

# **Action Selection and Mental Transformation Based on a Chain of Forward Models**

Heiko Hoffmann and Ralf Möller

In

S. Schaal, A. Ijspeert, S. Vijayakumar, J. C. T. Hallam und J. A. Meyer (Eds.)  
Proceedings of the 8th International Conference on the Simulation of Adaptive Behavior (pp.  
213-222). Cambridge, MA: MIT Press (2004).

# Action Selection and Mental Transformation Based on a Chain of Forward Models

Heiko Hoffmann\* and Ralf Möller\*\*

\*Cognitive Robotics  
Department of Psychology  
Max Planck Institute for  
Human Cognitive and Brain Sciences  
80799 Munich, Germany  
heiko.hoffmann@cbs.mpg.de

\*\*Computer Engineering Group  
Faculty of Technology  
Bielefeld University  
33594 Bielefeld, Germany

## Abstract

This study investigates whether goal-directed action selection and spatial mental transformations can be accomplished based on a chain of concatenated forward models. A forward model was trained with visual and motor data obtained from a mobile robot. A chain of identical forward models allowed predictions beyond one time step. We studied the accumulation of prediction errors theoretically and experimentally, and applied the chain model for two tasks. First, by applying an optimization procedure to the chain, a sequence of motor commands was determined that leads from an initial to a goal state. Second, we used the model for a mental transformation task driven by simulated motor commands.

## 1. Introduction

Forward models are internal sensorimotor models that predict how the sensory situation changes as a result of an agent's actions. The concept of forward models was originally introduced to motor control. However, recent results and new conceptual ideas broadly extended the range of cognitive capabilities in which internal sensorimotor models and specifically forward models are supposed to be involved. Wolpert et al. (2003) suggest that action observation, imitation, social interaction, and even 'theory of mind' could be based on these internal models. Hesslow (2002) presents a 'simulation theory of cognitive function', which 'postulates the existence of an associative mechanism that enables the preparatory stages of an action to elicit sensory activity that resembles the activity normally caused by the completed overt behavior'. Forward models could also play a role in spatial perception: understanding the behavioral meaning of an object may be an internal simulation of the sensory consequences of actions targeted at this object (Möller, 1999).

In this paper, we explore the capabilities of chaining forward models - concatenating input and output of two consecutive stages (see also Hesslow (2002), Jirnhed et al. (2001), and Tani (1996) for using a chain of forward models for prediction). Instead of trying to model humans or animals, we construct a simple model capable of performing goal-directed movements and mental transformation in a real environment. In this way, we hope to enlighten these 'human' capabilities. Further, we restrict our study by excluding any social ability like imitation or theory of mind - in that context, forward models were discussed by Wolpert et al. (2003) and Grush (2003), and implemented by Demiris and Hayes (2002).

Goal-directed planning usually requires not only the selection of a single motor command, but of a whole sequence. We suggest to understand the search for a motor sequence as an optimization in a chain of identical forward models. The input to the chain is the current sensory situation. The optimization criterion is that the sequence of actions results in an output of the last stage in the chain that matches the goal-situation. The free parameters are the motor commands for each stage. The idea is that the perceptual judgment leading to the selection of actions is not only based on visual data, but on visual data in the context of one's own behavior. We will show that the chain of forward models enables the selection of action sequences leading to a goal.

The chain can be further used for mental transformations. Here, the motor commands are given, and the forward model is used to predict the resulting sensory changes. However, the motor commands are not executed, but only drive the mental transformation of sensory situations. Forward models in this context were also discussed by Hesslow (2002) and Grush (2003). Here, we show that the knowledge of the sensorimotor relationships (incorporated in a trained forward model) can be the basis of perceptual judgment. Perceptual distortion is widespread in biological visual systems, for example due to inhomogeneous distribution of receptors or prop-

erties of the optical projection. Lines on a picture mostly do not look like lines on the retina or on the visual cortex. Thus, associating constant properties in the external world with a representation of the constancy may be difficult without a teacher who marks the changing image features as belonging to the same physical property. However, associating the visual changes with the movements causing these changes could be a means to accomplish perception of constancy. O’Regan and Noë (2001) suggested that the ‘mastery of sensorimotor contingencies governing visual exploration’ is the basis of visual perception. This seems similar to the above ideas. Different from our approach, O’Regan and Noë (2001) did not use forward models and robot experiments in their studies.

For our experiments, we use a mobile robot with omnidirectional vision. We deliberately select a simple experimental setup where the robot is surrounded by a circle of obstacles. Full panoramic vision and the specific setup guarantee that all relevant visual information is available. Moreover, depth information can be directly extracted from a single image. Therefore, no memory is necessary, but the setup is restricted. Other robot studies working with a chain of forward models (Tani, 1996; Jirnhed et al., 2001) use recurrent neural networks with context layers. These allow more complex environments. But, omitting context units allows us to do predictions without the need to move the robot to initialize the context values. Omitting memory will further ease the theoretical investigation.

Two tasks are studied. The first task is an action selection task, where the robot’s goal is to get close to an obstacle in a predefined direction. In the second task, mental transformation is used to decide if the robot is in the middle of the circle or not. This is based on the internal simulation of a rotation of the robot. The location of the omnidirectional vision system is chosen such that it does not coincide with the robot’s rotational axis. Otherwise, this would make the task trivial from the perspective of direct image processing.

