



————— Technical Report No. 060 —————

## Modeling biological sensorimotor control with genetic algorithms

Susanne A. Huber<sup>1</sup>, Hanspeter A. Mallot<sup>2</sup>, &  
Heinrich H. Bülthoff<sup>3</sup>

————— May 1998 —————

<sup>1</sup> AG Bülthoff, E-mail: susanne.huber@tuebingen.mpg.de

<sup>2</sup> AG Bülthoff, E-mail: hanspeter.mallot@tuebingen.mpg.de

<sup>3</sup> AG Bülthoff, E-mail: heinrich.buelthoff@tuebingen.mpg.de

# Modeling biological sensorimotor control with genetic algorithms

*Susanne A. Huber, Hanspeter A. Mallot, & Heinrich H. Bülthoff*

**Abstract.** Evolutionary optimization of sensorimotor control has led to matched filter neurons in the visual system of flies that are specialized to certain visual motion patterns. We apply the technique of genetic algorithms in order to model parts of the sensor system and behavior of an artificial agent. The agents are rather simple systems with only four sensors. We will show how genetic algorithms can be applied to evolve simple matched filters that analyze the visual motion information for the task of obstacle avoidance. We compare the agents' sensorimotor control to that of flies. Further we test the optimization performance of the genetic algorithms. We can show that the use of binary or Gray coding has no significant influence on our optimization results and the speed of convergence. Real value coding leads on average to slightly smaller maximal fitness values. The use of a combination of mutation and crossover leads to high fitness individuals and a high fitness population.

---

## 1 Introduction

Many authors have described the use of genetic algorithms (GA) for the design of autonomous agents (Floreano & Mondada, 1994; Sims, 1994; Harvey, Husbands, & Cliff 1994). Mostly, agents are situated in simple environments; however, progress has been made in the last several years, and researchers have advanced to evolving agents in dynamic environments where it is difficult to identify and specify in advance all interactions between agent and environment. Many hybrid techniques have been developed that combine different adaptive processes derived from nature: evolution, development (Eggenberger, 1996; Dellaert and Beer, 1996) and learning (Nolfi, Elman and Parisi, 1994; Floreano and Mondada, 1996). Nevertheless, for robot control genetic algorithms are mostly applied to optimize some given structure of an agent, regardless of a biological counterpart.

In our work we investigate the applicability of the GA technique to generate parts of the sensor systems and behavior which can be compared to structures and behavior de-

scribed for flies. It has been demonstrated that the principle of fly vision can be used for navigational tasks in simulated and real agents (Franceschini, Pichon & Blanes, 1992; Huber, 1997). Cliff, Husbands and Harvey (1993) show the efficacy of using genetic algorithms to evolve concurrently the visual sensor system along with the control networks. We attempt to combine these approaches by using principles of fly vision and genetic algorithms to generate autonomous agents. We evolve a competence for obstacle avoidance through simultaneous adaptation of sensor parameters and the sensorimotor coupling, with the goal to compare the resulting perceptual and behavioral properties of the agent to that of flies.

The performance of GAs depends critically on the encoding of the optimization problem and the choice of control parameters. Therefore we investigate the optimization performance of the GAs, with respect to the encoding of the optimization problem and the choice of control parameters. We compare the use of different parameter coding techniques, especially Gray-coding and binary coding. In sim-

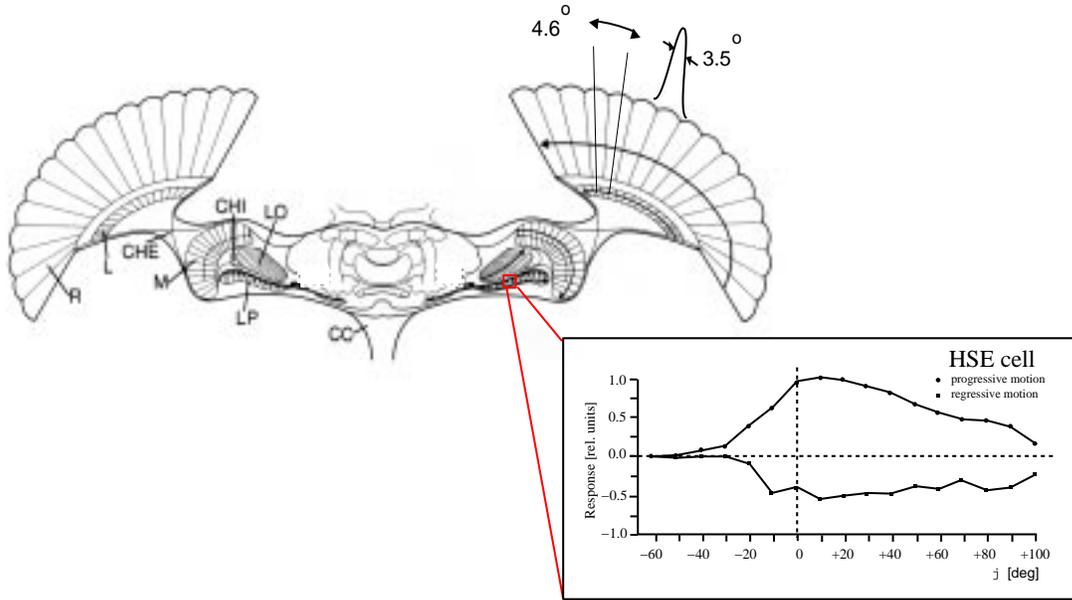


Figure 1: *Top: Cross-section through the fly's brain, with large compound eyes (adapted from Hausen, 1982), the retina (R), lamina (L), medulla (M), lobula (LO), lobula plate (LP) and the cervical connective (CC). The lamina and medulla are connected via the external chiasm (CHE) and the medulla and lobula complex via the internal chiasm (CHI). Bottom: Motion response of the HSE (horizontal equatorial cell) at an elevation of  $\theta = 0^\circ$ . Normalized mean response to stimulation with progressive and regressive motion (modified from Hausen 1982). The response is stronger to progressive image motion due to the motion detectors' asymmetric layout.*

ulations we test the GA under various parameter coding techniques (binary, Gray- and real parameter value coding), crossover and mutation probabilities, and scaling factors that define the range of fitness scaling before the selection process.

In Section 2, results from the research on the visual system of flies are reviewed and in Section 3 the architectures of the agents are described. In Section 4 the genetic algorithms are introduced and the properties of Gray- and binary coding are investigated. Then in Section 5 we present the agents that result from the optimization procedure. The optimization performance of the GA is presented in Section 6. The comparison of the evolved agents with flies and the evaluation of the applicability of GAs to evolve biologically inspired control structures are discussed in Section 7.

## 2 Visuomotor control in flies

The visually controlled orientation behavior of flies is particularly well-studied (Reichardt & Poggio, 1976; Heisenberg & Wolf, 1984; Egelhaaf & Borst, 1993; Bülthoff, Poggio,

& Wehrhahn, 1980; Wagner, 1986). The perceptual and motor system of flies did not develop independently from each other but adapted in the course of evolution in well-tuned interaction.

The resolution of the compound eyes of flies is much coarser than that of human eyes and thus the perception of shape is relatively poor. Hence, for visual orientation the detection of motion plays a more prominent role than pattern vision. Large field neurons (Fig. 1), the so-called tangential cells (Hausen, 1982), in the lobula plate – a section of the visual system of flies – specialized their receptive fields and sensitivity to certain motion patterns. They integrate the outputs of motion detectors with specialized matched filters analyzing the motion information.

