m) \).

The basic idea of learning by averaging is to collect not only perfect
Figure 2. Learning by averaging (left) and staged learning (right). These graphs show the quality function $Q_s(m)$ over motor space for a fixed sensory controller input $s$. In motor space, both the bold and dashed area are sampled randomly, but only examples within the bold area on the X-axis are included in the training set.

learning examples, but to include every learning example in the training set whose quality exceeds a certain threshold $\hat{Q}_s$ depending on the initial sensory state $s$. One obtains learning examples by sampling randomly in sensory and motor space. Whenever $Q_s(m)$ is larger than $\hat{Q}_s$, the example $(s, m)$ is included in the training set.

In the training, these imperfect learning examples are averaged. For this purpose, one needs a controller which adapts its output to the average of different motor commands corresponding to similar sensory inputs. A widely used architecture with such learning properties is the multi-layer perceptron. If the learning is successful, $Q_s(m_C)$ ($m_C$: Controller output) is close to maximum quality $Q_s^{\text{max}}$ for all $s$ (see Fig. 2, left).

The success of learning depends on the properties of $Q_s(m)$. Whenever $Q_s(m)$ is unimodal and symmetric for all $s$, success is guaranteed. The more $Q_s(m)$ deviates from this perfect form, the larger the expected difference between $Q_s(m_C)$ and $Q_s^{\text{max}}$.

2.3. Staged Learning

If $Q_s(m)$ is not unimodal and symmetric for all $s$, or if the number of learning examples is too small for proper averaging, the controller $C$ that evolved from learning by averaging will not be precise enough. In this case, we suggest “Staged learning” as additional procedure. Let us consider the general case where we have a controller $C_k$ trained in stage $k$. This controller produces output with a certain quality $Q_s(m_{C_k})$. For the generation of a new set of learning examples, sensory space is sampled randomly again. For each encountered sensory state $s$, a corresponding motor
command is generated by adding noise to the controller output $m_{C_k}(s)$:
$m = m_{C_k}(s) + \eta_k(m_{C_k}(s))$. Again, the example $(s, m)$ is included in the training set only if the quality of the motor command $Q_s(m)$ exceeds a certain threshold $\bar{Q}_{s,k}$. In staged learning, this threshold is computed by $\bar{Q}_{s,k} = Q_s(m_{C_k})$ (see Fig. 2, right). It is often useful to approximate $\bar{Q}_{s,k}$ because determining $Q_s(m_{C_k})$ requires the controller output $m_{C_k}$ and the execution of the respective movement.

Based on controller $C_k$, a training set $T_{k+1}$ is collected for stage $k+1$, and using $T_{k+1}$, a controller $C_{k+1}$ is trained. By repeating this procedure, controller performance increases from stage to stage. When staged learning is combined with learning by averaging as suggested here, the controller of the first stage evolves as depicted in Section 2.2. Moreover, in controller training the collected learning examples are averaged.

Staged learning in combination with learning by averaging works fine, whenever the quality function $Q_s(m)$ is unimodal for all $s$. The more $Q_s(m)$ deviates from this ideal form, the more difficult staged learning will be (but still may work). With a well-chosen noise function $\eta_k(m_{C_k}(s))$, the great advantage of staged learning is the significant reduction of search effort for good learning examples as we will show for saccade control.

3. Saccade Controller

3.1. Controller Design

Our robot camera system consists of two cameras mounted on a single “pan-tilt unit” (PTU) with a horizontal (pan) and a vertical (tilt) rotation axis. The cameras are mounted around 85 cm above a table surface with a size of $80 \times 80$ cm. The distance between the optical axes of the cameras is approximately 20 cm. The task of the proposed saccade controller is to fixate interesting objects (colored wooden blocks) on a table surface in a way that the respective target object is centered in both camera images. In this study, 42 different wooden blocks were scattered over the table surface. The arrangement of blocks differed between controller training and testing.

Two motor parameters (pan and tilt) are controllable via hardware. A third parameter, the vergence angle between both cameras, determines the horizontal offset of two regions extracted from the camera images. Altogether, the saccade controller has three motor outputs: pan, tilt, and vergence. All these values are scaled to the range $[-1.0; +1.0]$. In pan and tilt direction, this range corresponds to just under 60 degrees.

Figure 3 shows the overall control scheme. The input of the controller
Figure 3. Overview of the saccade controller (see text for details).

is both kinesthetic and visual. The kinesthetic part consists of the current motor state in time step $t$ ($\text{pan}_t$, $\text{tilt}_t$, and $\text{verg}_t$). The visual input is reduced to three values: $x_l$, $x_r$, and $y_{lr}$. These denote the position of the current target object in both the left and right "eye". Thus, altogether there are six input values (also scaled to the range $[-1.0; +1.0]$).