Training and test are separated. Training data are collected by random exploration. The forward model is realized by a multilayer perceptron (MLP) (see Jordan and Rumelhart (1992) for MLPs used as forward models). The network associates a motor velocity vector (comprising the velocity of left and right side of wheels) with a change in the preprocessed and compressed visual information. A chain of concatenated MLPs enables the prediction of a change in sensory situation induced by a sequence of short movements with given velocity vectors. Since the accumulation of prediction errors can be critical, the increase of the error with increasing chain length is studied theoretically and compared with the performance of the MLP.

The remainder of this article is organized as fol-

lows: section 2 describes the methods used, the robot setup, the data collection, the image processing, the forward model network, the optimization methods for the goal-directed movements, and the mental transformation task. Section 3 shows the results of the robot experiments, which are discussed in section 4. Section 5 draws conclusions. Appendix A describes theoretically the accumulation of prediction errors.

## 2. Methods

### 2.1 Robot setup

We used a *Pioneer 2 AT* four-wheel robot from *ActivMedia Robotics* (see Fig. 1). It has differential steering and was equipped with a panoramic vision system based on an omnidirectional hyperbolic mirror (middle size, wide view) from *Accowle* (Fig. 2)<sup>1</sup>. The camera’s optical axis was positioned 12 cm in front of the robot’s rotational axis. Images were grabbed at a resolution of  $640 \times 480$  pixels. A circular shaped cover on top of the mirror prevented light entering directly into the lens without reflexion from the mirror. The illumination of the room was kept constant for all of the training and tests.

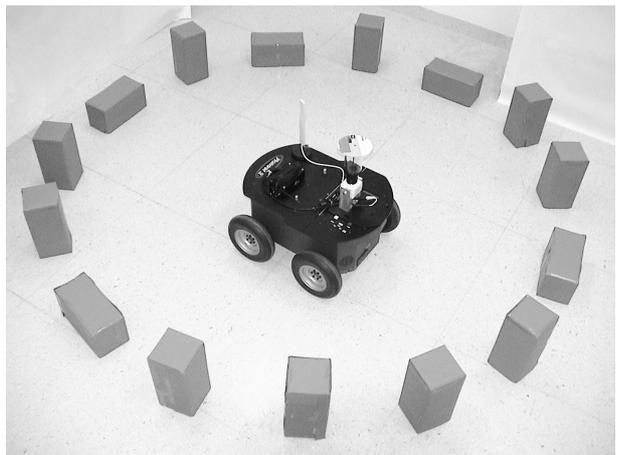


Figure 1: Pioneer robot with omnidirectional camera surrounded by 15 red obstacles

### 2.2 Collection of training data

Training data were collected by random exploration. The goal was to record the changing images induced by a given motor command. The robot was put within a circle with an inner diameter of  $180 \pm 2$  cm formed by red bricks (Fig. 1). A random velocity was chosen for the left and right side of wheels individually ( $v_L$  and  $v_R$ ). The velocities ranged from  $-60$  mm/sec to  $60$  mm/sec in steps of  $20$  mm/sec. The combination with both veloc-

<sup>1</sup>A *DFK 4303/P* camera and a *Pentax TS2V314A* lens were used.



Figure 2: Panoramic vision system

ities being zero was discarded. After a set of velocities was chosen, the robot maintained the given speed. Every two seconds, an image was recorded from the camera (Fig. 3) and stored. Recording started after guaranteeing the robot a one second acceleration phase. The series lasted up to a maximum of six shots (five 2 sec intervals) or until the robot got too close to one of the obstacles (this was determined with the help of the same kind of image preprocessing as described in section 2.4). In the first case, a new combination of velocities was chosen as above, and a new recording series started. In the latter case, the robot was either allowed to go only forward or backward, depending on if the obstacle was in the back or front, respectively. Forward movements were chosen randomly from a subset that fulfills  $v_L v_R \geq 0$  and  $v_L + v_R > 0$ . Backward movements were chosen in an analogue way (here  $v_L + v_R < 0$ ). This training scheme would result in more forward and backward movements vs. rotational movements (which fulfill  $v_L v_R < 0$ ) because when the robot was next to the obstacles it was not allowed to do turns. Therefore, when rotation was possible, the rotational movements were chosen with a higher probability to adjust toward a balanced distribution of velocity combinations.

The actual wheel velocity was recorded during the 2 sec intervals. If it deviated by more than 10 mm/sec from the given value the series was stopped, and the interval's was data discarded. After that, a new series started, as above. The robot was able to pursue this kind of random exploration automatically without getting into physical contact with any obstacle.

Totally, 5466 intervals with 6808 images were recorded. The velocities were roughly evenly distributed (Fig. 4). There was a slight dominance for straight



Figure 3: Image as seen through the mirror (part of the training set)

movements ( $v_L = v_R$ ) and a lack for combinations like  $v_L = 0$  and  $v_R = 20$  mm/sec. The reason is that the front and back wheels are connected by a chain, making the robot occasionally stick to the floor during slow turns.