For example, one class of tangential cells, the horizontal cells (HSN, HSE, and HSS cells) have their receptive fields in the dorsal, equatorial and ventral part of the fly's visual field. They respond strongest to motion from front to back (progressive) and are inhibited by motion from back to front (regressive) in the ipsilateral field. The HSN and HSE

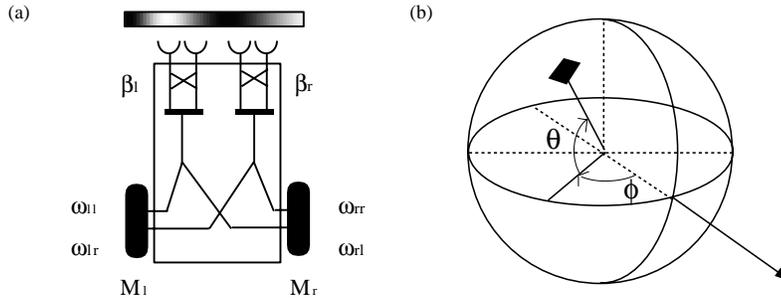


Figure 2: (a) The agent has four sensors, two of each form a motion detector. The outputs of the detectors  $\beta_l$  and  $\beta_r$  are connected to the motors  $M_l$  and  $M_r$  via the transmission weights  $(\omega_{ll}, \omega_{lr}, \omega_{rl}, \omega_{rr})$ , which are evolved with the GA technique. (b) The orientation of each sensor is given by the azimuth  $\phi$  and elevation  $\theta$  angles.

cells are designed to detect preferentially rotation around the vertical body axis, because they receive additional input from cells in the contralateral field, which respond selectively to horizontal regressive motion. Nevertheless, the entire HS system on each side and the HSS cells in particular are activated during straight flight. During translatory movements the HS system receives mainly progressive image motion, except for occasional disturbances due to for example the wind, which lead to rotatory image motion. As progressive image motion that results from translatory movements contains information about the structure of the environment<sup>1</sup>, the HS cells are most probably involved in the task of obstacle avoidance (Götz, 1980).

### 3 The agent

We simulate simple agents, which are inspired by the “Vehicles” of Braitenberg (1984). The agents that have four visual sensors and two motors (Fig. 2a) (Huber, Mallot and Bühlhoff, 1996). Two sensors form a movement detector and the outputs of the two detectors are coupled via transmission weights to the two motors. The autonomous agent gathers information about its egomotion and the environment by evaluating the visual motion signals. The orientations of the sensors (Fig. 2b) determine which part of the motion field is used to navigate through the unknown environment. The

<sup>1</sup>Objects nearby cause a larger image flow than objects further away (Longuet-Higgins & Prazdny, 1980).

resulting movement detectors form a particularly simple case of matched filters for the course control.

#### 3.1 The sensor system

The input to each sensor is computed by “ray-tracing” (Foley, van Dam, Feiner, & Hughes, 1987) where the intensities of single points – at the intersection of the line of sight with the visible surfaces – are averaged over a given number of sampling points (for more detail see Table 1). The orientations of the optical axes of the two sensors on one hemisphere of the visual field are evolved by the genetic algorithm. The other pair of sensors is oriented bilaterally symmetrical on the other hemisphere.

#### 3.2 The motion detectors

As a model for motion perception in insects, Reichardt & Hassenstein proposed a correlation detector (Hassenstein & Reichardt, 1956; Reichardt, 1961) which correlates temporal modulation of image intensities in two neighboring ommatidia. Here we use a version of the correlation-type movement detector, where the visual signals first are temporally highpass filtered ( $\tau_H = 2.0$  steps, one step corresponds to one simulation cycle), making the motion detector independent of background illumination. The visual signals are then processed in two mirror-symmetrical subunits. In each subunit the signals of two input channels are multiplied after the signals have been filtered by two lowpass filters with different time constants. The time constants of the

lowpass filters of the motion detector are fixed ( $\tau_{LP1} = 2.0$  steps,  $\tau_{LP2} = 5.0$  steps). Then the outputs of the two subunits are subtracted to obtain the direction of the motion stimulus.

### 3.3 The motor system

The agent is modeled as a simple kinematic system with two motors, ignoring its mass and inertia. In order to model the visuomotor control of flies this approximation can be made, because in flies the force produced by the wings is almost completely used to overcome the air friction. After an initial acceleration, within a short time the fly reaches a constant velocity as the applied force is balanced by the increasing air friction. The velocities  $v_l$  and  $v_r$  that result for the left and right motors are proportional to the force of the two motors. Each motor produces a constant basic velocity  $v_0$  (see Table 1), which is modulated by the visual information. In previous experiments (Huber, Mallot & Bühlhoff, 1996) we found that bilateral symmetry seems to be advantageous for a robust obstacle avoidance behavior. The sensorimotor coupling is bilaterally symmetrical as well and the matrix:

$$\mathbf{W} = \begin{pmatrix} \omega_i & \omega_c \\ \omega_c & \omega_i \end{pmatrix}, \quad (1)$$

contains the ipsi- and contralateral transmission weights ( $\omega_i$  and  $\omega_c$ ) for the coupling of the outputs  $\beta_l$  and  $\beta_r$  of the two motion detectors with the motor system. The velocity of the two motors is given by:

$$\begin{aligned} v_l &= v_0 - k(\omega_i\beta_l(t) + \omega_c\beta_r(t)) \\ v_r &= v_0 - k(\omega_c\beta_l(t) + \omega_i\beta_r(t)), \end{aligned} \quad (2)$$

where  $k = 10$  u/step is the scaling factor.

As the force produced by the wings of the fly never becomes negative, the velocities  $v_l$  and  $v_r$  are always above zero. The system has two degrees of freedom: translation in the heading direction and rotation around the vertical body-axis. The translatory and rotatory velocities are:

$$v_t = \frac{v_r + v_l}{2} \quad \text{and} \quad \dot{\psi} = \frac{v_r - v_l}{c}, \quad (3)$$

where  $c = 1$  u is the distance between the two motors. The distance  $c$  is given in units  $u$  of the agent's size:  $u = 10$  cm. Table 1 gives a short overview of the experiments with respect to the architecture of the agents.

exp.	$\Delta\phi$	$\Delta\theta$	$\Delta s$	$v_0$
1	$10^\circ$	$10^\circ$	$2.5^\circ$	0.5
2	ev.	1 pixel	$1^\circ$	ev.
3	ev.	1 pixel	$1^\circ$	ev.

Table 1: Overview for the experiments:  $\Delta\phi$  and  $\Delta\theta$  are the horizontal and vertical sensor aperture.  $\Delta s$  is the angular distance of the sampling points that are used to compute the visual input to each sensor;  $v_0$  ([u/step]) is the basic velocity; 'ev.' indicates parameters, optimized by the GA.

### 3.4 The agents' task

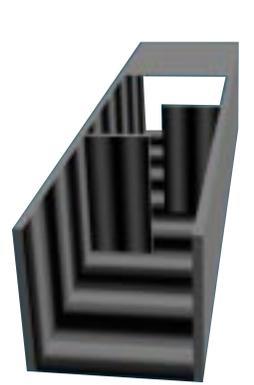


Figure 3: Tunnel with sinusoidal pattern and two obstacles.

The agents have to avoid the walls of the tunnel as well as additional obstacles. The tunnel is designed with two sidewalls, the floor, and the ceiling (only part of the ceiling is shown in Fig. 3). The width, height, and length are given in units  $u$  of the agent's size (Table 2). The number of obstacles  $m_o$ , the agent has to avoid during evolution varies between experiments. Either a sinusoidal or a random dot pattern is mapped onto the walls. The agents move at a constant height halfway between floor and ceiling. They all have to keep a safe distance  $d = 1$  u from the tunnel walls, otherwise their movement is judged as

a collision.

Nr.	$w \times h \times l$	$m_o$	pattern
1	$20 \times 20 \times 10^5$	4	S: 10 u
2	$20 \times 20 \times 1.1 \cdot 10^3$	6	S: 20 u
3	$20 \times 20 \times 1.1 \cdot 10^3$	6	RD

Table 2: Overview for the experiments: Width  $w$ , height  $h$  and length  $l$  ([u]) of the tunnel, the number of obstacles  $m_o$ , the pattern mapped onto the tunnel walls ( $S$  indicates a sinusoidal pattern, the wavelength is given in units of the agent’s size and  $RD$  a random dot pattern).