Image processing takes the current camera images and the current fixation goal as input. The fixation goal determines which objects are most salient, depending on their size and color. In the first step of image processing, quadratic regions are cut out from the camera images and subsampled to $55 \times 55$ pixels (corresponding to a diagonal angle of view of 56 degrees). Afterwards, the most salient object on the table surface is identified as target object. The position of its center in both the left and right subsampled image is determined.

3.2. Controller Training

Controller training follows the procedure of staged learning as outlined in Section 2.3. Each learning example in the training set is generated in the following way: First, a random motor command is generated and executed (consisting of $\text{pan}_t$, $\text{tilt}_t$, and $\text{verg}_t$). Then, one of the wooden blocks on the table surface within the visual input of the controller system is randomly selected as target object. The image coordinates of its center are computed for both eyes (resulting in $x_l$, $x_r$, and $y_{lr}$). In this way, the complete controller input is determined. For output generation ($\text{pan}_{t+1}$, $\text{tilt}_{t+1}$, $\text{verg}_{t+1}$), one has to distinguish two cases. In stage one, no controller is available. The output is just a random variation of the initial motor position. From stage two on, first the output of the already existing controller from the last stage is determined ($\text{pan}^*_{t+1}$, $\text{tilt}^*_{t+1}$, $\text{verg}^*_{t+1}$). Then, the vector $[(\text{pan}^*_{t+1}, \text{tilt}^*_{t+1}) - (\text{pan}_t, \text{tilt}_t)]$ is scaled with a random value around $y_{lr}$.

*Due to the geometry of the camera head, $y_l$ and $y_r$ are always equal; thus, they are represented by one value $y_{lr}$.
Figure 4. Points illustrate the performance of the first-stage controller resulting from learning by averaging. 200 random trials were carried out. The bold curve is a quadratic function fitting the functional relationship between the pre- and post-saccadic radial target distance (combined value for the left and right eye).

1.0. This scaled vector and random noise are added to \((\text{pan}^*_t + 1, \text{tilt}^*_t + 1)\), resulting in the final endpoint \((\text{pan}_{t+1}, \text{tilt}_{t+1})\). \(\text{verg}_{t+1}\) is determined by adding noise to \(\text{verg}^*_t\). All types of noise are uniformly distributed, and noise amplitude is reduced from stage to stage. The exact definition of the noise function \(\eta_k(m_{C_k}(s))\) does not seem to be critical for our specification of saccade control as long as the noise level decreases in each stage.

For the saccade controller, qualities and quality thresholds are always computed in pairs for the left and right side \((Q^l, Q^r)\), in the following we will refer to both values as \(Q^{l/r}\). Each random movement is only included as learning example in the training set if the quality of the movement is larger than a certain quality threshold on both sides. The quality \(Q^{l/r}_t(m_t)\) of movement \(m_t\) in sensory state \(s_t\) is the negative of the radial target distance \(r_{l/r}\) on the left/right side for the resulting sensory state \(s_{t+1}\). \(r_{l/r}\) is computed as the Euclidean distance between the center of the subsampled input images and the target object coordinates in these images. \(r_{l/r}\) is scaled to the range \([0.0; +1.0]\). For the first stage, the quality threshold \(\bar{Q}^{l/r}_t\) for sensory state \(s_t\) is the negative of \(r_{l/r}\). Acceptable estimates of the correct motor command are obtained by averaging between both under-

\(^b\)This quality function is unimodal over motor space for every sensory state.
and overshoot examples included in the resulting training set.

From stage two on, the quality threshold $Q^{l/r}_{s,k}$ is not exactly computed as $Q^{l/r}_{s,k}(m_{C_k})$. Instead, $Q^{l/r}_{s,k}$ is approximated (see Section 2.3) by fitting a quadratic function to the functional relationship between the radial target distance before a saccade, and after a saccade carried out by controller $C_k$ (fitting is based on a combined radial target distance value for the left and right side, see Fig. 4). $Q^{l/r}_{s,k}$ is the negative of this fitted function. The data for the fitting procedure is collected by carrying out 200 random trials with controller $C_k$. This kind of approximation saves implementational and computational effort. The quadratic function was chosen for the fitting procedure since it captures the relationship between the pre- and post-saccadic radial target distance sufficiently well.

In each stage, 10,000 learning examples were collected for training. As controller network, a multi-layer perceptron with 40 hidden units was used since this network architecture is well suited for staged learning in combination with learning by averaging (see Section 2.2). As outlined in Section 3.1, the controller has six input units and three output units. Training was carried out with 2,000 epochs of resilient propagation [7]. To avoid outliers, in each stage five controller networks were trained. The controller with medium performance was taken as basis for the next stage.