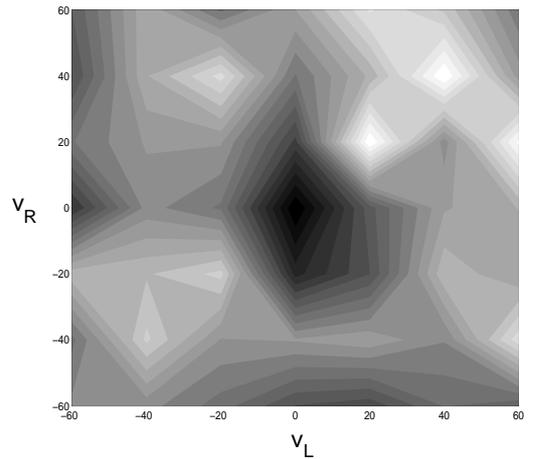


Figure 4: Distribution of velocities  $v_L$  and  $v_R$ , values in mm/s. Frequency values range from 0 (black) to 200 (white)

### 2.3 Collection of test data

Test data for evaluating the anticipation performance were collected separately. A slightly different random exploration scheme was used. The goal was to get random movement sequences instead of series with a constant motor command. For each interval, a new random velocity combination was used. A recording series consisted of eight 2 sec intervals starting from zero velocity. The first interval was discarded, thus leaving seven intervals each under identical conditions. The wheel velocity was monitored in the last 0.7 sec of each interval. If

its mean value deviated by more than 10 mm/sec from the given value the whole recording series was discarded. Therefore, the limited acceleration forced us to make the random choice of velocities slightly dependent on the previous choice. Absolute velocity changes for each wheel by 80 mm/sec or more were not allowed.

The choice of velocities in each interval further depends on the encounters with obstacles. If the obstacle was in the front the robot moved backward, and if it was in the back the robot moved forward (same way as done for the training data). Additionally, the robot responded to an obstacle on the left or right side by turning the front of the robot toward the obstacles (choosing  $|v_L| > |v_R|$ ,  $v_L > 0$ , and  $v_R < 0$  for obstacles on the right side, and accordingly for the left side, changing roles of  $v_L$  and  $v_R$ ). This behavior made the robot facing the obstacles with the front side, and thus, it was followed by a backward movement. Due to the location of the mirror in the frontal part of the robot (see Fig. 1), the bottom part of obstacles close to the rear part of the robot were occluded. Thus, this required to keep the back-side at a greater distance to the obstacles.

Totally, 138 series were recorded, with a total of 966 intervals and 1104 images.

## 2.4 Image processing

It proved to be impossible to use the original visual information in the training for the following reasons. First, the dimensionality is too high, and second, the pixels in the image change too rapidly. Therefore, the image was preprocessed to detect only a special class of objects and extract only a visual distance information in a few sectors. Image processing contained the following steps:

First, a contrast mechanism enhanced red objects ( $R - (G + B)/2$ ). The result was smoothed with a binomial filter. Then, a threshold function was applied on all pixel values. In the resulting image, the red obstacles appear as white region on an otherwise black background (Fig. 5).

In 10 sectors ( $36^\circ$  each), the distance from the center of the robot (within the image) to the closest object was determined (Fig. 5). These 10 distance values form the final representation of the sensory input to be processed by the network (Fig. 6).

The ‘motor commands’  $v_L$  and  $v_R$  together with two corresponding image representations (as in Fig. 6) from two consecutive recordings (2 sec apart) form one training pattern. Each pattern is therefore a 22-dimensional vector. Before network training, the set of patterns is normalized to have zero mean and unit variance.

## 2.5 Forward model network

The basic ingredient of our network architecture is a forward model. It gets as input the sensory information

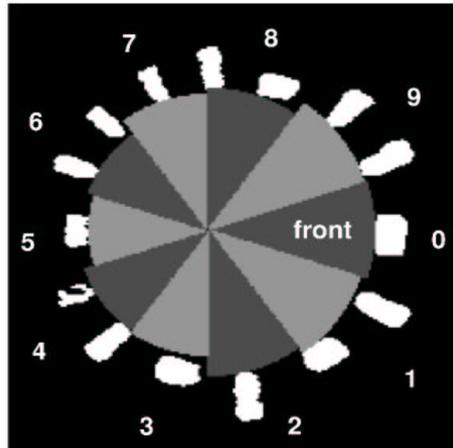


Figure 5: Distance information in 10 sectors derived from the camera image. Here, the data are taken from the image in Fig. 3

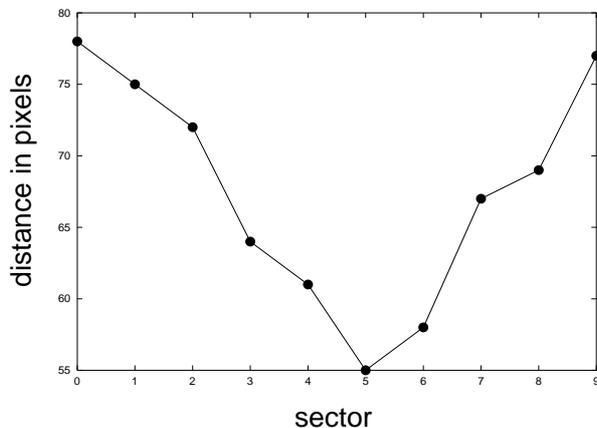


Figure 6: Visual distance in 10 sectors for the situation in Fig. 5

(Fig. 6) of one time step and the motor command consisting of the velocities  $v_L$  and  $v_R$ , and it predicts the sensory information of the next time step. Training data were collected as described in section 2.2.

To anticipate future sensory information beyond 2 sec, we feed the sensory output back into the sensory input (Fig. 7). This feedback completely overwrites the previous input. In each time step, the corresponding motor command  $M$  of the sequence is fed into the network. Thus, for illustration it seems more intuitive to replace the feedback by a chain of identical forward models (Fig. 8).