## 4 The optimization with genetic algorithms

### 4.1 Coding of the parameters

In order to test the performance of the GA with respect to the parameter encoding technique, we use either real parameter values or the parameters are encoded in a bitstring. For bitstring encoding, Forrest (1993) claimed that Gray-coded representations are often more successful than binary-coded representations for applications that optimize multi-parameter functions. Gray codes have the property that the incrementation or decrementation of the real parameter value by one step is always a 1 bit change. Therefore, point-mutation causing 1 bit changes results with a higher probability in small changes in the real parameter value for Gray-coding than for binary coding and hence a gradient approach to the optimum seems possible. However, the change of an arbitrary bit in the Gray coded string can cause step sizes in the corresponding real value that are much larger compared to those in binary codes. Nevertheless, for an entire parameter set on average, the expected number of steps has to be the same as for binary coding, because both code the same information. The question remains which of the coding schemes is more advantageous?

In order to investigate this question in more detail, we assume a real parameter value  $p$  which is represented in a binary coded string (of length  $l = nl_p$ , where  $n$  is the number of

real value parameters and  $l_p$  the number of bits per parameter). If the  $i^{\text{th}}$  bit of this string is inverted,<sup>2</sup> the new real parameter value that corresponds to the resulting bitstring is  $2^{i-1}$  steps away. The average steps needed to get to the corresponding real value for each bit inversion in a string for binary coding is (Wright, 1991):

$$\frac{1}{l} \sum_{i=1}^l 2^{i-1}. \quad (4)$$

If Gray code is used, the inversion of a bit results in a number of steps ranging between 1 and  $2^l - 1$  to reach the corresponding real value.

Hence the expected number of steps that occur per string in a parameter can be much larger ( $2^l - 1$  steps) but also smaller (1 step) for Gray coding ( $G$ ) than for binary coding ( $B$ ). Huber (1997) developed an iterative procedure in order to calculate the number of strings (with length  $l$ )  $r_l$  of a parameter set for which the expected average number of steps is *smaller* for Gray coding than for binary coding ( $G < B$ ):

$$r_l = 2^l - r_{l-1} - 2 \quad \text{with} \quad r_2 = 2. \quad (5)$$

The number of strings  $r_l$  for which the expected average number of steps is *equal or smaller* for Gray coding than for binary coding ( $G \leq B$ ), is given by

$$r_l = 2^l - r_{l-1} \quad \text{with} \quad r_2 = 2. \quad (6)$$

Although the percentage of smaller step sizes, is larger for Gray- than for binary coding, the maximal step size  $s_{\max}$  that can occur becomes very large for  $l > 3$  (Table 3). In our simulations the length  $l_p$  of the bitstring (coding a parameter  $p$ ) is either 3, 4 or 8. In cases of  $l_p = 3$  and 4, we run simulations with either Gray- or binary coding in order to test experimentally which coding procedure leads to a better performance of the GA. For  $l_p = 8$  binary coding is used because the step sizes can get extremely large for a Gray-code.

<sup>2</sup> $i=1$  is the first bit on the right counting from right to left

	$G < B$		$G \leq B$		G	B
1	$r_l$	in %	$r_l$	in %	$s_{\max}$	$s_{\max}$
2	2	50	2	50	3	2
3	4	50	6	75	7	4
4	10	63	10	63	15	8
5	20	63	22	69	31	16
6	42	66	42	66	63	32
7	84	66	86	67	127	64
8	170	66	170	66	255	128

Table 3: The number  $r_l$  and percentage of bit-changes resulting in smaller step sizes ( $G < B$ ) as well as equal or smaller step sizes ( $G \leq B$ ) for Gray-coding than for binary coding and the maximum step size  $s_{\max}$  that can occur in a set of bitstrings of length  $l_p$  for Gray coding ( $G$ ) and binary coding ( $B$ ).

## 4.2 Generation of offspring

The procedure for the generation of offspring is the same for all experiments. The new generation is obtained by the following procedure:

1. The fitness (for definition see Section 5.1) that results from the evaluation of the individuals is scaled linearly such that the average fitness  $\bar{f}$  is unchanged and maximal fitness is scaled to  $n\bar{f}$  for some constant  $n \geq 1$  (Goldberg, 1989). The coefficient  $n$  is set to:

$$n = \min\{n_c, n_0\} \quad (7)$$

where  $n_c$  is a constant value and

$$n_0 = \frac{f_{\max} - f_{\min}}{\bar{f} - f_{\min}}. \quad (8)$$

For the case  $n_c > n_0$ , scaling would cause negative fitness values if  $n_c$  were used. Therefore,  $n_0$  is applied instead. Here the scaling still leaves the average fitness  $\bar{f}$  unchanged but leads to a scaled  $f_{\min} = 0.0$  – preventing negative fitness values – and a scaled  $f_{\max} = n_0\bar{f}$ .

2. The number of offspring of each individual,  $N_i$ , is obtained by a random procedure such that the expectation  $E$  of  $N_i$  is proportional to the scaled fitness

(“roulette-wheel” selection). Then in terms of the raw fitness, we have

$$E(i) = \frac{N-2}{N} \left(1 + (n-1) \frac{f_i - \bar{f}}{f_{\max} - \bar{f}}\right), \quad (9)$$

where  $N$  is the total population size. The factor  $(N-2)/N$  is needed since only  $(N-2)$  individuals of the new generation are obtained by this scheme.

3. The selected parents exchange their genetic material by one-point crossover. The crossover-point lies between any two bits on the bitstring in the case of binary or Gray-coding. In addition point-mutation is used to introduce new genetic material into the population. In the case of a bitstring this causes a bit-inversion.

For real parameter values the crossover-point lies between the parameters. The individuals exchange parameters but no change of parameter values occurs during crossover. For real parameter values point-mutation is introduced by multiplication of the parameter  $P$  with a random factor  $(1+r)$  where  $r$  is a random number,  $r \in [-0.5, 0.5]$ :

$$\tilde{P} = P(1+r). \quad (10)$$

4. The individual with maximal fitness is transferred to the next generation automatically (“elitist-strategy”; Davis, 1991). The number of offspring is 2.

## 5 Experiments

We conducted three experiments with different parameters for crossover and mutation rates as well as the scaling factor (Table 4). For experiment 1 the optimization behavior of the GA was tested with respect to the parameter encoding technique, crossover and mutation rates as well as the scaling factor. The development of the maximal and average fitness for the various experiments are presented in Section 6. Furthermore, we varied the layout of the agents’ sensors, the number of optimization parameters and the tunnel layout between experiments.

Nr.	$p_c$	$p_m$	$n_c$
1	var.	var.	var.
2	0.3	0.01	3.0
3	0.7	0.1	2.0

Table 4: *Given are the crossover and mutation rate  $p_c$  and  $p_m$  (var. = varied in different trials of the experiment) and the scaling factor  $n_c$  for experiment 1, 2 and 3.*

We present the agents that resulted with the highest fitness values from these experiments.

### 5.1 Experiment 1: Agents with evolved sensor orientations and transmission weights

**Parameter optimization:** The free parameters of the system are the sensor orientations which implicitly define the preferred motion vector of the detectors and the transmission weights.<sup>3</sup> The parameters are encoded in a bitstring. The angles  $\phi_i$  and  $\theta_i$  ( $i = 1, \dots, 4$ ) are encoded with 4 bits within a range of  $5^\circ$  to  $175^\circ$  and a stepwidth of  $11.3^\circ$ . The weights of the sensorimotor coupling are encoded with 3 bits, with the decoded real values  $[\pm 0.01, \pm 0.05, \pm 0.10, \pm 0.50]$ . Thus 6 parameters are evolved. The length of the bitstring is  $4 \times 4$  bits +  $2 \times 3$  bits = 22 bits. The crossover and mutation probabilities are varied, in order to investigate the optimization behavior of the GA (see Section 6).

**Fitness function:** We designed the fitness function such that it provides a rough estimate of the agents' behavior, instead of describing the precise path the agent should take. The fitness function is:

$$f_1 = x(t_{\text{stop}}), \quad (11)$$

where  $x(t)$  is the actual position on the longitudinal axis of the tunnel and  $t = t_{\text{stop}}$  is the

<sup>3</sup>The orientations of the four sensors on the view sphere are given by the azimuth  $\phi_i$  and elevation  $\theta_i$  ( $i = 1, \dots, 4$ ), where  $\phi_i = 0^\circ$  gives the meridian in the heading direction and  $\phi_i = \pm 180^\circ$  the meridian backwards in the opposite direction, for the right and left hemisphere respectively. The elevation  $\theta_i$  is  $> 0^\circ$  for orientations above the horizon and  $< 0^\circ$  below.

number of steps the individual survived in the tunnel without colliding with walls. The maximum number of steps is 800.

