Fast learning mapping schemes for robotic hand-eye coordination

Hülse, Martin Siegfried; McBride, Sebastian Daryl; Lee, Mark Howard

Published in:
Cognitive Computation

DOI:
10.1007/s12559-009-9030-y

Publication date:
2010

Citation for published version (APA):
Hülse, M. S., McBride, S. D., & Lee, M. H. (2010). Fast learning mapping schemes for robotic hand-eye coordination. *Cognitive Computation*, *2*(1), 1 - 16. https://doi.org/10.1007/s12559-009-9030-y

Biological Cybernetics manuscript No.
(will be inserted by the editor)

Fast learning mapping schemes for robotic hand-eye coordination

Martin Hülse · Sebastian McBride · Mark Lee

Received: date / Accepted: date

Abstract In aiming for advanced robotic systems that autonomously and permanently readapt to changing and uncertain environments, we introduce a scheme of fast learning and readaptation of robotic sensorimotor mappings based on biological mechanisms underpinning the development and maintenance of accurate human reaching. The study presents a range of experiments, using two distinct computational architectures, on both learning and realignment of robotic hand-eye coordination. Analysis of the results provide insights into the putative parameters and mechanisms required for fast readaptation and generalization from both a robotic and biological perspective.

1 Introduction

Cross-modal development describes the integration of sensory systems to create a unified view of the sensory world [23]. The transition from inaccurate pre-reaching to accurate reaching observed in infants [22] is considered to be an example of this phenomenon, whereby links are established between the different coordinate reference frames of hand and eye. Once established, the relationship between these mapping systems also has the ability to change, for example, as the hand-eye relationship alters during child growth [2], suggesting long-term plastic attributes of the system to facilitate adaptive processes. Investigating the mechanism underlying these mapping and remapping processes may confer advantage from both a robotic and biological perspective. Robotically, it may provide a highly efficient (low learning cycle) method of initial mapping between hand and eye as well as additional adaptive remapping strategies where required. Biologically, it may identify a putative developmental mechanism that explains how this change from pre-reaching to reaching takes place as well as highlighting additional features of the remapping process. This study assesses what is currently known about the biological mechanism and from that derives a robotic model which is subsequently validated.

Reaching can be executed in many different situational and environmental contexts but, as a general rule, it occurs immediately after saccade to the target object and the subsequent determination of the object location in eye-centered coordinates. It has previously been considered that a common reference frame between hand and eye is required to derive a movement vector, which can be achieved by either a) coding both the target and hand in body-centered coordinates or b) coding hand position in eye-centered coordinates. Using single neuron cell recordings within the part of the brain responsible for reach (posterior parietal cortex) during various reach paradigms, data has suggested a common coding with regard to the latter [4]. However, the mechanism may also contain greater flexibility than this and other studies have suggested that different modalities (visual, proprioceptive, auditory) have their own spatial maps and that the predominating reference frame can change depending on the context [2,23,16]. From a developmental perspective, early learning to reach is associated with error of target location. This occurs up to 3-5 months of age and is often referred to as pre-reaching [22]. Thereafter, it is considered that infants have a ’unified coding system within which visual, auditory and proprioceptive stimulation is integrated’ to facilitate the reach process [6]. What appears to be important about this transition from inaccurate to accurate reaching is

This work was supported by EU-FP7 projects IM-CLeVeR and ROSSI, and by EPSRC, UK through grant EP/C516303/1.

M. Hülse
Dept. Computer Science, Aberystwyth University, Penglais, SY23 3DB, Wales, UK
Tel.: +44-1970-622441
Fax: +44-1970-628536
E-mail: msh@aber.ac.uk

that it is not dependent on visual guidance and that after visual or even auditory location of the target, proprioceptive information about hand position is sufficient to attain an accurate reaching action [6]. However, it is also known that the onset of reaching in blind infants in response to auditory cues is delayed (8-11 months compared to 3-4 months in sighted infants) [9], which may suggest that there is a predominant role of vision in the initial development of a common mapping system between modalities. In this context, the inability to accurately determine the location of a target during the prereaching stage may be the result of 1) inaccurate proprioceptive information about hand position, 2) inaccurate transformation of this data within the common reference frame (i.e. poor mapping between one reference frame and the other) and/or 3) inaccurate motor commands to realize the derived movement vector. Although proprioceptive development starts prenatally [17], predominant implementation of this system concurs with the onset of accurate reaching [20], thus this may account for some of the initial pre-reaching motor error. Early development is also associated with immense dampening and fine tuning of motor action [3] which may also explain some of the transition to accurate reaching. Relatively less work has been carried out in the second factor listed above, the developmental proprioceptive and visual feedback mechanisms that log correct and incorrect kinematic sequences at the point of accurate or inaccurate reach to target respectively. Research by Thelen et al. (see [19] for review) has suggested that this initial mapping by infants occurs through a ’trial and error’ process where a wide range of movement parameters and solutions in different contexts and modalities are explored over time in order to calibrate reach movements. For example, mapping between modalities could occur when 1) an object is touched and then saccaded to, 2) when an object is located through visual or auditory stimuli and then touched or 3) when an object is placed and then saccaded to. Biologically, a calibration process that could account for all of these different modalities and situations is still unknown. One possible method is a simple mapping strategy that links the location of the object identified through one modality with another. This is referred to here as a learning scheme for cross-modal mapping and the first aim of this study was to examine this strategy (using the aforementioned example 3 [known proprioceptive location through object placement mapped to eye-coordinates through saccade]), from the perspective of robotic efficiency (rate of learning) but also to critically assess it as a putative biological mechanism underlying the phenomenon of cross-modal mapping.

In parallel to developmental mapping, adults can also be forced to make mapping adjustments within prism and force field experiments. This readaptation is obviously in the context of an established visuo-motor map and, thus, although it is sometimes referred to as early learning, it is probably not an accurate representation of of the aforementioned developmental mapping but rather a secondary adaptive remapping process more in line with the changes in eye-hand relationship that take place during childhood growth [2]. Redding and Wallace [16] have postulated that adaptation can be divided into two separate components, calibration and alignment. Calibration is the result of cognitive learning processes and can operate through a range of available modalities in order to make the required short-term strategic adjustments at the time of the task. Alignment describes the relationship of the various modality-specific and common topographical reference maps where the relationships between these maps can change as a result of perceptual discrepancies (perceptual learning). The latter is considered to be the primary process that occurs during childhood growth as a result of slow and unperceived changes in the relationship between hand and eye [2]. An interesting and potentially useful characteristic of alignment from a robotic perspective is that it appears to generalize across space i.e. a global shift in the relationship between maps occurs as a result of perceptual learning at one point [2, 16]. Alignment (or realignment), may therefore be an extremely efficient method of readaptation whereby not all points in space have to be relearnt if the arm-eye relationship changes. The second aim of this study, therefore, was to investigate realignment as a potential form of readaptation and in particular to identify what factors need to be characterized as part of the implementation process.

2 Robot systems, control and realignment test

2.1 Introduction

The robot hand-arm system, the active vision system and their spatial organization on and around a table can be seen in Figure 1. The arm and active vision systems operate independently but are governed by a central unit which we describe later in detail. Hence, both sub-systems are without any direct connection or have any access to a shared world model.

Developing robust hand-eye coordination requires that the sensorimotor mapping which represents the relation between object position (known by the arm, but unknown by the vision system) and tilt-verge configuration of the vision system (unknown by the arm) is somehow learnt. Both, the location of the object on the table and the tilt-verge configuration are the interaction outcomes of the two independently working sensorimotor systems or modalities (arm, active vision system). The learning outcome should enable the complete system to “reach and grasp where it looks” and to “look where it reaches to.” In other words, the system needs to learn its hand-eye coordination based on already established but “unconnected” reach- and gaze-control.

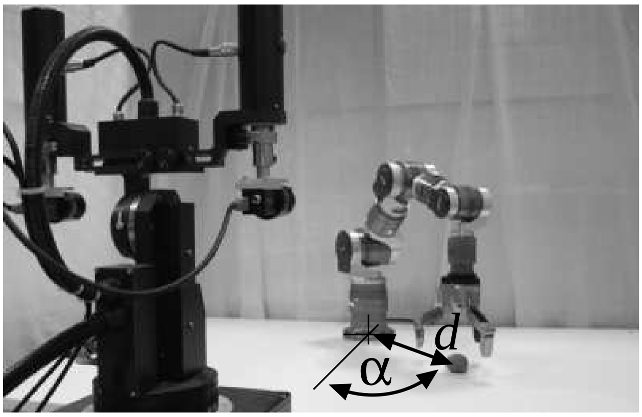


Fig. 1 Active vision and hand-arm system.

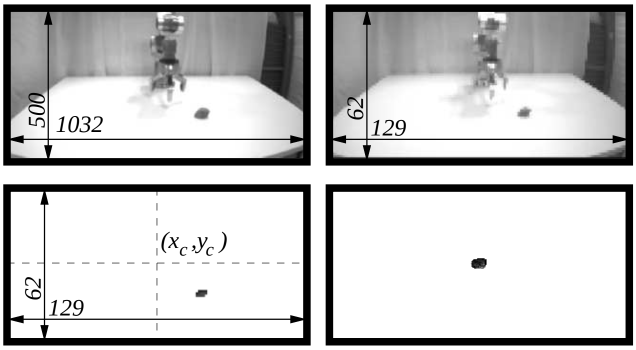


Fig. 2 Example for the reduction (top right) of the original image (top left) and the resulting filtered image date, before (bottom left) and after triggering gaze-control (bottom right).

2.2 Vision and hand-arm system

The active vision system integrates two cameras (both provide RGB image data, 1032x778, 25 frames per second) mounted on a pan-tilt-verge unit. In this experiment we didn't use the pan movement. Hence, the active vision system has 3 degrees of freedom (DOF), that is, one verge movement for each camera and one tilt which moves both cameras. Each motor can be controlled by determining the values for speed or position, given in radians (*rad*).