As a forward model we used an MLP with one hidden layer. The network’s activation functions were the identity on input and output, and the sigmoidal function in the hidden layer. The MLP had 12 input neurons (two velocity values and the 10 sector values) and 10 output neurons (10 sector values). The hidden layer

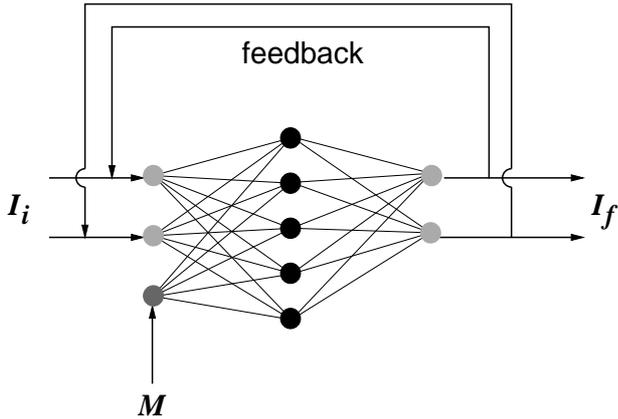


Figure 7: Forward model with feedback loop. The model maps the sensory information  $I_i$  to  $I_f$  in the context of the motor command  $M$

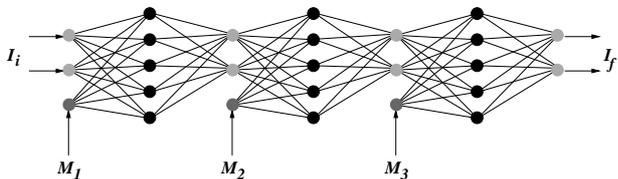


Figure 8: Concatenated chain of forward models. The sensory output of link  $t$  is the sensory input of link  $t + 1$

comprises 15 hidden units. This number seemed to be a good compromise between recall speed and accuracy. Higher numbers did not improve the performance noticeably. The network was trained on 5466 patterns with 3000 epochs of resilient propagation (RPROP) (Riedmiller and Braun, 1993).

The performance of anticipation was evaluated about the squared distance (squared error  $E^2$ ) between the output  $\mathbf{o}$  of a chain with  $n$  links and the real sensory information  $\mathbf{r}$  after  $n$  2 sec intervals,

$$E^2 = \sum_{i=1}^{10} (o_i - r_i)^2 \quad . \quad (1)$$

## 2.6 Goal-directed movements

The task in the planning of goal-directed movements is to find a series of motor commands (here velocity combinations) such that the final sensory information matches the desired value. Here, we treat this problem as an optimization task. The function to be optimized is the squared error between anticipated and desired goal. The free parameters are the velocities  $v_L$  and  $v_R$  for each time step (link in the chain).

We applied two different optimization methods, simulated annealing and Powell's method from the *Numerical Recipes* code (Press, 1992). The first is more suited to

find a global minimum, whereas the second might be caught in a local minimum.

Simulated annealing is a stochastic method for searching the minimum value, occasionally allowing jumps to higher function values. The probability of these jumps is given by the Boltzmann distribution. The temperature parameter of the distribution is slowly reduced during simulation time according to an annealing scheme. Here, we used an implementation (Carter Jr., 1994) incorporating a scheme called 'Fast Simulated Annealing' (Szu and Hartley, 1987). Further, this implementation includes at the beginning an increase in the temperature to a point where a large jump in the function value occurs, and only then starts decreasing the temperature. We used the default parameters given in the implementation, except for the learning rate, which was set to 0.1, and at each temperature value the number of random steps, which was set to 20 times the number of free parameters. Random numbers were generated using the *ran1* code from the Numerical Recipes (Press, 1992).

Powell's method is based on conjugate directions, but does not need the evaluation of a gradient. We used the parameters as given in the Numerical Recipes (Press, 1992). The fractional tolerance of the function value was set to  $10^{-4}$ .

Both optimization methods were initialized by setting all velocities to zero.

The treatment of the goal-directed movement as an optimization problem allows us to add penalty terms to the squared error to restrict the possible range of solutions. The choice of velocities beyond the range  $\pm 60$  mm/sec, used for training, was prohibited by punishing velocities outside this range with an additional term in the cost function. This was necessary because otherwise for goals out of the reach of one interval, the method could produce larger velocities as solutions for which no examples were available in the training set - there is no guarantee that the extrapolation of sensory predictions found by the network for these velocities are correct. Further, a penalty term was added to velocity series resulting in robot positions too close to the obstacles.

The above solution assumes a given number of chain links. Since the appropriate number of links - which corresponds to the duration of the movement sequence - is not known beforehand, we start with one link and increase the number of links in the optimization process. In each optimization step, we solve the optimization problem and test the optimization criterion. If the criterion is not yet met, the number of links is increased by one and the optimization is restarted (from zero velocities). This is repeated, until the anticipated sensory state matches the desired state (within 0.5 pixels - the resolution limit).

In our experiments, the goal state was not the complete sensory information, as in (1), but only the value

$t_k$  in a predefined sector  $k$ . Thus, the squared error to be optimized is  $E^2 = (o_k - t_k)^2$ , with  $o_k$  equal to the predicted output in sector  $k$ .

## 2.7 Mental transformation

In the mental transformation task the robot has to assess whether it is standing in the center of the circle. In Fig. 3, the robot is roughly in the middle of the circle, but apparently this cannot be decided from the image representation (Fig. 6). The reason for this asymmetry is that the center of the robot differs from the optical axis of the camera.