**Agent 1:** Agent 1 developed a peculiar detector with a baseline of  $77.9^\circ$  and traveled the tunnel T1 (the tunnel in which it evolved) successfully (Fig 4c (top)) when started from the center-line even with  $\pm 10\%$  noise added to the sensor signals and the signals modulating the motor output.<sup>4</sup> When started from other positions ( $x_s = 50, y_s = \pm 15, \pm 10, \pm 5$  u), the agent does not show a robust behavior, it successfully travels the tunnel only in 22% of the trials (these trails are not shown). This is due to the facts that (i) at starting positions off the center-line, both frontal sensors (Fig 4a) detect the same obstacle and the agent does not receive sufficient information for a successful avoidance behavior, and (ii) the turning response of the agent is not large enough to avoid all obstacles.

	$y_o$	$\Delta x_o$	$\Delta y_o$
T1	20.0	103	20
T2	7.0	127	11

Table 5: *Tunnel geometry of T1 (the tunnel the agents evolved in) and T2 in terms of the average width of obstacles  $\bar{y}_o$  [u], average gap ( $\Delta y_o$  [u]) between obstacles at the same  $x$ -position in the tunnel, and the average distance ( $\Delta x_o$  [u]) between the obstacles.*

In a second test the agent had to travel tunnel T2 which has a different layout (Fig. 4c (bottom)). This tunnel has 15 instead of only 4 obstacles. The average width  $y_o$  of these obstacles is smaller than in T1 (Table 5). On the one hand this may facilitate avoiding the obstacles but on the other hand the chance that a sensor is oriented towards the obstacle is smaller and thus the chance that an obstacle avoidance behavior is initiated, decreases. The average gap  $\Delta y_o$  between obstacles at the same  $x$ -position in the tunnel is reduced to about half of the size compared to T1. The average distance  $\Delta x_o$  between the obstacles

<sup>4</sup>The noise range is chosen as a given percentage of the current signal. The magnitude of noise at every step is uniformly distributed over this range.

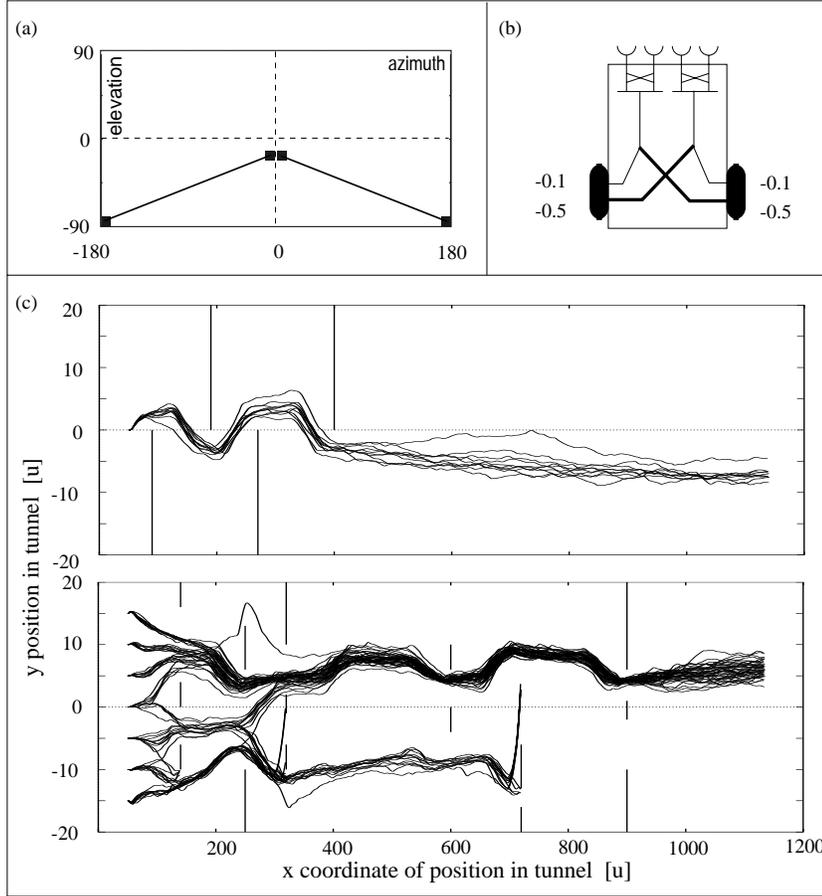


Figure 4: **Agent 1:** (a) The sensor orientations evolve at  $(5.0^\circ, -17.0^\circ)$  and  $(175.0^\circ, -85.0^\circ)$ . (b) Transmission weights coupling the detector output with the motor system. (c) Top: Paths through tunnel T1 that the agent evolved in (sinusoidal pattern  $\lambda = 10$  u). The obstacles are at  $x = 90$  u and  $270$  u on the right side ( $-20$  u  $\leq y \leq 0$  u) and at  $x = 190$  u and  $400$  u on the left side ( $0$  u  $\leq y \leq 20$  u). Bottom: Paths in tunnel T2 (sinusoidal pattern:  $\lambda = 10$  u) under variation of the starting position ( $x_s = 50$ ;  $y_s = \pm 15, \pm 10, \pm 5, 0$  u). In both cases  $\pm 10\%$  noise is added to the sensory input and the signals modulating the motor output. For every starting position the agent is tested 10 times for 2000 steps.

at different  $x$ -positions is slightly larger. Under variation of the starting position and with noise added to the input and motor signal (see Fig. 4 for more detail), agent 1 successfully navigates tunnel T2 in 73% of the trials (Fig. 4c (bottom)).

From the sensor input signals (Fig. 5a,b) and the detector outputs (Fig. 5c) one can see that the sensor oriented towards the floor does not detect the obstacles but responds to the sinusoidal pattern of the tunnel floor. The sensors oriented in the heading direction signal a sinusoidal pattern which is disturbed if an obstacle is detected. This leads to a change in the amplitude and phase re-

lation of the two sensor signals. Due to the stronger contralateral transmission weights, a larger positive detector output on the right leads to a larger motor output on the left and thus a turning response to the right and vice versa. Such a disturbance is indicated by the grey stripe in Fig. 5(a-d). A turning response results (Fig. 5e,f) such that the agent can avoid the obstacle. The angular velocity is  $\dot{\psi} = -4.0(\beta_r - \beta_l)/c$  rad/step and the velocity in the heading direction  $v_t = v_0 + 3.0(\beta_r + \beta_l)u$ /step.