The robot arm and hand systems (SCHUNK GmbH & Co. KG) have 7 DOF each. We make use of only five DOF of the arm in order to place the robot hand at certain positions on the table. The hand system has three fingers. All fingers have two segments each equipped with a pressure sensitive sensor pad. Since the control of the grasping is out of the scope of this paper we won't give any further details about the hand system and its control.

2.3 Reaching and gaze-control

The domain of the reach movement, referred to here as *reach space*, is represented as a 2-dimensional polar coordinate system because the objects are only located on a table, a 2-dimensional space. Taking the base of the arm as reference, a table location is fully determined by the distance d (cm) and the planar angle of the arm α (rad) (see Fig. 1). The inverse kinematics mapping between the 2-dimensional reach space and the 5-dimensional joint space of the arm is solved analytically which won't be described further. It is important to note that arm-control only places the hand on the table with respect to a given distance and relative angle (d, α). The actual table space the system is operating in is defined by the range of distance d and angle α, here we have: $-1.4 \leq \alpha \leq 1.4 (rad)$ and $30 \leq d \leq 60 (cm)$ spanning an area of 3944 cm^2.

The purpose of the gaze-control is to move the cameras in such a way that the visual stimuli, a colored ball, will be driven into the image center. To achieve reasonable performance,RGB image data of resolution 1032x500 were captured and reduced to 129x62. Each RGB color value in the reduced image represents the mean value of the color values in the corresponding 8x8 sub-field in the original image. The reduced RGB images were filtered with respect to a defined color, here blue. Non-zero values (grey pixels) in the filtered data indicate the appearance of the filtered color in the image. The x and y positions of the non-zero values in relation to the image center (x_c, y_c), relate to specific speed values s_x and s_y. These values are used to specify the speed values of the motors controlling verge and tilt (Fig. 2). The relation between x and y and speed values is linear:

$$s_x := \frac{2(x - x_c)}{2x_c},$$

$$s_y := \frac{2(y - y_c)}{2y_c}$$

where $2x_c$ and $2y_c$ are the horizontal and vertical resolutions of filtered image, respectively. The actual speed values of the verge v_x and tilt v_y motors are calculated as follows:

$$v_x := c_x \cdot \overline{s_x},$$

$$v_y := c_y \cdot \overline{s_y}$$

where $c_{x,y}$ are constants for normalization while $\overline{s_x}$ and $\overline{s_y}$ represent the mean values of all non-zero values of s_x and s_y values in the color filtered image data, respectively.

In order to avoid conflicts between different tilt values resulting from the different visual input of left and right camera, the left camera controls two DOF (its verge and the tilt) whilst the right camera determines only its own verge movement. The consequence of this is that the right camera cannot always drive the visual stimuli completely into its center.

The signals coming from the gaze control drive the motors of the tilt and verge axis directly until the stimulus is shifted into the image center. At this point, the motor signals

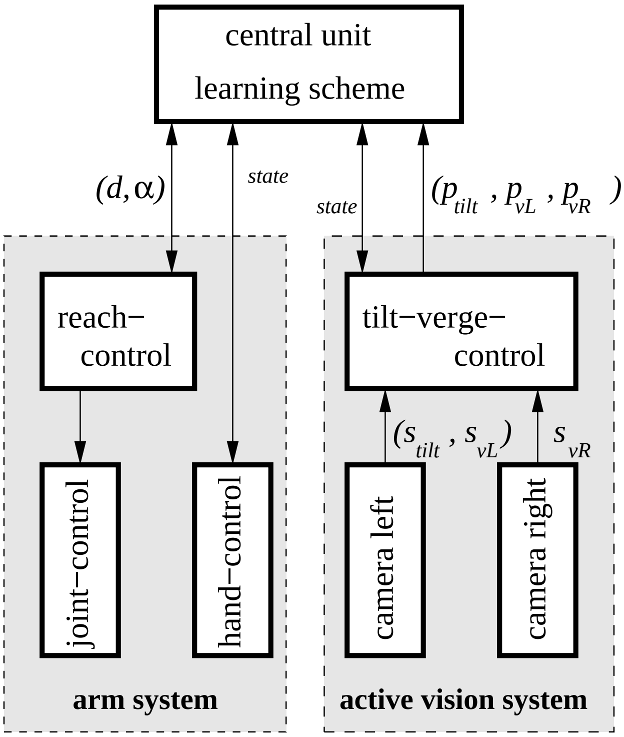


Fig. 3 System architecture

become zero and the system comes to a standstill indicating that it has focused on the stimulus. Such a halt position is fully determined by the motor positions of the tilt, left and right verge axis, $(p_{tilt}, p_{vL}, p_{vR})$, and is referred to here as the *vision space*.

2.4 Overall system architecture and learning substrate

Figure 3 illustrates the general system architecture which combines the arm and the active vision systems. Each system acts independently and thus can be seen as separate sensorimotor systems. Coordination between them is achieved by the central unit which provides the substrate for learning the relation between arm and vision system. In the case of the arm system, the central unit can set target coordinates (d, α) in the reach space which triggers specific hand movements. In addition, it can request state information, for example, the current hand states or whether the target position has been reached. Regarding the active vision system the central unit switches on and off the gaze-control and can read and set positions of the motors driving the tilt, left and right verge axis, $(p_{tilt}, p_{vL}, p_{vR})$.

Bridging the arm and vision system, the central unit establishes hand-eye coordination by learning the relation between the reach and vision space resulting from the interaction of the vision and arm systems in their shared environment, the table.

Learning the cross-modal mapping between two spaces is provided by a form of case-based learning strategy. Assuming two spaces $X \subseteq R^n$ and $Y \subseteq R^m$ of arbitrary dimension, where

$$x = (x_1, x_2, \dots, x_n) \in X$$

and

$$y = (y_1, y_2, \dots, y_m) \in Y.$$

A mapping $\mathscr{M}$ stores the pairs $[x^t, y^t]$ representing concrete examples at time t indicating how one point in one space X is related to space Y. This is referred to as a *link*. Since there is direct coding between links we also have an additional property of bi-directionality i.e. x^t refers to y^t and vice versa. This learning scheme is inspired by a previous methodology [15] used to learn the sensorimotor mapping for saccadic eye movements in a robot system.

It is important to note at this stage that, to achieve a standardized relationship between X and Y, a metric is required for both spaces. Stored links within the mapping are unlikely to occur again, thus, a definition of distance (i.e. a metric) between the points in space is required to allow a search for the ’closest neighbor’ stored in the mapping (where ’closest neighbor’ leads to the best estimation of the corresponding point the mapping can provide).

For each space in a mapping a different metric can be applied. Within the system described here, the distance measure in the reach space is based on a transformation from polar coordinate system to the 2-dimensional Euclidean space representing the table. This measure ensures that the distance between two points in reach space represents the actual distance on the table. In the vision space, however, it is harder to derive *a priori* a metric which matches with the actual distance in table domain. Moreover, because it is the simplest and most commonly used metric, all three dimensions of the vision space were represented by the same physical dimension, radians.

3 Two computational architectures for learning and realignment

In the following, we introduce the two computational architectures. Both provide the facility to learn the robotic hand-eye coordination and also its re-adaption or realignment if the physical hand-eye configuration or other external entities change.

In the first architecture $A_{\mathscr{R}}$ (Fig. 4) vision and reach space are directly coupled by mapping $\mathscr{R}$. Starting with the fixation of an object on the table, the gaze-control delivers a concrete point in vision space, $(p_{tilt}, p_{vL}, p_{vR})$. For this point, the mapping $\mathscr{R}$ provides an estimation in reach space (d^E, α^E) which determines the target for the next reach action. Once the robot arm has reached the target coordinates, it executes the grasping routine, i.e. it picks up the object.

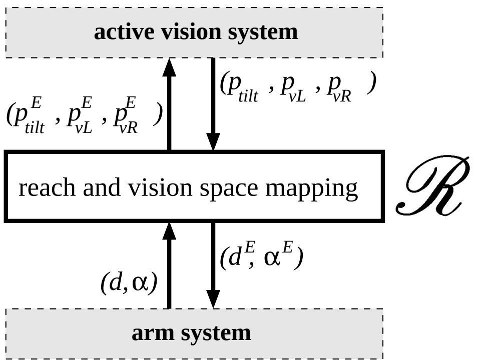

active vision system
$(p^E_{tilt}, p^E_{vL}, p^E_{vR})$
$(p_{tilt}, p_{vL}, p_{vR})$
reach and vision space mapping
$\mathscr{R}$
(d,α)
(d^E, α^E)
arm system

Fig. 4 Computational architecture $A_{\mathscr{R}}$ which directly links vision and reach space.

In other words the robot picks up objects which the vision system has previously saccaded to.

Because of the bi-directionality of the mapping, the arm system can also guide the vision system. If the arm places an object on the table then the corresponding coordinates (d,α) in the reach space can be used to generate an estimation of saccade $(p^E_{tilt}, p^E_{vL}, p^E_{vR})$. This point in vision space tells the vision system where to "look" for the object. Hence, saccadic eye movements can be modulated proprioceptively by the arm system. In the following, we will use symbol $\mathscr{R}$ as a general reference for mappings linking reach and vision space directly. Links of such mappings are written as:

$$[(d,\alpha),(p_{tilt}, p_{vL}, p_{vR})].$$

With regard to realignment, whereby the spatial relation between active vision and arm system has altered, the complete mapping $\mathscr{R}$ must be re-learned. In an attempt to overcome the problem, a second architecture is introduced which allows the system to learn the shift in visual space only and thus compensate for the changed relationship between reach and vision space. This architecture, referred to as $A_{\mathscr{S}}$, is based on the already learned and fixed sensorimotor mapping $\mathscr{R}$, but works in conjunction with a second mapping $\mathscr{S}$ which provides the substrate to learn the relative shift in the vision space.