Thus, instead of deciding the above question with pure image processing, we propose a different strategy. After observing the current image, the robot simulates a left and a right turn (around its rotational axis, i.e.,  $v_L = -v_R$ ), and anticipates the effect of these movements on the image representation. From the current position the robot simulated five rotational steps (2 sec each) to the left with the velocity  $v_L = -40$  mm/sec and  $v_R = 40$  mm/sec, and, also from the current position, five steps to the right at the opposite velocity. Five steps at this speed corresponded to a rotation of  $72^\circ$ . Since our environment always contained a full circle, it was not necessary to cover the entire  $360^\circ$  in the mental simulation. Then, the values of the frontal sector for the different representations were compared (altogether 11 values). If they had a variance of less than one pixel squared it was concluded that they were the same, and therefore, the position of the robot was in the center of the circle (which is the only point having same distance to the circle boundary in all directions).

## 3. Results

In this section, we first analyze the anticipation performance achieved by the chain of MLPs. Second, this chain is applied to goal-directed action-selection, and third it is used in a mental transformation task. All results presented in the following were obtained by using the same MLP as a forward model (see section 2.5).

### 3.1 Performance of anticipation

As we show in appendix A, the squared error is expected to increase linearly with the number of anticipation steps, under the assumption of randomly independent errors for each step. This is not likely for series with a constant motor command. Therefore, the test patterns were collected during random walks. Figure 9 shows the mean squared distance about the sensory values (distances in the 10 sectors) from the starting point of a sequence (see section 2.3). A pure random walk would result in a linear increase (same derivation as the one leading to (6), see appendix A). But, as the solid curve

indicates, the squared distance increases stronger than linear. This is due to the slight dependence of velocities, as mentioned in section 2.3. The limited maximum range for random walks within the circle of obstacles reduces finally the increase in distance. For intermediate interval numbers, starting after 1 up to 6, the squared distance increases roughly linear (dashed line), as expected for random movement sequences. Therefore, we will only include the second to the sixth anticipation step for the test of the theoretic prediction of the linear increase of the squared error.

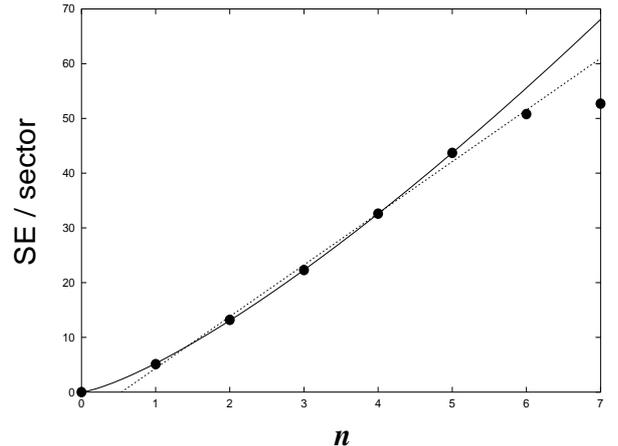


Figure 9: The mean squared distance (in pixels squared) of image sector values from the starting image representation for increasing length of the test series

The performance of the anticipation on the 138 test pattern series is shown in Fig. 10. The mean squared error was below 2.4 pixels squared for all tested chain lengths. This comes close to the image resolution. Starting from the second to the sixth step the squared error per sector increases linearly as predicted (a solid line is fitted to the data).

The MLP network was only trained on points lying on the manifold of valid sensory information embedded in a 10-dimensional space. This manifold is only two-dimensional (the robot has only two degrees of freedom, namely the distance to the center of the circle and the orientation of the robot). Therefore, the network's behavior on points slightly outside the manifold is not clear. Thus, we have a look at the effect of a small change in the input of the forward model. Let  $\mathbf{f}(\mathbf{x})$  be the transformation the MLP does on the input  $\mathbf{x}$ . Figure 11 displays the mean change of the output,  $\mathbf{f}(\mathbf{x}_0 + \mathbf{e}) - \mathbf{f}(\mathbf{x}_0)$ , as a function of a small distortion  $\mathbf{e}$ . The sensory input to the network was taken from the collected test patterns. The velocities were chosen randomly within the range of the training values. For each test pattern, 100  $\mathbf{e}$  values were chosen randomly, ranging in magnitude from 0.02 to 2 pixels. The slope  $s$  in the diagram was evaluated with a linear fit,  $s = 0.47$ . It follows that the network

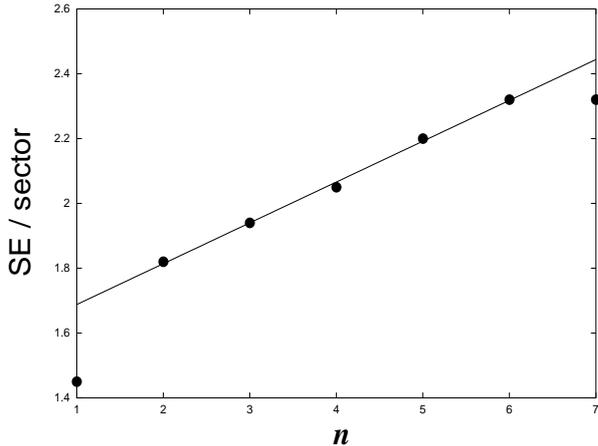


Figure 10: The anticipation performance on the 138 test series. The mean squared error (in pixels squared) per sector is shown as a function of the number  $n$  of chain links

reduces the magnitude of deviations from a valid input to about one half. See the discussion in section 4.