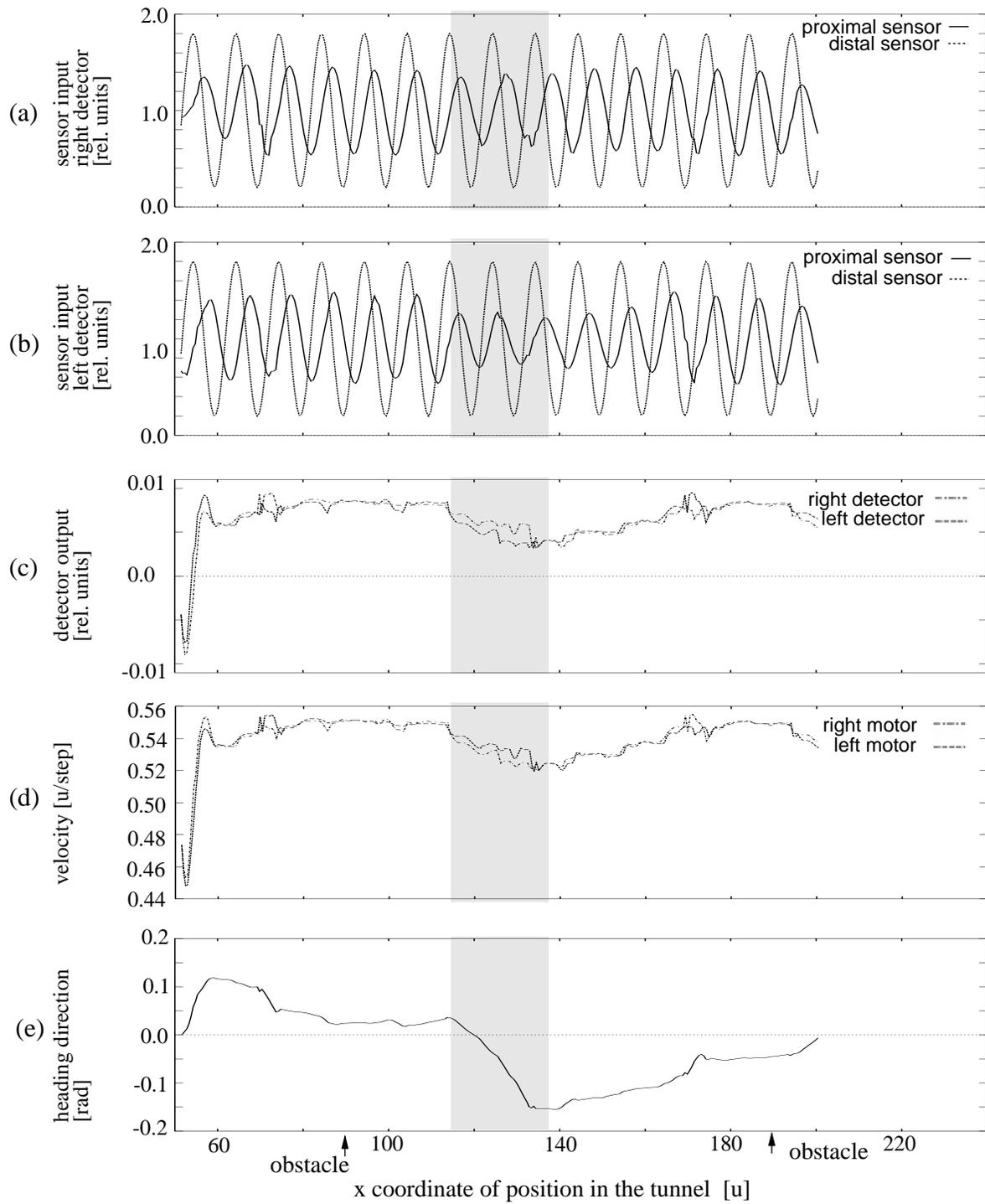


Figure 5: **Agent 1:** Sensor input to (a) the pair of sensors forming a motion detector on the right hemisphere and (b) on the left hemisphere, (c) detector outputs, (d) motor signals and (e) the heading direction of the agent (for 280 steps). The obstacles are at  $x = 90$  u on the right side ( $-20$  u  $\leq y \leq 0$  u) and  $x = 190$  u on the left side ( $0$  u  $\leq y \leq 20$  u). The intensity of the pattern ranges between  $[0, 2.0]$  in relative units. The grey stripe indicates the agent's response to the obstacle at  $x = 190$  u.

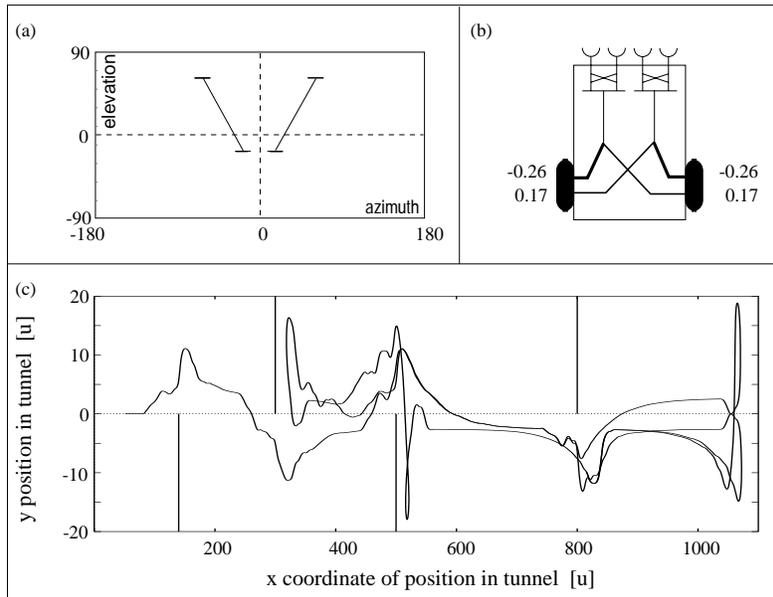


Figure 6: **Agent 2:** (a) The sensor orientations evolve at  $(16.3^\circ, -17.0^\circ)$  and  $(61.7^\circ, 62.3^\circ)$ . The sensor aperture is  $17.5^\circ$ . (b) The transmission weights couple the detector output with the motor system. (c) Path of the agent through the tunnel (5000 steps) it was evolved in. The tunnel is 1100 u long and closed by a wall at both ends. Four obstacles are placed at  $x = 140$  u and  $500$  u on the right side ( $-20 \text{ u} \leq y \leq 0 \text{ u}$ ) and  $x = 300$  u and  $800$  u on the left side ( $0 \text{ u} \leq y \leq 20 \text{ u}$ ).

## 5.2 Experiment 2: Agents with evolved sensor aperture

**Parameter optimization:** In the following simulation the angular aperture – being the same for all four sensors – together with the orientation of the sensors is optimized. In addition the constant basic velocity  $v_0$  of the two motors is a parameter of the optimization process.

For this simulation a temporal lowpass filter with a time constant of 3 steps is included to model the inertia of the motor. Azimuth and elevation angles ( $\phi_i$  and  $\theta_i$ ) of the four sensors are again encoded with 4 bits in the range of  $[5.0^\circ, 175.0^\circ]$  with a stepwidth of  $11.3^\circ$ . The angular aperture is encoded with 3 bits in the range of  $10.0^\circ$  to  $27.5^\circ$  with a stepwidth of  $2.5^\circ$ . The parameters for the transmission weights and the basic velocity are encoded with 8 bits each. The basic velocity is  $v_0 \in [0.5, 0.99] \text{ u/step}$  and the transmission weights are evolved in the interval  $[-0.39, 0.39]$ . The length of the resulting bit-string is  $4 \times 4 \text{ bits} + 1 \times 3 \text{ bits} + 3 \times 8 \text{ bits} = 43 \text{ bits}$ .

**Fitness function:** The fitness function is:

$$f_2 = asx_{\max}, \quad (12)$$

where  $s$  is the length of the agent's path, and  $x_{\max}$  the maximum value of the  $x$ -component of the agent's position. Collision is punished by dividing the fitness the agent received at the point of collision by a factor of 2 ( $a = 0.5$ ). If the agent travels the tunnel without collision for 5000 steps,  $a$  is set to 1.

**Agent 2:** The aperture of the sensors is  $17.5^\circ$  which is much larger than that of agent 1. The resulting motion detector once again has a very large baseline of  $90.0^\circ$ , with one sensor oriented in the heading direction slightly below the horizon and the other oriented laterally. Due to the orientation of the motion detectors, after initialization of the high- and lowpass filters, their motion signal is negative if no obstacle is in the agent's way. In addition the output signals of detectors on both hemispheres are equal and the agent follows a straight line. As soon as the image of the obstacle falls onto the right