The second architecture $A_{\mathscr{S}}$ is illustrated in Figure 5, it contains the two mappings, $\mathscr{S}$ and $\mathscr{R}$ where the latter $\mathscr{R}$ remains unchanged (grey box). Mapping $\mathscr{S}$ (white box) links absolute tilt-verge values with tilt-verge offsets values and thus essentially maps between two 3-dimensional spaces. The corresponding links are written as:

$$[(p_{tilt}, p_{vL}, p_{vR}),(\Delta p_{tilt}, \Delta p_{vL}, \Delta p_{vR})].$$

The summation of the tilt-verge values representing the current configuration of the active vision system and the offset values, determine a new point P in the vision space. P is the point in $\mathscr{R}$ which relates to the point in reach space representing an estimate of the required reach position (d^E, α^E).

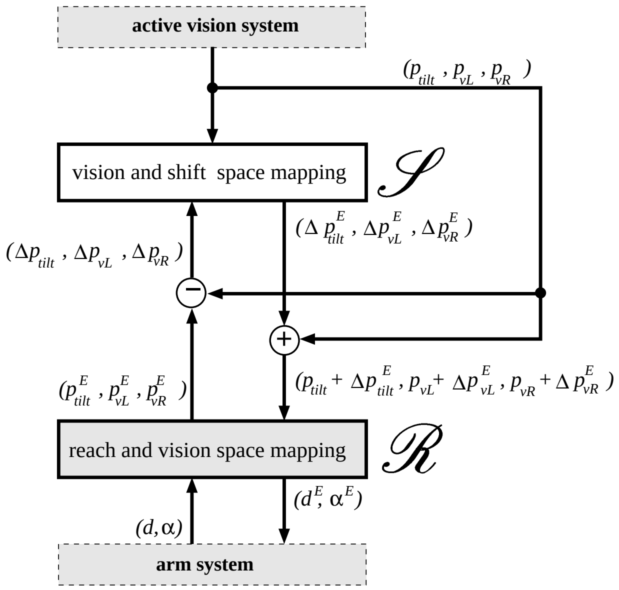

active vision system
$(p_{tilt}, p_{vL}, p_{vR})$
vision and shift space mapping
$\mathscr{S}$
$(\Delta p^E_{tilt}, \Delta p^E_{vL}, \Delta p^E_{vR})$
$(\Delta p_{tilt}, \Delta p_{vL}, \Delta p_{vR})$
−
+
$(p^E_{tilt}, p^E_{vL}, p^E_{vR})$
$(p_{tilt} + \Delta p^E_{tilt}, p_{vL} + \Delta p^E_{vL}, p_{vR} + \Delta p^E_{vR})$
reach and vision space mapping
$\mathscr{R}$
(d^E, α^E)
(d,α)
arm system

Fig. 5 Computational architecture $A_{\mathscr{S}}$ which learns the alteration of the vision and arm system configuration via the relative shift in the visual space.

The assumption behind this architecture is that the offsets in the vision space which are needed to compensate changes in arm-vision configuration are highly structured and thus potentially constant across space. Theoretically therefore, mapping $\mathscr{S}$ will contain fewer links compared with $\mathscr{R}$ and therefore realignment for $A_{\mathscr{S}}$ would be expected to be much faster with fewer learning examples required.

The general data flow of this architecture starts with a specific active vision system configuration $(p_{tilt}, p_{vL}, p_{vR})$. This is the input for $\mathscr{S}$ generating the estimated offset for the three vision system components tilt, verge left and verge right:

$$(\Delta p^E_{tilt}, \Delta p^E_{vL}, \Delta p^E_{vR}).$$

The sum of offset and current configuration

$$(p_{tilt} + \Delta p^E_{tilt}, p_{vL} + \Delta p^E_{vL}, p_{vR} + \Delta p^E_{vR}),$$

is the input for the fixed mapping $\mathscr{R}$, leading to the final position estimation in the reach space.

The links for mapping $\mathscr{S}$, however, are derived from the current absolute tilt-verge configuration $(p_{tilt}, p_{vL}, p_{vR})$ *and* the estimated absolute tilt-verge configuration $(p^E_{tilt}, p^E_{vL}, p^E_{vR})$. The latter is derived from the fixed mapping $\mathscr{R}$ in relation to the current arm position (d,α), i.e. the proprioception of the robotic system:

$$\begin{pmatrix} \Delta p_{tilt} \\ \Delta p_{vL} \\ \Delta p_{vR} \end{pmatrix} = \begin{pmatrix} p^E_{tilt} \\ p^E_{vL} \\ p^E_{vR} \end{pmatrix} - \begin{pmatrix} p_{tilt} \\ p_{vL} \\ p_{vR} \end{pmatrix}.$$

4 Schema for permanently adapting mappings

In order to enable the system to learn or to build up autonomously the mappings that link reach and vision space without human intervention, the arm system ‘presents’ to the vision system an object at a known position (i.e. proprioception) and initiates the gaze control leading to a specific tilt-verge configuration. This true or absolute link between the vision and reach space is the cornerstone for developing the mapping for both computational architectures ($A_{\mathscr{R}}$ or $A_{\mathscr{S}}$). In the following we introduce the complete protocol for the simpler architecture, $A_{\mathscr{R}}$. This protocol (see Box 1) is able to adapt to changes of the environment because it a) combines the acquisition and the evaluation of new links and b) alters or deletes links already present in the mapping depending on their age (Q).

Box 1. Protocol for learning and re-learning mappings:

```
1     Initialize mapping $\mathscr{R}$
1.1   set tolerance level $T$ and
      minimum age $Q$ of $\mathscr{R}$;

2     Pick up object
2.1   move arm to pre-defined object
      position;
2.2   pick up object;
2.3   move arm away;

3     Select a target position in reach space $(d,\alpha)$

4     Execution of reach and gaze
4.1   move the arm to the target
      position $(d,\alpha)$;
4.2   put the object on the table;
4.3   move the arm away;
4.4   start the gaze control;
4.5   wait until vision system has
      fixated the object on the table;
4.6   read the active-vision system
      configuration $(p_{tilt}, p_{vL}, p_{vR})$ ;
4.7   stop gaze control;

5     Learning
5.1   $(d^E,\alpha^E) := E(\mathscr{R}, p_{tilt}, p_{vL}, p_{vR})$;
5.2   calculate estimation error $z$, the
      distance between estimated and
      actual position $(d^E,\alpha^E)$ and $(d,\alpha)$ ;
5.3   for each link in $\mathscr{R}$ increase its
      age by 1;
5.4   if $(z \leq T)$
      then: set age of the link to 0 which
          has provided the good estimation;
      else: add new link  $[(d,\alpha),(p_{tilt}, p_{vL}, p_{vR})]$
          to  $\mathscr{R}$;
5.5   remove link in $\mathscr{R}$ with highest age
      value, if its age is larger than $Q$ ;

6     Pick up the object
6.1   move arm to the target position;
6.2   pick up object;
6.3   move arm away;

7     Go back to 3
```

In the first step of the protocol a mapping $\mathscr{R}$ is initialized. Mapping $\mathscr{R}$ can either be empty or it can already contain links. Additionally, the tolerance value T for the minimum allowable error between the estimated and actual object position and the minimum age Q of links are set. T defines the threshold which determines whether a new link needs to be added to the mapping in order to improve mapping accuracy. Estimation errors beyond this threshold are counted as wrong estimations and are removed.

After initialization the arm picks up the object from a pre-defined position on the table (step 2) and selects a target position (d,α). The selection strategy we apply guarantees equal distribution of links in reach space. This is achieved by taking into account the minimum distance to all links within the current mapping [13].

Having selected a new target position (d,α), the arm reaches to this position, places the object on the table and initiates the gaze-control (3 and 4). The execution of the gaze-control results in fixation of the object which is represented by a specific tilt-verge configuration $(p_{tilt}, p_{vL}, p_{vR})$. This process creates a new example of how vision and reach space are related and is represented in form of a link:

$$[(d,\alpha),(p_{tilt}, p_{vL}, p_{vR})].$$

At this point, however, it is not confirmed that the new link will be added to mapping $\mathscr{R}$. This occurs at step 5 where the system evaluates this new link according to the error value z (5.2) which is the distance between actual (d,α) and estimated (d^E,α^E) table position. The estimation is based on the current mapping $\mathscr{R}$ and the tilt-verge configuration $(p_{tilt}, p_{vL}, p_{vR})$, written in step 5.1 as:

$$(d^E,\alpha^E) := E(\mathscr{R}, p_{tilt}, p_{vL}, p_{vR}).$$

Hence, z indicates the current performance of $\mathscr{R}$ for this concrete example.

Having calculated the discrepancy between estimated and actual table position (z), the age value of each link in $\mathscr{R}$ is increased by 1 (step 5.3) and evaluated with respect to the allowed tolerance T (5.4). If error z is smaller than the defined tolerance, then no link is added, however, the age value of the link generating the ’good’ estimation is set back to zero. If the mapping delivers an insufficient estimation, $z > T$ then the new link $[(d,\alpha),(p_{tilt}, p_{vL}, p_{vR})]$ is added having age value zero.

After evaluating and performing the corresponding updates, the oldest link in $\mathscr{R}$ is examined. If its age value is larger than the minimum age value Q, it will be removed from $\mathscr{R}$. This process guarantees that all links are removed from the mapping which haven’t been contributing to the mapping’s performance over the last Q learning cycles.

Box 2. Protocol for testing a given mapping:

```
1     Initialize given mapping $\mathscr{R}$
2     Pick up object
3     Select a target positions in reach space $(d,\alpha)$
4     Execution of reach and gaze
5     Test
5.1   $(d^E,\alpha^E) := E(\mathscr{R}, p_{tilt}, p_{vL}, p_{vR})$;
5.2   calculate estimation error $z$, the
      distance between estimated and
      actual position $(d^E,\alpha^E)$ and $(d,\alpha)$ ;
5.3   print  $z$
6     Pick up the object
7     Go back to 3
```

At this stage the mapping $\mathscr{R}$ is updated according to the given example and its current estimation performance. The robotic system can then prepare itself for the next *learning cycle* by picking up the object (step 6) and restarting at step 3.