4. Results

The left part of Table 1 shows the results for the sets of training examples. From stage to stage, the average radial target distance after each saccade in the training set decreases. In stage seven, it reaches 0.0. This means, that the respective training set consists exclusively of perfect learning examples, i.e. perfect saccades resulting in optimum fixation after only one move.

On the other hand, the average number of required random movements for the collection of one learning example increases from 4.2 to 19.1. Thus, the higher the performance level already reached, the more difficult it is to find even better learning examples. Altogether, in the overall learning history, for each of the perfect examples in the training set of stage seven, around 60 random movements had to be carried out.

To assess controller performance, first we tested fixation success (right part of Table 1). In each trial, a random motor starting position and a

---

*Resilient propagation needs significantly less epochs for training than standard backpropagation. Moreover, it has virtually no adjustable parameters and results in well-generalizing networks.*
Table 1. Results over seven stages of learning, regarding both the sets of training examples and the trained saccade controllers. The results of the controllers with medium performance for each stage are reported here.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Random movements per pattern</th>
<th>Training set</th>
<th>Controller</th>
<th>Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Fixation</td>
<td>One-step</td>
<td></td>
</tr>
<tr>
<td>One-step</td>
<td>After</td>
<td>Average</td>
<td>Percent.</td>
<td>Average</td>
</tr>
<tr>
<td></td>
<td></td>
<td>one saccade</td>
<td>successful</td>
<td>needed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>number of</td>
<td>trials</td>
<td>saccades</td>
</tr>
<tr>
<td>1</td>
<td>4.2</td>
<td>0.38 / 0.38</td>
<td>11.2 (5.05)</td>
<td>36 %</td>
</tr>
<tr>
<td>2</td>
<td>3.7</td>
<td>0.22 / 0.21</td>
<td>6.98 (3.99)</td>
<td>48 %</td>
</tr>
<tr>
<td>3</td>
<td>4.8</td>
<td>0.17 / 0.17</td>
<td>7.89 (5.11)</td>
<td>53 %</td>
</tr>
<tr>
<td>4</td>
<td>5.7</td>
<td>0.13 / 0.13</td>
<td>5.84 (4.04)</td>
<td>84 %</td>
</tr>
<tr>
<td>5</td>
<td>11.8</td>
<td>0.07 / 0.07</td>
<td>4.49 (3.88)</td>
<td>84 %</td>
</tr>
<tr>
<td>6</td>
<td>11.6</td>
<td>0.03 / 0.03</td>
<td>4.60 (3.56)</td>
<td>76 %</td>
</tr>
<tr>
<td>7</td>
<td>19.1</td>
<td>0.0 / 0.0</td>
<td>3.49 (3.01)</td>
<td>99 %</td>
</tr>
</tbody>
</table>

random target object were determined. Then, the controller carried out a series of saccades towards the target object until it was successfully fixated in the center pixel in both the left and right subsampled input image (this corresponds to a precision of 0.033% of the overall image area). The results in Table 1 are obtained from 200 fixation trials. The maximum number of saccades within one trial was restricted to 20. When this number was exceeded, the trial was counted as non-successful. The percentage of successful fixation trials increases from 36% in stage one to 99% in stage seven. The average number of needed saccades in one trial drops down from 11.2 to 3.49 (for successful trials). Moreover, we looked at the radial target distance after the first controller saccade. The average radial target distance decreases from 0.48 in stage one to 0.06 in stage seven.

5. Discussion

The results show that staged learning is successful in the domain of saccadic eye movements. Herein, the main result is the number of 60 random movements needed in the overall learning history for one of the perfect learning examples in the final training set. When searching from scratch for a single perfect example, around 60,000 movements would be required.

\[d\]

The number of 60,000 random movements was determined by building up a training set as described in Section 3.2 for stage one. The quality threshold was set to a constant...
instead. Thus, with staged learning, there is a huge gain in the speed of building up a very good training set. We expect that this gain will be the larger the higher the dimensionality of sensorimotor space.

Moreover, the missing teacher signal is replaced by a quality function over sensorimotor space. Here applies the restriction that ideally, this quality function is unimodal. In our ongoing work, we want to show that it is possible to define such quality functions for many different motor tasks, not only for saccade control. In this paper, we considered staged learning always in combination with learning by averaging. Actually, we think that staged learning will also work without averaging, e.g. in combination with recurrent neural networks. With such controller architectures, the quality function is not restricted to a unimodal form any longer.

If further research as outlined above is successful, staged learning will be a more general approach for the learning of controllers (and inverse models) than the procedures described in Section 2.1. Saccade control provides a promising start as shown in this work. Moreover, staged learning could be used to model human learning in many areas of motor control.

References


value (0.01) which allowed only perfect saccades to be included in the training set.