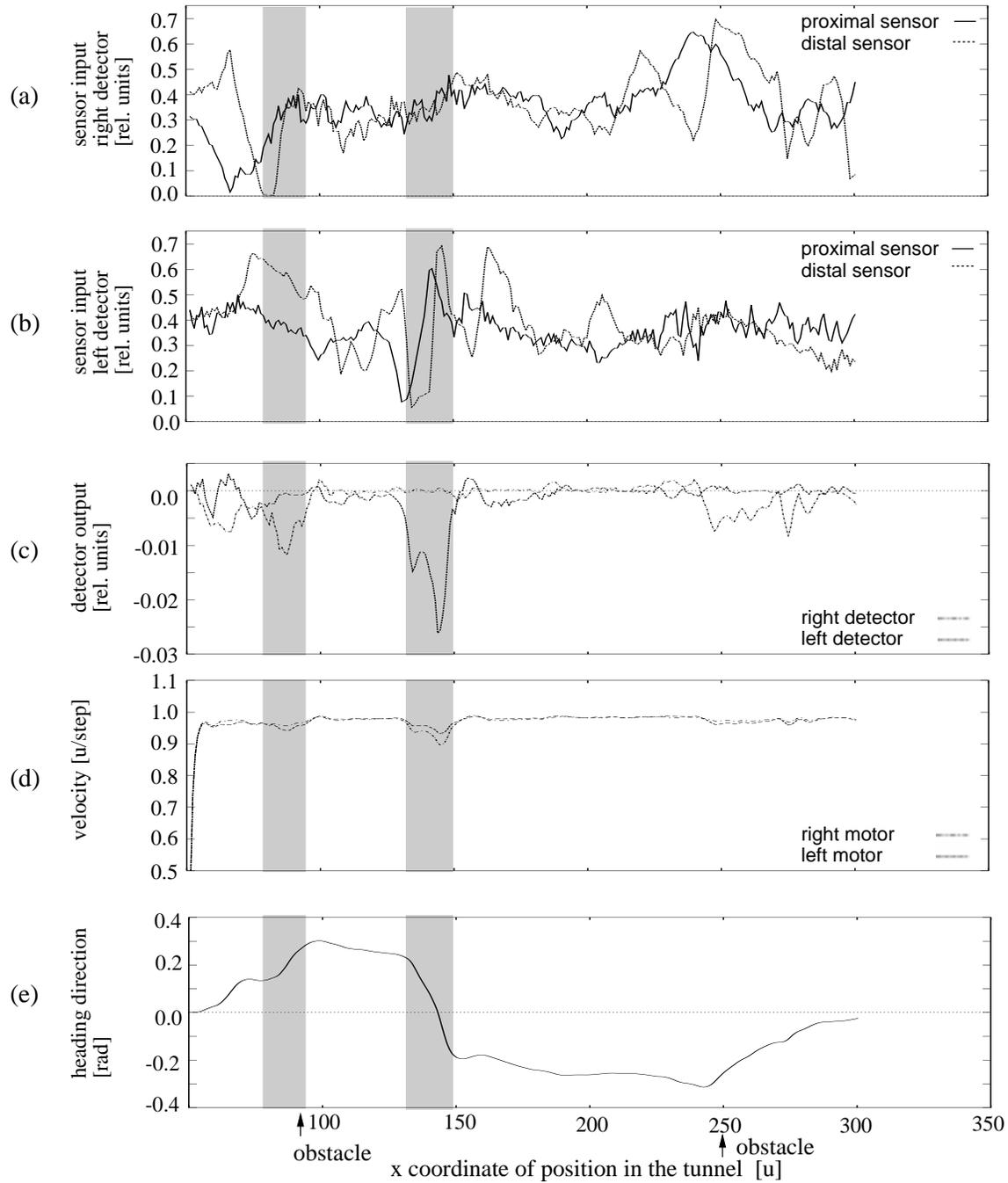


Figure 9: **Agent 3**: Sensor input to (a) the pair of sensors forming a motion detector on the right hemisphere and (b) the left hemisphere, (c) detector outputs, (d) motor signals and (e) the heading direction (for 260 steps). The intensity of the pattern ranges between  $[0, 0.75]$  in relative units. Obstacles are at  $x = 90$  u; on the right side ( $-20 \text{ u} \leq y \leq 0 \text{ u}$ ) and  $x = 250$  u; on the left side ( $0 \text{ u} \leq y \leq 20 \text{ u}$ ). The grey stripes indicates the agent's response to the obstacles at  $x = 90$  u and the tunnel wall at around  $x = 140$  u;  $y = 20$  u.

related and a large negative detector output results. Due to the orientation of the detector, it responds maximally to motion from back to front. The transmission weights for the contralateral connections are stronger than for the ipsilateral. Because of the negative sign of both the transmission weights and the detector outputs, the contralateral motor signal is more reduced than the ipsilateral and the agent is directed away from the obstacle.

The basic velocity evolved to  $v_0 = 0.98$  u/step; the angular velocity is  $\dot{\psi} = -1.5(\beta_r - \beta_l)/c$  rad/step and  $v_t = v_0 + 2.8(\beta_r + \beta_l)$  u/step. The agent is highly adapted to its environment, but cannot travel the tunnel from different starting points or with additional intrinsic noise, because the information provided by the two detectors is not sufficient in such a complex world and a larger number of motion detectors is necessary (Huber & Bülthoff, 1997). In fact, no agent could be evolved in a tunnel without bilateral symmetric random pattern on the walls.

## 6 The optimization performance of genetic algorithms

### 6.1 GA for experiment 1

The performance of GAs depends critically on the choice of control parameters. Therefore we investigated the performance of the GA for the choice of the crossover and mutation rates ( $p_c$  and  $p_m$ ) and the scaling factor  $n_c$  that defines the range of fitness scaling before the selection process. In addition we tested the performance with respect to the parameter coding techniques of binary, Gray- and real parameter value coding.

We ran four blocks of simulation. In the first two blocks we applied binary encoding (with  $n_c = 1.2$  and  $n_c = 2.0$ ), in block 3 Gray-coding ( $n_c = 2.0$ ) and in block 4 real value coding ( $n_c = 1.2$ ), where the parameters are restricted to the same intervals as the bitstring coded parameters ( $[5^\circ, 175^\circ]$  and  $[-0.5, 0.5]$ ). In each block, the crossover and mutation rates ( $p_c$  and  $p_m$ ) were varied according to Table 6. The population size was

100 and the initial population was held constant for all blocks.

	$p_c$	$p_m$
$C_{++}M_+$	0.7	0.01
$C_+M_+$	0.3	0.01
$C_{++}M_0$	0.7	–
$C_0M_+$	–	0.01
$C_0M_{++}$	–	0.05
$C_0M_{+++}$	–	0.10

Table 6: Probabilities for crossover  $p_c$  and mutation  $p_m$ .

### 6.2 Results

**Maximal fitness:** Figure 10a shows the fitness of the best individual after 100 generations averaged over 10 trials for all blocks. The averaged maximal fitness is about the same for all blocks, except for real value coding, the maximal fitness tends to be smaller and shows a much higher variation. The agent 1 with the highest fitness ( $F_1(99) = 481$  u) results in 15% of the cases from the simulations with a combination of crossover and mutation, in 10% of the cases from simulations without crossover and only in 3% from simulations without mutation. In the rest of the cases the maximal fitness is slightly smaller than 481 u.

**Average fitness:** Figure 10 shows the average fitness averaged over the maximal fitness in the population and over 10 trials. With a small scaling factor (block 1 and block 4), the population hardly increases its fitness after 100 generations, except for binary coding and trials without mutation. Increasing the scaling factor (block 2 and 3) leads to a high average fitness except for very high mutation rates ( $C_0M_{++}$  and  $C_0M_{+++}$ ). The disruptive power of mutation causes a lot of changes in the bitstrings of a population, and destroys a large number of individuals, independent of their fitness.

**Comparison to random search:** A random search technique, where the fitness of

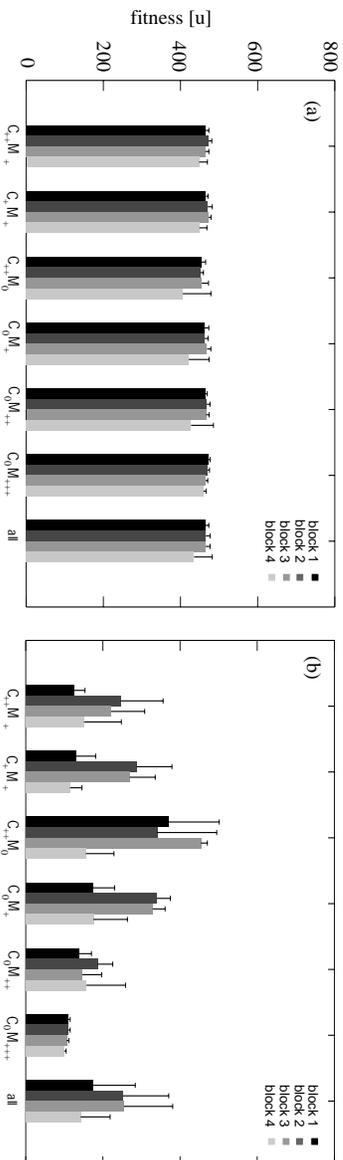


Figure 10: *Experiment 1: (a) Maximal fitness at generation 99 and (b) average fitness at generation 99. Both averaged over 10 simulation runs for block 1, 2, 3 and 4 (1: Gray-coding,  $n_c = 1.2$ , 2: Gray-coding,  $n_c = 2.0$ , 3: binary coding,  $n_c = 2.0$ , 4: real valy coding,  $n_c = 1.2$ ).*

30,000 individuals (10,000 with Gray coded, with binary coded and with real value parameters, respectively) is evaluated, finds only individuals with  $f_1 \leq 452$  u. The genetic algorithm finds on average individuals with  $f_1 \geq 452$  u 77% of the cases after 22 generations, i.e. after testing 2200 individuals.