The protocol for a global test of a learned mapping utilizes the same protocol for learning with some alterations (Box 2). For testing the evaluation of the estimation and the mapping update, processes in step 5 are obviously not needed anymore and are therefore removed with only distance z between estimated and actual object position being calculated.

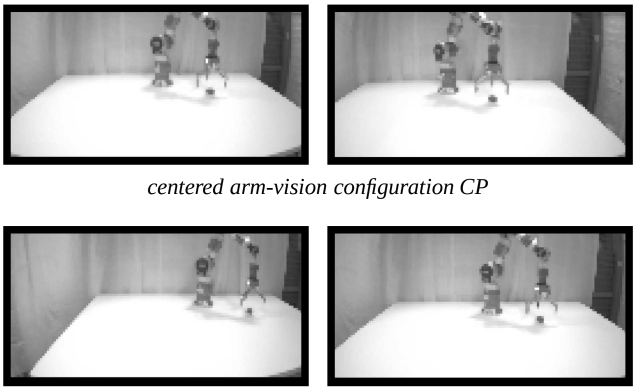
The overall protocol for architecture $A_{\mathscr{S}}$ is the same as that for $A_{\mathscr{R}}$ i.e. the evaluation of the estimated object position is performed exactly the same way. In addition, the update of mapping $\mathscr{S}$ is also exclusively determined by estimation of error value z, tolerance values T and minimum age Q. However, the estimation of the table position does need an additional step because two mappings are now involved ($\mathscr{S}$ and $\mathscr{R}$). The calculation of the links for the adapting mapping $\mathscr{S}$ also requires additional processing because it is now based on the estimated and the actual tilt-verge values (compare Fig. 5).

5 Experiments on learning and realignment

5.1 Introduction

The sequence of experiments carried out on learning (construction of an initial map) and realignment (alteration of an existing map) are presented in the following sections. The latter was achieved by moving the vision system to a different location (shift of approximately 30 *cm*). Figure 6 illustrates the difference in visual input between both positions, referred to here as *centered position CP* and *shifted position SP*. The following set of experiments were carried out:

1. architecture $A_{\mathscr{R}}$; optimization of the learning process in the context of the two variables Q (minimum age of link) and T (tolerance);
2. architecture $A_{\mathscr{R}}$; used in both the shifted and centred positions to quantitatively describe the offset in space when the eye-arm physical relationship is shifted;
3. architecture $A_{\mathscr{S}}$; integration and validation of the independently generated offset values;
4. architecture $A_{\mathscr{S}}$; learning the offset value with an empty mapping $\mathscr{S}$ and optimization of the $A_{\mathscr{S}}$ architecture in the context of the two variables Q (minimum age of link) and T (tolerance);
5. architecture $A_{\mathscr{R}}$ and $A_{\mathscr{S}}$; assessment of realignment performance for both architectures in the context of already existing mappings within the architectures and also in the context of the two variables Q (minimum age of link) and T (tolerance);



centered arm-vision configuration CP

shifted arm-vision configuration SP

Fig. 6 Visual input of left and right camera in the centered arm-vision configuration *CP* (top) and after the shift *SP* (bottom) in the starting position, i.e. before triggering gaze-control.

5.2 Learning with architecture $A_{\mathscr{R}}$ and $A_{\mathscr{S}}$

5.2.1 Introduction

The following sections detail learning (i.e. without realignment) of the two positions *CP* and *SP* using architecture $A_{\mathscr{R}}$ in order to obtain a) a baseline measure of accuracy in the context of variables Q and T and b) to allow a mathematical description of the required transformation. The latter is then used as the offset value (Δ) within the $A_{\mathscr{S}}$ architecture and this is subsequently compared, in terms of error values, to the same architecture when offset value(s) are obtained through the learning process.

5.2.2 Learning mapping $\mathscr{R}$ in $A_{\mathscr{R}}$

Learning the mapping in a specific position without featuring realignment was done for different parameter settings for capacity Q (minimum age) and tolerance T in both arm-vision configurations *CP* and *SP*. Each run had 300 learning

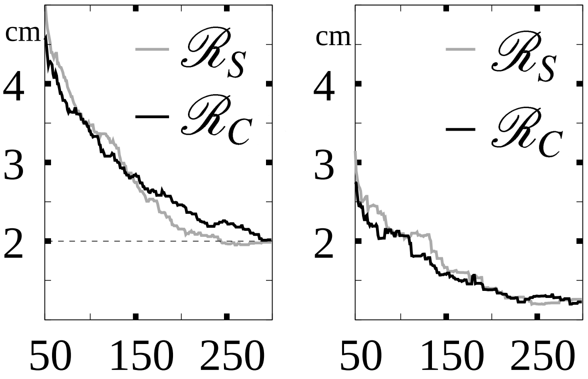


Fig. 7 Average error (left) and standard deviation (right) of the of table location (*cm*) over the learning cycles for the centered ($\mathscr{R}_C$) and the shifted position ($\mathscr{R}_S$) for architecture $A_{\mathscr{R}}$.

cycles. The results, based on an additional independent test set of 100 examples, are summarized in Table 1.

Figure 7 also presents the average and standard deviation for the run with minimal tolerance and highest meaningful minimum age ($T = 0.0, Q = 300$) for both both arm-vision configurations *CP* and *SP* (respective mappings $\mathscr{R}_C$ and $\mathscr{R}_S$). For both mappings, the lowest average error is achieved after 300 learning cycles, which means the mapping contains 300 links. This concurs with previous findings [13] where the error curve saturated between 250 and 300 learning cycles achieving a maximal accuracy of hand-eye coordination of 2.0 +/- 1.2 *cm*.

A visual representation of the relationship between vision and reach can be gained through plotting and color coding the points in reach and vision space (i.e. 2-dimensional sub-space) (Fig. 8A). The shape of the set of points in the vision space of the mapping is surprisingly similar to the actual working space of the robot arm.

Fig. 8B presents the difference bewteen the two mappings $\mathscr{R}_C$ and $\mathscr{R}_S$ in vision space (Fig. 8B). It is interesting to see their relation in the vision space and the figure suggests that a 3-dimensional translation should shift one map onto the other. This observation highly supports the original hypothesis that realignment might be provided through learning the translation in the vision space only and, that learning the parameter for this translation and representing it through the mapping $\mathscr{S}$ for architecture $A_{\mathscr{S}}$ should substantially reduce the number of links required for accurate reaching as compared to complete re-learning of the mapping in architecture $A_{\mathscr{R}}$.

5.2.3 Approximating mapping $\mathscr{S}$ in $A_{\mathscr{S}}$

Based on data provided by the two mappings $\mathscr{R}_C$ and $\mathscr{R}_S$ resulting from architecture $A_{\mathscr{R}}$, we manually derived an approximation of the offset value (Δ) for all three components $(p_{tilt}, p_{vL}, p_{vR})$ that determine the vision space.

$$\begin{aligned} p'_{tilt} &= p_{tilt} + \Delta_{tilt}(p_{tilt}), \\ p'_{vL} &= p_{vL} + \Delta_{vL}(p_{vL}), \\ p'_{vR} &= p_{vR} + \Delta_{vR}(p_{vR}), \end{aligned}$$

$\Delta_{\{tilt,vL,vR\}}(x)$ is the function determining the offset for a specific value position value $p_{\{tilt,vL,vR\}}$ and $p'_{\{tilt,vL,vR\}}$ is the shifted $\{tilt, vL, vR\}$-component of p' which is the point fed into $\mathscr{R}$ in order to get the estimation for the target coordinates in the reach space (compare Fig. 5). The generated Δ values are presented in Appendix B (Fig. 14). Three approximations of the offsets were applied, regression based on quadratic polynomial, linear function and the overall mean value. The error values achieved with these approximations are presented in Table 2. In comparison to the original data (Table 1), each offset approximation shows a drop in performance (increased error). However, all approximations were still significantly better than having no offset at all; see the “no realignment” entry in Table 2.

5.2.4 Learning the mapping $\mathscr{S}$ in $A_{\mathscr{S}}$

To investigate the capability of the system to learn the offset in the visual space, several runs were conducted systemati-

learning results for $A_{\mathscr{R}}$			error (*cm*)		
parameters *Q*	 *T*	arm-vision configuration	 avg.	 dev.	nmb. of links
300	0.0	*CP*	2.0	1.2	300
		SP	2.0	1.2	300
	2.0	*CP*	2.0	1.2	283
		SP	2.0	1.2	283
	4.0	*CP*	2.5	1.4	168
		SP	2.6	1.4	175
	6.0	*CP*	2.9	1.5	110
		SP	3.3	1.7	108
	8.0	*CP*	3.7	2.1	68
		SP	3.8	2.1	76
	10.0	*CP*	4.2	2.4	48
		SP	4.0	2.4	49
250	0.0	*CP*	2.5	1.6	250
		SP	2.2	1.4	250
200	0.0	*CP*	2.7	1.7	200
		SP	2.7	1.7	200
100	0.0	*CP*	4.1	2.7	100
		SP	4.1	2.9	100

Table 1 Average error and standard deviation of the estimation error (in *cm*) after 300 learning cycles for different learning parameter setting for architecture $A_{\mathscr{R}}$. In addition number of links in the resulting mapping are given as well.