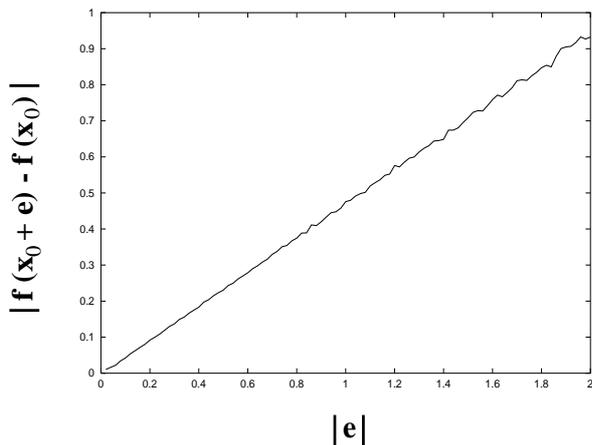


Figure 11: Stability of the forward model. The response to small deviations from a test pattern input is given. All values in pixels

### 3.2 Goal-directed action selection

The goal for the robot is to move in a way that the sensory value in a single given sector reaches a given value. Figure 12 illustrates the result of two typical movements. In example *A*, the goal was to make the sector in the back right (number 4 in Fig. 5) attain a low value (e.g. 50 pixels). The robot moved backward in a rightward curve. In example *B*, the front sector should attain a low value. Thus, the robot moved from the middle of the circle straight toward the obstacles.

To test the performance quantitatively, a random series of goals was chosen. A run consisted of choosing a

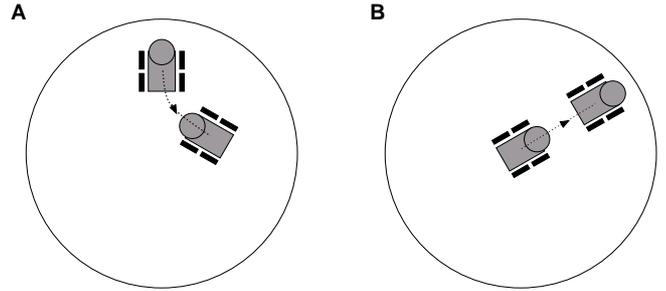


Figure 12: Typical goal-directed movements. In *A*, the goal is to have a short distance to an obstacle in the back right sector, and in *B*, in the front sector

goal and executing the resulting movement. The goal sector was chosen among the 10 sectors, and its value was chosen from the interval  $[50, 65]$ . With the given shape of the robot and the obstacles, it was physically possible for each sector to attain these values. At the beginning of each run, an image was taken, which was used as the starting point of the anticipation. At the end of a run, another image was taken for comparison with the desired goal. We did not change the position of the robot between two consecutive runs. The robot did two sequences including 50 runs. At the beginning of each sequence the robot was placed in the middle of the circle. This was done to increase the variety of movements, because during a sequence the robot happened to spend most of its time near the obstacles. Table 1 shows the result of these tests. Collisions with the obstacles during the test runs did not occur.

Table 1: The result of 100 goal-directed movements for each optimization method is shown

Optimization method:	Sim. Anneal.	Powell
Found solution:	99%	96%
Exact hits:	18%	15%
Close within one pixel:	41%	46%
Closer to goal:	85%	83%
Right direction:	91%	91%
Mean squared error/link:	4.4	2.6
Mean number of links:	2.8	2.5

The two optimization methods gave very similar results. In almost all trials both optimization methods found a solution (96 to 99%). In 15 to 18% of the trials the final sector value matched exactly the desired value, and in almost half of the trials it was within one pixel of the desired value. In more than 80% of the trials, the final sector value was closer to the desired than to the initial value, and in more than 90% of trials, the final sector value was changed in the right direction (increasing or decreasing).

### 3.3 Mental transformation

The robot was placed in 20 random positions (concentrated in the center) with random orientation. For each position the robot had to decide, with the simulation strategy described in section 2.7, if it is standing in the middle of the circle. Figure 13 shows the result of this classification. Among the positions that were classified as center, the maximum distance to the center was 10 cm.

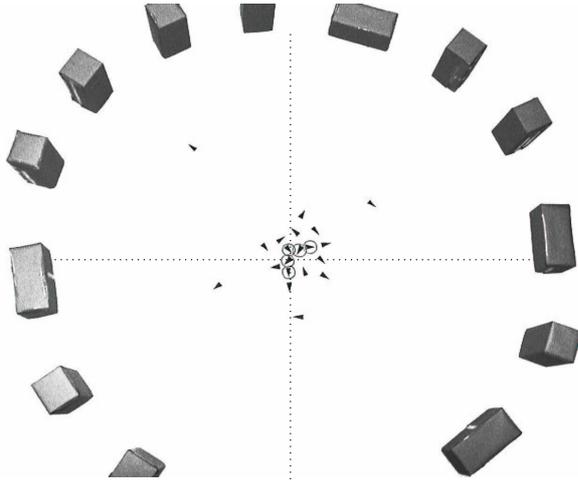


Figure 13: The performance based on mental transformation for detecting the center of the circle. Markers indicate position of the robot's rotational axis and direction the robot was facing. Markers surrounded by a circle represent trials in which the position was classified as center. The obstacle circle shown is the same as in Fig. 1

## 4. Discussion

Goal-directed motion planning requires a search in a high-dimensional motor space defined by a sequence of movements. Nothing is known about the structure of the optimization function defined over this space. The fact that Powell's method, which is a local minimization method, showed a similar performance than simulated annealing suggests that the presented task does not provide many local minima that are not global. Whether other environments have similar properties is not known.