**The evolved detectors:** With a sinusoidal pattern on walls and obstacles, the agents evolve movement detectors with a very large baseline (Fig. 11). The detector orientations and sensor spacing that result for the best individual at generation 99 show a high variability. This indicates that the fitness function has many local optima. The distribution for the proximal sensor when bistring coding is applied shows two maxima at  $\pm 17.0^\circ$  in the heading direction. This is an orientation where the obstacles can be detected very well. In addition, the distribution of the detector orientations is in all three blocks symmetrical to the horizon ( $\theta = 0^\circ$ ), as one would expect for a tunnel which is symmetrical above and below the agent.

### 6.3 Results of GA performance for experiment 2 and 3

Previous results indicate that a combination of crossover and mutation with a high scaling factor  $n_c$  is a good strategy to find rapidly individuals with high fitness and a population with high average fitness. As can be seen in

Fig. 12 this result holds true for experiment 2 as well. Comparing the optimization behavior with the random search technique, the GA finds agents with much higher fitness. The maximal fitness found by the random search is  $f_2 = 59.5 \cdot 10^4$  u<sup>2</sup>.

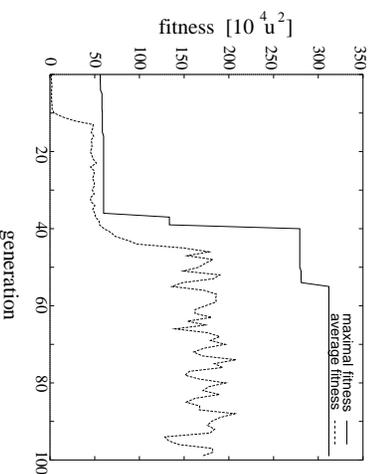


Figure 12: **Agent 2:** *Maximal and average fitness for the generations 0 to 99, that result from a GA with  $n_c = 3.0$ ,  $p_c = 0.3$ ,  $p_m = 0.01$ , and  $N = 100$  with the maximal fitness  $f_2(99) = 311.9 \cdot 10^4$  u<sup>2</sup>.*

For the optimization of the agent moving through a tunnel with random pattern mapped onto the tunnel walls and obstacles, the same control parameters for the GA are applied as in the previous experiment, except of the scaling factor ( $n_c = 2.0$ ) and as a result the average fitness of the population (Fig.13) does not increase to the same extend as in the previous experiment. The maximal fitness that is obtained is  $f_2(99) = 245.8 \cdot 10^5$  u<sup>2</sup>.

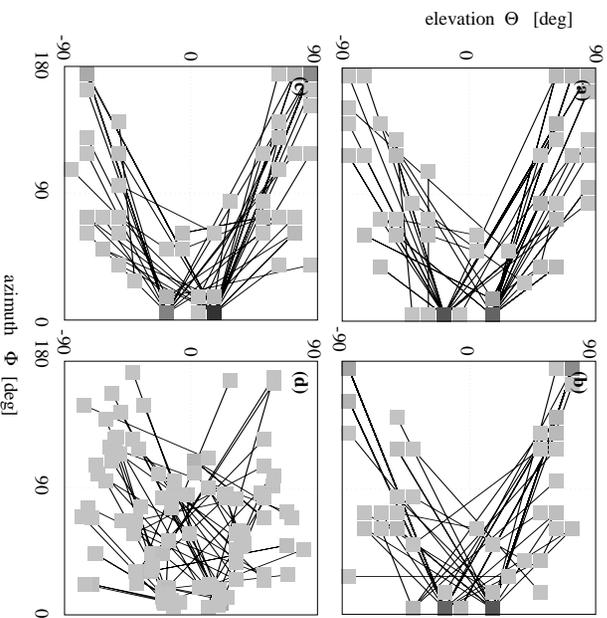


Figure 11: *Experiment 1: Distribution of the detectors for the best individual at generation 99 for the simulation (a) block 1, (b) block 2 and (c) block 3 (d) block 4. Average angular baseline for (a)  $83.0^\circ \pm 28.2^\circ$ , (b)  $91.7^\circ \pm 27.1^\circ$ , (c)  $83.5^\circ \pm 22.3^\circ$ , (d)  $82.8^\circ \pm 34.3^\circ$ . Darker grey-values indicate that this orientation appeared more often. Only the sensors that form a detector on one hemisphere are shown where  $\phi = 0^\circ$  indicates the agents heading direction and  $\phi = 180^\circ$  the opposite direction. The other sensor pair is oriented bilaterally symmetrical on the other hemisphere.*

With the random search technique the highest fitness that is found is  $f_2 = 6.2 \cdot 10^5 \text{ u}^2$ , this is much smaller than the fitness that results for individuals evolved with the GA.

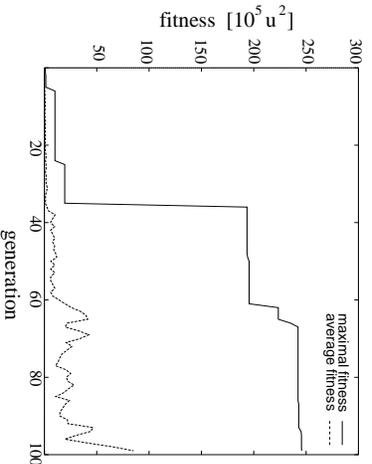


Figure 13: **Agent 3: Maximal and average fitness for generation 0 to 99 that result from a GA with  $n_c = 2.0$ ,  $p_c = 0.7$ ,  $p_m = 0.01$ , and  $N = 100$  with the maximal fitness  $f_2(99) = 245.8 \cdot 10^5 \text{ u}^2$ .**

## 7 Summary and discussion

### 7.1 Agent with two motion detectors

We presented agents that generate an obstacle avoidance behavior during a simulated evolution with GAs. They develop the viewing direction of their sensors and the transmission weights for the sensorimotor-coupling in a closed loop. We designed the fitness functions such that the agent's exploration behavior evolved along with the obstacle avoidance behavior. Rather than defining a fitness function that gives the precise path the agent should take, the performance of the agents is measured by some estimate for the distance they can travel without colliding with an obstacle. In experiment 1 agents evolve from the GA that are robust against variations of the environment including a different tunnel layout, variation of the starting position, and additional noise on sensor and motor signals. The agents that evolve in a tunnel with sinusoidal pattern on the walls perform very well and are highly adapted and specialized to this environment; however, they cannot sur-

live in any other environment. In most cases agents are optimized with one sensor oriented in the heading direction, that detects obstacles in front of the agent, and the other sensor oriented towards the floor, ceiling or side wall. The resulting motion detectors correlate signals from sensors at a large angular separation. With their very large baseline, the layout of the detectors differs drastically from that of flies. If the sensors do not detect an obstacle, the symmetric layout of agent and environment leads to symmetric sensory inputs and motor outputs and hence to pure translations. As soon as an obstacle appears, the symmetry of the internal control signals does no longer exist. The sensor layout and the transmission weights evolved such that the agent turns away from obstacles.