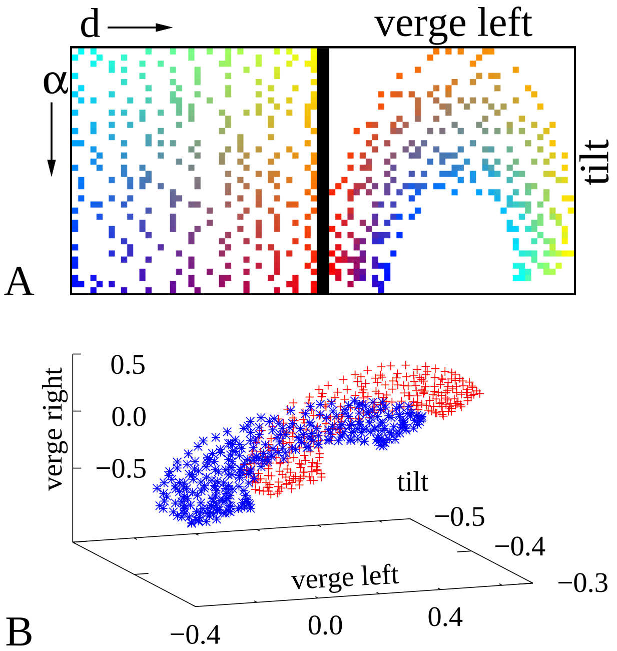


Fig. 8 **A:** Points in reach (left) vision space (right) establishing the links in $\mathscr{R}_C$. The same color indicates the same link. **B:** Points in vision space establishing the links of the learned mappings $\mathscr{R}_C$ (blue) and $\mathscr{R}_S$ (red). Both mappings are plotted in the same reference frame clearly indicating the different location in the vision space.

cally using parameter values for T (tolerance) and Q (minimum age). Since the learned mapping $\mathscr{S}$ in $A_{\mathscr{S}}$ is also determined by the underlying mapping $\mathscr{R}$ in this architecture, each parameter setting was tested in both arm-vision configurations, centered and shifted i.e. when the system learned the offsets in the centered position $\mathscr{R}_S$ was used and conversely, when the system learned the offsets in the shifted position $\mathscr{R}_C$ was used.

300 learning cycles were conducted during the learning phase of the experiment with each run starting with an empty mapping $\mathscr{S}$. During the test phase, 100 examples were used.

Learning results using low values for age (Q=10, 30, 60) and lowest possible tolerance ($T = 0.0$) are illustrated in Figure 9. One can see that the smaller the Q value, the higher the fluctuations of the average error over the whole learning process. Moreover, the final estimation performance after 300 cycles doesn't indicate any improvement compared with earlier learning cycles in these runs. Increasing Q reduced substantially the number of fluctuations but average error remained high (see Table 3). In contrast, starting with $Q = 300$ and increasing the tolerance value lead to small improvements of the average error (Table 3). Interestingly, there appeared to be an optimal value for T (8cm for both configurations) with the average error significantly smaller compared to other tolerance values.

approximation results in $A_{\mathscr{S}}$		error (*cm*) avg.	dev.
$\mathscr{S}$	quadratic	3.7	2.2
	linear	4.1	2.1
	mean value	4.5	2.2
no realignment		29.4	8.5

Table 2 Average error and standard deviation of the estimation resulting from the approximation of mapping $\mathscr{S}$ in architecture $A_{\mathscr{S}}$. For comparison error values are provided if the system is not adapting to the new configuration at all ("no realignment").

learning results for $A_{\mathscr{S}}$			error (*cm*)		
parameters Q	 T	arm-vision configuration	 avg.	 dev.	nmb. of links
100	0.0	*SP*	5.0	4.7	100
		CP	5.3	4.6	100
200		*SP*	5.3	4.9	200
		CP	5.0	4.2	200
250		*SP*	5.6	5.5	250
		CP	5.1	4.3	250
300	0.0	*SP*	6.0	5.7	300
		CP	4.6	3.8	300
	2.0	*SP*	5.7	5.6	255
		CP	4.5	3.6	248
	4.0	*SP*	5.0	5.3	156
		CP	4.8	4.2	164
	6.0	*SP*	5.0	4.2	107
		CP	4.2	3.3	70
	8.0	*SP*	3.5	1.8	21
		CP	3.6	1.7	10
	10.0	*SP*	5.5	3.0	12
		CP	4.6	2.4	6
100	8.0	*SP*	3.6	1.9	16
		CP	3.6	1.7	9
50		*SP*	3.5	1.8	12
		CP	3.6	1.7	8
40		*SP*	3.6	1.8	9
		CP	4.0	2.1	9
30		*SP*	4.0	2.2	8
		CP	4.1	2.3	7

Table 3 Average error and standard deviation of the estimation error (in *cm*) after 300 learning cycles for different learning parameter setting for architecture $A_{\mathscr{S}}$. In addition number of links in the resulting mapping are given as well.

With the tolerance value fixed at $T = 8.0$, reducing the Q resulted in a drop in estimation performance but only for Q values less than 50 (Table 3). Using these optimal T and Q values ($T = 8.0$, $Q = 50$), the learning process was assessed (in terms of average error and number of links) for both $\mathscr{S}_C$ (centered configuration *CP*) and $\mathscr{S}_S$ (shifted configuration *SP*). Both mappings are presented in Figure 10A and it is

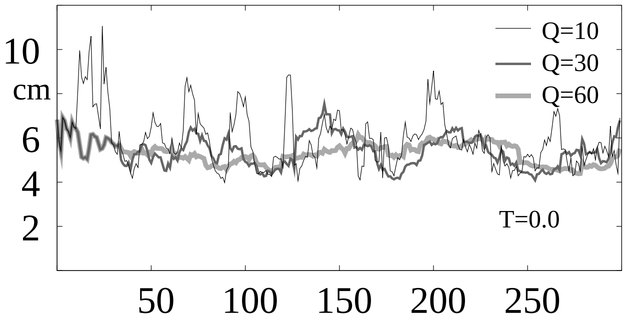


Fig. 9 Average error of different learning of architecture $A_{\mathscr{S}}$. Each run is over 300 learning cycles for different Q values while $T = 0.0$.

evident that, in both cases, the final mapping contains only a very few links (8 in $\mathscr{S}_C$ and 12 in $\mathscr{S}_S$) but that it requires between 50 and 100 learning cycles to attain this optimal arrangement. The points in visual space required to establish the links in $\mathscr{S}_C$ and $\mathscr{S}_S$ are also plotted in relation to the points in the mappings $\mathscr{R}_C$ and $\mathscr{R}_S$ (Fig. 10B, C).

5.3 Experiments on realignment

Having established optimal parameter settings for both architectures we come to the experiments on realignment. For all of the following experiments it is important to understand that the alteration of the arm-vision configuration is an external event. The robotic system doesn't have any trigger telling the process that something has changed. Internally the external change is going to manifest itself by the occurrence of poor estimations only. These poor estimation results determine specific alterations of $\mathscr{R}$ in $A_{\mathscr{R}}$ or $\mathscr{S}$ in $A_{\mathscr{S}}$, respectively. Hence, the mapping is permanently driven to self-adjustment according to the match of internal prediction and actual object position.

5.3.1 Realignment in $A_{\mathscr{R}}$

In applying architecture $A_{\mathscr{R}}$ for realignment, the same protocol as used for the experiments above were implemented except that the arm-vision system configuration changed after 200 learning cycles and there were 450 learning cycles in total over both the learning and realignment phases. Learning took place using six permutations of T and Q values (2.0, 4.0, and 6.0; 100 and 200) that were then subsequently tested using 100 test samples as before.

Figure 11 illustrates the development of average error and standard deviation of the table position estimation for the six different parameter settings. The jump to much larger average error values at learning cycle 200 indicates the shift in arm-vision relationship. Driven by the estimation errors, $\mathscr{R}$ is updated, and step by step, the system is adapted to the new relationship between reach and vision space.

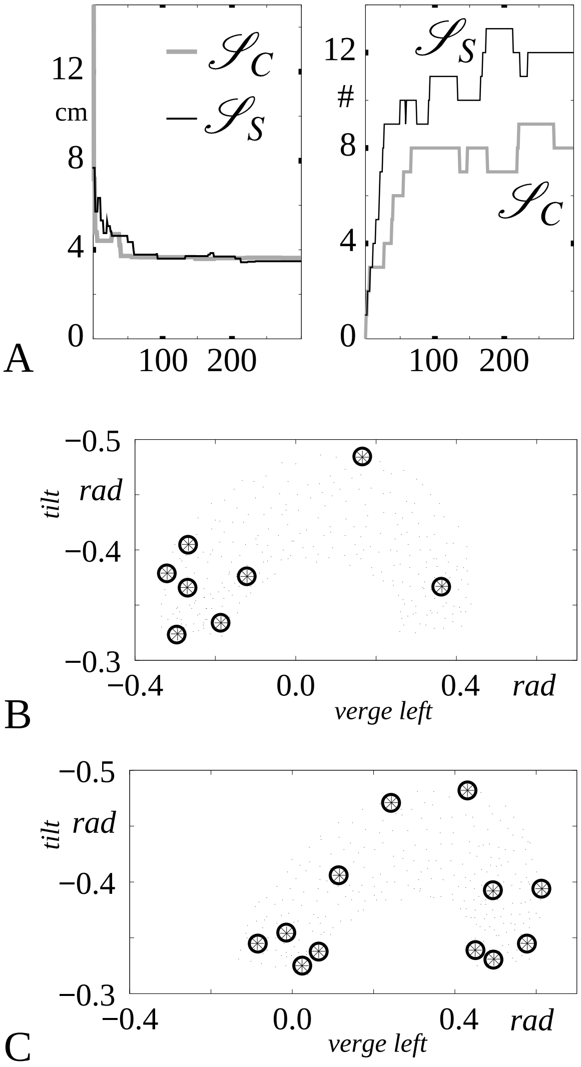


Fig. 10 **A**: Learning process of $\mathscr{S}_C$ and $\mathscr{S}_S$ ($T = 8.0\ cm, Q = 50$) indicated by the average estimation error (left) and their number of links (right) over the learning cycles. **B, C**: The points in visual space (black circles) that establish the mapping in $\mathscr{S}_C$ (**B**) and $\mathscr{S}_S$ (**C**) in relation to those points (black dots) establishing the mappings $\mathscr{R}_C$ (**B**) and $\mathscr{R}_S$ (**C**).

As previously observed, the minimum age Q and tolerance values determine the level of accuracy; lower Q values produced higher average error whilst settings for T had the opposite effect.

However, more importantly is the impact of Q on realignment. After the shift, it is apparent that the system requires exactly Q learning cycles to reattain the minimal error level. This is clearly indicated by the drop of the standard deviation at learning cycle 400 or 300 (for Q=200 and 100 respectively) despite the fact that at the point of shift (learning cycle 200) the number of links is very different for each mapping (see Fig. 12).