Mental transformation could be used to judge the physical properties of the center position. No motor commands needed to be executed (this is not possible with the use of context layers (Tani, 1996)). No extra training, especially no teacher, was necessary to achieve a successful detection of the center of the circle. The maximum distance of a location, classified as center, to the real center (10 cm) is low compared to the circle diameter (180 cm), and to the length of the robot (40 cm). The remaining inaccuracy might be attributed to predic-

tion errors, and to deviations from perfect symmetry in the circle of obstacles.

Distance comparisons in our mental transformation task were solely based on the compact image representation. Alternatively, given the demonstrated performance of the anticipation, the distance could be readily obtained about the time the robot needs to reach an obstacle with a constant straight forward movement. The time could be obtained with the same mental simulation as done for the rotational movements. This alternative was not chosen in the presented paper because of efficiency. First, the image representation was enough to tell if two distances are the same. Second, an additional simulation of a straight movement would add more errors. On the one side, if a high wheel velocity would have been chosen then the time resolution would have been poor because of the few 2 sec intervals. On the other side, a low wheel velocity results in many steps, and therefore, it leads to a higher prediction error (see the linear increase in squared error).

The overall performance was limited by mainly two technical deficiencies. First, the image resolution in the region showing the obstacles was low, about 140 pixels in diameter. This could be improved by using a lens with a smaller viewing angle, or a mirror with a viewing range more constrained to the neighborhood of the robot. Second, the robot did not very accurately reproduce the given wheel velocities. This problem was avoided in the collection of training and test sets by discarding inappropriate movements. However, we could not avoid it during the execution of the goal-directed movements. That explains the higher squared errors occurring there. An improvement would be a robot using only two driven wheels and caster wheels, which does turns on a floor more easily.

The presented tasks depend on the quality of prediction. Appendix A provides an estimation of the error accumulation. It relies on the assumption that a small deviation from a given point does not change the direction of the transformation. This seems plausible after comparing the size of the anticipation error - below 2.4 pixels squared (see Fig. 10) - with the range of sector values in Fig. 6. However, a test, as done for the MLP network (see Fig. 11) to reveal the effect of a small deviation in starting position, could not be done since the true transformation is unknown, and the training patterns do not lie dense enough to serve as a substitute.

In Fig. 10, the squared error grows at a smaller rate than expected with increasing chain length  $n$ , when compared to the squared error of a single step. According to (6) the slope should be the squared error of a single step. This discrepancy might be explained by considering that the 10-dimensional sensory data lie on a two-dimensional manifold. Thus, for a single anticipation step, errors could point in all 10 dimensions. But, since the an-

anticipated states move only within the two-dimensional manifold, there is an increased chance for error compensation (as illustrated in Fig. 14). This works only if the network maps data outside its training domain onto the two-dimensional manifold (or close to) - which is not a guaranteed feature. But, luckily our network showed this behavior. The MLP decreases the distance to the manifold (see Fig. 11). To test the above idea, we computed the percentage of improvements in squared error during the anticipation series. The resulting value of 45% is much higher as expected for a random walk in 10 dimensions, and therefore supports our argument.

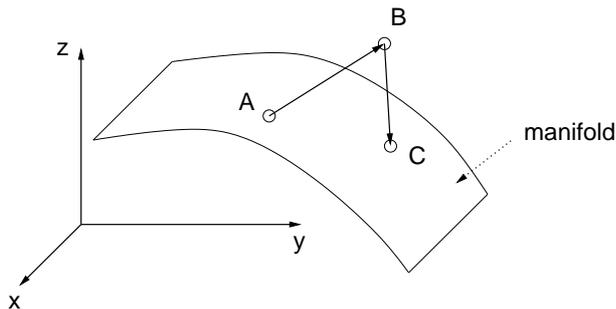


Figure 14: Compensation of the error in two prediction steps. The low dimensional data-manifold constrains the position of point  $C$

The presented anticipation method relies on the complete current sensory information given. Therefore, the method could not cope with occluded vision. But, there is a solution to this problem. Given a partially occluded image, an association network model could be used to restore the complete sensory representation. In a previous study, we developed a model capable of this task (Hoffmann and Möller, 2003).

## 5. Conclusions and future work

Although the experimental setup used in our experiments is very simple and widely artificial, the results demonstrate that forward models can be used for both planning of goal-directed actions and mental transformation.

The agent learns from the interaction with the environment, no teacher is necessary. As a result, it obtains an internal sensorimotor model. This model is then applied to interpret sensory information and to select actions leading to a desired goal. An amalgamation of sensor processing and motor control might overcome the shortcomings of ‘representationalist’ approaches to visual perception, which, in contrast, restrict perceptual processes to the sensory domain (see Möller (1999)). We suggest that a sensorimotor approach as the one described in this work could explain spatial perceptual capabilities like the interpretation of the shape and the

physical properties of objects or, as shown above, perceptual judgments of the spatial relation to objects.

We presented only a first simple example, but the approach offers extensions for future work:

- Obstacles arranged in a straight line could be detected based on mental simulating of straight movements.
- The robot could also learn a different environment like a triangle instead of a circle. It could be tested how well the model generalizes, for example, tackling the question if the robot can also operate in a square if it was only trained in circles and triangles.
- The above extensions will still work without the use of memory. But, we are also thinking of more complex mobile robot tasks, like detecting a dead-end based on internal simulation. Here, not all visual information will be available at any time.