From experiment 3 where a random pattern is mapped onto the walls and obstacles, agents with a small motion detector baseline evolve. These detectors exploit the correlation of the signals of neighboring sensors and image motion is detected. Motion detectors with a large angular baseline do not provide a signal that is useful for obstacle avoidance due to uncorrelated input signals. The detector layout of the agent 3 is comparable to that of flies. Like flies the agent turns into the direction of a smaller image flow. The agents are able to avoid obstacles in comparatively simple environments; however, they would fail under real world conditions, because the information provided by their four sensors is very sparse. In a complex environment visual motion information from a larger field of view has to be averaged in order to obtain meaningful information about movements relative to the visual surroundings.

## 7.2 The optimization behavior of genetic algorithms

We tested the optimization performance of the GA with respect to (i) the encoding of the optimization problem (ii) different parameter coding techniques (binary, Gray- and real parameter value coding), (iii) different scaling factors that define the range of fitness scaling before the selection process, and (iv) variation

of the crossover and mutation probabilities. In addition, the optimization performance of the GA that falls into the class of guided random search techniques was compared to pure random search.

Our results give insight to the influence of the control parameters for this optimization problem. The sensor orientations that result from the various simulation trials are very diverse, as there are many different comparable good solutions to the optimization problem. This indicates that the fitness function has many local optima. Using a GA is advantageous because the search is performed in parallel from many starting points in addition to a guided random search.

Most important for a successful optimization procedure is the careful parameterization of the problem. The encoding of the optimization problem influences the result drastically (compare agent 1 and 2). In experiment 2 the agent's sensor aperture is included as an optimization parameter. Larger apertures evolve, which enhance the performance of the agents considerably.

We can conclude that high fitness values can be found for bitstring coding (Gray- and binary coding). The use of Gray-coding or binary coding has no significant influence on the optimization results and the speed of convergence. With real value coding, where the parameters are restricted to the same intervals that were used for bitstring coding, the resulting maximal fitness values are on average slightly smaller and the variance is higher.

Using crossover only or low mutation rates the GA may converge to a local optimum if the diversity of the initial population is small. The combination of mutation and crossover leads most of the time to high fitness individuals. However, with a low scaling factor the population does not gain much fitness, because many individuals are destroyed under these circumstances. The effect of the high disruption rate of the GA can be reduced by a larger scaling factor. Now, the high fitness individuals produce a larger number of offspring and the population gains more fitness. Nev-

ertheless one has to choose the scaling factor carefully, in order to prevent a premature convergence of the GA. The assumption that the GA is transformed to a random search, when high mutation rates are applied, can be rejected. The GA technique is faster than random search.

## 8 Conclusions

We can conclude that the genetic algorithms can be very useful to model the sensorimotor control of autonomous agents. If the goal is to design systems that can be compared to biological counterparts, the desing of the agent as well as the choice of the optimization parameters which describe the agent, are crucial. Besides the description and design of the agent, also the environment in which the agents are evolving, has to be realistic. Optimization of agents in a simplified world results in very specialized agents. A complex architecture of the agent necessitates a complex environment and vice versa.

## References

- Braitenberg, V. (1984). *Vehicles - experiments in synthetic psychology*. Cambridge, MA: The MIT Press.
- Bülthoff, H., Poggio, T., & Wehrhahn, C. (1980). 3-D analysis of the flight trajectory of flies *Drosophila melanogaster*. *Zeitschrift für Naturforschung*, **35c**, 811 – 815.
- Cliff, D., Harvey, I., & Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, **2(1)**, 73 – 110.
- Davis, L. (1991). *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold.
- Dellaert, F., & Beer, R. D. (1996). A developmental model for the evolution of complete autonomous agents. In *Proc. 4th Int. Conf. Simulation of Adaptive Behavior*, pp. 393 – 401 Cambridge, MA. The MIT Press/Bradford Books.
- Egelhaaf, M., & Borst, A. (1993). Motion computation and visual orientation in flies. *Comparative Biochemical Physiology A*, **104(4)**, 659 – 673.
- Eggenberger, P. (1996). Cell Interactions as a control tool of developmental processes for evolutionary robotics. In *Proc. 4th Int. Conf. Simulation of Adaptive Behavior*, pp. 440 – 448 Cambridge, MA. The MIT Press/Bradford Books.
- Floreano, D., & Mondada, F. (1994). Automatic creation of an autonomous agent: Genetic evolution of a neural – network driven robot. In *Proc. 3rd Int. Conf. Simulation of Adaptive Behavior*, pp. 421 – 430 Cambridge, MA. The MIT Press/Bradford Books.
- Floreano, D., & Mondada, F. (1996). Evolution of plastic neurocontrollers for situated agents. In *Proc. 4rd Int. Conf. Simulation of Adaptive Behavior*, pp. 402 – 410 Cambridge, MA. The MIT Press/Bradford Books.
- Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1987). *Computer graphics – principles and practice*. Reading, MA: Addison – Wesley.
- Forrest, S. (1993). Genetic Algorithms: Principles of natural selection applied to computation. *Science*, **261**, 872 – 878.
- Franceschini, N., Pichon, J. M., & Blanes, C. (1992). From insect vision to robot vision. *Phil. Trans. Royal Soc. Lond. B*, **337**, 283 – 294.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass: Addison Wesley.
- Götz, K. G. (1980). Visual guidance in *Drosophila*. In *O. Siddiqi, P. Babu, L. M. Hall, J. C. Hall (eds.): Development and neurobiology of Drosophila*. NY: Plenum Publishing Corp.

- Harvey, I., Husbands, P., & Cliff, D. (1994). Seeing the light: Artificial evolution, real vision. In *Proc. 3rd Int. Conf. Simulation of Adaptive Behavior*, pp. 392 – 401. The MIT Press/Bradford Books, Cambridge, MA.
- Hassenstein, B., & Reichardt, W. (1956). Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenbewertung bei der Bewegungsperzeption des Rüsselkäfers *Chlorophanus*. *Zeitschrift für Naturforschung*, **11b**, 513 – 524.
- Hausen, K. (1982). Motion sensitive interneurons in the optomotor system of the fly. *Biological Cybernetics*, **45**, 143 – 156.
- Heisenberg, M., & Wolf, R. (1984). *Vision in Drosophila*. Berlin: Springer Verlag.
- Huber, S. A. (1997). *Studies of the visual orientation behavior in flies using the artificial life approach*. Schwangau, Germany: Ingeborg Huber Verlag.
- Huber, S. A., & Bülthoff, H. H. (1997). Modelling obstacle avoidance behavior of flies using an adaptive autonomous agent. In *Proc. 7th Int. Conf. on Artificial Neural Networks*, pp. 709 – 714 Berlin. Springer.
- Huber, S. A., Mallot, H. A., & Bülthoff, H. H. (1996). Evolution of the sensorimotor control in an autonomous agent. In *Proc. 4th Int. Conf. Simulation of Adaptive Behavior*, pp. 449 – 457 Cambridge, MA. The MIT Press/Bradford Books.
- Longuet-Higgins, H. C., & Prazdny, K. (1980). The interpretation of a moving retinal image. *Proc. Royal Soc. Lond. B*, **208**, 385 – 397.
- Nolfi, S., Elman, J. L., & Parisi, D. (1994). Learning and evolution in neural networks. *Adaptive Behavior*, **3(1)**, 5 – 28.
- Reichardt, W. (1961). Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In *W. A. Rosenblith (ed.): Sensory Communication* (pp. 303 – 317). New York: The MIT Press and John Wiley and Sons.
- Reichardt, W., & Poggio, T. (1976). Visual control of orientation behavior in the fly. Part 1: A quantitative analysis. *Quart. Rev. Biophys.*, **9(3)**, 311 – 375.
- Sims, K. (1994). Evolving virtual creatures. In *Proc. Ann. Conf. Computer Graphics*, pp. 15 – 22 Reading, MA. Addison – Wesley.
- Wagner, H. (1986). Flight performance and visual control of flight of the free-flying housefly (*Musca domestica L.*), III. Interactions between angular movement induced by wide- and smallfield stimuli. *Phil. Trans. Royal Soc. Lond. B*, **312**, 581 – 595.
- Wright, A. H. (1991). Genetic algorithms for real parameter optimization. In *Foundations of Genetic Algorithms 1* (pp. 205–218). San Mateo, CA: Morgan Kaufmann.