Tolerance and minimum age also influence the growth of links in a mapping. For example, when Q is set at 200 (see

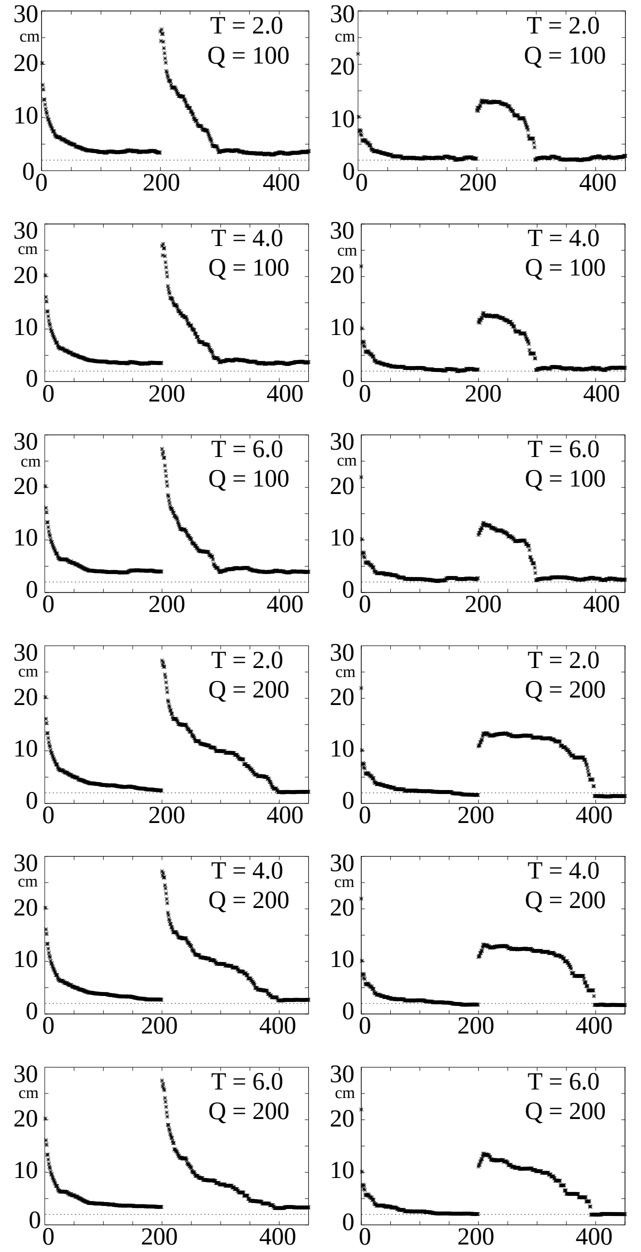


Fig. 11 Average error (left) and standard deviation (right) of the of table location (*cm*) and its evolution over 450 cycles of the re-learning protocol for different *Q* and *T* values. The arm-vision configuration was altered after 200 cycles. These values result from an additional test sets (100 samples each). The base line in the diagrams on the left indicates the error level of 2.0 *cm*.

Fig. 12) the difference in the number of links for the three T values is very distinct during the first 200 learning cycles; the larger T values producing slower increases in the number of links. In addition, comparing $Q = 100$ versus $Q = 200$ when T is set at 6, the number of links saturates between learning cycle 100 and 200 for the former but continues to increase in the latter. Hence, the Q values determines the increase of links in the mappings before and during realignment.

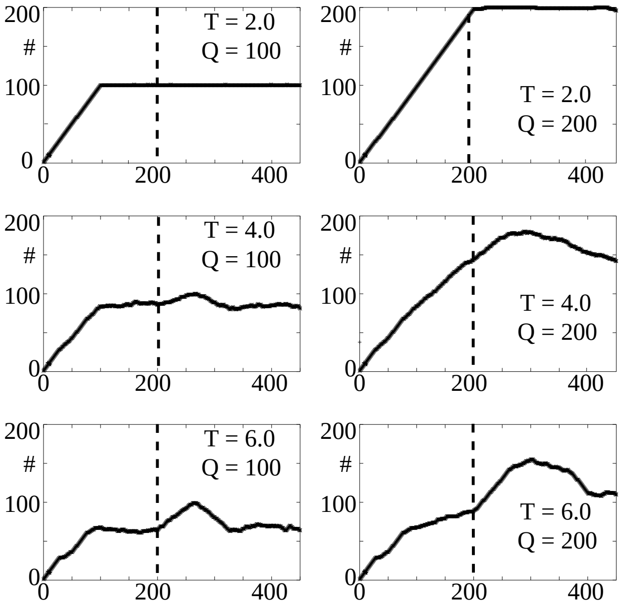


Fig. 12 Number of links and its evolution over 450 cycles of the realignment protocol for different *Q* and *T* values, same runs as in Figure 11.

5.3.2 Realignment in $A_{\mathscr{S}}$

Experiments on realignment for architecture $A_{\mathscr{S}}$ were conducted for optimal T and Q values ($T = 8.0$, $Q = 50$) derived from the experiments on learning above (average error values of ≈ 3.5 *cm*)(section 5.2.3). The system started in the centered arm-vision configuration with an empty mapping for $\mathscr{S}$ while $\mathscr{R}$ was initialized with $\mathscr{R}_C$. For 1800 learning cycles the arm-vision configuration was changed every 300 cycles with the system repeatedly altered between the same configurations: centered, shifted, centered and so forth.

The results for the average error and number of links are presented in Fig. 13. The dark grey regions in the diagram indicate the shifts in arm- vision configuration and the subsequent 50 learning cycles. The light grey regions indicate that the robot system is acting in the shifted position (*SP*) where it learns the the non-zero offsets. Whilst in the white regions, it is within the centred position (*CP*) and thus should develop a mapping generating offset values of zero since the underlying mapping is $\mathscr{R}_C$. This should, therefore, also provide an equivalent estimation performance to that previously reported for architecture $A_{\mathscr{S}}$ (2.0 +/- 1.2 *cm*). However, the results show that the system isn't able to learn a mapping with zero offset values after a shift; error values of 3.5 +/- 1.8 *cm* for both *CP* and *SP*. The plots also show that reaching this error level requires more than $Q = 50$ learning cycles, in contrast to $A_{\mathscr{R}}$. Similar results were recorded when $A_{\mathscr{S}}$ was initialized using mapping $\mathscr{R}_S$ and thus are not presented here.

Finally, the data also show that, in contrast to the $A_{\mathscr{R}}$ architecture, the number of learning cycles required to reach

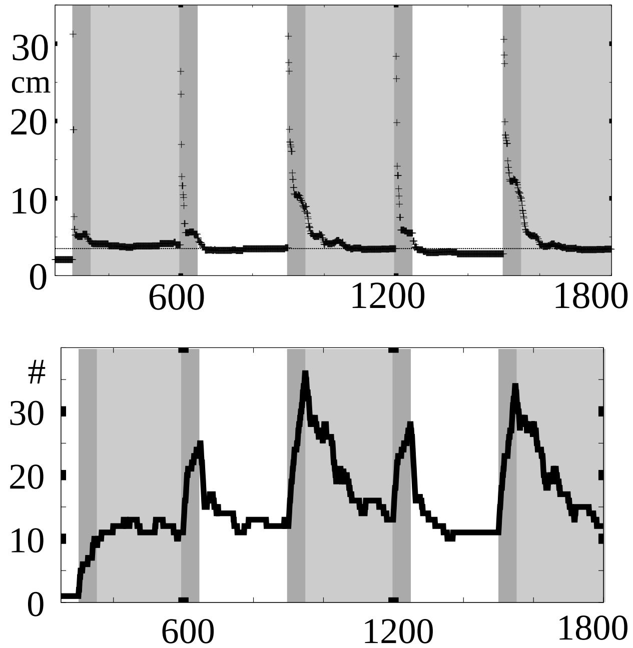


Fig. 13 Evolution of average error (top) and number of links (bottom) resulting from architecture $A_{\mathscr{S}}$ testing realignment over 1800 learning cycles for $T = 8.0$ and $Q = 50$ in the *CP* and *SP*. The base line in the top diagram indicates the value 3.5 *cm*. White regions represent the *CP*, light grey the *SP*.

minimum error is not determined by the Q value, with significantly more than 50 learning cycles required. Moreover, the number of learning cycles and the number of links required to obtain the minimum error value are substantially higher for *SP* compared to *CP* configuration.

6 Discussion

6.1 The architectures and the accumulation of uncertainty

Summarizing the results of robotic hand eye coordination with respect to the achieved error levels (Table 4), architecture $A_{\mathscr{R}}$ performed significantly better than $A_{\mathscr{S}}$ for learning and realignment. The reason that $A_{\mathscr{S}}$ did not perform as well as $A_{\mathscr{R}}$ is due to the the combination of the two mappings $\mathscr{R}$ and $\mathscr{S}$. Both result from physical measurements and therefore they inherently carry uncertainty. The experiments on learning with $A_{\mathscr{R}}$ indicate an uncertainty of 2 +/- 1.2 *cm* for $\mathscr{R}$. Having two mappings combined in a way as it is done in $A_{\mathscr{S}}$ is inevitably accompanied by an accumulation of uncertainty which leads to the increased overall mean errors for this architecture.

Interestingly, the error values were the same for both learning and realignment for each architecture. This was not expected since realignment requires much more advanced mechanisms in order to 'get rid' of the old links in a mapping while new links are added. In contrast to a learning task where an empty mapping is exposed to a non-changing arm-vision configuration which doesn't need to involve an element of 'forgetting'. Thus, the similarity of the mean errors for learning and realignment show that it is the architecture and not necessarily the context that determines the level of accuracy that hand-eye coordination can achieve.