## 6. Acknowledgments

The authors thank Wolfram Schenck for providing an implementation of the MLP network, Bruno Lara and Helmut Radrich for their effort in getting the robot running initially, Rainer Giedat for implementing a client-server interface for the optimization routines, Fiorello Banci for building a mounting for the camera, Karl-Heinz Honsberg for his assistance with electronic fixes, and the two anonymous reviewers for their comments.

### A Estimate of error accumulation

We will show that the expectation value of the squared error of the anticipated sensor input only increases linearly with the number of chain links. Let  $\mathbf{e}$  be the error of the feed-forward output of a single link.  $\mathbf{e}$  is a vector with one component for each output neuron. We assume that the probability distribution of this error does not depend on the input of the network. Thus, all errors are independent of each other. Further, we assume that the error for each output neuron has zero mean and the same standard deviation  $\sigma$ .

On this basis, we compute the expectation value of the squared error. The total error of the chain output is the sum of the errors of the outputs of each link. To illustrate this, think of each correct transformation at one link as a line in a  $d$ -dimensional space, with  $d$  equal to the number of output neurons (Fig. 15).

A line connects an input point with an output point (of the transformation). The error at link  $i$  can be drawn as an arrow  $\mathbf{e}_i$  at the end of a line (output point). This will result in a different starting point for the next line. If the error is small and the transformation function sufficiently smooth we can approximate that the displacement of the starting point does not change the direction

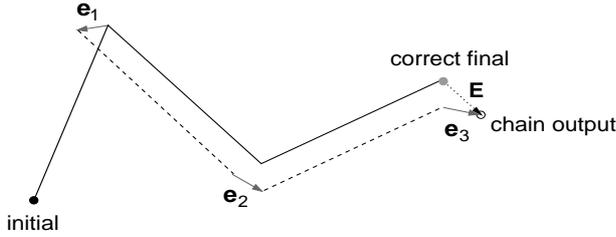


Figure 15: Error accumulation in a feed-forward chain. Each solid black line is the correct transformation for one link. The dashed black lines are the correct transformations for a slightly different starting point

and length of the next line, which would be then the correct transformation at the new starting point. Thus, the displacement  $\mathbf{E}$  of the final point is the sum of the vectorial errors of each stage. Therefore, given  $n$  links, the total error  $E$  can be written as

$$E = \left| \sum_{i=1}^n \mathbf{e}_i \right| \quad . \quad (2)$$

We want to compute the expectation value of  $E^2$ ,

$$\langle E^2 \rangle = \left\langle \left( \sum_{i=1}^n \mathbf{e}_i \right)^2 \right\rangle \quad . \quad (3)$$

Doing the square operation on the sum gives

$$\langle E^2 \rangle = \left\langle \sum_i \mathbf{e}_i^T \mathbf{e}_i + \sum_{i,j \neq i} \mathbf{e}_i^T \mathbf{e}_j \right\rangle \quad , \quad (4)$$

and using the linear property of the expectation value results in

$$\begin{aligned} \langle E^2 \rangle &= \sum_i \langle \mathbf{e}_i^T \mathbf{e}_i \rangle + \sum_{i,j \neq i} \langle \mathbf{e}_i^T \mathbf{e}_j \rangle \\ &= \sum_i \langle \mathbf{e}_i^T \mathbf{e}_i \rangle \quad . \end{aligned} \quad (5)$$

The last term vanishes because  $\mathbf{e}_i$  and  $\mathbf{e}_j$  are independent random variables, for  $i \neq j$ , and each variable has zero mean. The remainder is a sum over the variances for each link and dimension. Therefore,

$$\langle E^2 \rangle = n d \sigma^2 \quad . \quad (6)$$

Thus, the expectation value of the squared error increases only linearly with the chain length.

## References

Carter Jr., E. F. (1994). A general purpose simulated annealing class. <http://www.taygeta.com/Classes.html>.

Demiris, Y. and Hayes, G. (2002). Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model.

In Dautenhahn, K. and Nehaniv, C., (Eds.), *Imitation in Animals and Artifacts*, pages 327–361. MIT Press.

Grush, R. (2003). The emulation theory of representation: motor control, imagery, and perception. *Behavioral and Brain Sciences*. in press.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6(6):242–247.

Hoffmann, H. and Möller, R. (2003). Unsupervised learning of a kinematic arm model. In Kaynaka, O., Alpaydin, E., Oja, E., and Xu, L., (Eds.), *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003, LNCS*, volume 2714, pages 463–470. Springer.

Jirenhed, D.-A., Hesslow, G., and Ziemke, T. (2001). Exploring internal simulation of perception in mobile robots. *Lund University Cognitive Studies*, 86:107–113.

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354.

Möller, R. (1999). Perception through anticipation. a behavior-based approach to visual perception. In Riegler, A., Peschl, M., and von Stein, A., (Eds.), *Understanding Representation in the Cognitive Sciences*, pages 169–176. Plenum Academic / Kluwer Publishers.

O'Regan, J. K. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24:939–1031.

Press, W. (1992). *Numerical recipes in C: The art of scientific computing*. Cambridge University Press, Cambridge.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591.

Szu, H. and Hartley, R. (1987). Fast simulated annealing. *Physics Letters A*, 122(3,4):157–162.

Tani, J. (1996). Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26(3):421–436.

Wolpert, D. M., Doya, K., and Kawato, M. (2003). A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London. Series B*, 358:593–602.