6.2 The impact of T and Q on accuracy and speed of learning

Regarding architecture $A_{\mathscr{R}}$ where a direct coupling of vision and reach space is implemented, the experiments show that tolerance T and minimum age value Q directly influence the final accuracy of the mapping. Overall, lower T and higher Q values produced lower mean errors. This relation holds because each additional link in $\mathscr{R}$ increases the average accuracy and, the number of links is in turn determined by T (link addition) and Q (link subtraction) values. This is the case for both learning and realignment in $A_{\mathscr{R}}$. However, with particular regard to speed of learning during realignment, it is also apparent that larger values of Q increase the overall number of links (and thereby reduce the speed of learning). This is because increasing Q increases the number of learning cycles required for links to reach the minimum age thereby increasing the length of time that they are held in the mapping. Because link removal is critical to the readaption process (i.e only when the last link in the mapping representing the old / wrong arm-vision configuration is deleted is the system completely adapted to the new situation) this means that there will always be a tradeoff between between accuracy and number of learning cycles during the realignment process.

The tolerance value T in architecture $A_{\mathscr{R}}$, however, has no direct impact in the number of learning cycles needed for complete realignment. Therefore, the minimum value $T = 0.0cm$ can be applied in $A_{\mathscr{R}}$ providing optimal error levels for a given Q without increasing the learning cycles for realignment.

Regarding architecture $A_{\mathscr{S}}$ during the learning phase, there was also an optimal range for T(8cm) and Q (50) in terms of generating the lowest possible mean errors. As previously stated, these same values, although not presented, were also generated from the realignment experiments. Interestingly, the data derived from the architecture $A_{\mathscr{R}}$ exper-

	$A_{\mathscr{R}}$	$A_{\mathscr{S}}$
learning	2.0 +/- 1.2	3.5 +/- 1.8
realignment	2.0 +/- 1.2	3.5 +/- 1.8
approximation	×	3.7 +/- 2.2

Table 4 Mean errors in cm resulting from the two architectures tested for learning and realignment. In addition best approximation results as presented for architecture $A_{\mathscr{S}}$.

iments did not predict in anyway these T and Q values for the $A_{\mathscr{S}}$ architecture *a priori*. In addition, again in contrast to architecture $A_{\mathscr{R}}$, set Q values did not predict the number of learning cycles for complete realignment using architecture $A_{\mathscr{R}}$; whilst 50 was the optimal Q value, the required number of learning cycles required before the error value plateaued (complete realignment), was much greater.

Finally, data for all learning and realignment experiments for both architectures demonstrate that the tolerance value T does not match directly with the final mean error. For example, with $T = 8.0$ *cm* the resulting mean errors were ≈ 3.6 +/-2.0 *cm* for both architectures. Hence, for large tolerance values the final average error will always be much smaller. Nevertheless, no matter how small the value of T, the average error can never go beyond the error levels which are determined by the system's intrinsic level of uncertainty.

6.3 Generalization in $A_{\mathscr{R}}$ and $A_{\mathscr{S}}$

With respect to the error levels, $A_{\mathscr{R}}$ performed better than $A_{\mathscr{S}}$. Although there may appear to be an argument for using $A_{\mathscr{S}}$ if the number of learning cycles needed for realignment is considered, this is in fact not the case since similar small learning cycle numbers can be expected for $A_{\mathscr{R}}$ if that same high level of error is considered acceptable. Interestingly for architecture $A_{\mathscr{S}}$, data showed that only a few links are actually necessary (see Fig. 10 and 13) to build up a mapping $\mathscr{S}$ but unfortunately, many more learning cycles than links are required in order to discover the latter.(80-200 vs. 8-12 respectively). Unfortunately because it is not known *a priori* which links best represent a specific sensorimotor relation, there is no indicator of how many learning cycles are required to find them.

The optimal number of links to achieve the best form of generalization for architecture $A_{\mathscr{S}}$, like $A_{\mathscr{R}}$ was determined by the Q and T parameters. However, unlike $A_{\mathscr{R}}$, increasing the number of links in the architecture did not increase but rather decreased the accuracy (e.g. 300 links gave an error value of 6.0+/-5.7cm versus 3.5+/-1.8cm for an equivalent 12 links)

The data also demonstrated that generalization capability is predominantly determined by the tolerance value T. The smaller the T value, the more likely the mapping overfits, i.e. it learns the noise. This issue was highly relevant for architecture $A_{\mathscr{S}}$ where the accumulation of uncertainty became higher with lower T values. The high tolerance value of $T = 8.0$ *cm* compensated for this effect and thus produced the the lowest error value. This is also apparent in architecture $A_{\mathscr{R}}$, but to a lesser extent, where we see that at $Q = 300$ the same level of accuracy is achieved for tolerance values 0.0 and 2.0 *cm*. The key point here is that the number of links are different (300 vs. 283 respectively)(Table 1). Hence, the mapping learned with $Q = 300$ and $T = 2.0$ provides better generality (if we accept that fewer links is a measure of generalization).

In summary, the tolerance value T determines the degree of overfitting which is highly relevant when mappings are used in combination. Unfortunately, our data did not provide any evidence as to how to estimate optimal T-values *a priori*; even knowing the uncertainty of each single mapping in architecture $A_{\mathscr{S}}$ did not allow the question about why $T = 8.0$ *cm* provided the optimal error value to be answered. However, having found the optimal T and Q values our mapping learning schema was able to achieve generalization which generated similar mean errors compared with the approximation of $\mathscr{S}$ for architecture $A_{\mathscr{S}}$ (Table 4).

6.4 Model-free learning

Our learning scheme as a case-based strategy doesn't come with any assumptions about a model. Hence, the number of learning cycles needed for learning and realignment is not biased by the number of free parameters of an underlying model. Consider, as an example, learning similar sensorimotor mappings with artificial neural networks by optimizing the weighted connections. In such cases the number of weights to be optimized has a big impact on the amount of training data needed. In this respect, we can compare our results for architecture $A_{\mathscr{R}}$ with the work of Hoffmann et al. [14] who presented a similar robotic setup with learning but not realignment experiments. Using a reach space covering an area of 40x30cm^2 with 3371 training examples Hoffmann et al. achieved a similar level of accuracy. Thus, the ratio between training examples and reach space was $\frac{3371 samples}{1200 cm^2} \approx 2.8$ samples per cm^2. With regard to our results for $A_{\mathscr{R}}$, we had 300 samples in a reach space of 3944 cm^2 and, thus, our learning results were based on a ratio of ≈ 0.08 samples per cm^2. Therefore, we can say that similar precision was achieved with 35 times less data. Bearing in mind that a single learning run (300 examples) requires almost 5 hours, the approach of Hoffmann et al. appears rather inapplicable for autonomous robot systems and particularly so if realignment is to be considered.

6.5 Cross-modal mapping

The attraction of case-based learning from a biological perspective is that it embraces two central concepts within the field of developmental research; firstly Thelen's idea of exploration and selection [19] and secondly Piaget's original dogma on schemas whereby both case-based learning and schemas can be described as non-modality specific, context driven opportunities for learning [10]. Although this provides a strong argument for biological plausibility, it cannot be logically extended to infer information about the actual underlying biological mechanism which, due to the non-neural network nature of the model, is likely to be quite different. However, the model may actually suggest greater similarities to the biology since the learning speed and the low number of learning cycles are much closer to biological systems compared to that of neural networks [13].

6.6 Readaptation

Realignment was investigated as a possible readaptive strategy whereby a global transformation (Δ) could be applied to the original learnt data based on one estimated error function. Although graphically it appeared that a linear global transformation function might be applicable (Fig. 8), it subsequently became apparent that this was not the case with both verge left and verge right (Δ) values varying non-linearly across the space. A quadratic function improved the fit of data (Table 2), however, baseline error values could still not be achieved (3.7 +/- 2.2cm vs. 2.0 +/- 1.2cm). This suggested that realignment based on a global transformation is not a completely adequate remapping strategy. Biologically, the phenomenon of alignment has been reported to generalize but on a similarly low best-fit linear (r2=0.17) [16]. Redding and Wallace interpreted this error as being indicative of untrained corresponding points left over from the original mapping. In this sense, for complete and accurate remapping to take place, the majority of points still have to be individually remapped thus embracing a strategy similar to the original robotic solution. From an evolutionary perspective, alignment as an adaptive process exists to deal with the changes that occur between sensory and motor systems during pre-adult growth [2]. However, in this context incremental changes between systems would be relatively small and there would be extended periods of time for perceptual error and learning to occur within all of the egocentric space. Thus, although realignment as a phenomenon may exist to accelerate the readaptation process, it must operate through the majority of egocentric spatial points for accurate remapping to occur. Biological data in fact suggests that this readaptation process may lie somewhere between generalization and the remapping of every point and may be algorithmically akin to a 'plastic map' solution where many but not all points are required for accurate remapping across a non-linear space [2]. This map may also provide a slightly more efficient solution for readaptation within robotic systems and thus warrants further investigation.

7 Conclusion

In this paper we have introduced a schema which allows a permanent adaptation of sensorimotor mappings in changing environments. The schema was applied to two computational architectures providing learning and realignment of a robotic hand-eye coordination. Systematic robotic experiments have demonstrated the impact of the adaptation process parameters on fast readaptation, accuracy and generalization. The analysis of the data have provided valuable insights for the application of this schema for other domains in robotics where fast adaptation processes are required. The data also provides some insights into the factors and parameters that might need to be considered to gain a full understanding of the equivalent biological system, in particular 1) the types of transformation required to deal with non-linear space and 2) the issue of deleting and creating new links between mappings as part of a realignment process.

With respect to realignment of hand-eye coordination we have shown that complete realignment can be achieved by only a few examples but to find these examples might require a test of as many examples as is needed for relearning the complete mapping which directly links reach and vision space. The reason for this is that realignment in our robotic setup requires a highly non-linear mapping in order to compensate the changes that result from an alteration of the arm-vision system configuration. This wasn't anticipated since we had started with the assumption that alterations can be represented by linear transformations in the vision space. However, the application of a physical robotic system has shown that even simple alterations of the system might require non-linear compensation mechanisms.

Furthermore, the two computational architectures examined here have also indicated that combinations of learned sensorimotor mappings can lead to an accumulation of uncertainty because each mapping itself is a result of physical measurements which inherently carry uncertainty. Hence, mechanisms must be developed which compensate for this effect. Continuously adapting sensorimotor mappings are the core element of our approach towards complex adaptive robot systems. Therefore, our robot systems will integrate different sensorimotor mappings on different levels of abstraction. It is in this context that generalization and accumulation of uncertainty are highlighted as key issues. Future research on sensorimotor learning and adaptation for autonomous systems will need to address these issues.

A Uncertainty of the active vision system

In order to evaluate these results in relation to the uncertainty of the active vision system the average tilt-verge values for specific object locations were also derived (Table 5). One can see that the vision system has a uncertainty of ≈ 0.002 *rad* in average for tilt and verge motors. The complete vision domain, however, is approximately 0.2 *rad*^{3}. Therefore, out of 2.5×10^7 distinguishable samples in the vision space our method requires only 300 examples to achieve the given performance in robotic hand-eye coordination.

centered arm-vision configuration

coord.		tilt		verge left		verge right	
d	α	avg.	dev.	avg.	dev.	avg.	dev.
30	-1.4	-0.319	0.001	-0.183	0.001	-0.364	0.002
	-0.7	-0.365	0.001	-0.116	0.001	-0.333	0.001
	0.0	-0.395	0.001	0.064	0.001	-0.178	0.001
	0.7	-0.367	0.001	0.216	0.001	-0.005	0.001
	1.4	-0.324	0.000	0.263	0.001	0.071	0.002
60	1.4	-0.336	0.000	0.429	0.002	0.249	0.001
	0.7	-0.435	0.001	0.386	0.001	0.138	0.000
	0.0	-0.496	0.001	0.086	0.001	-0.224	0.001
	-0.7	-0.430	0.001	-0.266	0.001	-0.496	0.006
	-1.4	-0.330	0.001	-0.346	0.002	-0.513	0.001

shifted arm-vision configuration

coord.		tilt		verge left		verge right	
d	α	avg.	dev.	avg.	dev.	avg.	dev.
30	-1.4	-0.317	0.001	0.043	0.001	-0.147	0.001
	-0.7	-0.361	0.001	0.126	0.001	-0.095	0.000
	0.0	-0.390	0.001	0.311	0.001	0.079	0.000
	0.7	-0.367	0.001	0.429	0.002	0.237	0.022
	1.4	-0.324	0.001	0.449	0.001	0.282	0.001
60	1.4	-0.333	0.002	0.594	0.001	0.445	0.001
	0.7	-0.431	0.001	0.596	0.001	0.396	0.001
	0.0	-0.491	0.001	0.379	0.001	0.091	0.001
	-0.7	-0.427	0.001	0.001	0.001	-0.260	0.001
	-1.4	-0.327	0.001	-0.134	0.004	-0.331	0.002

Table 5 Average and standard deviation of tilt, verge left and verge right values delivered by active vision system resulting from the saccade towards an object on the table at specific positions. (10 examples for each data point.)

B Approximation of $\mathscr{S}$ in $A_{\mathscr{S}}$

The parameters for approximating the shifts in visual space via quadratic and linear regression as well as the simple mean value are summarized in Table 6. The quadratic functions provide the best match, which are plotted in Figure 14 overlaid by the actual offset values derived from the mappings $\mathscr{R}_C$ and $\mathscr{R}_S$.

comp.	grad.	coefficients			R^2
		a	b	c	matching value
tilt	quad.	-1.408	-1.165	-0.239	0.07
	lin.	0.0	-0.053	-0.021	0.03
	mean	0.0	0.0	0.001	
verge l.	quad.	0.527	-0.208	-0.246	0.30
	lin.	0.0	0.073	-0.257	0.12
	mean	0.0	0.0	-0.236	
verge r.	quad	0.478	-0.084	-0.268	0.22
	lin.	-0.019	-0.243	0.010	0.0103
	mean	0.0	0.0	-0.244	

Table 6 Parameters for the functions $\Delta(x) = ax^2 + bx + c$ approximating the offset in visual space.

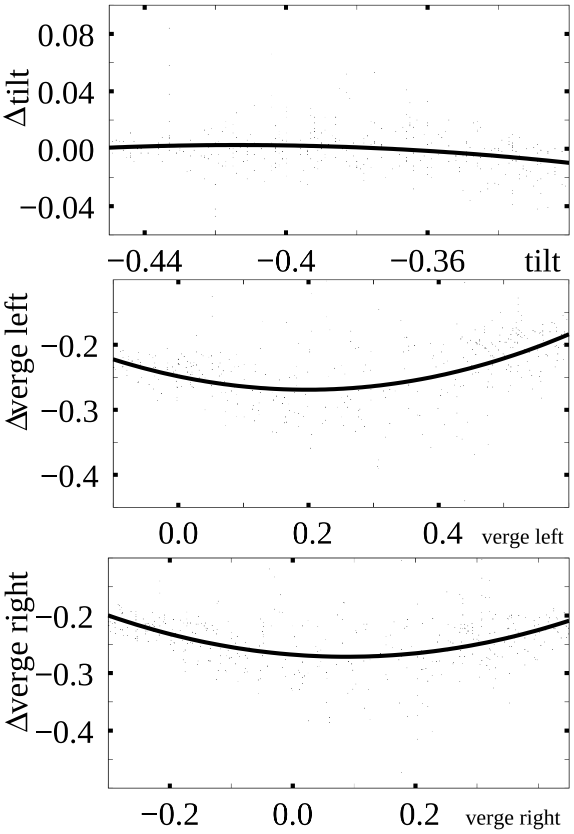


Fig. 14 Empirical data representing the offset needed to shift the points in visual space in order to compensate the change from the centered to the shifted arm-vision configuration. The three diagrams show the offsets of a component over its absolute value for tilt (top), verge left (middle) and verge right (bottom). In addition the trend lines for the quadratic approximation is shown.

References

1. R.N. Aslin, P. Salapatek, *Saccadic Localization of visual targets by very young human infant*, Perception & Psychophysics 17, 293-302, 1975.
2. F.L. Bedford, *Keeping perception accurate*, Trends in Cognitive Sciences 3, 4-11, 1999.
3. N.E. Berthier, R. Keen, *Development of reaching in infancy*, Experimental Brain Research 169, 507-518, 2006.
4. C.A. Buneo, C.A., M.R. Jarvis, A.P. Batista, R.A. Andersen, *Direct visuomotor transformations for reaching* Nature, 416, 632-636, 2002.
5. F. Chao, M.H. Lee, J.J. Lee, *A developmental algorithm for oculomotor coordination*, TAROS, Edinburgh, 72-78 2008.
6. R.K. Clifton, D.W. Muir, D.H. Ashmead, M.G. Clarkson, *Is visually guided reaching in early infancy a myth*, Child Development 64, 1099-1110, 1993.
7. J. Diedrichsen, Y. Hashambhoy, T. Rane, R. Shadmehr, *Neural correlates of reach errors*, Journal of Neuroscience 25, 9919-9931, 2005.
8. J. Findlay, I. Gilchrist, *Active Vision: the psychology of looking and seeing*, Oxford University Press, 2003.
9. S. Fraiberg, B.L. Siegel, R. Gibson, *The role of sound in the search behavior of a blind infant*, Psychoanal Study Child 21, 327-357, 1966.
10. F. Guerin, D. McKenzie, *A Piagetian Model of Early Sensrimotor Development*, Epigenetic Robotics, Brighton, UK, 2008.
11. I.D. Gilchrist, C.A. Heywood, J.M. Findlay, *Visual sensitivity in search tasks depends on the response requirement*, Spatial Vision 16, 277-293, 2003.
12. P. Harris, A. MacFarland, *Growth of effective visual-field from birth to 7 weeks*, Journal of Experimental Child Psychology 18, 340-348, 1974.
13. M. Hülse, S. McBride, M. Lee, *Robotic hand-eye coordination without global reference: A biologically inspired learning scheme*, In: Proc. Int. Conf. on Developmental Learning, ICDL 2009, Shanghai, China, 2009.
14. H. Hoffmann, W. Schenk, R. Möller, *Learning visuomotor transformations for gaze-control and grasping*, Biol Cybern, 93: 119130, 2005.
15. M.H. Lee, Q. Meng, F. Chao, *Developmental Learning for Autonomous Robots*, Robotics and Autonomous Systems, 55(9), 750-759, 2007.
16. G.M. Redding, B. Wallace, *Generalization of prism adaptation*, Journal of Experimental Psychology-Human Perception and Performance 32, 1006-1022, 2006
17. S.R. Robinson, G.A. Kleven, *Learning to move before birth, In: Hopkins, B., Johnson, S.P. (Eds.), Prenatal development of postnatal functions* Praeger Publishers, Westport, pp. 131-175, 2005
18. A. Roucoux, C. Culee, M. Roucoux, *Development of fixation and pursuit of eye-movements in human infants*, Behavioural Brain Research 10, 133-139, 1983.
19. J.P. Spencer, M. Clearfield, D. Corbetta, B. Ulrich, P. Buchanan, G. Schöner, *Moving toward a grand theory of development: In memory of Esther Thelen*, Child Development 77, 1521-1538, 2006.
20. E. Thelen, D. Corbetta, K. Kamm, J.P. Spencer, K. Schneider, R.F. Zernicke, *The transition to reaching-mapping intention and intrinsic dynamics*, Child Development 64, 1058-1098, 1993.
21. E. Thelen, D. Corbetta, J.P. Spencer, *Development of reaching during the first year: Role of movement speed*, Journal of Experimental Psychology-Human Perception and Performance 22, 1059-1076, 1996.
22. C. Von Hofsten, *Developmental changes in the organization of prereaching movements*, Developmental Psychology 20, 378-388, 1984.
23. M.T. Wallace, *Cross-modal Neural Development, In: P.T. Quinlan (Ed.), Connectionist Models of Development*, Biddles Ltd., Guildford, pp. 312-343, 2003.